RV COLLEGE OF ENGINEERING®
BENGALURU – 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

"COVID-19 Text Classifier"

MINI-PROJECT REPORT
OBJECT ORIENTED PROGRAMMING USING JAVA (18CS45)
IV SEMESTER

2020-21

Submitted by

Sahil Sharma – 1RV19IS046
Shivam Prajapati – 1RV19IS049

(Scan QR code for video presentation)

Under the Guidance of

Prof. Priya D
Department of ISE, RVCE,
Bengaluru - 560059

ABSTRACT

Today entire world and India in particular is suffering from the COVID-19 pandemic. Keeping this in mind a machine learning based model is implemented using java openNLP library. The machine learning model is trained to classify text as to if it is related to COVID-19 or not. If the text is related to corona, then further classification is done as to whether the text contains request for essentials or offers for essentials or it simply has statistics or is just information regarding corona. If the text is not related to COVID-19 other important details in the text like locations, names and dates mentioned will be displayed. For user to use this model a smooth interface is provided using javaFX GUI. The interface will also have other information like results from different ML models, instructions on how to use the application and real time information of confirmed, recovered and expired cases of COVID-19 in India represented. For a better visual feel, information and model results have also been displayed as pie charts and line graphs. Apart from using as a simple text classifier, the application can further be extended by coupling it with photo to text converting model so that it can be implemented in real life for example on various social media platforms. Overall, the project aims at being helpful to society in some or the other manner.

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the **Mini-**project work titled "COVID-19 Text Classifier" has been carried out by **Sahil Sharma (1RV19IS046) and Shivam Prajapati (1RV19IS049)**, bonafide students of RV College of Engineering, Bengaluru, have submitted in partial fulfillment for the **Assessment of Course: OBJECT ORIENTED PROGRAMMING USING JAVA (18SC45) – Open-Ended Experiments** during the year 2020-2021. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report.

Priya D
Faculty Incharge
Department of ISE,
RVCE., Bengaluru –59

B M Sagar
Head of Department
Department of ISE,
RVCE, Bengaluru–59

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059
(Autonomous Institution Affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

DECLARATION

We, **Sahil Sharma (1RV19IS046) and Shivam Prajapati (1RV19IS049)**, the students of 4th Semester B.E., Department of Information Science and Engineering, RV College of Engineering, Bengaluru hereby declare that the Mini-Project titled **"**COVID-19 Text Classifier**"** has been carried out by us and submitted in partial fulfillment for the **Assessment of Course: OBJECT ORIENTED PROGRAMMING USING JAVA (16CS44) - Open-Ended Experiment** during the year 2020-2021.

Place: Bengaluru **Sahil's Signature and Shivam's Signature**

Date:

TABLE OF CONTENTS

INTRODUCTION

Object Oriented Concepts

Traditional vs Object Oriented Approach

There are majorly two ways of developing any software and those are Traditional (Procedural) approach or the Object Oriented approach.

• The traditional approach to develop software systems that used procedural programming, could not be used to develop software because of some drawbacks that affected the efficiency and maintainability of the software.

• Object oriented approach to software engineering is used to cover up the drawbacks and develop software projects that use object oriented programming.

• In procedural programming, functions are described and called to perform the specific tasks, wherein the data is not encapsulated with the functions. This was a major problem that comes up with the traditional approach where data is global and not encapsulated within any model object.

• Object oriented programming / paradigm was developed to solve or cover up the problem where all the components of the system as a real entity having attributes and functions linked with it. With this object oriented approach, a blueprint or prototype of any entity can be described and is called a class, and various objects can be created from this.

• Object oriented programming solved some other problems related with traditional procedural programming such as security, effectiveness, data hiding, abstraction, inheritance etc. Object oriented programming also promotes communication within objects through the means of message passing.

• Considering all the points, it is not always that an Object oriented approach will be easier in terms of difficulty level than the traditional approach. For traditional approaches,

procedural programming depends on the size of the software programs, whereas difficulty levels of the object oriented approach depend on the experience of the development team and complexity of the programs.

To ensure a highly efficient and scalable softwares, object oriented approach was a necessary development over traditional approaches. For this reason, this project also is also developed based on object oriented approach using the object oriented language java.

Object Oriented Analysis, Object Oriented Design and Object Oriented Programming

- Object–Oriented Analysis (OOA): is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system's object model, which comprises of interacting objects.
- The main difference between object-oriented analysis and other forms of analysis is that in object-oriented approach, requirements are organized around objects, which integrate both data and functions. They are modelled after real-world objects that the system interacts with. In traditional analysis methodologies, the two aspects - functions and data - are considered separately.
- Grady Booch has defined OOA as, "Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain".
- The primary tasks in object-oriented analysis (OOA) are −
  1. Identifying objects
  2. Organizing the objects by creating object model diagram
  3. Defining the internals of the objects, or object attributes
  4. Defining the behavior of the objects, i.e., object actions
  5. Describing how the objects interact
  6. The common models used in OOA are use cases and object models.

- Object–Oriented Design (OOD): involves implementation of the conceptual model produced during object-oriented analysis. In OOD, concepts in the analysis model, which are technology−independent, are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the

solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.

- The implementation details generally include −

  1. Restructuring the class data (if necessary),
  2. Implementation of methods, i.e., internal data structures and algorithms,
  3. Implementation of control, and
  4. Implementation of associations.

- Grady Booch has defined object-oriented design as "a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design".

- Object-oriented programming (OOP): is a programming paradigm based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.
- The important features of object–oriented programming are −

  1. Bottom–up approach in program design
  2. Programs organized around objects, grouped in classes
  3. Focus on data with methods to operate upon object's data
  4. Interaction between objects through functions
  5. Reusability of design through creation of new classes by adding features to existing classes

- Some examples of object-oriented programming languages are C++, Java, Smalltalk, Delphi, C#, Perl, Python, Ruby, and PHP.

Features of Object Oriented System

- Encapsulation: is a process of information hiding. It is simply the combination of process and data into a single entity. Data of an object is hidden from the rest of the system and available only through the services of the class. It allows improvement or modification of methods used by objects without affecting other parts of a system. In this project different classes for training, predictions and GUI has been used for the implementation of encapsulation.

- Abstraction: It is a process of taking or selecting necessary method and attributes to specify the object. It focuses on essential characteristics of an object relative to perspective of user. In this project variables used that are unnecessary for other classes have been made private and those needed by other classes are public.

- Relationships: All the classes in the system are related with each other. The objects do not exist in isolation, they exist in relationship with other objects.
  There are three types of object relationships −
  1. Aggregation − It indicates relationship between a whole and its parts.
  2. Association − In this, two classes are related or connected in some way such as one class works with another to perform a task or one class acts upon other class.
  3. Generalization − The child class is based on parent class. It indicates that two classes are similar but have some differences.

- Inheritance: is a great feature that allows to create sub-classes from an existing class by inheriting the attributes and/or operations of existing classes. Several classes have inherited various other classes in this project for example Application classs being inherited to run the javaFX GUI to name one.

