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, similar to 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.

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 allogorithm 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 algorithm for handling data, this helps in getting through different algothms available including TFIDF, NLP algorithms, word2vec algorithm, kmean algorithm, classification of data using all these process 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 using use of all these algorithms simplify the management of data.

1.3 Specific Significance:

This application helps in fetching the answer to particular 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.

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

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 algorithm already existing but the main problem in the existing algorithms are completemess 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 probem. Hence we should integrate multiple algorithm for high accuracy in desiging 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 are 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 souces that are available to us are limited. We can increase the accuracy to provide the best awswer to any question is by considering all the data souces that are available on the web. The soluction for this approach is the knowdege 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

resolvng the problem of extraction of data and its integration to fetech 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 contruction 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.

4. Specific Datasets

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

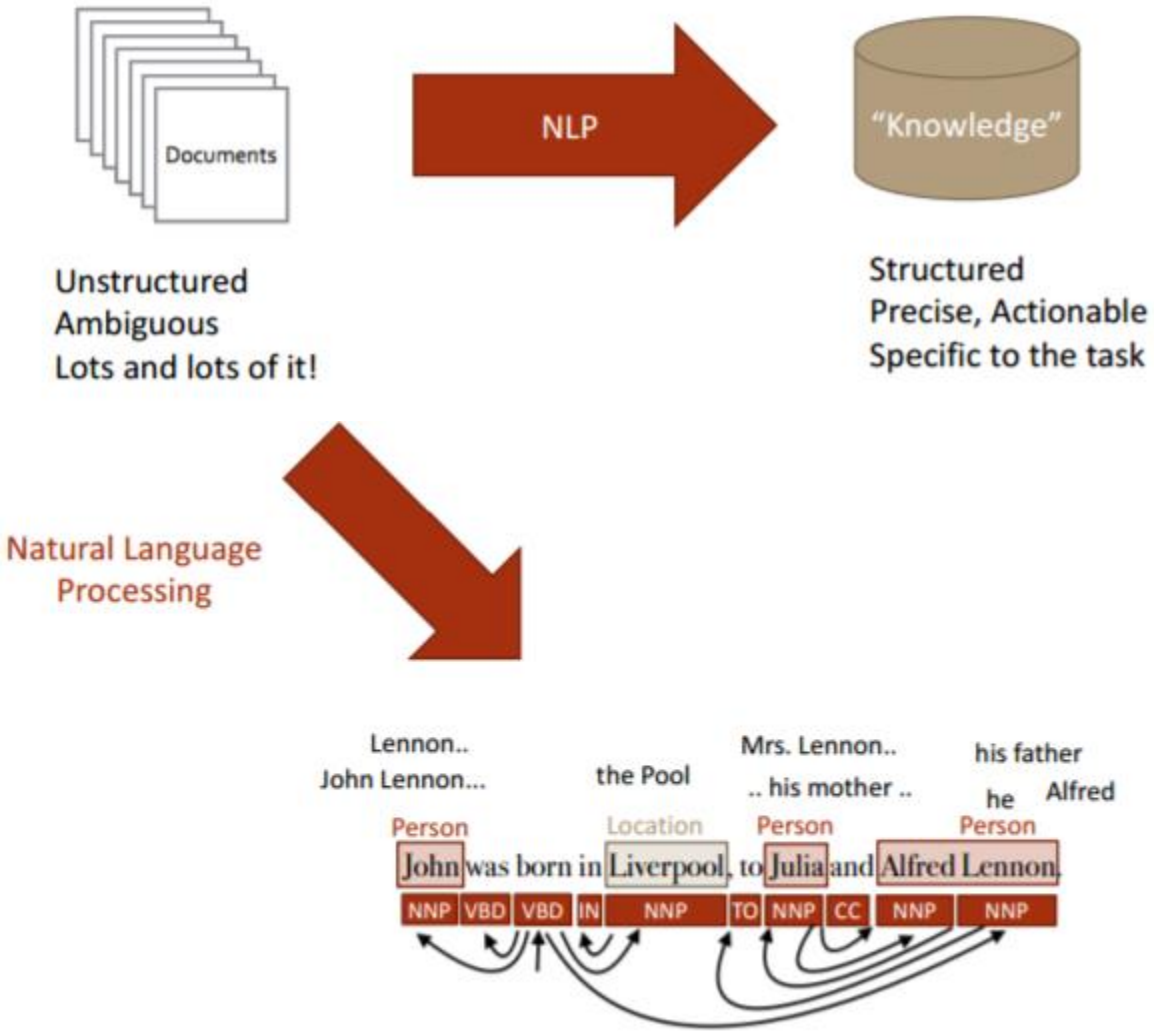
- WikiRef220
    - WikiRef220 is the collection of the news article, taken from the Wikipedia pages.
    - This dataset includes the information in the form of text data.
    - The articles included in this dataset are November 2015 paris attack, Flight 370 Malaysian Airlines, premier league, Michelle Obama, Samsung Galaxy.

- BBC News-In this especially we have selected politics area and sports.
    - This datasets 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 pollical area and sports area of the BBC news dataset.

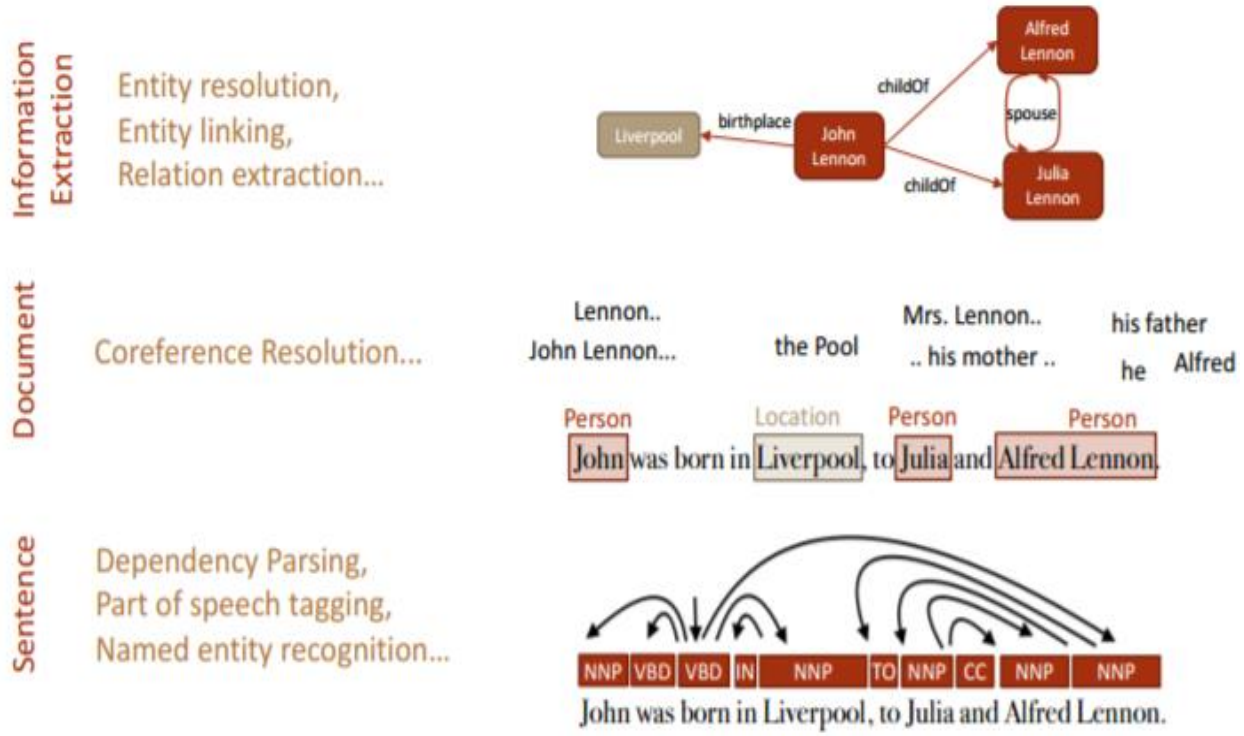
5. Design

5.1 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.



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

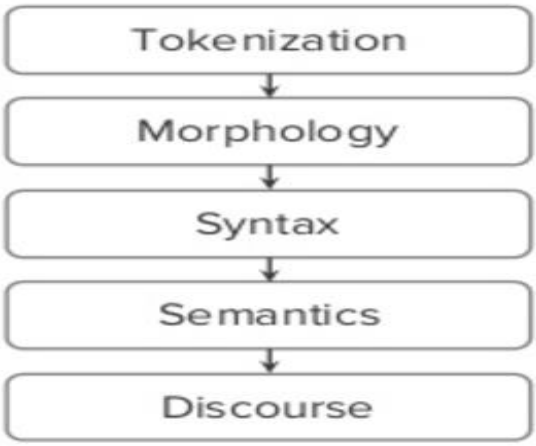
Step 4: Knowledge Graph construction – Construction of the knowledge graph from generated NER.

5.2 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 in order to perform automatic aummarization, named entity recognization, translation, relationship extraction, speech recognization, 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 reconization , stemming i.e., excluding the postfix words to get the base root word, Named entity recognization.
- Syntax – Parsing Constituency or dependency
- Semantic – Coreference resolution i.e., finding the context that belongs to same entity.



5.3 Information Retrieval

Information retrieval is the process of tracing through the stored data and recoving 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.

5.3.1 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 increses with the repeteation 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 particular document and Inverse document frequency i.e., log value number of document the word exists divided by the total number of documents.

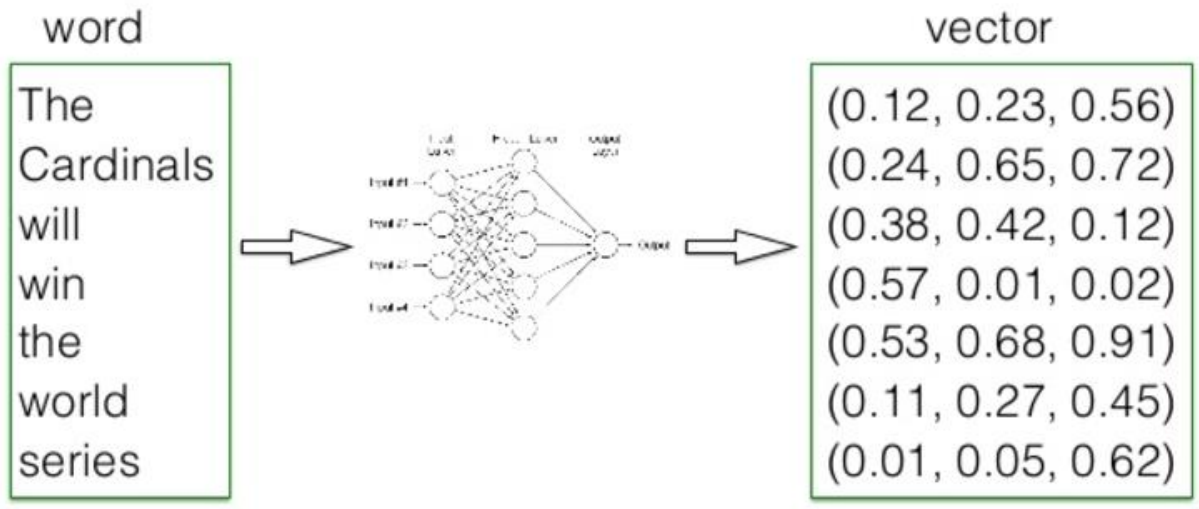
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

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

5.3.2 Word2Vector

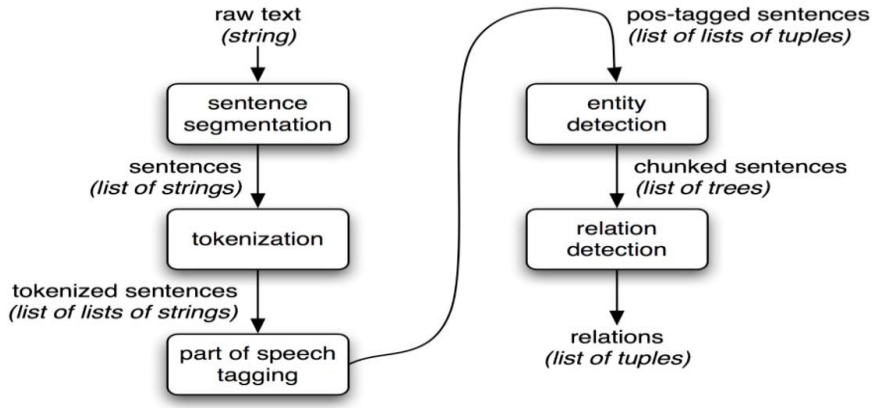
Word2Vec is the process of construction of the vector from the huge text document. All the word vectors are marked in the vector space where the closely meaning words are very close to each other. Thus mean that they are the same grouped words.

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



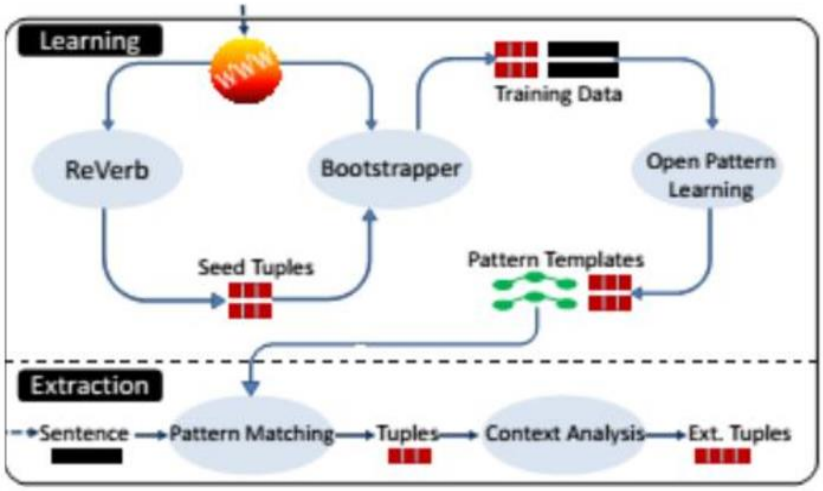
5.4 Information Extraction

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.



5.4.1. 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.

5.4.2 WordNet

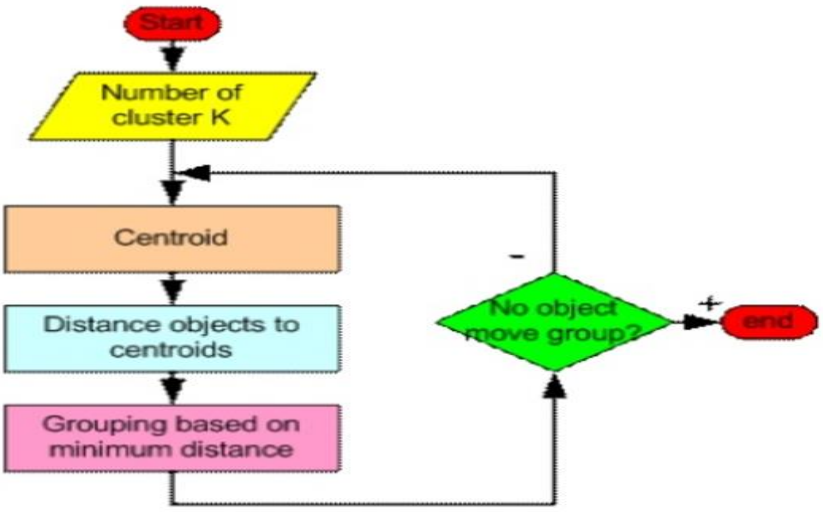
WordNet involves the generation of the synonym for a particular token of word. WordNet algorithm in analyze the data to extract the correct information though we use the synonym of the word. WordNet generate the synsets, which is the group of words with similar meaning.

