

# SENTIMENT ANALYSIS USING RECURRENT NEURAL NETWORKS

## A PROJECT REPORT

Submitted to



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR,  
ANANTHAPURAMU**

*In partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY  
IN  
COMPUTER SCIENCE AND ENGINEERING**

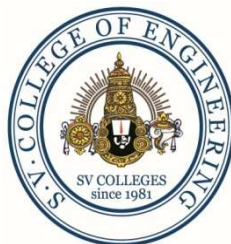
By

**K PoornaChandra  
V Bhuvitha  
G Harsha Vardhan  
C Ramya Sree Reddy**

**16BF1A0549  
16BF1A0555  
16BF1A0537  
16BF1A0522**

**Under the Guidance of  
Dr.P.Dileep Kumar Reddy, Ph.D.**

Associate Professor



**Department of Computer Science and Engineering**

**SRI VENKATESWARA COLLEGE OF ENGINEERING**

Permanent Affiliation to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UGC act 1956. Accredited by NBA, New Delhi & NAAC Bangalore with 'A' Grade, Tirupati-517507, Chittoor, A.P.

**2016-2020**

# **SRI VENKATESWARA COLLEGE OF ENGINEERING**

Permanent Affiliation to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UGC act 1956. Accredited by NBA, New Delhi & NAAC Bangalore with 'A' Grade, Tirupati-517507, Chittoor, A.P.

## **Department of Computer Science and Engineering**



### **CERTIFICATE**

This is to certify that the project entitled “**SENTIMENT ANALYSIS**” is a bonafide work done by **K POORNACHANDRA 16BF1A0549,**  
**V BHUVIDHA 16BF1A0555, G HARSHA VARDHAN 16BF1A0537,**  
**C RAMYA SREE REDDY 16BF1A0522** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, from Jawaharlal Nehru Technological University Anantapur, Ananthapuramu during the year 2019-2020.

#### **Project Guide**

**Dr. P. Dileep Kumar Reddy, Ph.D.**

Associate Professor  
Department of CSE  
SV College of Engineering  
Tirupati – 517507

#### **Head of the Department**

**Dr. P.E.S.N. Krishna Prasad, Ph.D.**

Professor  
Department of CSE  
SV College of Engineering  
Tirupati – 517507

**Viva-Voce date:** \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

We hereby declare that the project report entitled “**SENTIMENT ANALYSIS**” done by us under the esteemed guidance of **Dr.P.DILEEP KUMAR REDDY, Ph.D., Associate Professor** and is submitted in partial fulfillment of the requirements for the award of the Bachelor’s of Technology in **Computer Science and Engineering**.

Date:

K PoornaChandra (16BF1A0549)

Place:

V Bhuvudha (16BF1A0555)

G Harsha Vardhan (16BF1A0537)

C Ramya Sree Reddy(16BF1A0522)

## ACKNOWLEDGEMENT

We are thankful to our guide **Dr.P.Dileep Kumar Reddy** for his valuable guidance and encouragement. His helping attitude and suggestions have helped us in the successful completion of the project.

We are thankful to project coordinator **Dr.P.Dileep Kumar Reddy, Assoc Professor** and **Mr.N.Phani Kumar, Asst Professor** for his guidance and regular schedules.

We would like to express our gratefulness and sincere thanks to **Dr.P.E.S.N. Krishna Prasad, Head, Dept of CSE**, for his kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to our beloved **Principal Dr.N.Sudhakar Reddy** for spending his valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thank the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

**K PoornaChandra (16BF1A0549)**

**V Bhuvridha (16BF1A0555)**

**G Harsha Vardhan (16BF1A0537)**

**C Ramya Sree Reddy (16BF1A0522)**

# CONTENTS

CHAPTER NO.	DESCRIPTION	PAGENO.
	Abstract	i
	List of Abbreviations	ii
	List of Figures	iii
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>PROJECT DESCRIPTION</b>	<b>2</b>
	2.1 Problem Definition	2
	2.2 Project Details	2
<b>3</b>	<b>COMPUTATIONAL ENVIRONMENT</b>	<b>6</b>
	3.1 Software Specification	6
	3.2 Hardware Specification	6
	3.3 Software Features	6
<b>4</b>	<b>FEASIBILITY STUDY</b>	<b>10</b>
	4.1 Technical Feasibility	10
	4.2 Operational Feasibility	11
	4.3 Economical Feasibility	11
<b>5</b>	<b>SYSTEM ANALYSIS</b>	<b>12</b>
	5.1 Existing System	12
	5.2 Proposed System	12
	5.2.1 Algorithm for proposed system	12
<b>6</b>	<b>SYSTEM DESIGN</b>	<b>13</b>
	6.1 UML Diagrams	
<b>7</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>20</b>
	7.1 Modules	
<b>8</b>	<b>TESTING</b>	<b>21</b>
	8.1 Unit Testing	21
	8.2 Integration Testing	22
	8.3 Functional Testing	22
	8.4 System Testing	22
	8.5 Acceptance Testing	23