- Polymorphism and Dynamic Binding: Polymorphism is the ability to take on many different forms. It applies to both objects and operations. A polymorphic object is one who true type hides within a super or parent class. In polymorphic operation, the operation may be carried out differently by different classes of objects. It allows us to manipulate objects of different classes by knowing only their common properties

Overview of Java Programming Language

Features of Java

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are :

- **Simple**
- **Object Oriented**
- **Robust**
- **Platform Independent**
- **Secure**
- **Multi Threading**
- **Architectural Neutral**
- **Portable**
- **High Performance**
- **Distributed**

Inheritance

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

Terms used in Inheritance:

- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new

class. You can use the same fields and methods already defined in the previous class.

- ❖ In our project i.e. Covid Text Classifier, we have used inheritance extensively to follow oops programming paradigm, since it is a java application it is extending Application class in main.java and other classes are also being extended in DashController.java.

Interfaces

- Interface is similar to a class, but it contains only abstract methods.
- By default the variables declared in an interface are public, static and final.
- Interface is a mechanism to achieve full abstraction.
- An interface does not contain any constructor.

Syntax:

```
interface InterfaceName
{
  public void method1();
  public void method2();
  <type> variableName = value;
}
```

- ❖ We have made use of interfaces wherever possible to inhance code portability.
- ❖ In DashController.java , we are implementing Initializable interface which has an abstract method initialize() which is called automatically on loading the java file.
- ❖ There is an interface named “othersnlp” for dealing with NLP related classes.

Packages

A package is a mechanism to group the similar type of classes, interfaces and sub-packages and provide access control. It organizes classes into single unit.
In Java already many predefined packages are available, used while programming.
For example: java.lang, java.io, java.util etc.

Advantages of Packages:

- Packages provide code reusability, because a package has group of classes.
- It helps in resolving naming collision when multiple packages have classes with the same name.

- Package also provides the hiding of class facility. Thus other programs cannot use the classes from hidden package.
- Access limitation can be applied with the help of packages.
- One package can be defined in another package.

Types of Packages

There are two types of packages available in Java.

1. Built-in packages

   Built-in packages are already defined in java API. For example: java.util, java.io, java,lang, java.awt, java.applet, java.net, etc.

2. User defined packages

   The package we create according to our need is called user defined package.

- ❖ In our project we are using several built in packages like java.util, javafx.collections and many more.
- ❖ We have bundled our whole project in multiple packages to inhance code reusability and interconnectivity of different java files.

Exception Handling

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

Advantage of Exception Handling:

The core advantage of exception handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application; that is why we need to handle exceptions.

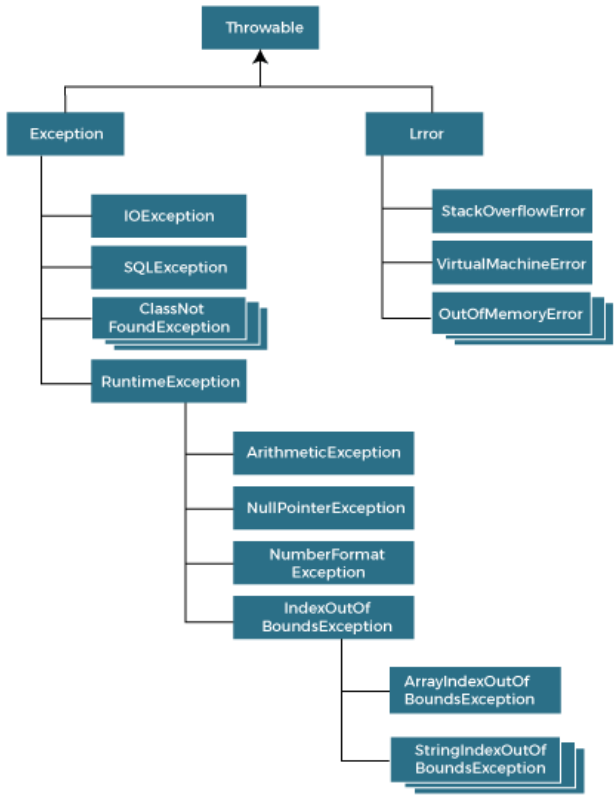
Types of Java Exceptions:

There are mainly two types of exceptions: checked and unchecked. An error is considered as the unchecked exception. However, according to Oracle, there are three types of exceptions namely:

1. Checked Exception
2. Unchecked Exception
3. Error

Hierarchy of Java Exception classes:

The java.lang.Throwable class is the root class of Java Exception hierarchy inherited by two subclasses: Exception and Error. The hierarchy of Java Exception classes is given below:



Multithreaded Programming

Multithreading in Java is a process of executing multiple threads simultaneously.
A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking. However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Advantages of Java Multithreading:

1) It **doesn't block the user** because threads are independent and you can perform multiple operations at the same time.

2) You **can perform many operations together, so it saves time**.

3) Threads are **independent**, so it doesn't affect other threads if an exception occurs in a single thread.

- In our program also we have used multithreading to create a new thread on giving the input and calculating the data for the given input using different algorithm other than our default algorithm and showing the performance comparison of both the algorithms.

Lambda Expressions

It provides a clear and concise way to represent one method interface using an expression. It is very useful in collection library. It helps to iterate, filter and extract data from collection.

The Lambda expression is used to provide the implementation of an interface which has functional interface. It saves a lot of code. In case of lambda expression, we don't need to define the method again for providing the implementation. Here, we just write the implementation code. Java lambda expression is treated as a function, so compiler does not create .class file.

Why use Lambda Expression?

1. To provide the implementation of Functional interface.
2. Less coding.

Syntax:

(argument-list) -> {body}

- In our project we have used lambda expression to provide the implementation of Runnable functional interface (used for thread creation).
- The body of run() method has been provided using lambda expression.

Regular Expressions

The **Java Regex** or Regular Expression is an API to define a pattern for searching or manipulating strings. It is widely used to define the constraint on strings such as password and email validation. After learning Java regex tutorial, you will be able to test your regular expressions by the Java Regex Tester Tool. Java Regex API provides 1 interface and 3 classes in **java.util.regex** package.

java.util.regex package:

The Matcher and Pattern classes provide the facility of Java regular expression. The java.util.regex package provides following classes and interfaces for regular expressions.

1. MatchResult interface

2. Matcher class
3. Pattern class
4. PatternSyntaxException class

- Our project uses regular expression to filter the input text and remove unnecessary words from the text which may lead to inaccuracy of ML algorithm for the purpose of classification.

Strings

In Java, string is basically an object that represents sequence of char values. An array of characters works same as Java string.

Java String class provides a lot of methods to perform operations on strings such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc. The java.lang.String class implements *Serializable*, *Comparable* and *CharSequence* interfaces.