5.5 Machine Learning

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

5.5.1 Clustering

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

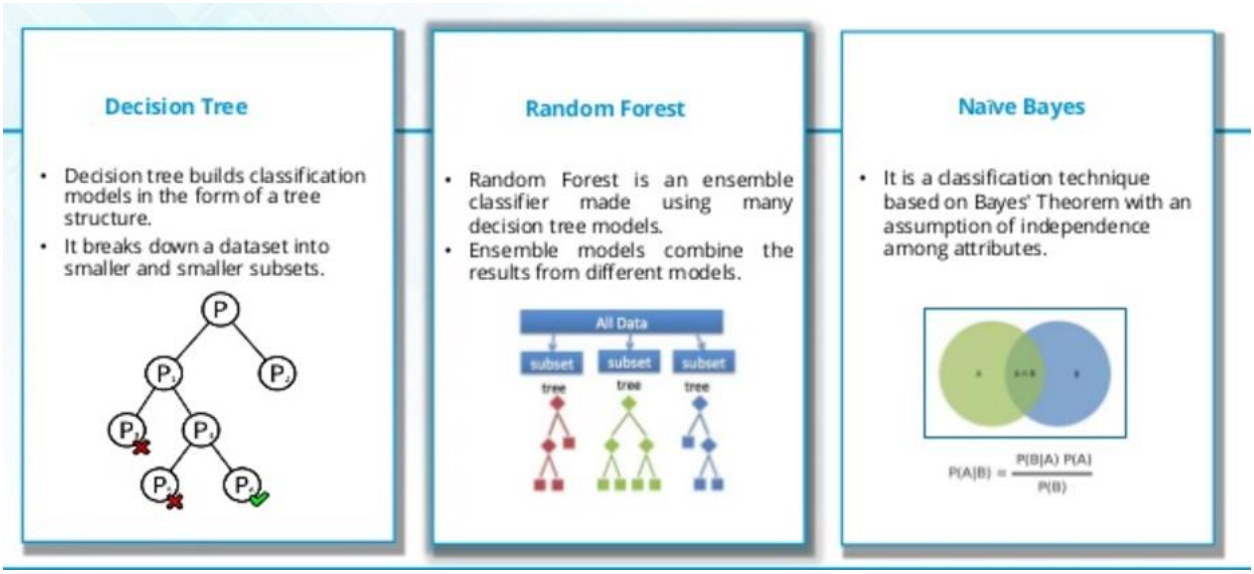


Steps involved in k-mean clustering:

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

5.5.2 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.



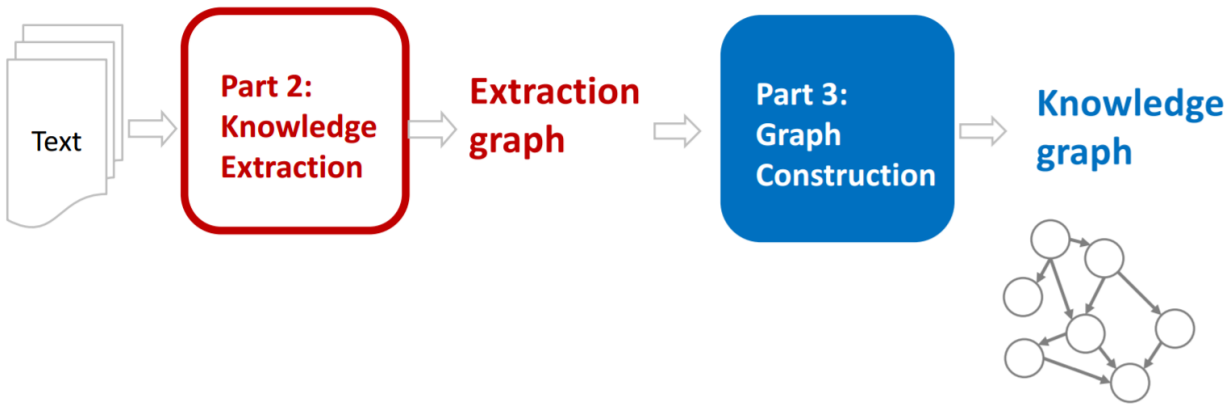
Steps involved in classification:

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

5.6 Knowledge Graph

Knowlwdge Graph is used to simplify the search results. This graph represent the graphical representation of the flow of the text data. The main advatange of using this knowledge graph is simplified diagrammatical representation of the huge data, helps in easy knowledge transfer and documenetaio easy.

5.6.1 Design workflow of knowledge Graph



Steps followed in desiging this knowledge graph:

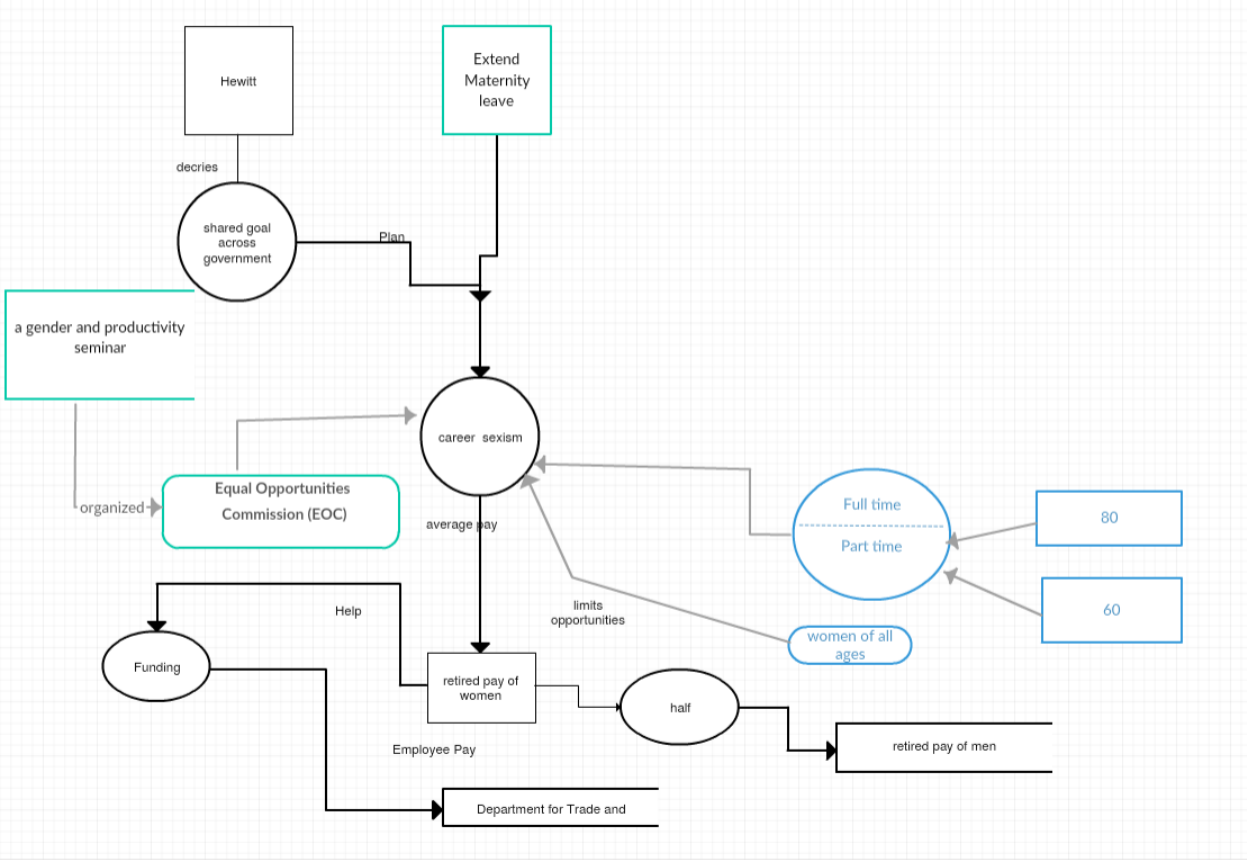
1. Recognizing the named entity reference including the people, organization, location, date etc.
2. Desiging the data schema i.e., finding the relationship between these entities.
3. Representing them in diagramical graph

5.6.2 Knowledge Graph for our dataset:

We do not have any specified rules for desiging this knowledge graph. Different companies have there own knowedge graph construction and follows there own rules.

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

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



5.7 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

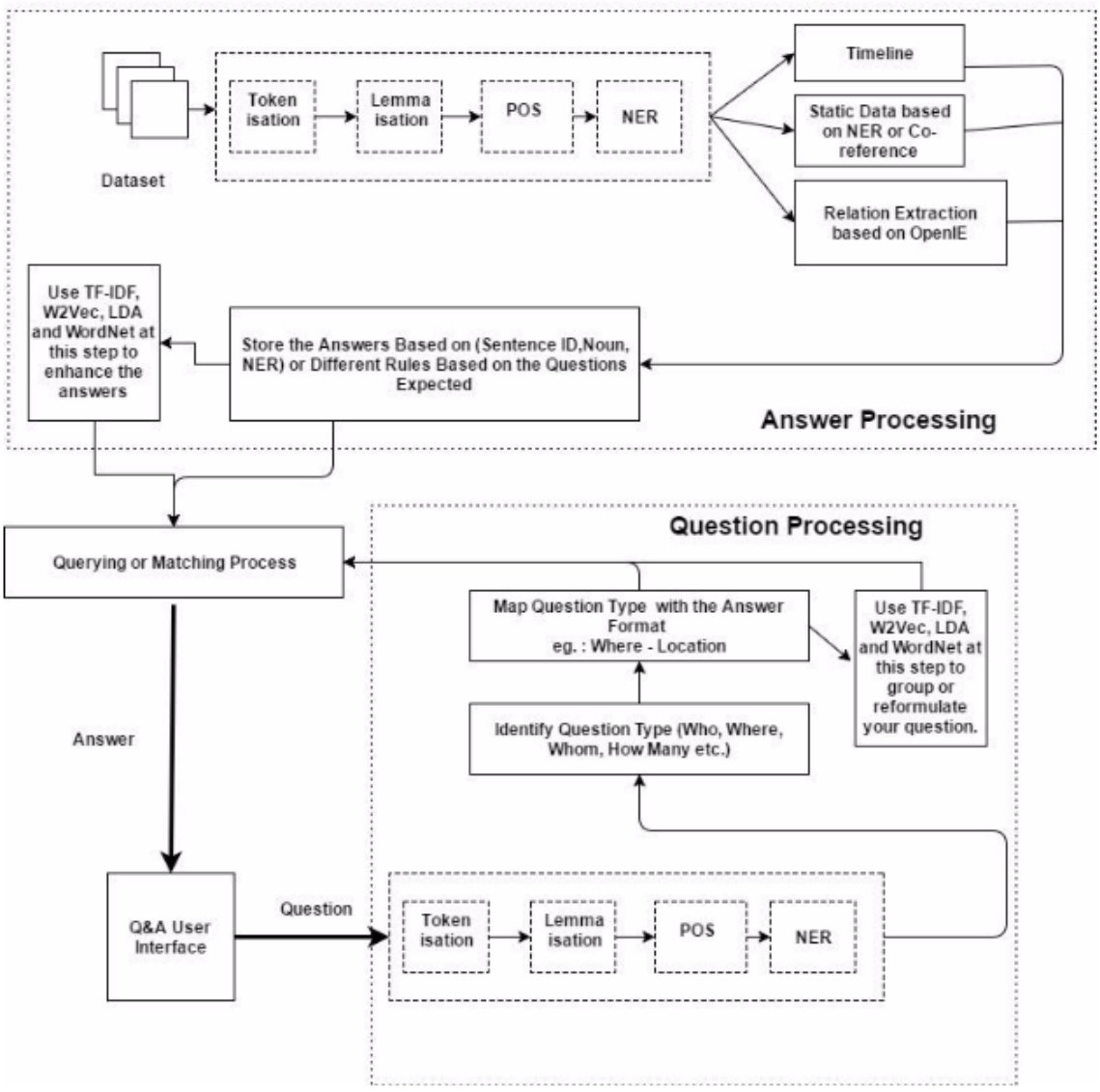
9. What is the average pay for part-time women.
   60p.

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

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 tha dataset and then we are storing the result of the NLP and on top of it applying the other shown approeches for the better performance.



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:

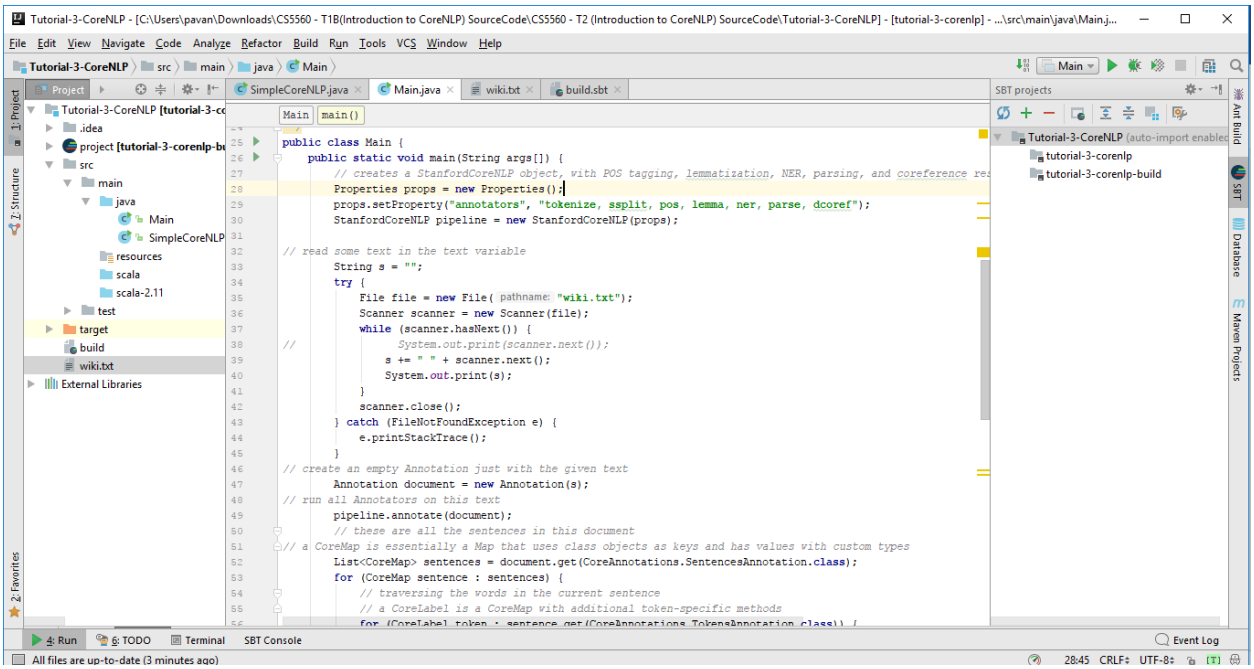
Lemmatization:

POS Tagging:

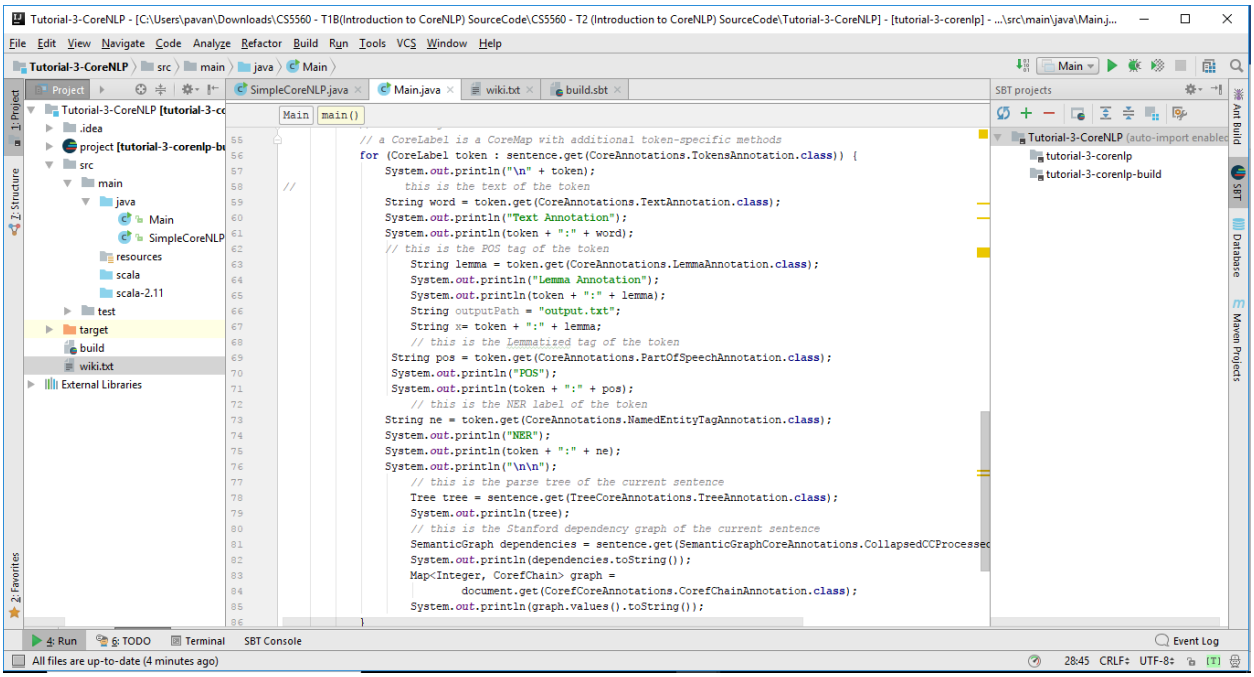
NER:

Coreference Resolution:

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

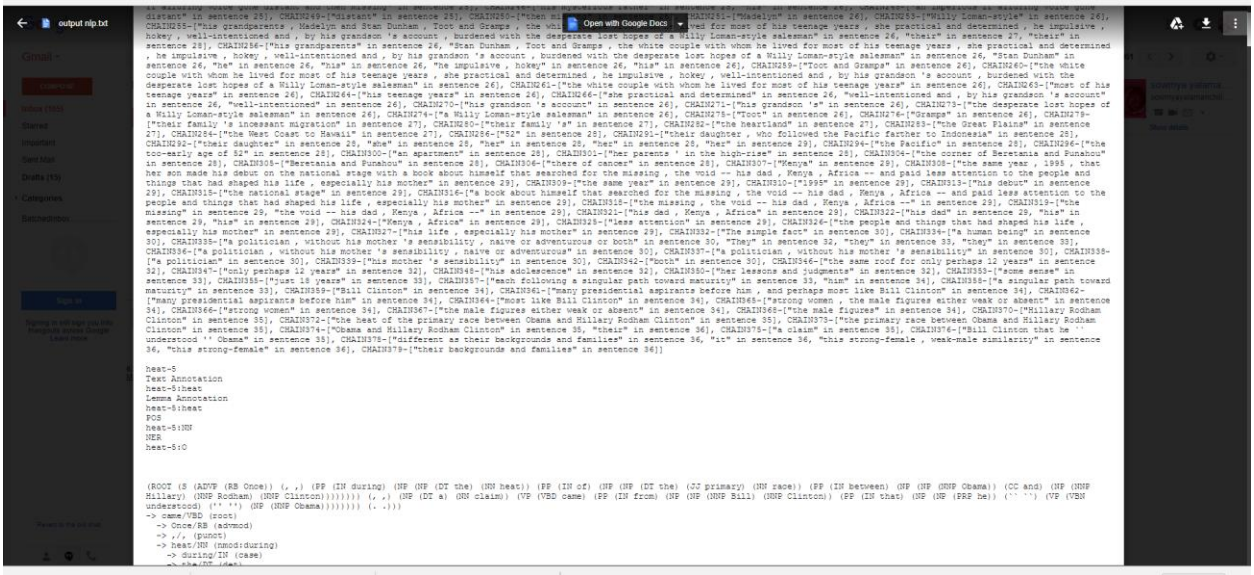


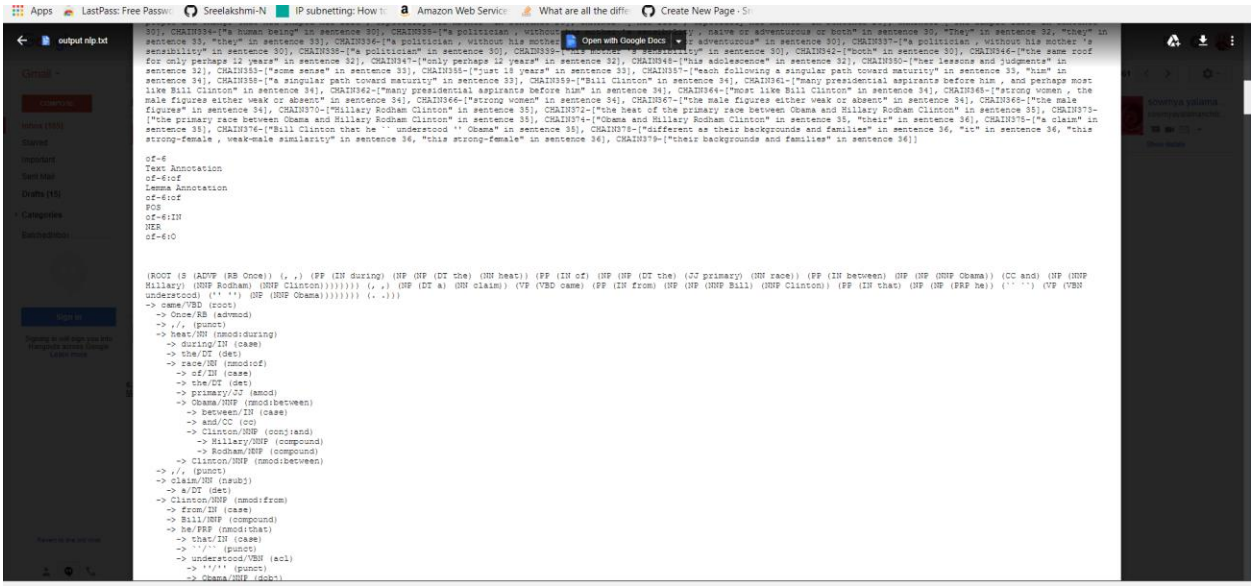
```
public class Main {
    public static void main(String args[]) {
        // creates a StanfordCoreNLP object, with POS tagging, lemmatization, NER, parsing, and coreference res
        Properties props = new Properties();
        props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");
        StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

// read some text in the text variable
        String s = "";
        try {
            File file = new File( pathname: "wiki.txt");
            Scanner scanner = new Scanner(file);
            while (scanner.hasNext()) {
//                System.out.print(scanner.next());
                s += " " + scanner.next();
                System.out.print(s);
            }
            scanner.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
// create an empty Annotation just with the given text
        Annotation document = new Annotation(s);
// run all Annotators on this text
        pipeline.annotate(document);
        // these are all the sentences in this document
// a CoreMap is essentially a Map that uses class objects as keys and has values with custom types
        List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);
        for (CoreMap sentence : sentences) {
            // traversing the words in the current sentence
            // a CoreLabel is a CoreMap with additional token-specific methods
            for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
```



```
            // a CoreLabel is a CoreMap with additional token-specific methods
            for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
                System.out.println("\n" + token);
//                this is the text of the token
                String word = token.get(CoreAnnotations.TextAnnotation.class);
                System.out.println("Text Annotation");
                System.out.println(token + ":" + word);
                // this is the POS tag of the token
                    String lemma = token.get(CoreAnnotations.LemmaAnnotation.class);
                    System.out.println("Lemma Annotation");
                    System.out.println(token + ":" + lemma);
                    String outputPath = "output.txt";
                    String x= token + ":" + lemma;
                    // this is the Lemmatized tag of the token
                String pos = token.get(CoreAnnotations.PartOfSpeechAnnotation.class);
                System.out.println("POS");
                System.out.println(token + ":" + pos);
                    // this is the NER label of the token
                String ne = token.get(CoreAnnotations.NamedEntityTagAnnotation.class);
                System.out.println("NER");
                System.out.println(token + ":" + ne);
                System.out.println("\n\n");
                    // this is the parse tree of the current sentence
                    Tree tree = sentence.get(TreeCoreAnnotations.TreeAnnotation.class);
                    System.out.println(tree);
                    // this is the Stanford dependency graph of the current sentence
                    SemanticGraph dependencies = sentence.get(SemanticGraphCoreAnnotations.CollapsedCCProcessed
                    System.out.println(dependencies.toString());
                    Map<Integer, CorefChain> graph =
                            document.get(CorefCoreAnnotations.CorefChainAnnotation.class);
                    System.out.println(graph.values().toString());
            }
```

Below are the outputs of the operations after applying the NLP.





output nlp - Notepad

File Edit Format View Help

Chicago" in sentence 18, "Chicago" in sentence 19, "it" in sentence 19, "Chicago" in sentence 21], CHAIN158-["Chicago , the antipode of remote Honolulu , deep in the fold of the mainland" :
, CHAIN185-["Hawaii and Chicago" in sentence 16, "the two main threads weaving through the cloth of Barack Obama 's life" in sentence 16], CHAIN187-["the cloth of Barack Obama 's life" in
ner conflicts" in sentence 19], CHAIN211-["the subtle , coolly ambitious persona" in sentence 19], CHAIN212-["the presidential election" in sentence 19], CHAIN214-["first" in sentence 20],
sentence 24, "It" in sentence 25, "It" in sentence 26, "It" in sentence 27, "that" in sentence 28], CHAIN232-["community work" in sentence 24], CHAIN233-["lives of public service" in sente
HAIN255-["his grandparents , Madelyn and Stan Dunham , Toot and Gramps , the white couple with whom he lived for most of his teenage years , she practical and determined , he impulsive , h
ve , hokey , well-intentioned and , by his grandson 's account , burdened with the desperate lost hopes of a Willy Loman-style salesman" in sentence 26], CHAIN261-["the white couple with wh
HAIN284-["the West Coast to Hawaii" in sentence 27], CHAIN286-["52" in sentence 28], CHAIN291-["their daughter , who followed the Pacific farther to Indonesia" in sentence 28], CHAIN292-["
in sentence 29], CHAIN310-["1995" in sentence 29], CHAIN313-["his debut" in sentence 29], CHAIN315-["the national stage" in sentence 29], CHAIN316-["a book about himself that searched for
a politician , without his mother 's sensibility , naive or adventurous or both" in sentence 30, "They" in sentence 32, "they" in sentence 33, "they" in sentence 33], CHAIN336-["a politici
re him , and perhaps most like Bill Clinton" in sentence 34], CHAIN362-["many presidential aspirants before him" in sentence 34], CHAIN364-["most like Bill Clinton" in sentence 34], CHAIN3
ale" in sentence 36], CHAIN379-["their backgrounds and families" in sentence 36]]

heat-5
Text Annotation
heat-5:heat
Lemma Annotation
heat-5:heat
POS
heat-5:NN
NER
heat-5:O

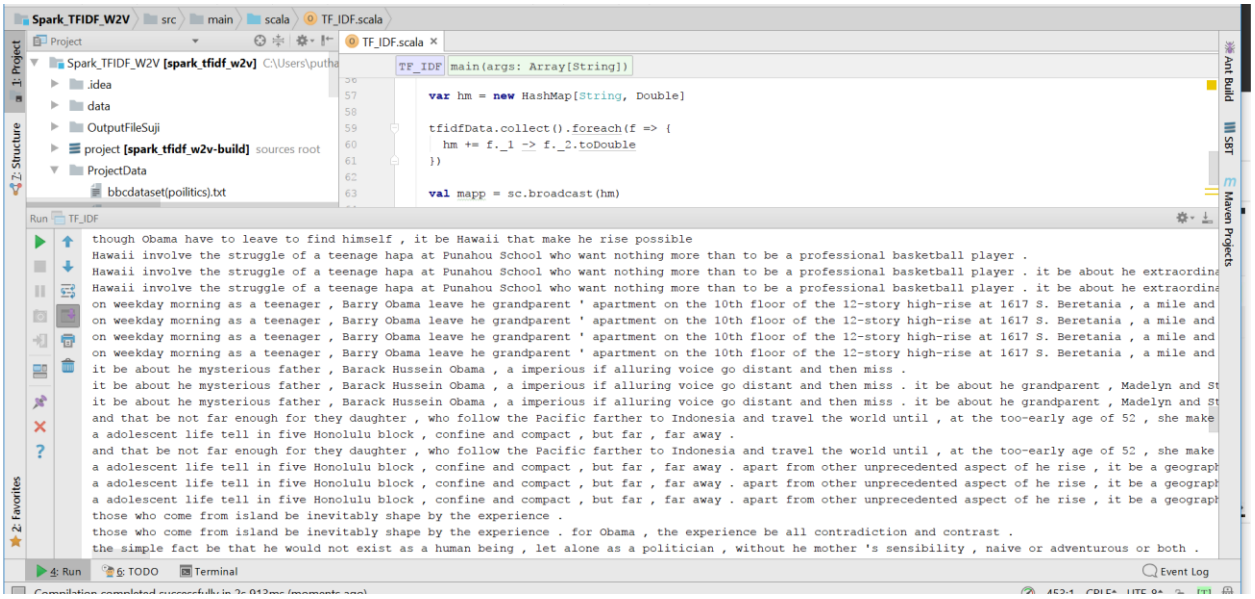
(ROOT (S (ADVP (RB Once)) (, ,) (PP (IN during) (NP (NP (DT the) (NN heat)) (PP (IN of) (NP (NP (DT the) (JJ primary) (NN race)) (PP (IN between) (NP (NP (NNP Obama)) (CC and) (NP (NNP Hil
-> came/VBD (root)
-> Once/RB (advmod)
-> ,/, (punct)
-> heat/NN (nmod:during)
-> during/IN (case)
-> the/DT (det)
-> race/NN (nmod:of)
-> of/IN (case)
-> the/DT (det)
-> primary/JJ (amod)
-> Obama/NNP (nmod:between)
-> between/IN (case)
-> and/CC (cc)
-> Clinton/NNP (conj:and)
-> Hillary/NNP (compound)
-> Rodham/NNP (compound)
-> Clinton/NNP (nmod:between)
-> ,/, (punct)