There are two ways to create String object:

1. By string literal (String s="welcome")
2. By new keyword (String s=**new** String("Welcome"))

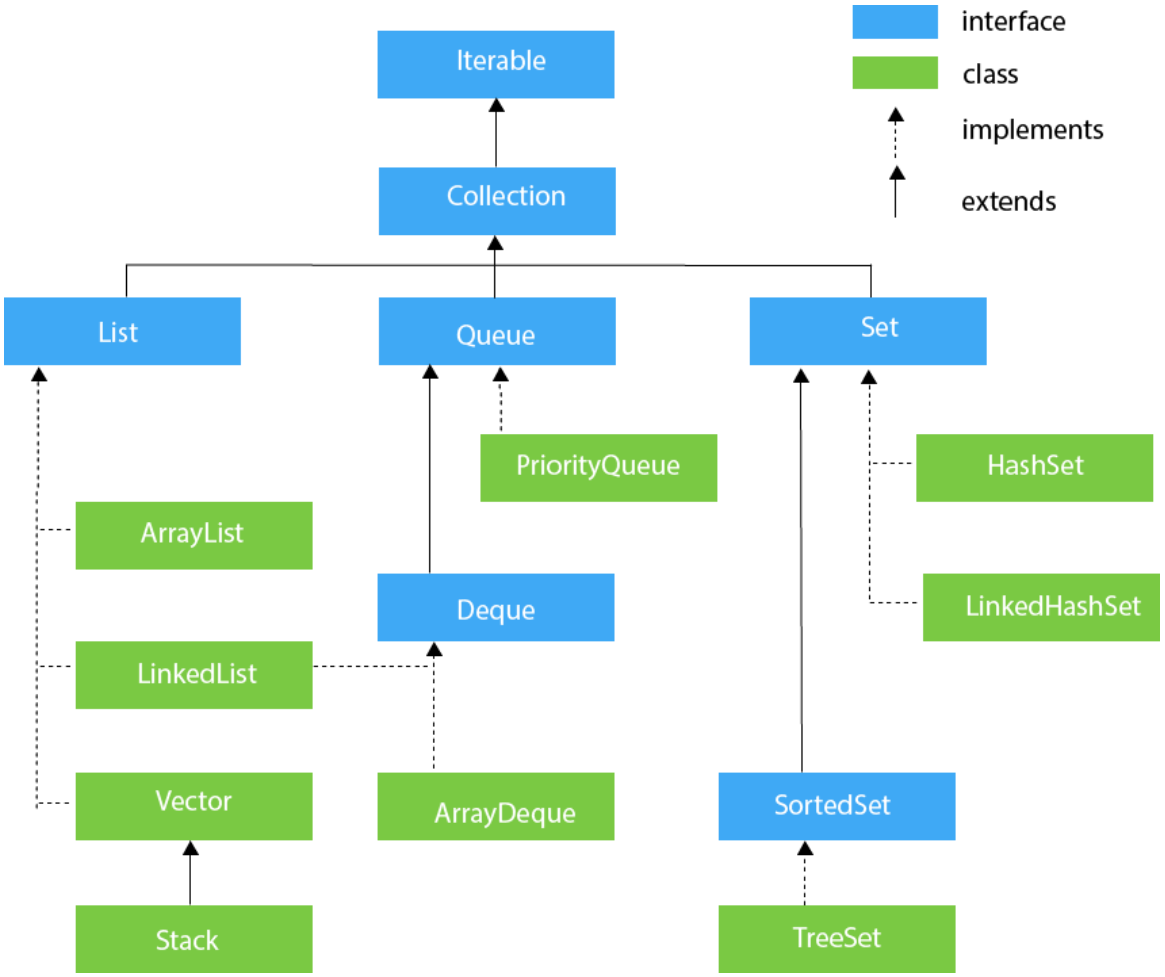
- We have used String to take input from the user and storing it in a String variable and used various inbuilt String methods to work with it.

Collection Framework

The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects. Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Hierarchy of Collection Framework

The **java.util** package contains all the classes and interfaces for the Collection framework.



- ❖ The project uses classes and iterfaces of collection framework wherever requires.
- ❖ It has been used for defining the variables and values for graphs and pie charts used in the project.

JavaFX framework

JavaFX is a Java library that is used to develop Desktop applications as well as Rich Internet Applications (RIA). The applications built in JavaFX, can run on multiple platforms including Web, Mobile and Desktops.

Features of JavaFX:

Feature	Description
Java Library	It is a Java library which consists of many classes and interfaces that are written in Java.
FXML	FXML is the XML based Declarative mark up language. The coding can be done in FXML to provide the more enhanced GUI to the user.
Scene Builder	Scene Builder generates FXML mark-up which can be ported to an IDE.
Web view	Web pages can be embedded with JavaFX applications. Web View uses WebKitHTML technology to embed web pages.
Built in UI controls	JavaFX contains Built-in components which are not dependent on operating system. The UI component are just enough to develop a full featured application.
CSS like styling	JavaFX code can be embedded with the CSS to improve the style of the application. We can enhance the view of our application with the simple knowledge of CSS.
Swing interoperability	The JavaFX applications can be embedded with swing code using the Swing Node class. We can update the existing swing application with the powerful features of JavaFX.
Canvas API	Canvas API provides the methods for drawing directly in an area of a JavaFX scene.
Rich Set of APIs	JavaFX provides a rich set of API's to develop GUI applications.
Integrated Graphics Library	An integrated set of classes are provided to deal with 2D and 3D graphics.
Graphics Pipeline	JavaFX graphics are based on Graphics rendered pipeline(prism). It offers smooth graphics which are hardware accelerated.
High Performance Media Engine	The media pipeline supports the playback of web multimedia on a low latency. It is based on a Gstreamer Multimedia framework.
Self-contained application deployment model	Self Contained application packages have all of the application resources and a private copy of Java and JavaFX Runtime.

- This whole project uses javaFX for GUI element.
- So, many javaFX elements has been used to enhance user experience.
- javaFX graphs, charts, buttons, shapes, text fields etc has been used.

Proposed System

Objectives

- To Train a models to identify Covid reated texts and sub-categories and use other available models for extracting persons, locations and dates from the given text.
- To display real time statistics of confirmed, recovered and expired COVID-19 cases in India.
- To implement the above mentioned functionalities in an user friendly way both numerically and graphically using JavaFX GUI.

Methodology

- The project is built on a dynamic and architecture neutral object oriented programming language – Java.
- As the project falls under natural language processing domain, openNLP library of java is being utilised here. Apache OpenNLP is an open-source Java library which is used to process natural language text. OpenNLP provides services such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution.
- OpenCSV library is also used in this project to read real time statistics of India from an up to date csv file having all such data from the web.
- Information grouping feature is used extensively to arrive at solution. Text is classified as to if it is related to COVID-19 or not, if it is then further classification is done as to whether the text contains request for essentials or offers for essentials or has statistics or general information. If text is not COVID-19 related important information from text like names, location and dates are extracted and displayed.
- Provision of an interactive and clean user interface is done using JavaFX framework. It will be responsible for displayinf real time COVID-19 statistics of India, taking data from user and displaying output based on trained model on screen along with graphical representation of the same in the form of pie charts and line charts.

Scope

This project has a wide variety of application in real world wherein it can be integrated with various emailing platforms or social media platforms like Instagram, Facebook, Twitter, etc. Whenever a person will get any message from anyone, it can be classified using the models trained. Such a classification will help people to respond to people in need with at most priority hence saving many lives.

The project can also be integrated with image to text conversion models which will further increase the application of this project.

Apart from above mentioned application of this project, the model still has scope of improvement. The trained model's accuracy can be further improved by feeding it with better training and testing dataset.

REQUIREMENT SPECIFICATIONS

Hardware Requirements

- Windows

    Windows 10 (8u51 and above)

    Windows 8.x (Desktop)

    Windows 7 SP1

    Windows Vista SP2

    Windows Server 2008 R2 SP1 (64-bit)

    Windows Server 2012 and 2012 R2 (64-bit)
- RAM: 128 MB
- Disk space: 124 MB for JRE; 2 MB for Java Update
- Processor: Minimum Pentium 2 266 MHz processor

- Mac OS X

    Intel-based Mac running Mac OS X 10.8.3+, 10.9+

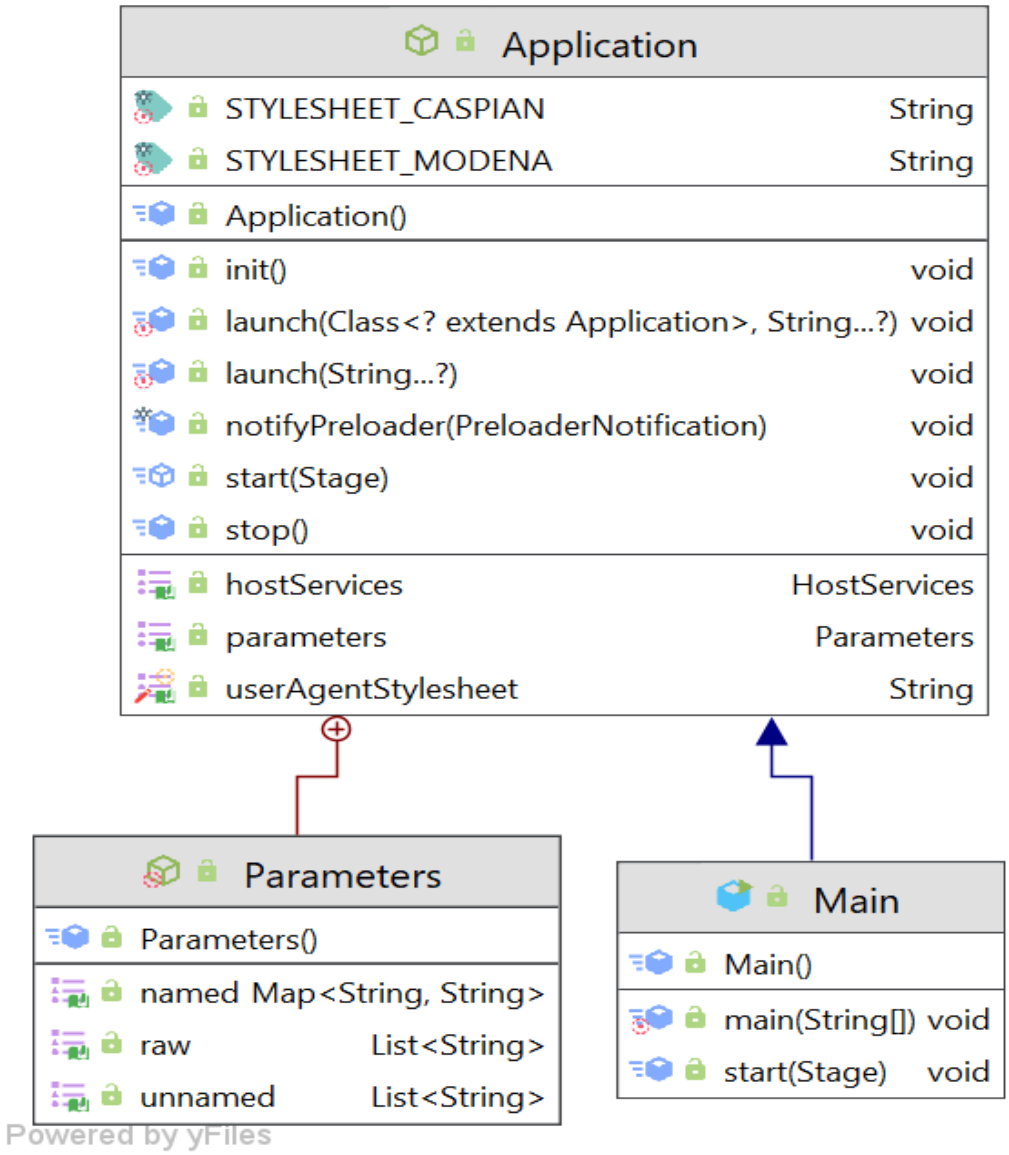
Software Requirements

- JDK 1.6
- Support for javafx by linking it with the project
- Internet access
- OpenCSV external library
- OpenNLP external library