Type here to search

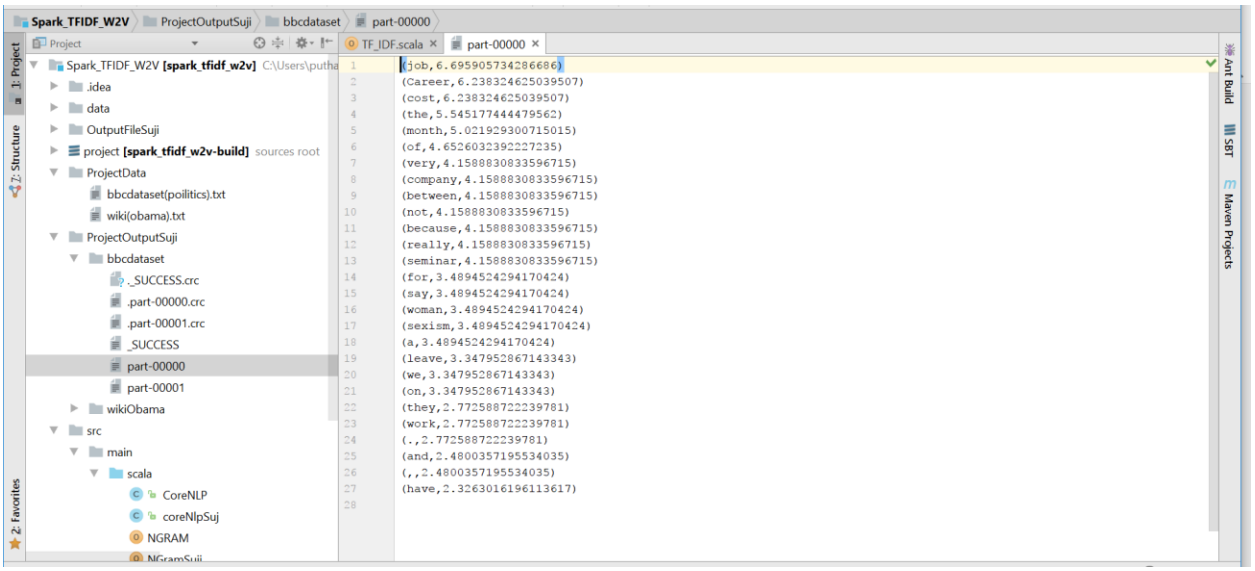
11:14 PM
6/23/2017

6.b TF-IDF for our dataset

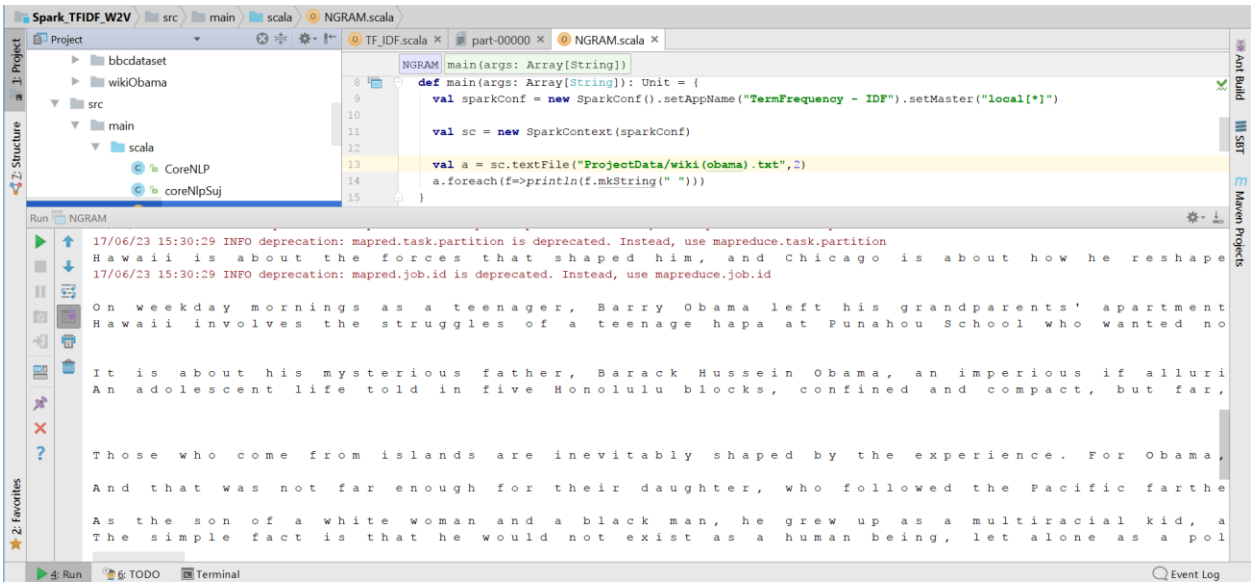
Generating the term frequency for the words in the dataset.



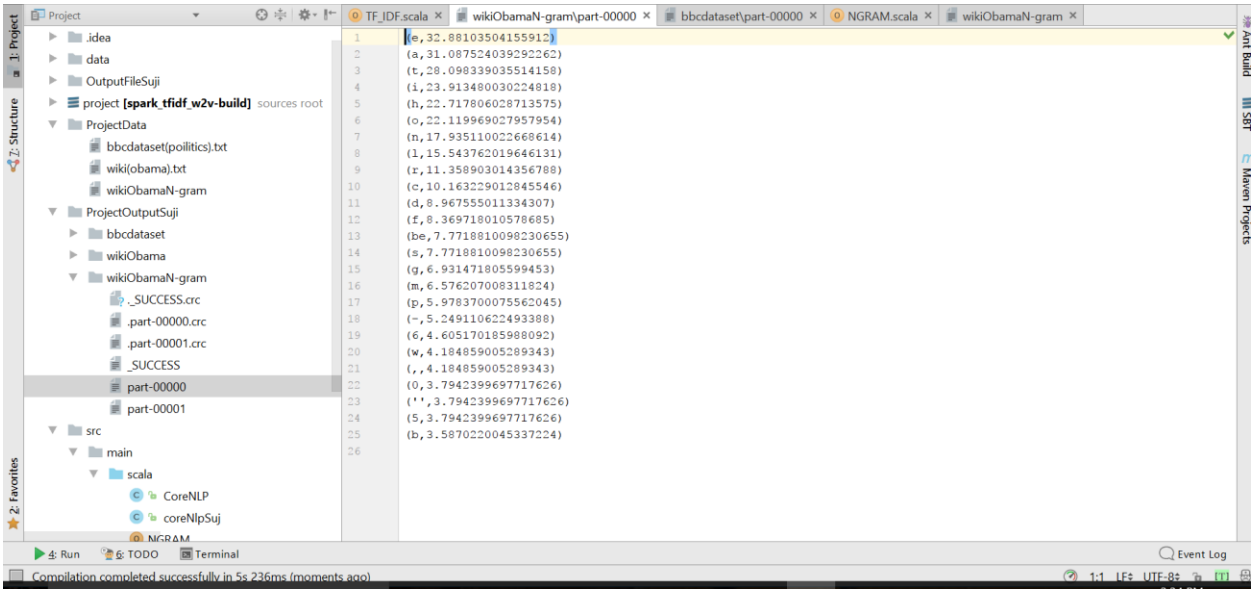
Generated Output.



Generation N-gram for the dataset

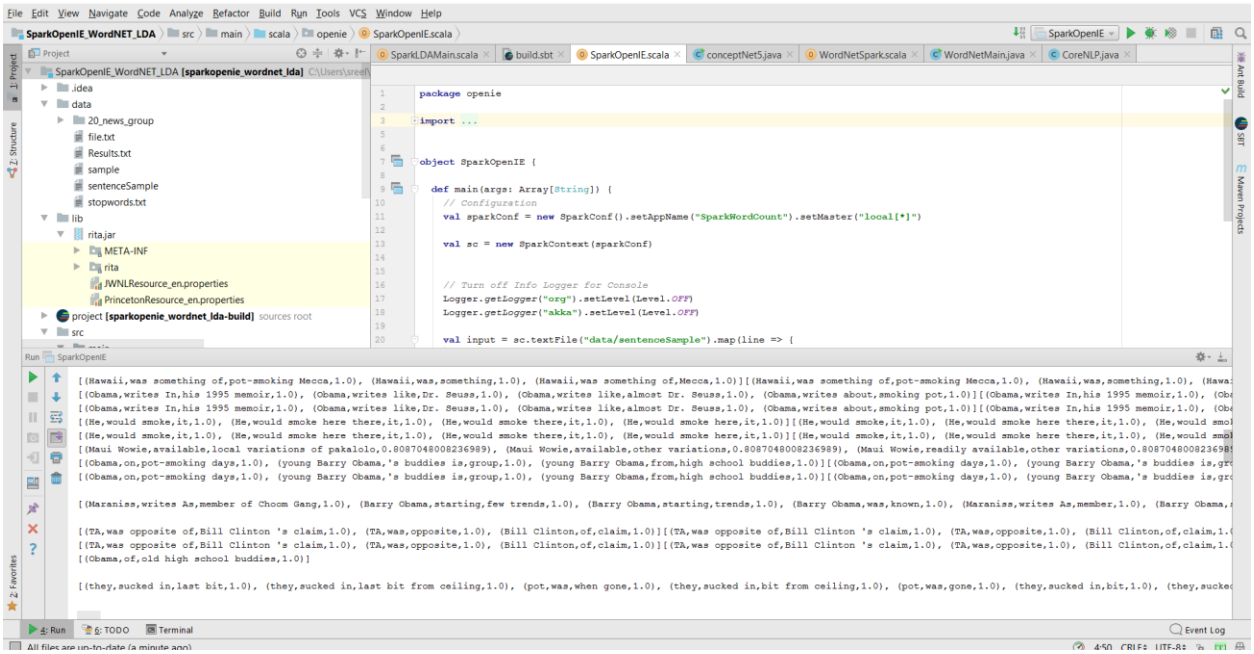


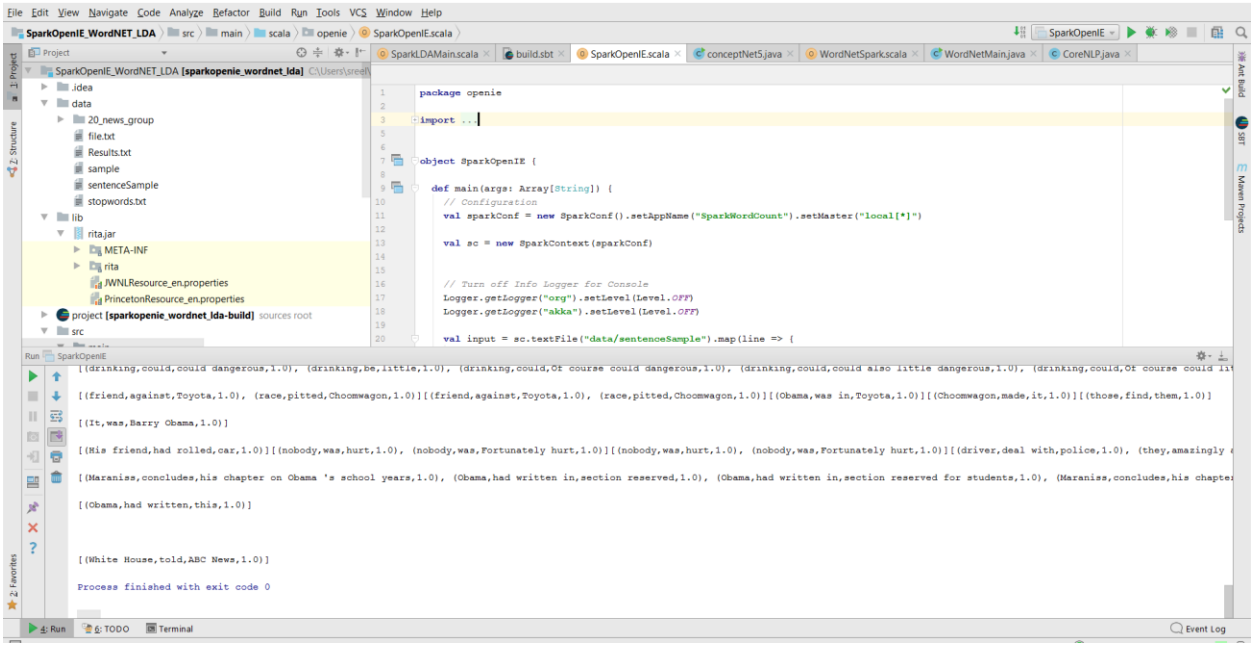
Generating the TF_IDF for N-gram output.

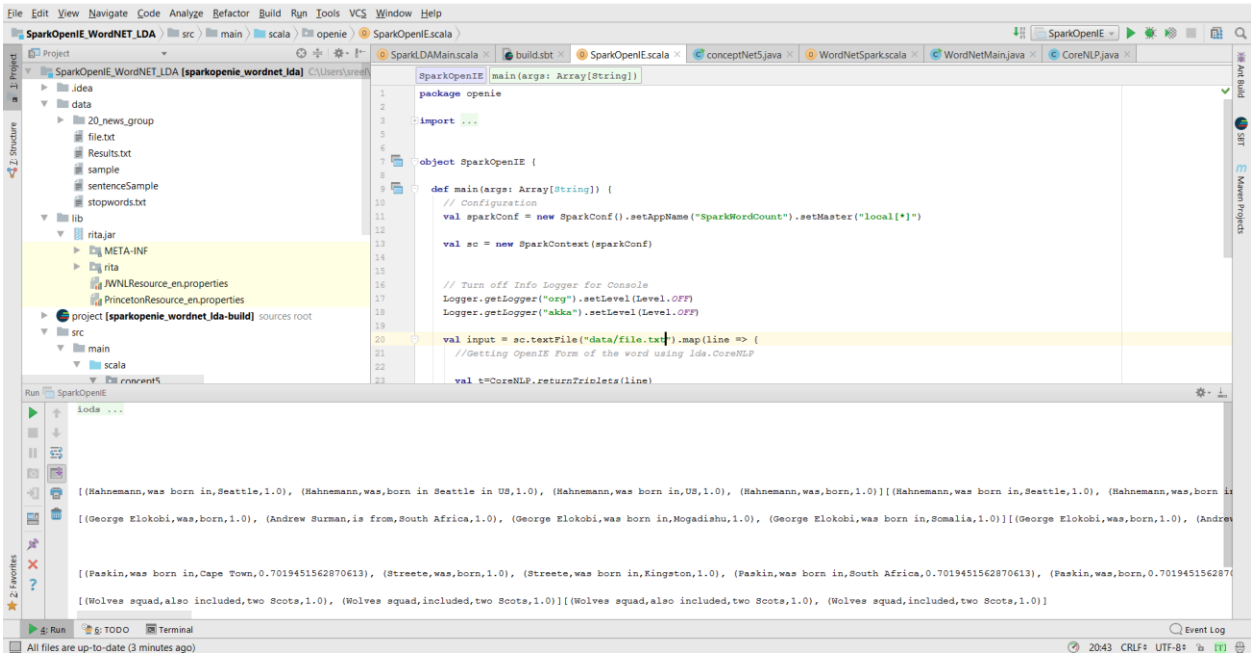


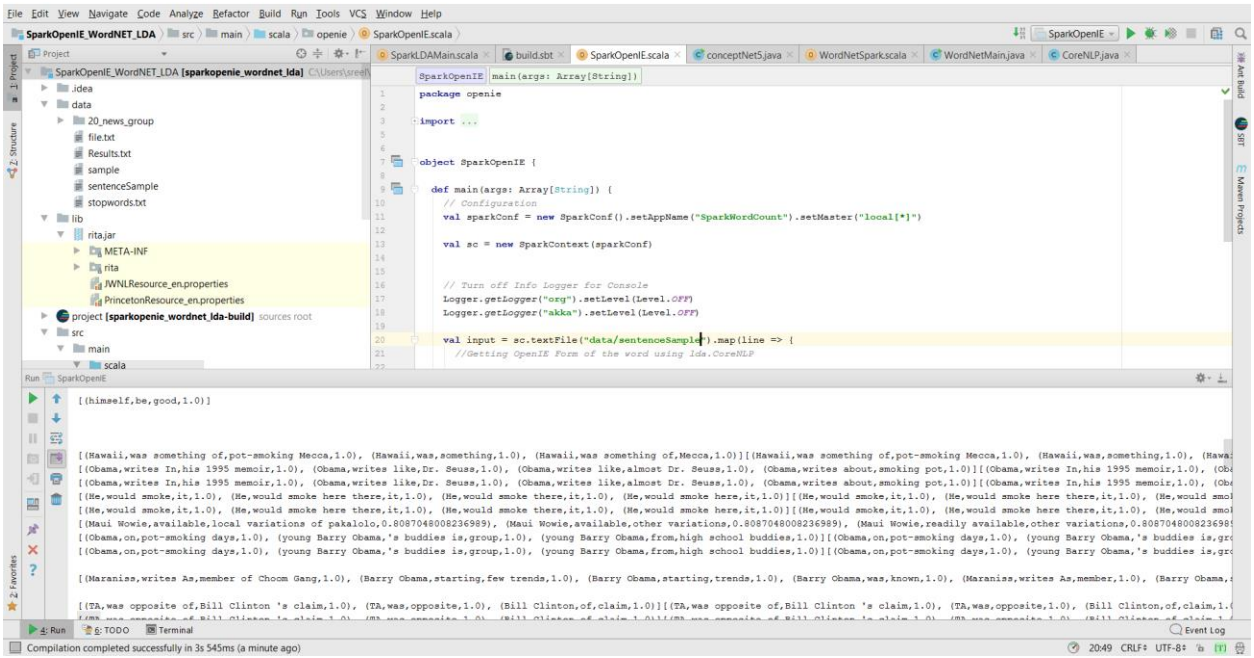
6.c Information Extraction

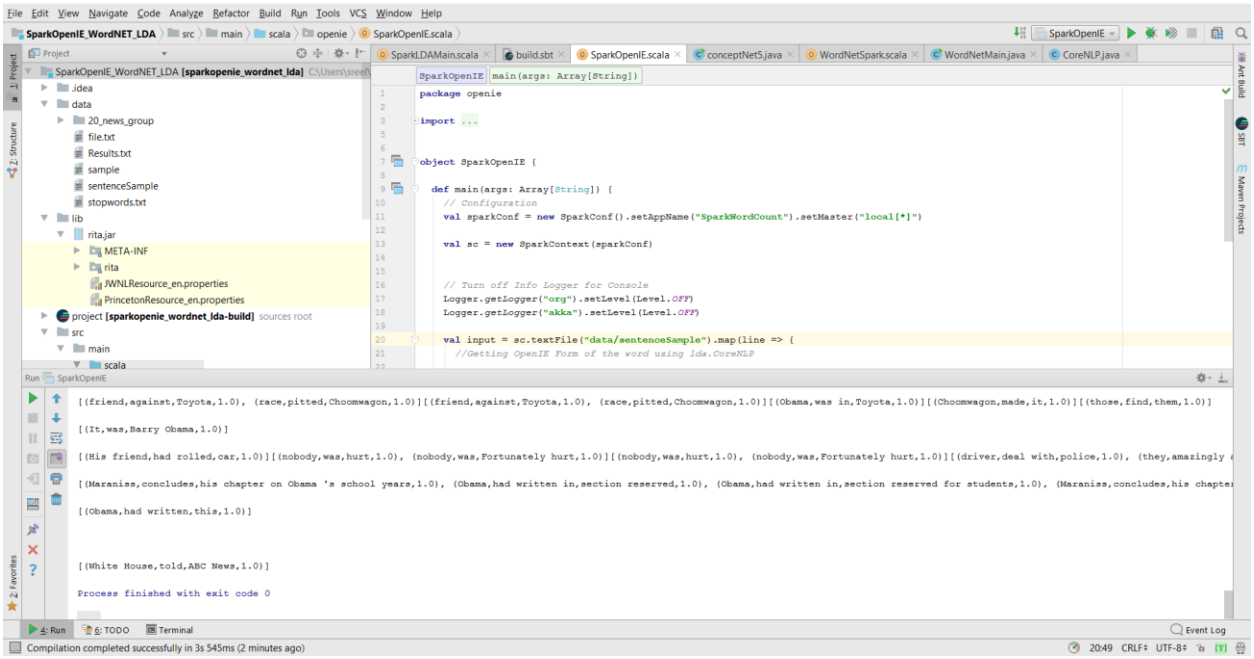
OpenIE for our dataset:

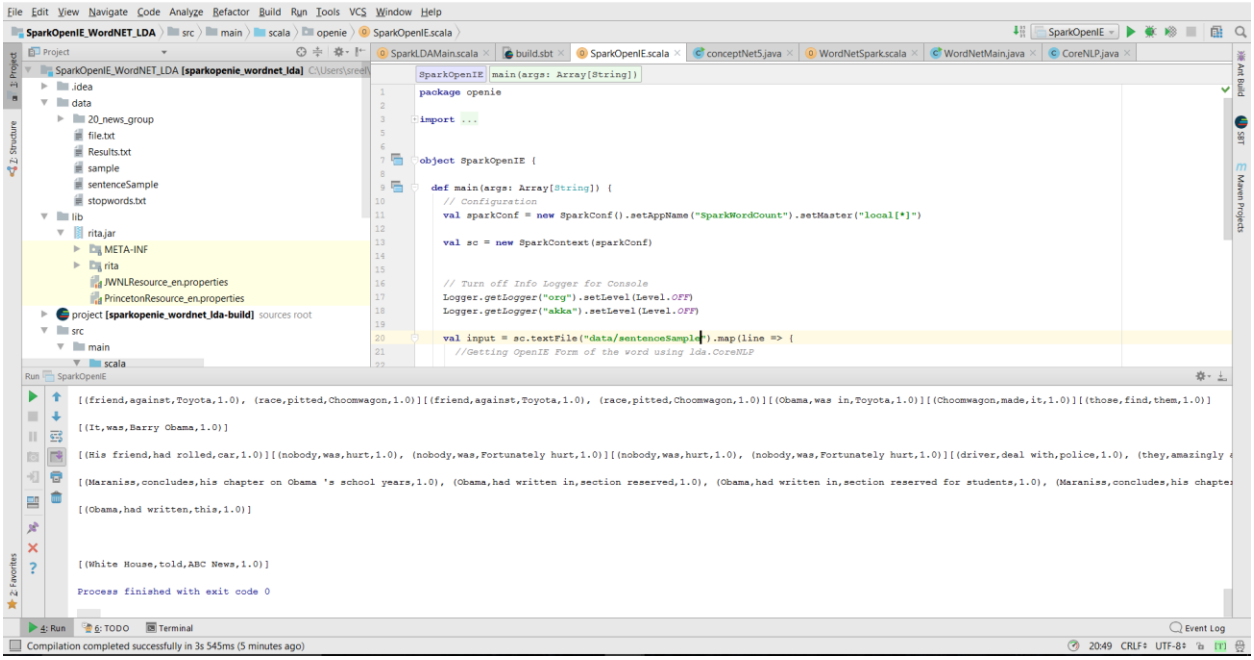




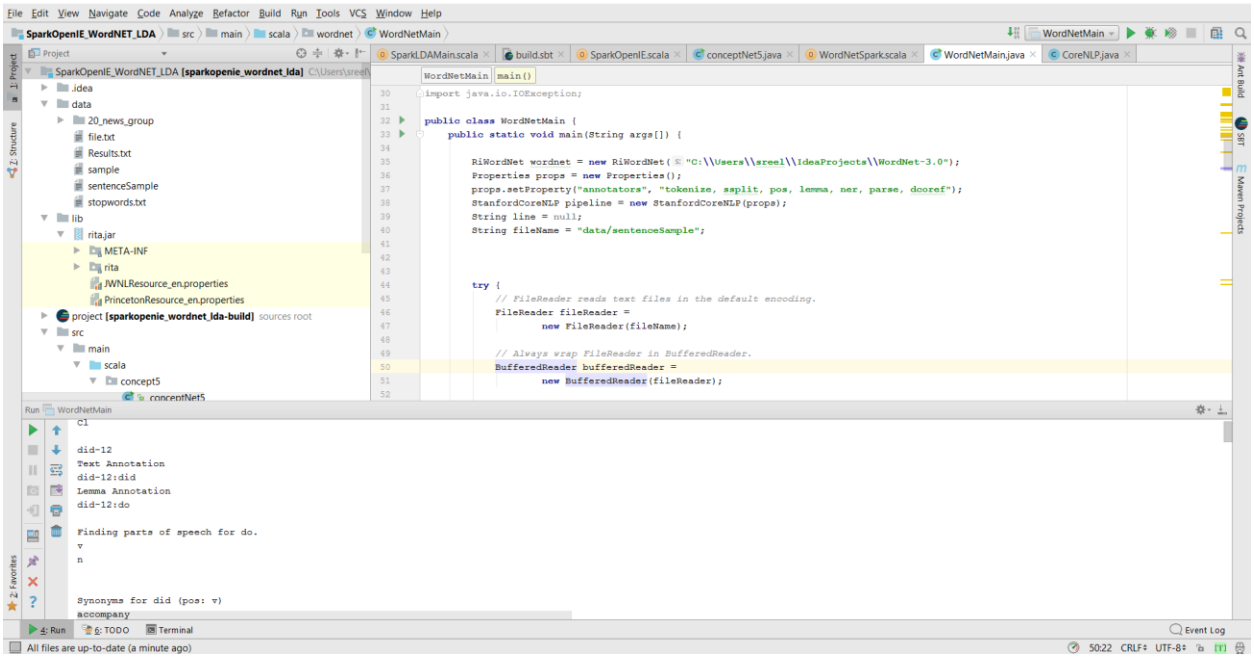

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA src main scala openie SparkOpenIE.scala
SparkOpenIE
SparkLDAMain.scala build.sbt SparkOpenIE.scala conceptNet5.java WordNetSpark.scala WordNetMain.java CoreNLP.java
SparkOpenIE main(args: Array[String])
Project
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda] C:\Users\sreeh
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
package openie
import ...
object SparkOpenIE {
def main(args: Array[String]) {
// Configuration
val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
val sc = new SparkContext(sparkConf)
// Turn off Info Logger for Console
Logger.getLogger("org").setLevel(Level.OFF)
Logger.getLogger("akka").setLevel(Level.OFF)
val input = sc.textFile("data/file.txt").map(line => {
//Getting OpenIE Form of the word using lda.CoreNLP
Run SparkOpenIE
4: Run 6: TODO Terminal Event Log
All files are up-to-date (3 minutes ago)
20:43 CRLF UTF-8

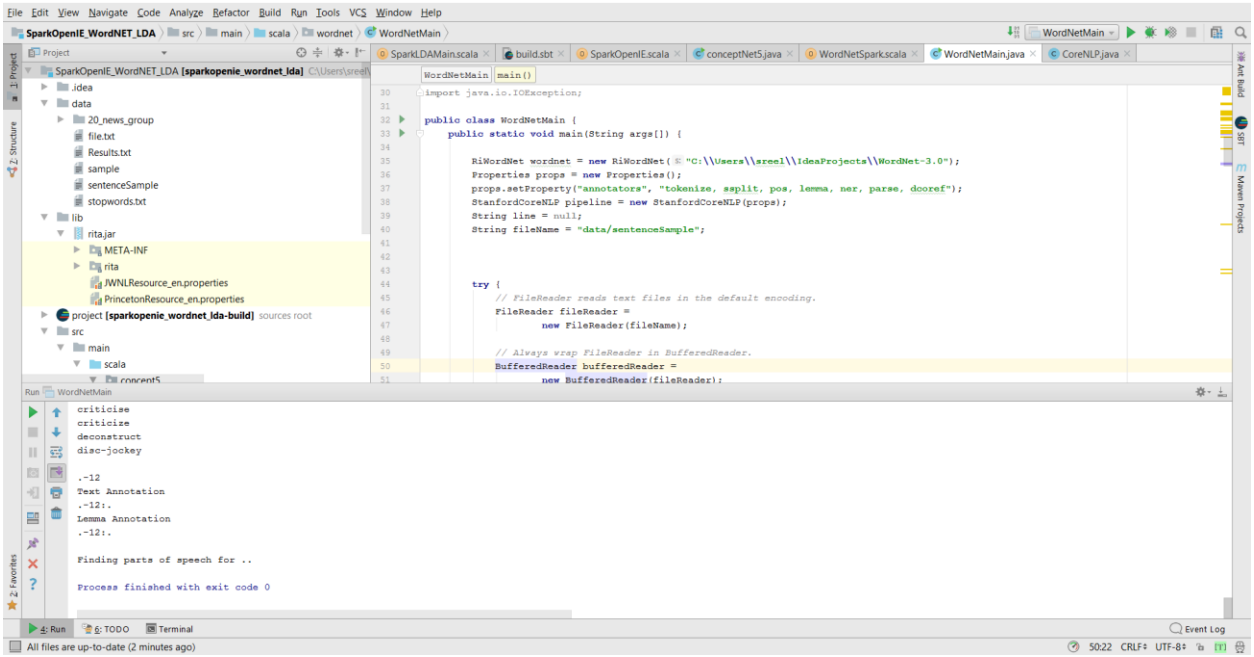

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA src main scala openie SparkOpenIE.scala
SparkOpenIE
SparkLDAMain.scala build.sbt SparkOpenIE.scala conceptNet5.java WordNetSpark.scala WordNetMain.java CoreNLP.java
SparkOpenIE main(args: Array[String])
Project
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda] C:\Users\sreeh
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
package openie
import ...
object SparkOpenIE {
def main(args: Array[String]) {
// Configuration
val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
val sc = new SparkContext(sparkConf)
// Turn off Info Logger for Console
Logger.getLogger("org").setLevel(Level.OFF)
Logger.getLogger("akka").setLevel(Level.OFF)
val input = sc.textFile("data/sentenceSample").map(line => {
//Getting OpenIE Form of the word using lda.CoreNLP
Run SparkOpenIE
4: Run 6: TODO Terminal Event Log
Compilation completed successfully in 3s 545ms (a minute ago)
20:49 CRLF UTF-8

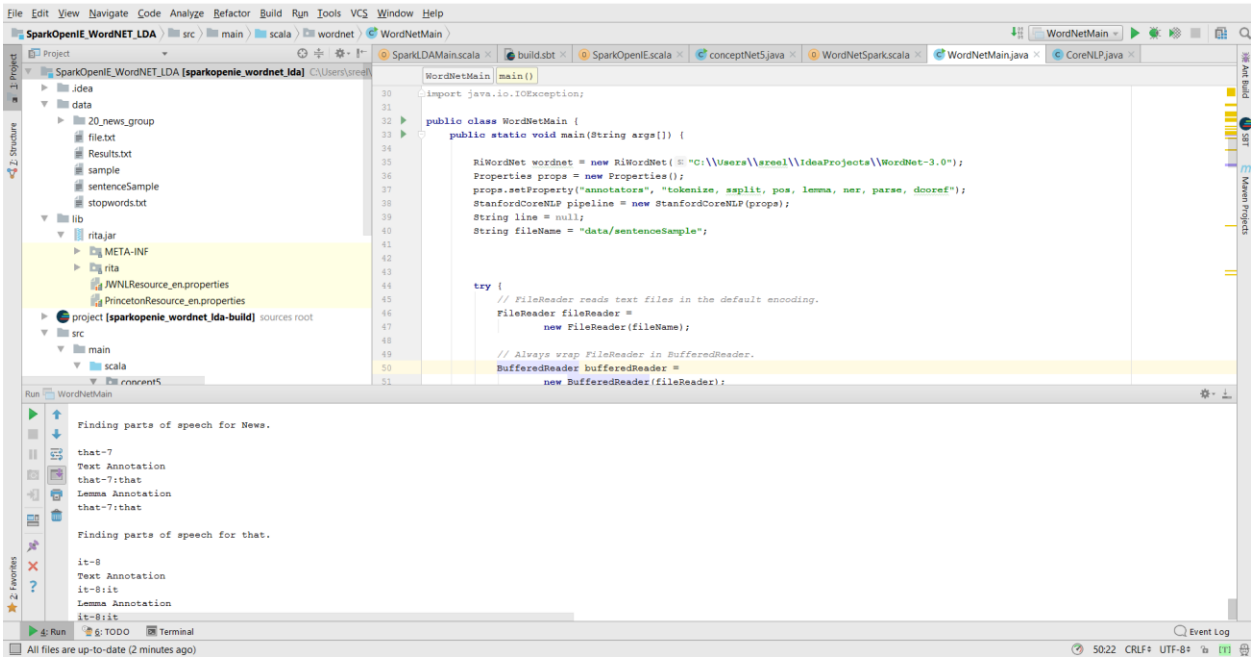

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA
SparkOpenIE.scala
SparkLDAMain.scala
build.sbt
SparkOpenIE.scala
conceptNet5.java
WordNetSpark.scala
WordNetMain.java
CoreNLP.java
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda]
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
SparkOpenIE
main(args: Array[String])
package openie
import ...
object SparkOpenIE {
def main(args: Array[String]) {
// Configuration
val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
val sc = new SparkContext(sparkConf)
// Turn off Info Logger for Console
Logger.getLogger("org").setLevel(Level.OFF)
Logger.getLogger("akka").setLevel(Level.OFF)
val input = sc.textFile("data/sentenceSample").map(line => {
//Getting OpenIE Form of the word using lda.CoreNLP
[(friend,against,Toyota,1.0), (race,pitted,Choomwagon,1.0)][(friend,against,Toyota,1.0), (race,pitted,Choomwagon,1.0)][(Obama,was in,Toyota,1.0)][(Choomwagon,made,it,1.0)][(those,find,them,1.0)]
[(It,was,Barry Obama,1.0)]
[(His friend,had rolled,car,1.0)][(nobody,was,hurt,1.0), (nobody,was,Fortunately hurt,1.0)][(nobody,was,hurt,1.0), (nobody,was,Fortunately hurt,1.0)][(driver,deal with,police,1.0), (they,amazingly
[(Maraniss,concludes,his chapter on Obama 's school years,1.0), (Obama,had written in,section reserved,1.0), (Obama,had written in,section reserved for students,1.0), (Maraniss,concludes,his chapte
[(Obama,had written,this,1.0)]
[(White House,told,ABC News,1.0)]
Process finished with exit code 0
4: Run
6: TODO
Terminal
Event Log
Compilation completed successfully in 3s 545ms (2 minutes ago)
20:49 CRLF UTF-8