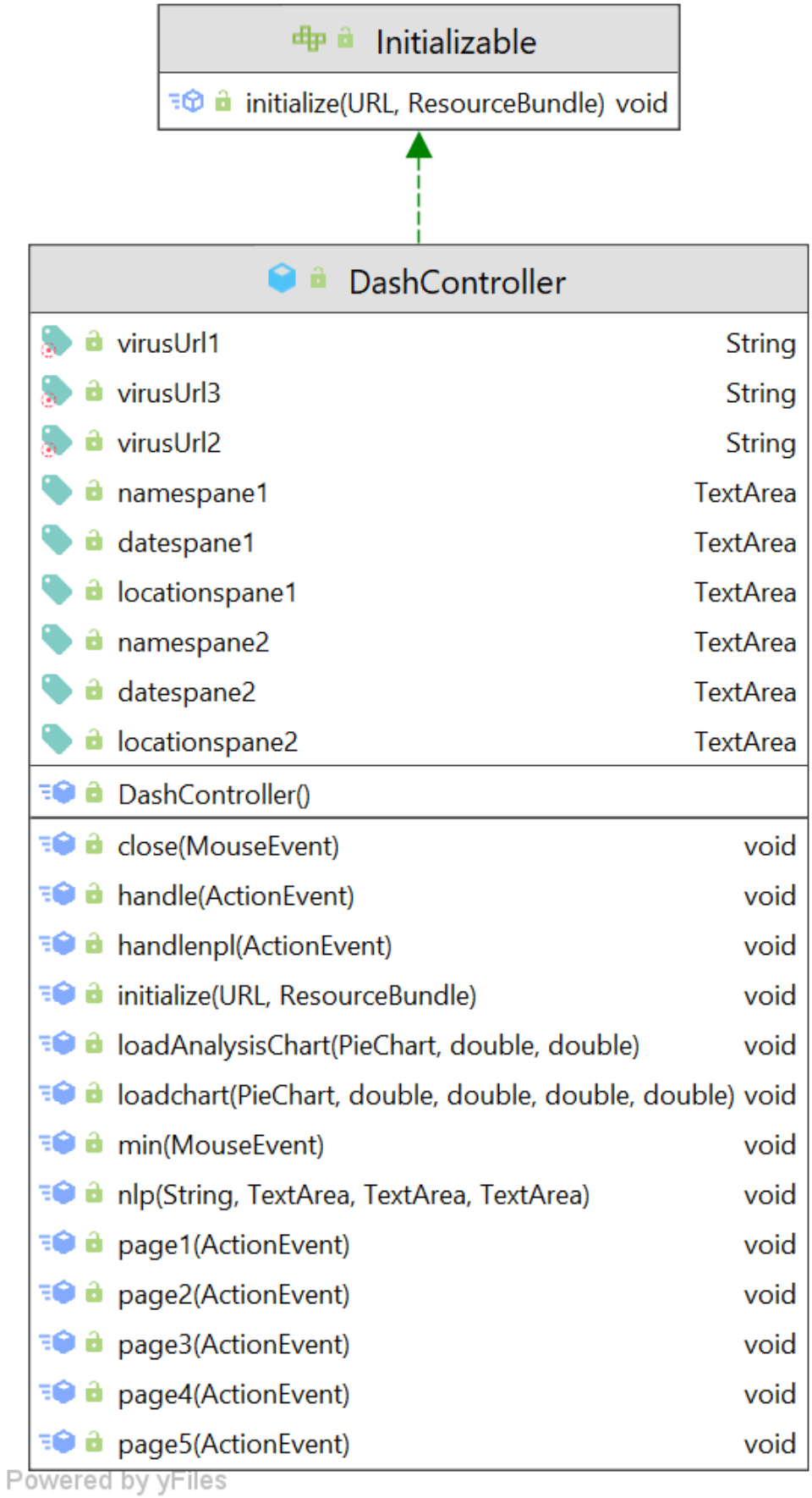
SYSTEM DESIGN AND IMPLEMENTATION

Class Diagram

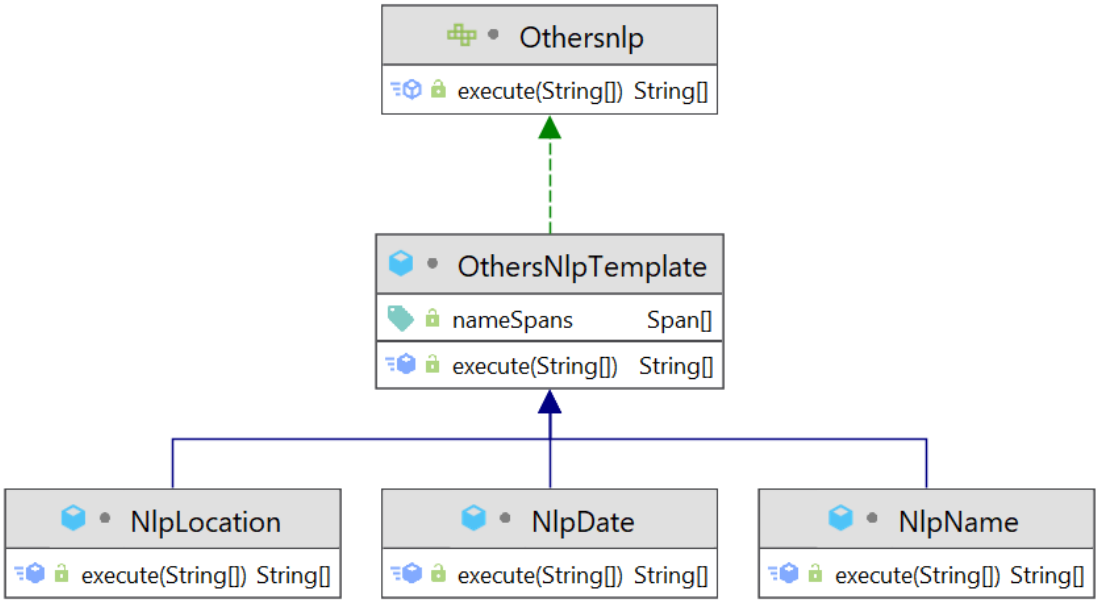
Main.java



DashController.java



NLP


Othersnlp
execute(String[]) String[]
OthersNlpTemplate
nameSpans Span[]
execute(String[]) String[]
NlpLocation
execute(String[]) String[]
NlpDate
execute(String[]) String[]
NlpName
execute(String[]) String[]

Modular Description/ Pseudo-code

- Training models
  - Reading training dataset and feeding it into doccategorizer along with other parameters set such as algorithm type, iterations and cutoff value.
  - Storing the trained model in a specified location.

- Improving trained models
  - Evaluating accuracy of trained model and improving it by enhancing train dataset and training parameters.
  - Finalizing the best model

- Usage of model
  - Using the models trained and other available models.
  - Developing a GUI for taking the input.
  - Feeding the input taken into the model for prediction and sending back result to the GUI.
  - Displaying the result in an meaningful and attractive way.

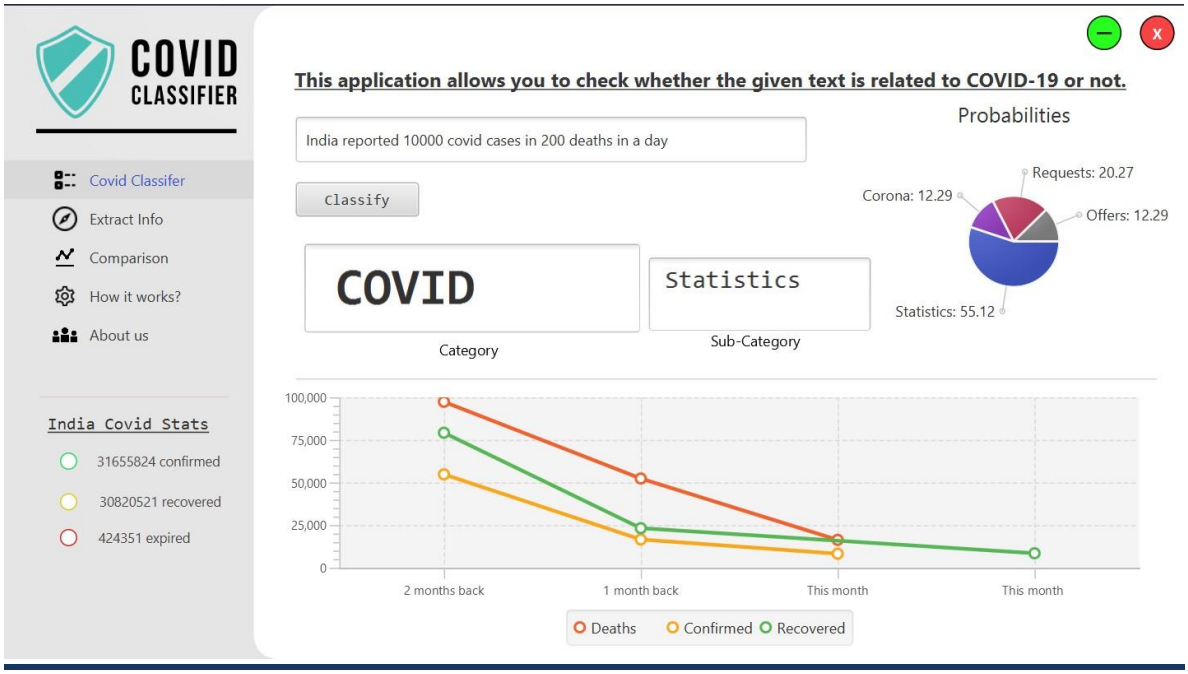
- Real time data
  - Accessing the dynamic csv file through internet.
  - Reading the file and identifying useful data to be displayed.
  - Feeding the obtained data to GUI
  - GUI presenting the data numerically and graphically.

RESULTS AND SNAPSHOTS

This is a covid text classifer which classifies the entered text into two categories, i.e. COVID or NON COVID.

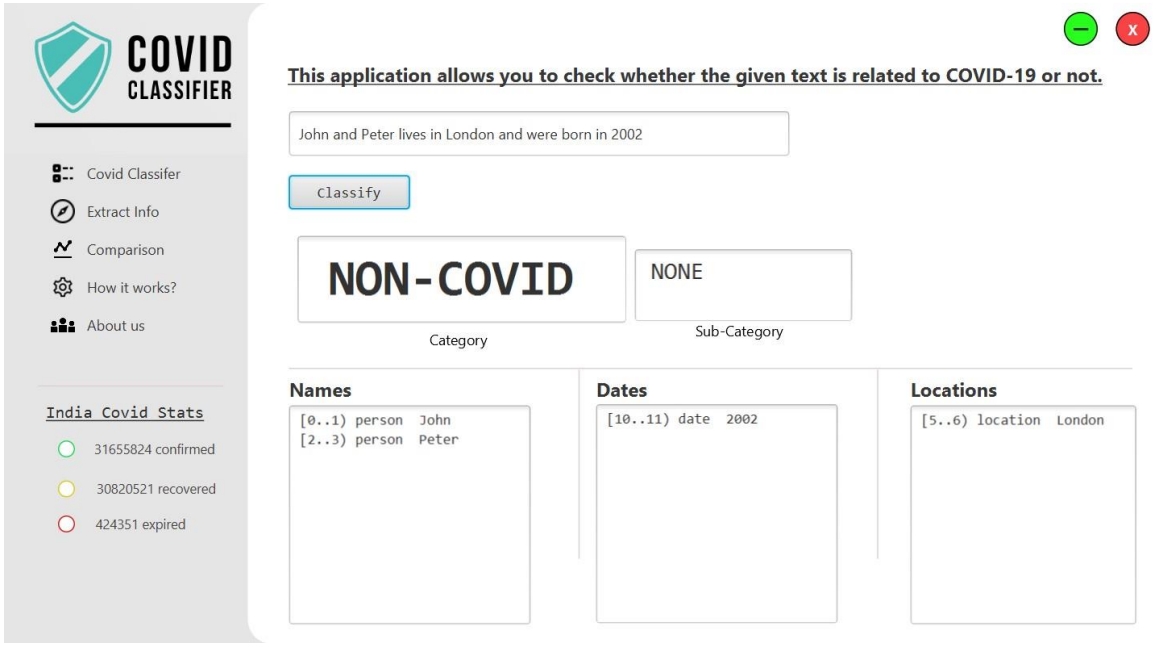
COVID:

- If the entered text is determined to be of class COVID, then the application also determines the sub-category of the text (i.e. Corona, Requests, Offers & Statistics)
- A Pie chart is also displayed, showing the probability percentages of each sub-category determined by the ML model.



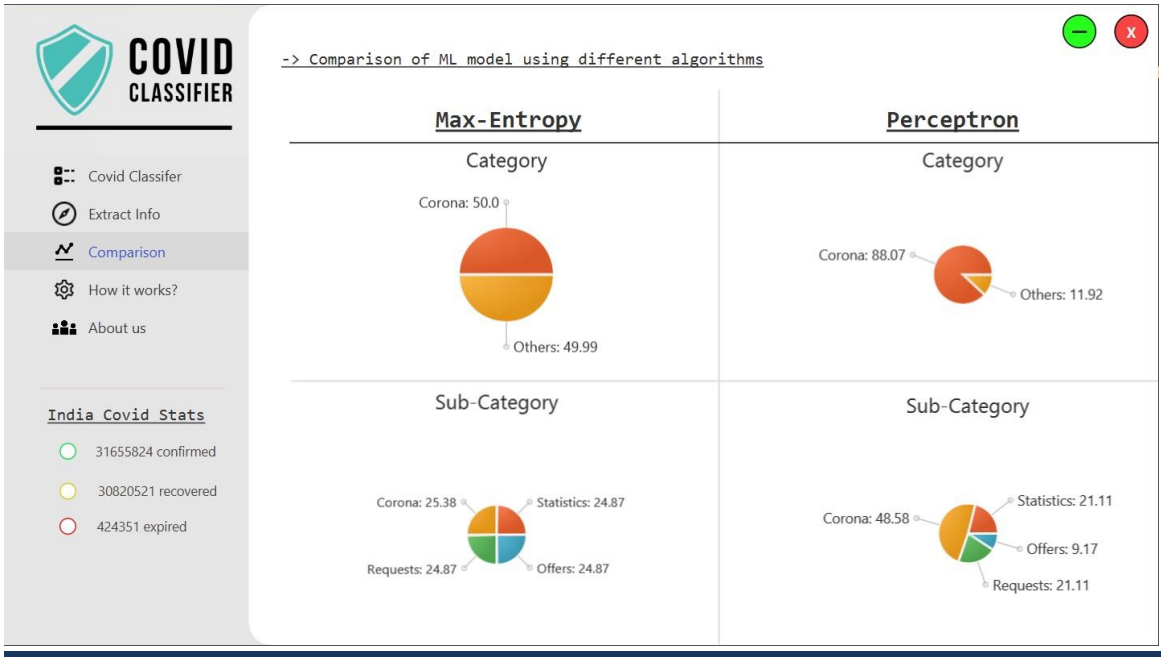
NON-COVID:

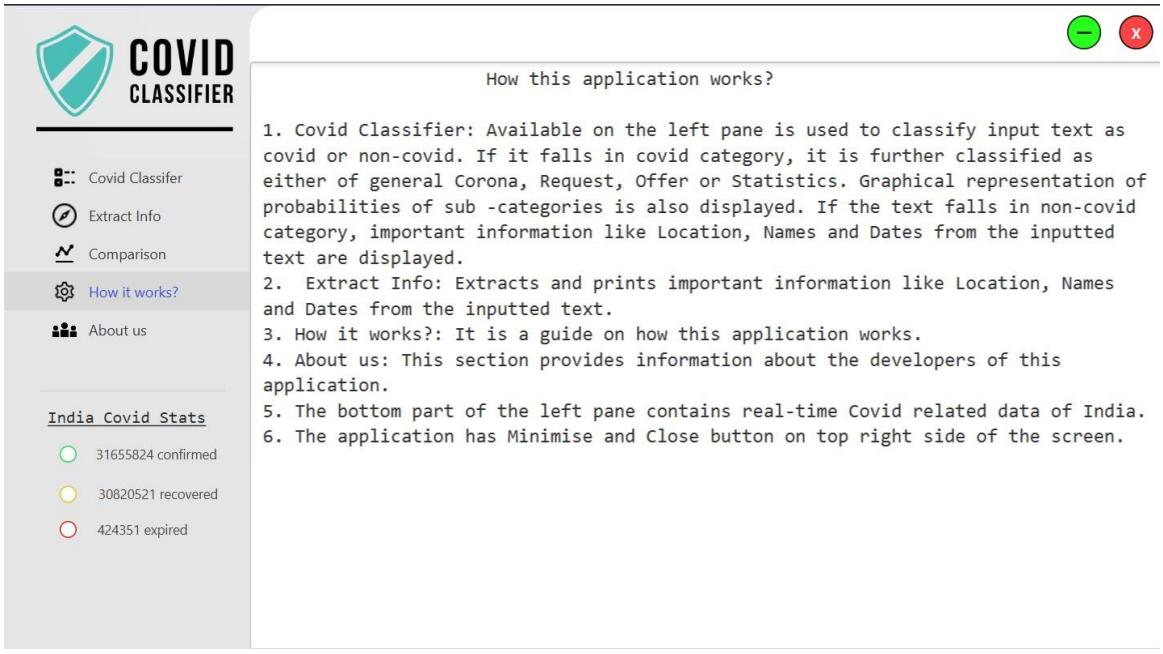
- If the entered text is classified as NON-COVID, then the application uses NLP models for extracting useful information from the text such as Names, Dates and Locations and displays them.