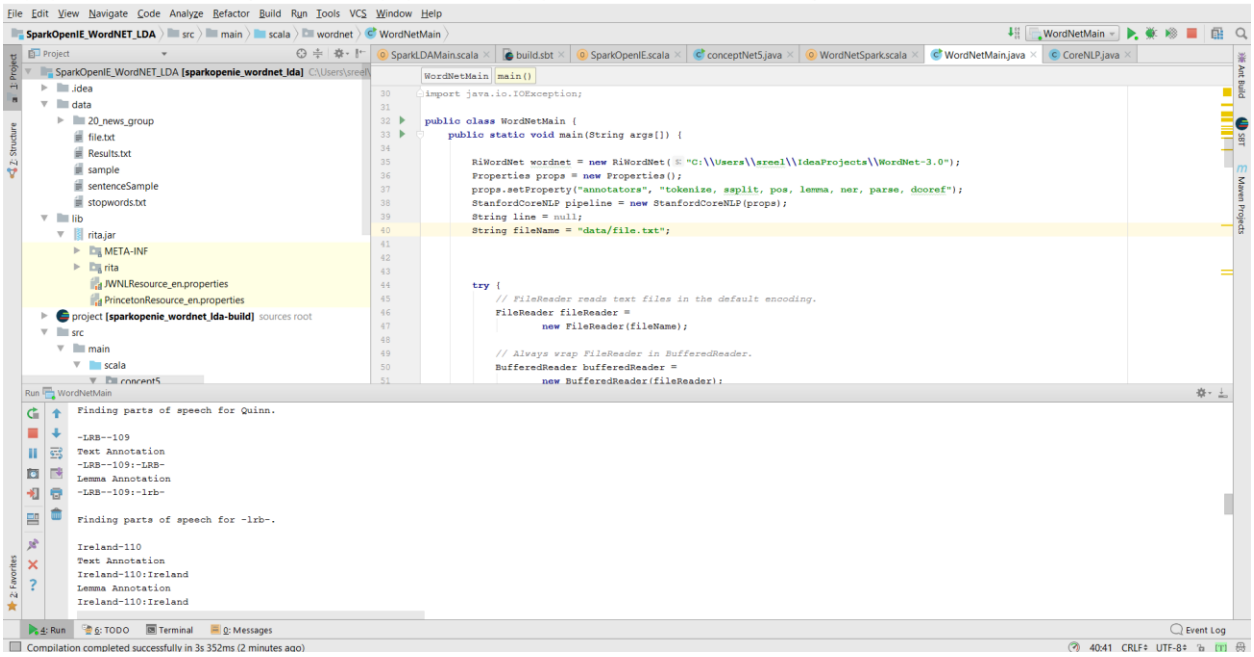

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA
SparkOpenIE.scala
SparkLDAMain.scala
build.sbt
SparkOpenIE.scala
conceptNet5.java
WordNetSpark.scala
WordNetMain.java
CoreNLP.java
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda]
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
SparkOpenIE
main(args: Array[String])
package openie
import ...
object SparkOpenIE {
def main(args: Array[String]) {
// Configuration
val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
val sc = new SparkContext(sparkConf)
// Turn off Info Logger for Console
Logger.getLogger("org").setLevel(Level.OFF)
Logger.getLogger("akka").setLevel(Level.OFF)
val input = sc.textFile("data/sentenceSample").map(line => {
//Getting OpenIE Form of the word using lda.CoreNLP
[(friend,against,Toyota,1.0), (race,pitted,Choomwagon,1.0)][(friend,against,Toyota,1.0), (race,pitted,Choomwagon,1.0)][(Obama,was in,Toyota,1.0)][(Choomwagon,made,it,1.0)][(those,find,them,1.0)]
[(It,was,Barry Obama,1.0)]
[(His friend,had rolled,car,1.0)][(nobody,was,hurt,1.0), (nobody,was,Fortunately hurt,1.0)][(nobody,was,hurt,1.0), (nobody,was,Fortunately hurt,1.0)][(driver,deal with,police,1.0), (they,amazingly
[(Maraniss,concludes,his chapter on Obama 's school years,1.0), (Obama,had written in,section reserved,1.0), (Obama,had written in,section reserved for students,1.0), (Maraniss,concludes,his chapte
[(Obama,had written,this,1.0)]
[(White House,told,ABC News,1.0)]
Process finished with exit code 0
4: Run
6: TODO
Terminal
Event Log
Compilation completed successfully in 3s 545ms (5 minutes ago)
20:49 CRLF UTF-8

Wordnet for our dataset



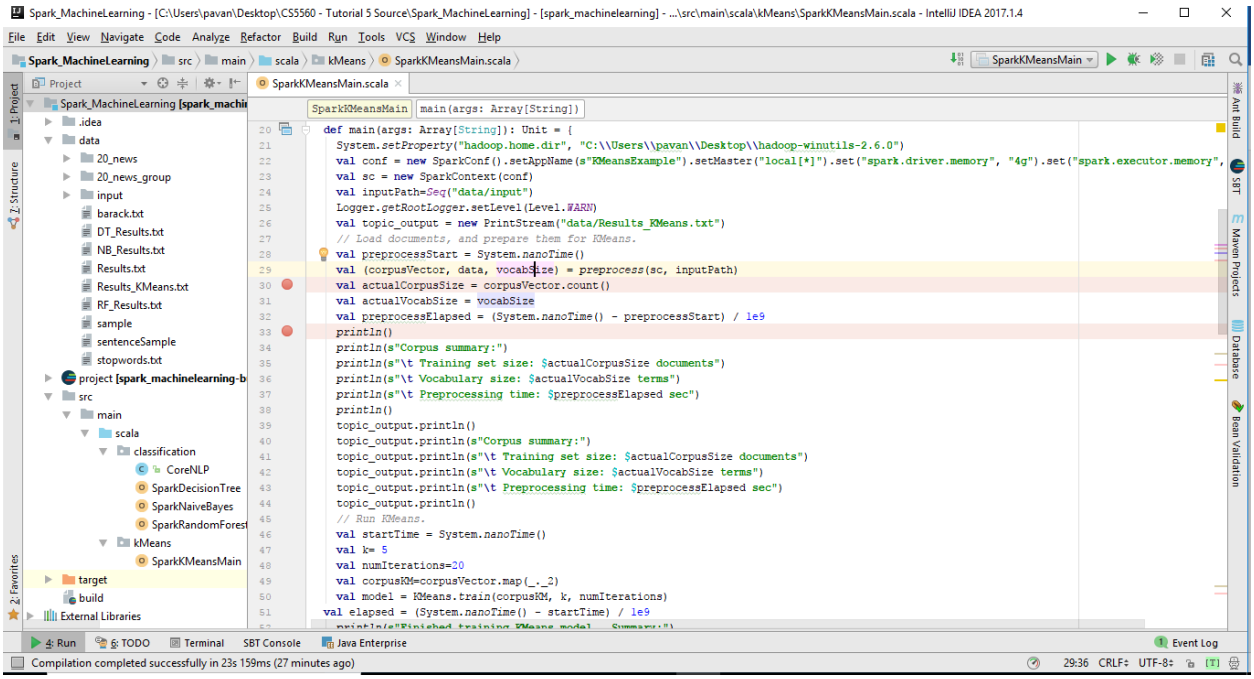


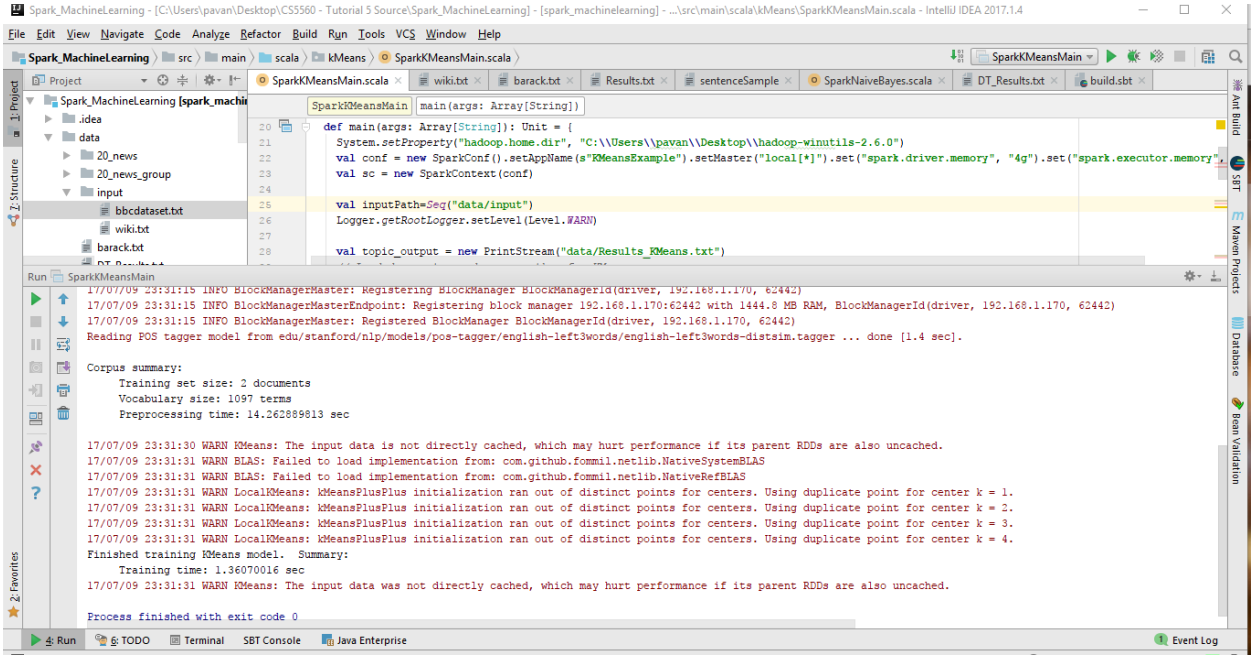

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA src main scala wordnet WordNetMain
WordNetMain
Project
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda] C:\Users\sreel\
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
concept5
SparkLDAMain.scala
build.sbt
SparkOpenIE.scala
conceptNet5.java
WordNetSpark.scala
WordNetMain.java
CoreNLP.java
WordNetMain
main()
import java.io.IOException;
public class WordNetMain {
public static void main(String args[]) {
RiWordNet wordnet = new RiWordNet("C:\\Users\\sreel\\IdeaProjects\\WordNet-3.0");
Properties props = new Properties();
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
String line = null;
String fileName = "data/sentenceSample";
try {
// FileReader reads text files in the default encoding.
FileReader fileReader =
new FileReader(fileName);
// Always wrap FileReader in BufferedReader.
BufferedReader bufferedReader =
new BufferedReader(fileReader);
Run WordNetMain
Finding parts of speech for News.
that-7
Text Annotation
that-7:that
Lemma Annotation
that-7:that
Finding parts of speech for that.
it-8
Text Annotation
it-8:it
Lemma Annotation
it-8:it
4: Run
6: TODO
Terminal
Event Log
All files are up-to-date (2 minutes ago)
50:22 CRLF UTF-8

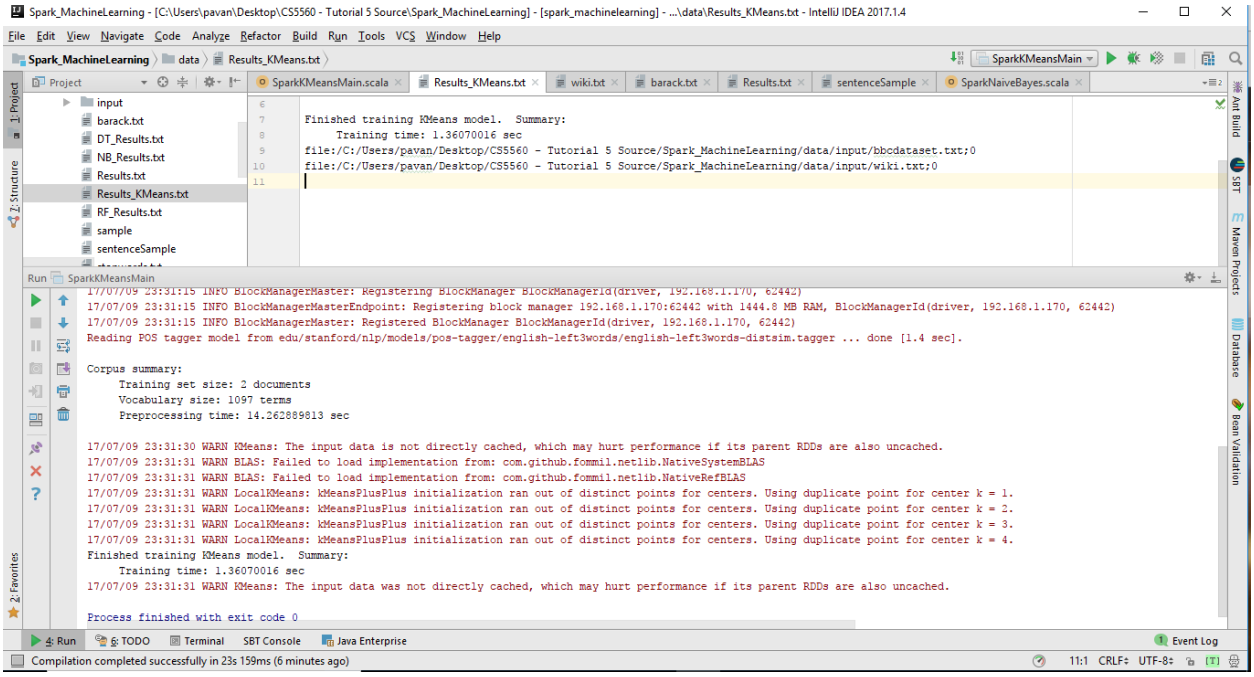

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkOpenIE_WordNET_LDA src main scala wordnet WordNetMain
WordNetMain
Project
SparkOpenIE_WordNET_LDA [sparkopenie_wordnet_lda] C:\Users\sreel\
.idea
data
20_news_group
file.txt
Results.txt
sample
sentenceSample
stopwords.txt
lib
rita.jar
META-INF
rita
JWNLResource_en.properties
PrincetonResource_en.properties
project [sparkopenie_wordnet_lda-build] sources root
src
main
scala
concept5
SparkLDAMain.scala
build.sbt
SparkOpenIE.scala
conceptNet5.java
WordNetSpark.scala
WordNetMain.java
CoreNLP.java
WordNetMain
main()
import java.io.IOException;
public class WordNetMain {
public static void main(String args[]) {
RiWordNet wordnet = new RiWordNet("C:\\Users\\sreel\\IdeaProjects\\WordNet-3.0");
Properties props = new Properties();
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
String line = null;
String fileName = "data/file.txt";
try {
// FileReader reads text files in the default encoding.
FileReader fileReader =
new FileReader(fileName);
// Always wrap FileReader in BufferedReader.
BufferedReader bufferedReader =
new BufferedReader(fileReader);
Run WordNetMain
Finding parts of speech for Quinn.
-LRB--109
Text Annotation
-LRB--109:-LRB-
Lemma Annotation
-LRB--109:-lrb-
Finding parts of speech for -lrb-.
Ireland-110
Text Annotation
Ireland-110:Ireland
Lemma Annotation
Ireland-110:Ireland
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 3s 352ms (2 minutes ago)
40:41 CRLF UTF-8