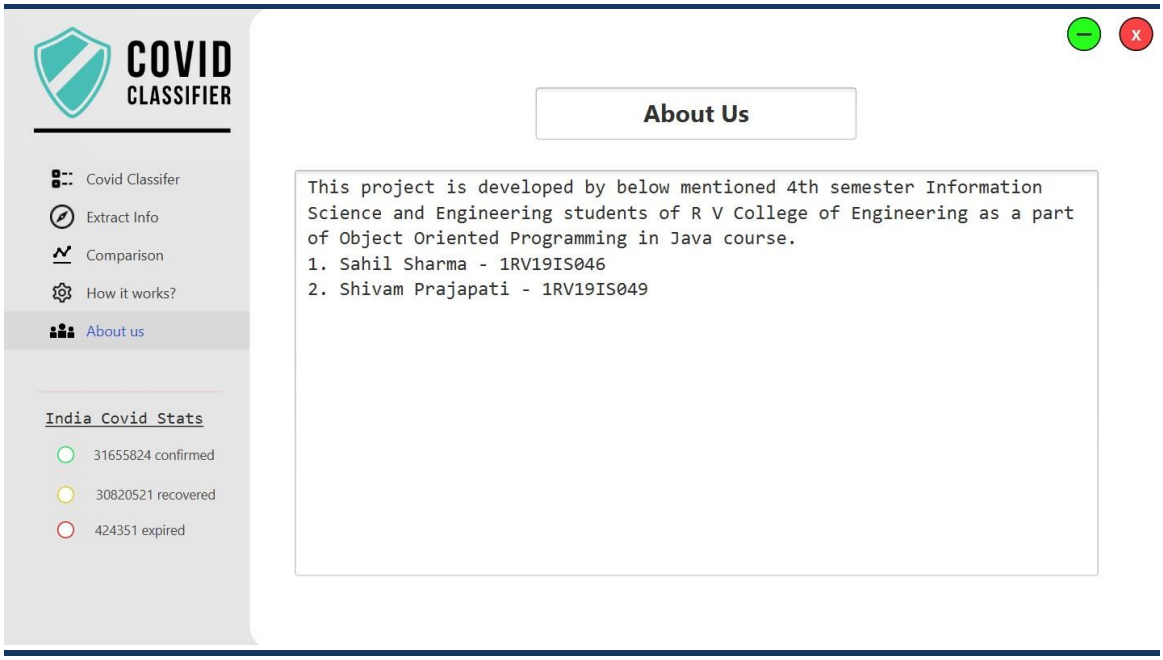


OTHER FEATURES:

- Besides the text classification feature, this application also displays realtime covid stats of India (i.e. Number of confirmed, recovered cases and deaths) and the past 3 months trend of covid in India is displayed using line chart as a visual element.
- There is also a dedicated page of Named Entity Recognition feature.
- A separate pane displays the comparison between two major alogrithms used for the classification of text (i.e. Max-entropy and Perceptron) using pie charts.







ACCURACIES OF THE MODELS USED:

- All together 8 ML models are used in this project, their accuracy is shown below:

1. Maxent model for binary classification into corona or others

en-is-covid-maxent-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:      46
    Min sentence size:       8
    Max sentence size:     663
Average sentence size: 319.74
           Tags count:       2
             Accuracy: 93.48%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|    Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
| Corona |      3 |     26 | 0.115   | 1         | 0.885  | 0.939     |
| Others |      0 |     20 | 0       | 0.87      | 1      | 0.93      |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

   a    b | Accuracy | <-- classified as
 <23>   3 |    -100% |      a = Corona
   .  <20>|    -100% |      b = Others

<-end> Confusion matrix
```

1. Maxent model for sub-classification of corona into corona, offers, requests or statistics

en-covid-category-maxent-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:      30
    Min sentence size:       6
    Max sentence size:      43
Average sentence size:   19.47
           Tags count:       4
             Accuracy:     80%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|        Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
|     Corona |      3 |     17 | 0.176   | 0.824     | 0.824  | 0.824     |
| Statistics |      3 |      3 | 1       | 0         | 0      | 0         |
|     Offers |      0 |      5 | 0       | 1         | 1      | 1         |
|   Requests |      0 |      5 | 0       | 0.625     | 1      | 0.769     |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

   a    b    c    d | Accuracy | <-- classified as
 <14>   .    3    . |    -100% |      a = Corona
   .   <5>   .    . |    -100% |      b = Offers
   .    .   <5>   . |    -100% |      c = Requests
   3    .    .   <.>|    -100% |      d = Statistics

<-end> Confusion matrix
```

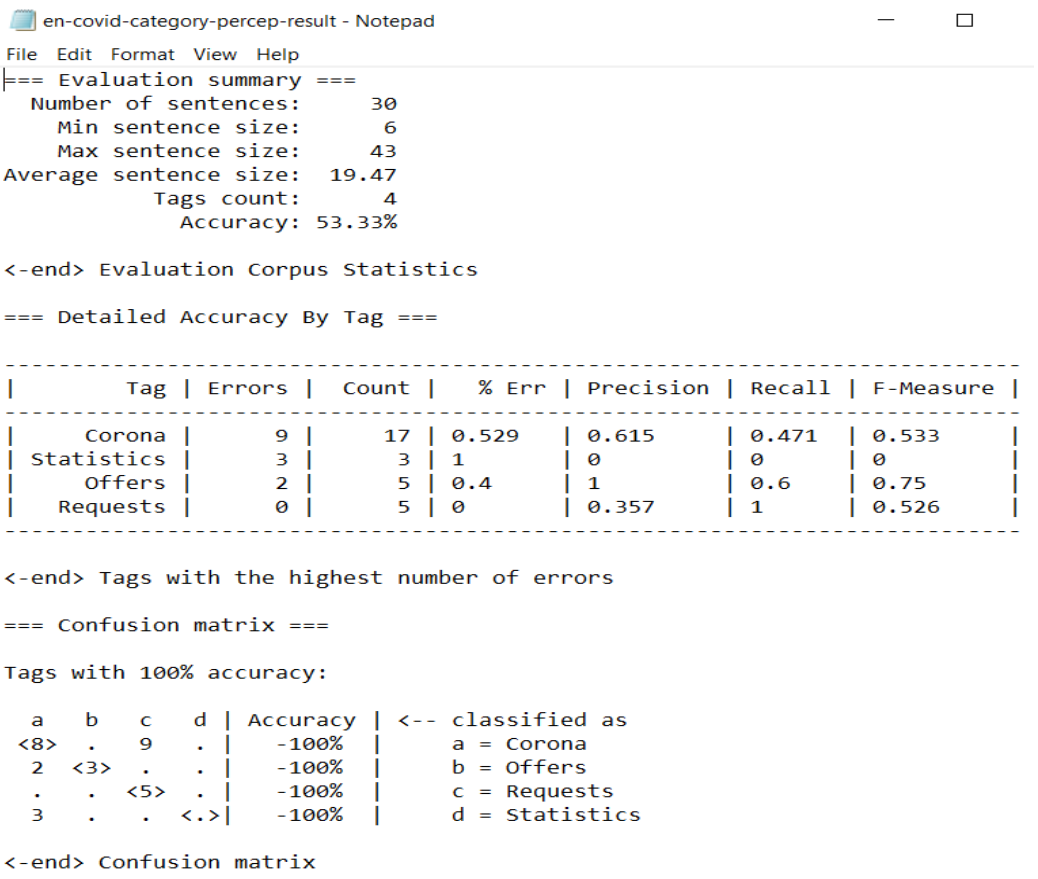
3.Perceptron model for binary classification into corona or others

en-is-covid-percep-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:     46
    Min sentence size:      8
    Max sentence size:    663
Average sentence size: 319.74
           Tags count:      2
             Accuracy: 80.43%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|    Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
| Others |      7 |     20 | 0.35    | 0.867     | 0.65   | 0.743     |
| Corona |      2 |     26 | 0.077   | 0.774     | 0.923  | 0.842     |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

   a    b | Accuracy | <-- classified as
 <24>   2 |    -100% |     a = Corona
   7  <13>|    -100% |     b = Others

<-end> Confusion matrix
```

4. Perceptron model for sub-classification of corona into corona, offers, requests or statistics



en-covid-category-percep-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:      30
    Min sentence size:       6
    Max sentence size:      43
Average sentence size:   19.47
           Tags count:       4
             Accuracy: 53.33%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------------
|        Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------------
|     Corona |      9 |     17 | 0.529   | 0.615     | 0.471  | 0.533     |
| Statistics |      3 |      3 | 1       | 0         | 0      | 0         |
|     Offers |      2 |      5 | 0.4     | 1         | 0.6    | 0.75      |
|   Requests |      0 |      5 | 0       | 0.357     | 1      | 0.526     |
-------------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

  a   b   c   d | Accuracy | <-- classified as
 <8>  .   9   . |   -100%  |     a = Corona
  2  <3>  .   . |   -100%  |     b = Offers
  .   .  <5>  . |   -100%  |     c = Requests
  3   .   .  <.>|   -100%  |     d = Statistics

<-end> Confusion matrix
```

5. Tokenizer

Precision: 0.9544554455445544

Recall: 0.9601593625498008

F-Measure: 0.9572989076464747

Execution time: 0.337 seconds

6. Person recognizer

en-ner-person-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:     43
    Min sentence size:      4
    Max sentence size:     25
Average sentence size:  11.98
           Tags count:      1
             Accuracy: 97.67%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|   Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
| other |     12 |    515 | 0.023   | 1         | 0.977  | 0.988     |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

    a     b     c | Accuracy | <-- classified as
 <503>    5     7 |   -100%  |     a = other
    .    <.>    . |   0%     |     b = person-cont
    .     .    <.>|   0%     |     c = person-start

<-end> Confusion matrix
```

7. Location tokenizer

en-ner-location-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:     43
    Min sentence size:      4
    Max sentence size:     25
Average sentence size:  11.98
           Tags count:      1
             Accuracy: 99.42%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|   Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
| other |      3 |    515 | 0.006   | 1         | 0.994  | 0.997     |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

    a     b | Accuracy | <-- classified as
 <512>    3 |   -100%  |     a = other
    .    <.>|   0%     |     b = location-start

<-end> Confusion matrix
```

8. Date tokenizer

en-ner-date-result - Notepad

File Edit Format View Help

```
=== Evaluation summary ===
  Number of sentences:     43
    Min sentence size:      4
    Max sentence size:     25
Average sentence size:  11.98
           Tags count:      1
             Accuracy: 99.22%

<-end> Evaluation Corpus Statistics

=== Detailed Accuracy By Tag ===

-------------------------------------------------------------------------
|   Tag | Errors |  Count |   % Err | Precision | Recall | F-Measure |
-------------------------------------------------------------------------
| other |      4 |    515 | 0.008   | 1         | 0.992  | 0.996     |
-------------------------------------------------------------------------

<-end> Tags with the highest number of errors

=== Confusion matrix ===

Tags with 100% accuracy:

    a     b     c | Accuracy | <-- classified as
 <511>    3     1 |   -100%  |     a = other
    .    <.>    . |   0%     |     b = date-start
    .     .    <.>|   0%     |     c = date-cont

<-end> Confusion matrix
```

CONCLUSION

World has seen unprecedented times in the recent past, a pandemic like no other. Public gatherings became fatal, colleges and schools got closed, that is the reason we are completing this project remotely from our homes. Many people lost their lives in first wave and the new variants of the coronavirus made people struggle for every single breath. Second wave came with acute and urgent rise in essentials needed to fight the virus. Those who could not get help lost lives and those who could not help the people in need got a lifelong repentance of not being able to help. This helplessness was sometimes because of overwhelming amount of requests and offers going ignored in a large and all messed web. To make sure this does not repeat in future, this application is built for the sole purpose of saving lives by highlighting such requests and offers when used in an appropriate manner. Highlighting and prioritizing response can bridge the gap of those in need and those who are capable of helping, hence ultimately saving lives because nothing is more precious.

This project has a wide variety of application in real world wherein it can be integrated with various emailing platforms or social media platforms like Instagram, Facebook, Twitter, etc. Whenever a person will get any message from anyone, it can be classified using the models trained. Such a classification will help people to respond to people in need with at most priority hence saving many lives.

REFERENCES

[1] https://codingee.com/traditional-vs-object-oriented-software-development-approach/

[2] https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_object_oriented_approach.htm

[3] http://opennlp.apache.org/docs/1.9.3/manual/opennlp.html

[4] https://www.tutorialspoint.com/opennlp/index.htm

[5] https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/

[6] Java Complete Referrence ninth edition - Herbert Schildt

[7] Object-Oriented Analysis and Design with Applications Third Edition - Grady Booch, Ivar Jacobson, and James Rumbaugh

APPENDIX –A: SOURCE CODE

https://github.com/sahilsharma0223/covid-classify