6.d Machine learning Techinques

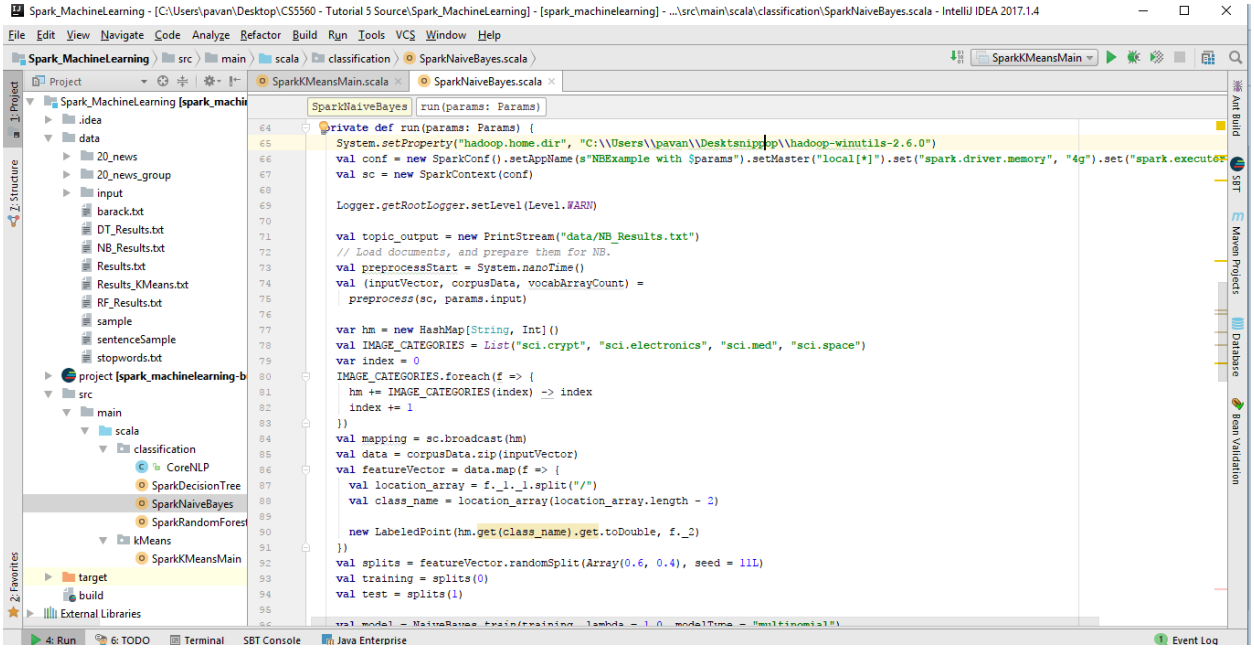
KMeans







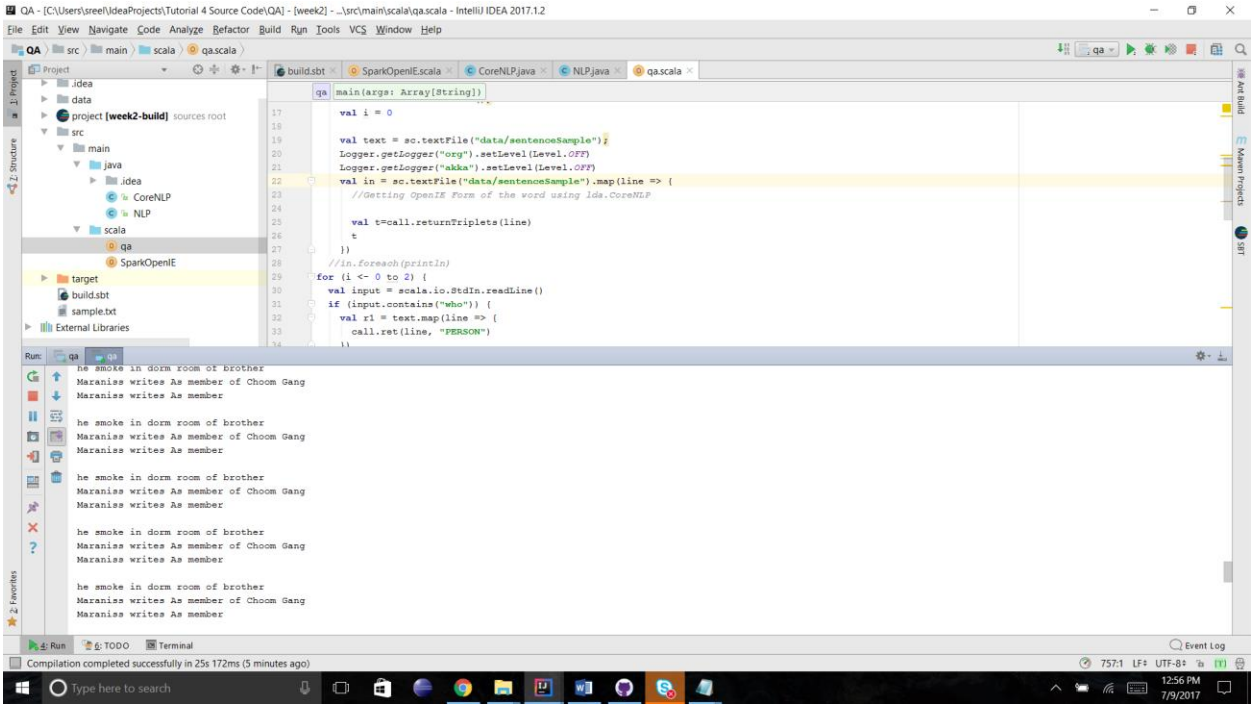
NaiveBias

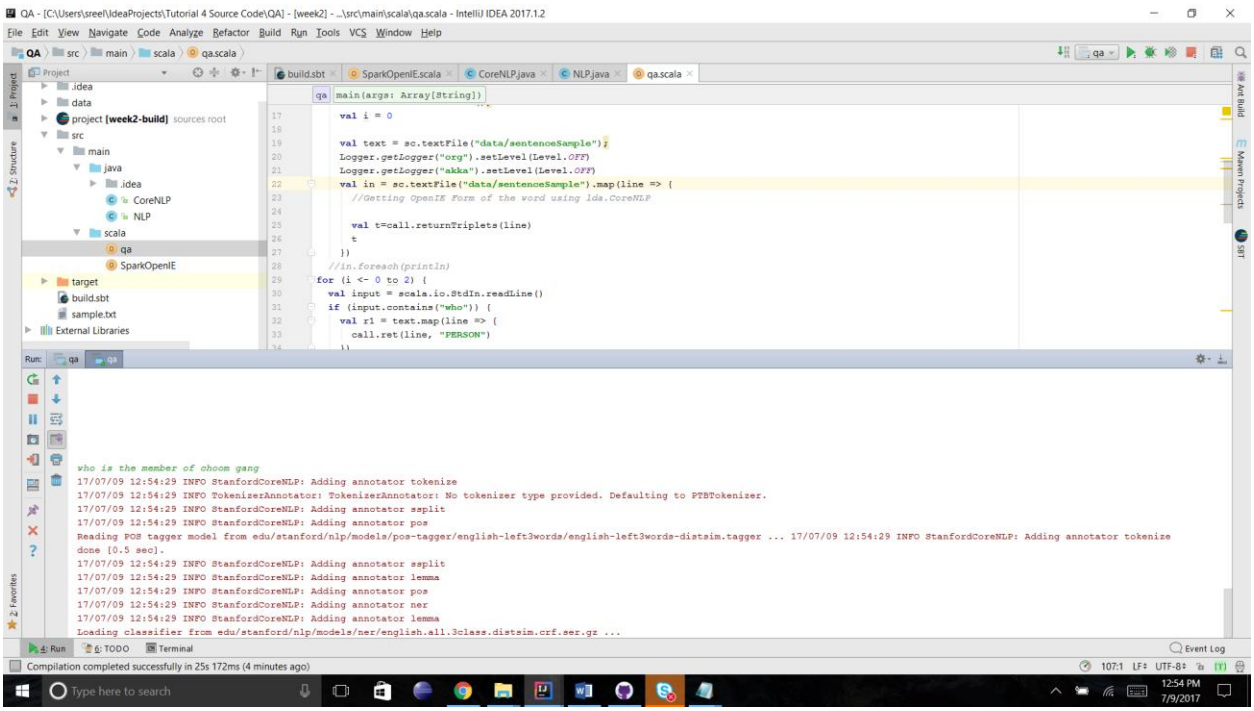


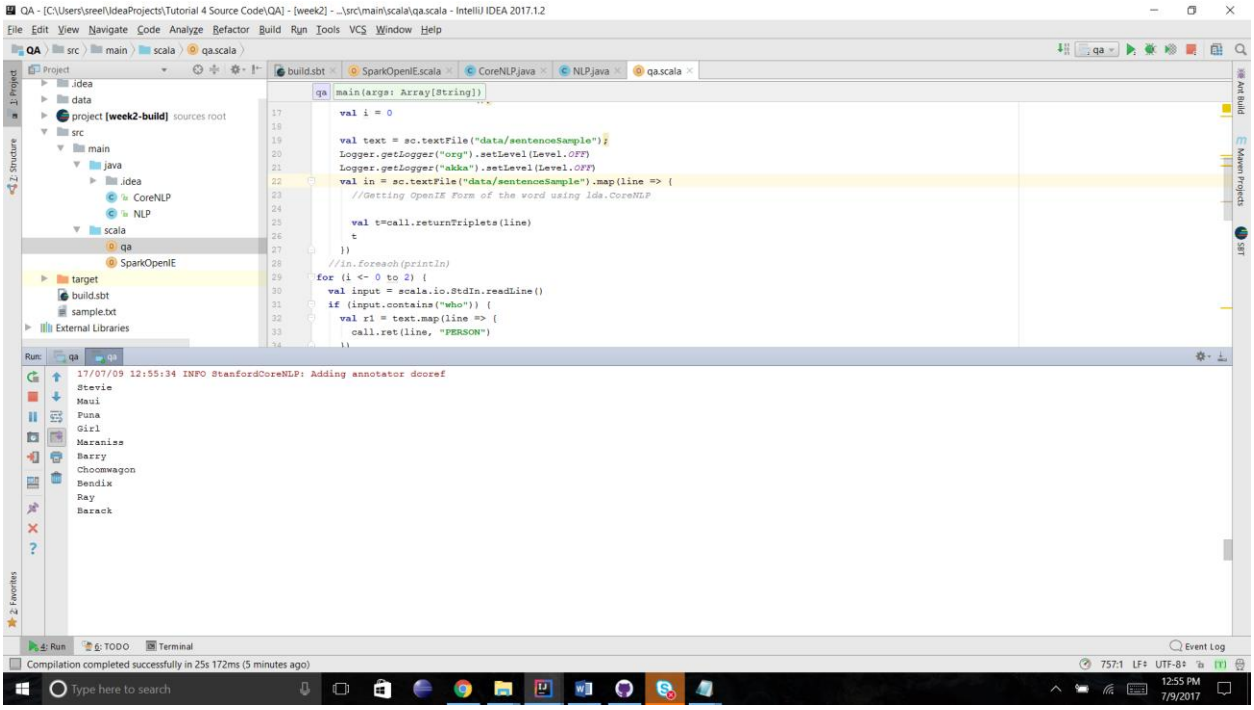
6.e Question Answering for our dataset

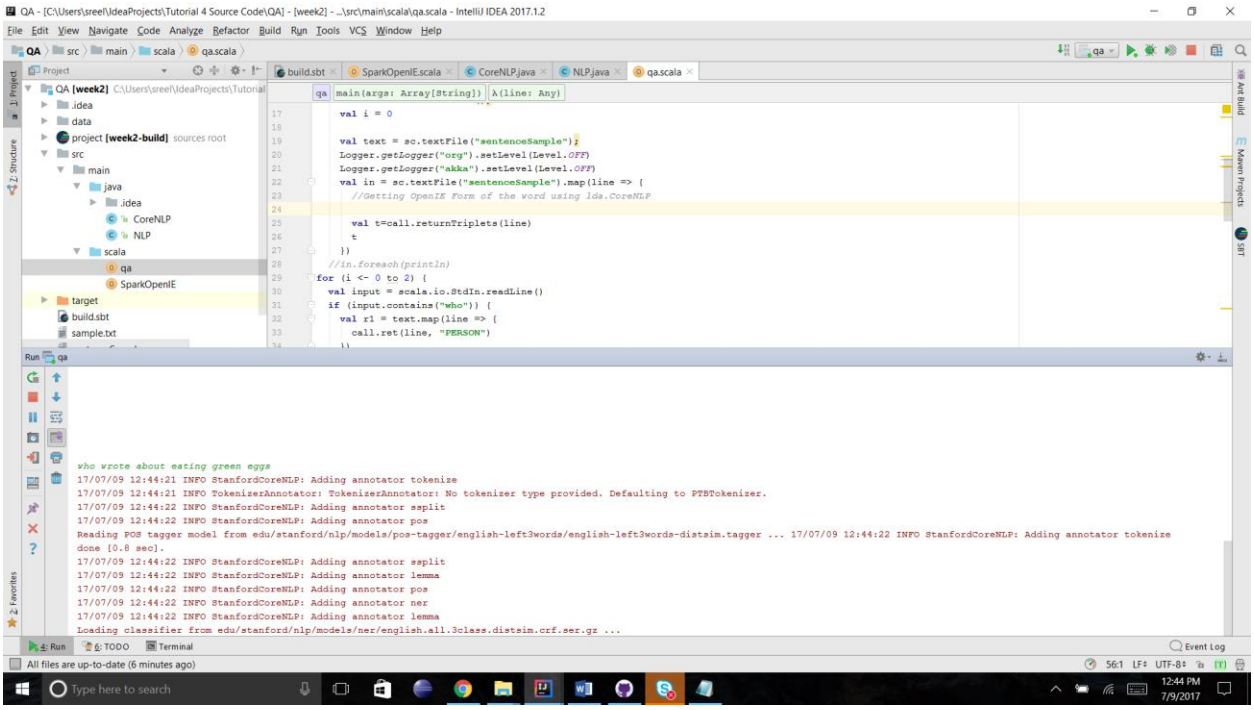
After performing the NLP operations, we have taken the post processed dataset and we have separately stored the result of NER output like from the NER result we have the all person

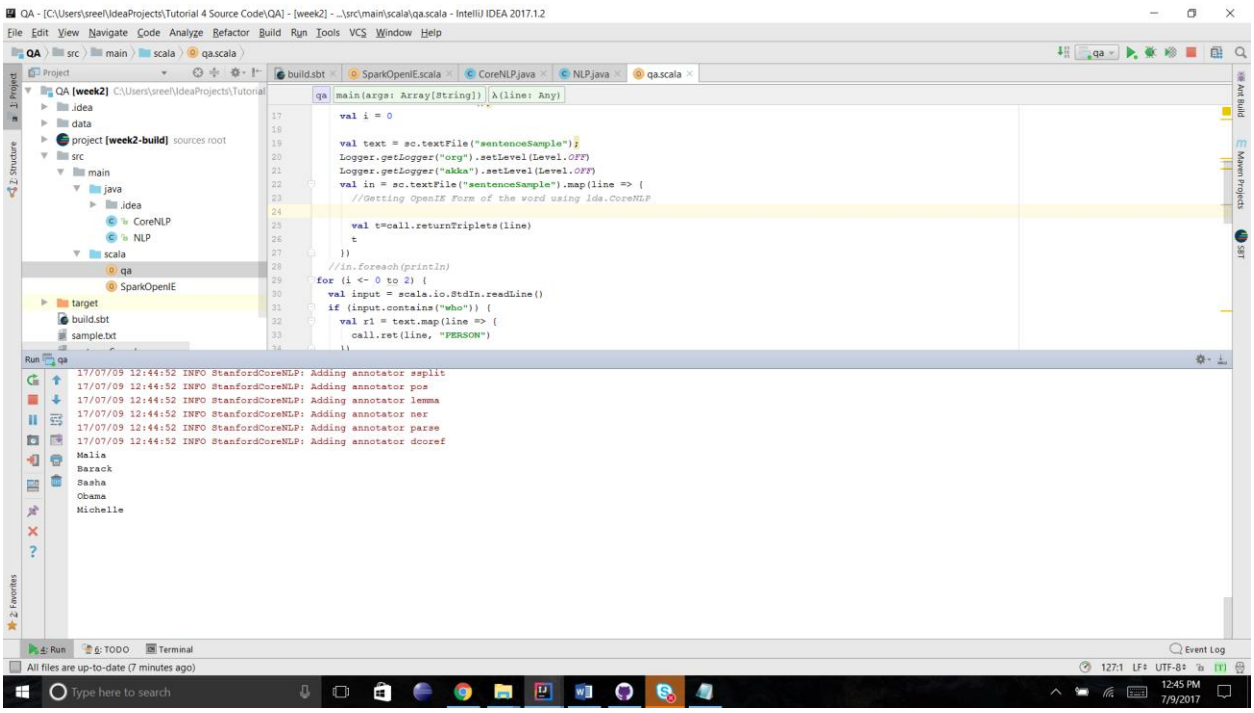
related entities to one file and similarly we have done for every group and based on that we have generated answers for the questions we choose. The below screenshots will depict the same.






QA - [C:\Users\sreel\IdeaProjects\Tutorial 4 Source Code\QA] - [week2] - ...\src\main\scala\qa.scala - IntelliJ IDEA 2017.1.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
build.sbt SparkOpenIE.scala CoreNLP.java NLP.java qa.scala
qa main(args: Array[String])
val i = 0
val text = sc.textFile("data/sentenceSample")
Logger.getLogger("org").setLevel(Level.OFF)
Logger.getLogger("akka").setLevel(Level.OFF)
val in = sc.textFile("data/sentenceSample").map(line => {
//Getting OpenIE Form of the word using lda.CoreNLP
val t=call.returnTriplets(line)
t
})
//in.foreach(println)
for (i <- 0 to 2) {
val input = scala.io.StdIn.readLine()
if (input.contains("who")) {
val r1 = text.map(line => {
call.ret(line, "PERSON")
17/07/09 12:55:34 INFO StanfordCoreNLP: Adding annotator dcoref
Stevie
Maui
Puna
Girl
Maraniss
Barry
Choomwagon
Bendix
Ray
Barack
Compilation completed successfully in 25s 172ms (5 minutes ago)
12:55 PM
7/9/2017


QA - [C:\Users\sreel\IdeaProjects\Tutorial 4 Source Code\QA] - [week2] - ...\src\main\scala\qa.scala - IntelliJ IDEA 2017.1.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
build.sbt SparkOpenIE.scala CoreNLP.java NLP.java qa.scala
qa main(args: Array[String]) λ(line: Any)
who wrote about eating green eggs
17/07/09 12:44:21 INFO StanfordCoreNLP: Adding annotator tokenize
17/07/09 12:44:21 INFO TokenizerAnnotator: TokenizerAnnotator: No tokenizer type provided. Defaulting to PTBTokenizer.
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator ssplit
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator pos
Reading POS tagger model from edu/stanford/nlp/models/pos-tagger/english-left3words/english-left3words-distsim.tagger ... 17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator tokenize
done [0.8 sec].
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator ssplit
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator lemma
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator pos
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator ner
17/07/09 12:44:22 INFO StanfordCoreNLP: Adding annotator lemma
Loading classifier from edu/stanford/nlp/models/ner/english.all.3class.distsim.crf.ser.gz ...
All files are up-to-date (6 minutes ago)
12:44 PM
7/9/2017



7. 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 comprehensinve integereated enevironment.

- IntelliJ
- PyCharm

7.a Contributors

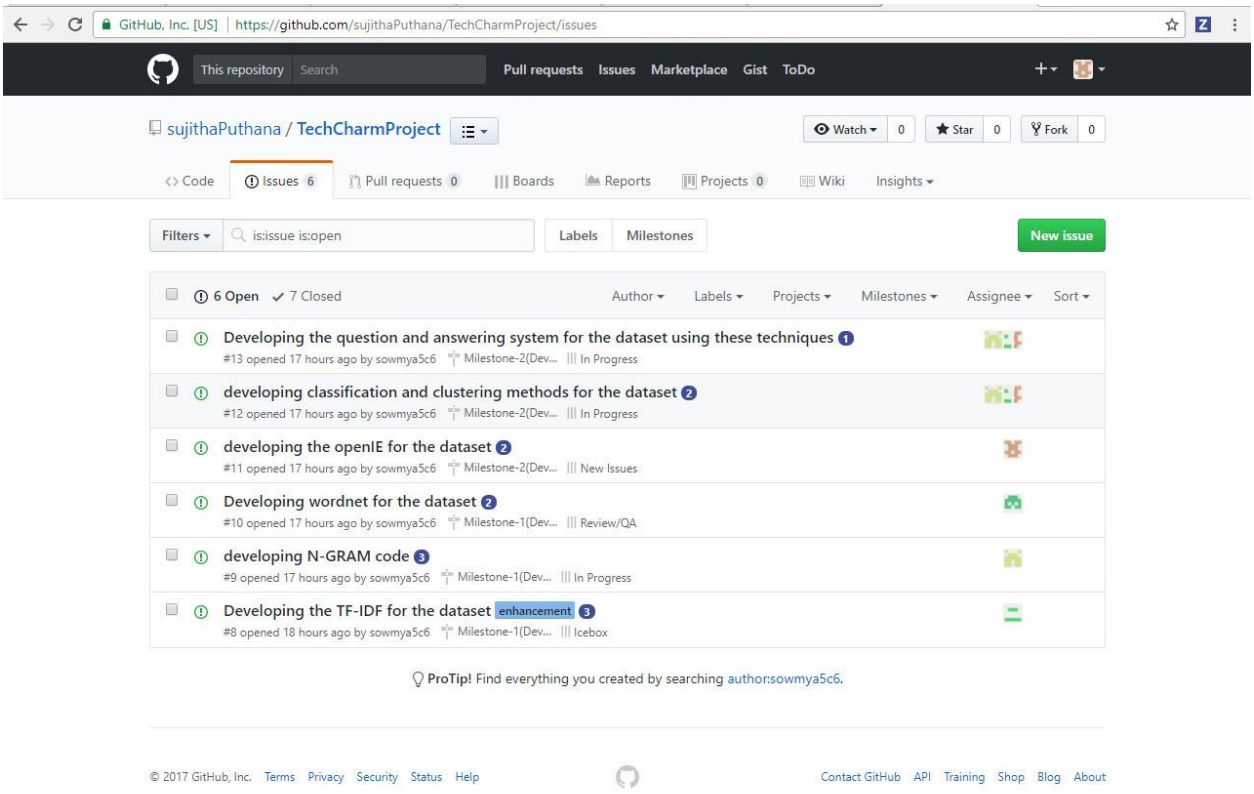
- Jakkepalli, Rama Charan Pavan - **25%**
- Puthana, Sujitha - **25%**
- Yalamanchili, Sowmya - **25%**
- Nandanamudi, Sreelakshmi - **25%**

Name	Implementation	Documentation
Pavan	**Increment-1:** Basic Question Answer System **Increment-2** K-means Question answer using TF-IDF	**Increment-1:** Domain, Specific Dataset, Future Work **Increment**-2: Added more description to document, related work, machine learning
Sujitha	**Increment-1:** TF-IDF **Increment-2** Classification Algorithm TF-IDF question and answer	**Increment-1:** Design workflow, Question Answer, Knowledge Graph **Increment-2**

		Added more description to document, related work, design of Information Extraction
Sowmya	**Increment-1:** Core NLP **Increment-2** Wordnet Question answer using openIE	**Increment-1:** Project Motivation, Objective, Significance. **Increment-2** Design of Information Retrieval
Sreelakshmi	**Increment-1:** Named Entity Recognition **Increment-2** OpenIE Question answer using openIE	**Increment-1:** Contribution, Milestone, issues creation, Work Completed **Increment-2** Design of Machine Learning

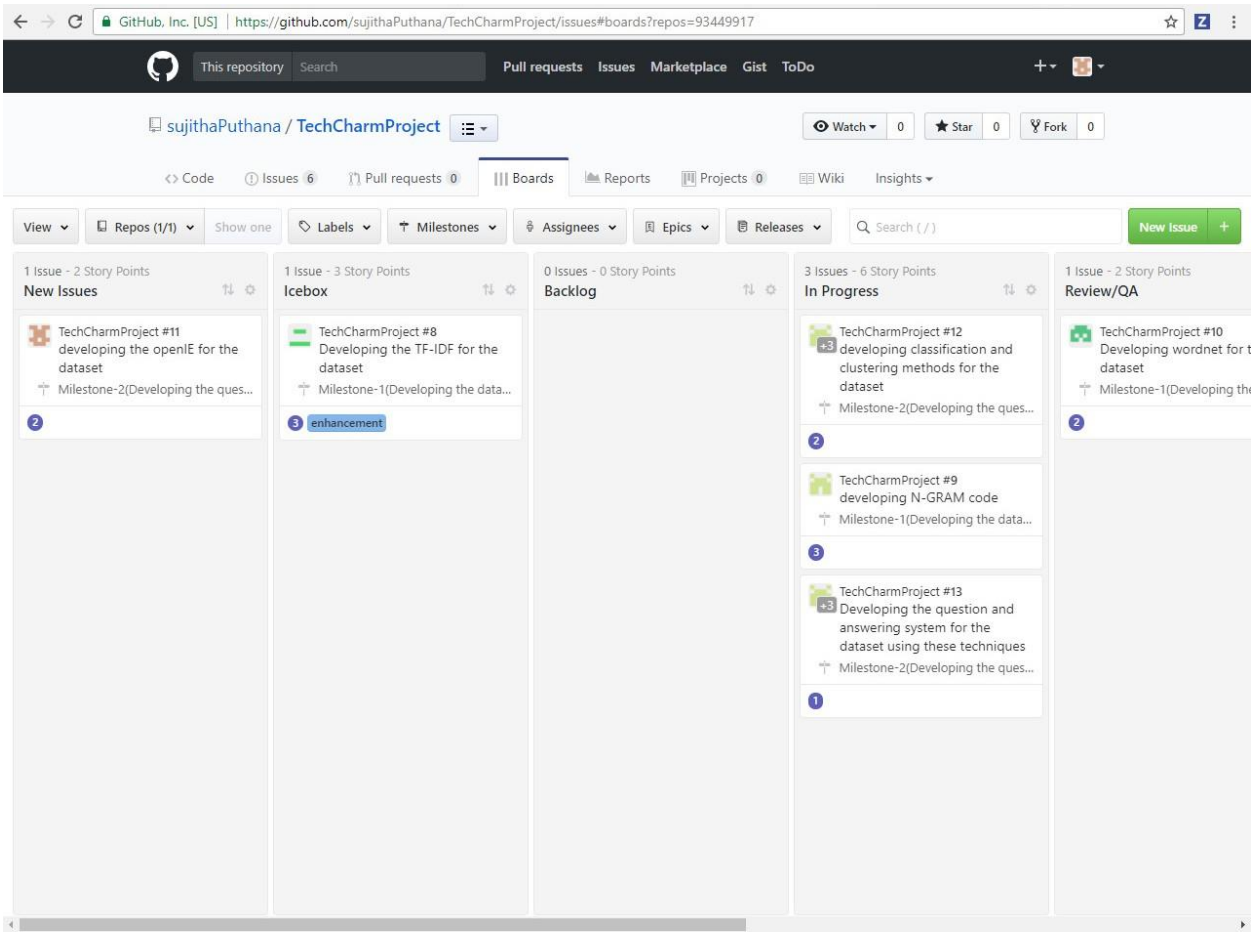
7.b Zen-Hub Screenshots

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.



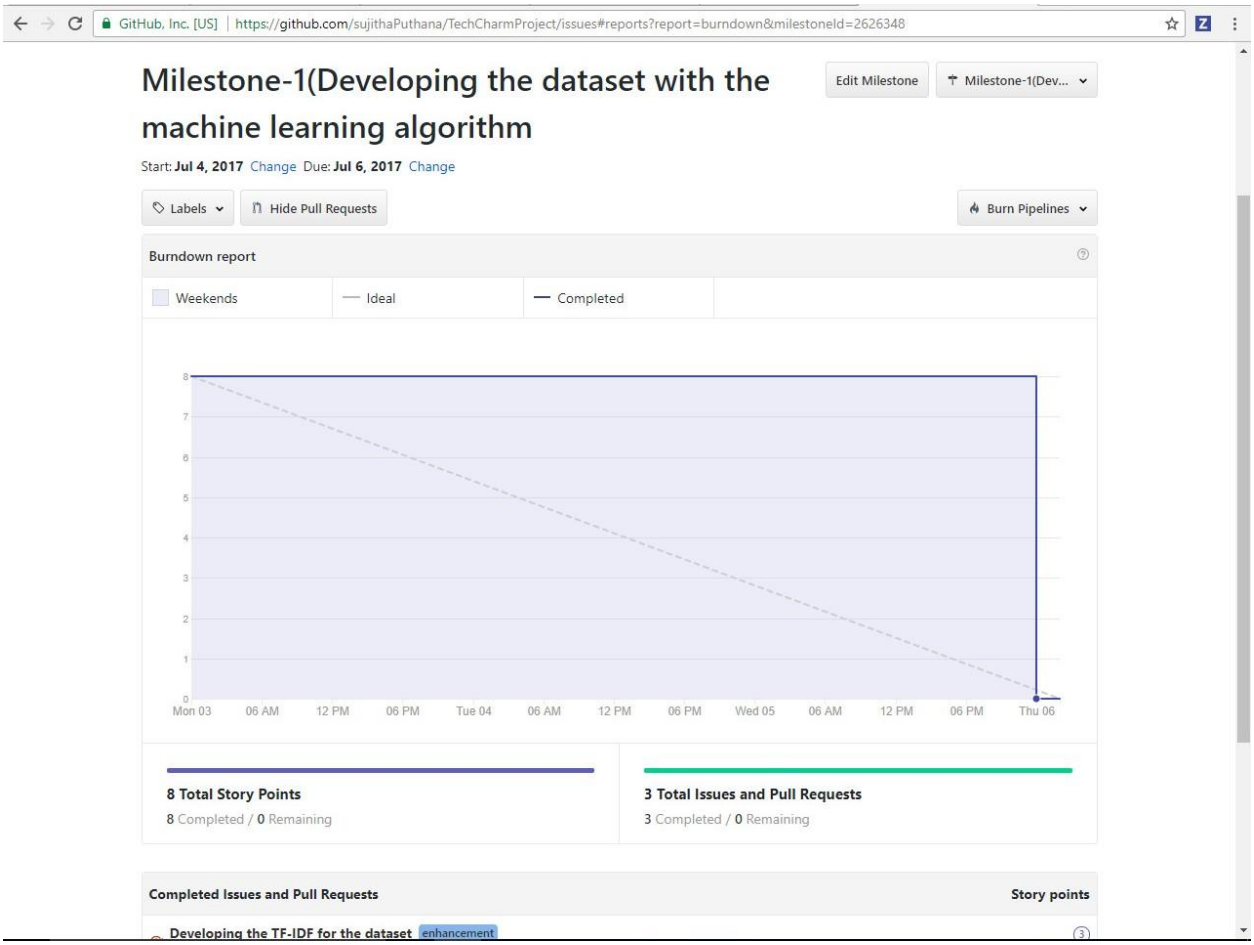
Project Timeline, Members, and Task Responsibility

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.



Burn-Down Chart:

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


GitHub, Inc. [US] | https://github.com/sujithaPuthana/TechCharmProject/issues#reports?report=burndown&milestoneId=2626348
Milestone-1(Developing the dataset with the machine learning algorithm
Edit Milestone
Milestone-1(Dev...
Start: Jul 4, 2017 Change Due: Jul 6, 2017 Change
Labels
Hide Pull Requests
Burn Pipelines
Burndown report
Weekends
Ideal
Completed
8
7
6
5
4
3
2
1
0
Mon 03
06 AM
12 PM
06 PM
Tue 04
06 AM
12 PM
06 PM
Wed 05
06 AM
12 PM
06 PM
Thu 06
8 Total Story Points
8 Completed / 0 Remaining
3 Total Issues and Pull Requests
3 Completed / 0 Remaining
Completed Issues and Pull Requests
Story points
Developing the TF-IDF for the dataset
enhancement
3

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 techinques for the dataset.
- Question and answer system using the OpenIE.

7.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.

7.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 techiques 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
3. https://nlp.stanford.edu/