<b>9</b>	<b>SAMPLE SOURCECODE</b>	<b>24</b>
<b>10</b>	<b>SCREEN LAYOUTS</b>	<b>42</b>
<b>11</b>	<b>CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>44</b>
<b>12</b>	<b>BIBLIOGRAPHY</b>	<b>45</b>

## **ABSTRACT**

Sentiment Analysis is a major subject in machine learning which aims to extract subjective information from the textual reviews. The field of sentiment analysis is closely tied to natural language processing and text mining. It is the task of identifying whether the opinion expressed in a text is positive or negative or neutral. It is used to determine the attitude of the reviewer with respect to the various topics or the overall polarity of reviews. The objective of this paper is to extract features from the product reviews and classify reviews into positive, negative and neutral. Social media websites have emerged as one of the platforms to raise user's opinions and influence the way any business is commercialized. Opinion of people matters a lot to analyze how the propagation of information impacts the lives in a large-scale network. It is simple and efficient Neural Language Model approach for text classification that relies only on unsupervised word representation inputs. For the classification of sentiments whether being positive, negative or neutral we use the best algorithm of Neural networks. By using all these techniques we try to find the maximum accuracy than others.

## LIST OF ABBREVIATIONS

1	<b>JVM</b>	:	Java virtual machine
2	<b>JDBC</b>	:	Java database connectivity
3	<b>UML</b>	:	Unified Modeling language
4	<b>IDE</b>	:	Integrated Development Environment
5	<b>XML</b>	:	Extension Markup Language
6	<b>DFD</b>	:	Data Flow Diagram
7	<b>API</b>	:	Application Program Interface



## LIST OF FIGURES

FIG NO.	FIGURE NAME	PAGE NO.
6.1	Class Diagram	14
6.2	Use case diagram	15
6.3	Sequence Diagram	16
6.4	Activity Diagram	17
6.5	Component Diagram	18
6.6	Deployment Diagram	19

## 1. INTRODUCTION

Internet is one of the drastically developing areas. People are often communicating, discussing and sharing information through internet. Due to these reasons internet is one of the essential part of human life. The information in it covers a wide range of areas such as academic information, feedback or opinion about products, comments about social issues etc. It helps people to think and make decision in many things. Majority of people always listen to others opinion before taking analysis is one of the research areas. In this information gathering is performed. And the information which is gathered will be analysed in order to determine the sentiment of the information such as negative sentiment or positive sentiment. One of the applications of this area is in product purchasing, before purchasing a product people will often enquire about the opinion of the product by other people. In this paper we are presenting methods to extract the sentiment of text data about movie or particular product. The objective of this paper is to predict the sentiment of review about the movie or particular product. Also we are interested in knowing the effective feature that can provide better result as well as the best feature selection method. It is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. To the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Currently, sentiment analysis is a topic of great interest and development since it has many practical applications. Since publicly and privately available information over Internet is constantly growing, a large number of texts expressing opinions are available in review sites, forums, blogs, and social media. It is the process of using natural language processing, text analysis, and statistics to analyze customer sentiment. The best businesses understand the sentiment of their customers—what people are saying, how they're saying it, and what they mean. Customer sentiment can be found in tweets, comments, reviews, or other places where people mention your brand. It is the domain of understanding these emotions with software, and it's a must-understand for developers and business leaders in a modern workplace. Customer service agents often use sentiment or intent analysis to automatically sort incoming user email into "urgent" or "not urgent" buckets based on the sentiment of the email, proactively identifying frustrated users. The agent then directs their time toward resolving the users with the most urgent needs first. As customer service becomes more and more automated through machine learning, understanding the sentiment and intent of a given case becomes increasingly important.

## 2. PROJECT DESCRIPTION

### PROJECT DEFINITION

Sentimental analysis is a way to analyze the subjective information in the text, which interpret and classify the emotions (positive or negative) within the text data using text analysis techniques. It is a procedure by which information is extracted from the appraisals and emotions of the people in regards to entities, events and attributes. In decision making, the opinions of others have a significant effect on customers ease, choices with regards to online shopping, choosing events, products, entities.

### PROJECT DETAILS

- **DATA**
- **RESOURCES**
- **PREPROCESSING**
- **DEEP LEARNING**

In this process, it identifies and categorizes opinions expressed in piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product is positive or negative.

### DATA

To gather the data many options are possible. In some previous paper researches, they built a program to collect automatically text based on two classes (positive and negative). Others make their own datasets etc.

Data should be collected from any open source websites into two classes positive and negative.

- Positive with 1.
- Negative with 0.

Item ID	Sentiment	Sentiment text
1	1	Although I didn't like Stanley and Iris tremendously as a film, I did admire the acting. Jane Fonda and Robert De Niro are great in this movie. I haven't been a fan of Fonda's work but here she is delicate and strong at the same time. De Niro has the ability to make every role he portrays into acting gold. He gives a great performance in this film and there is a great scene where he has to take his father to a home elderly people because he can't care for

## Sentimental Analysis using RNN

		him anymore that will break your heart. I wouldn't really recommend this as a great cinematic entertainment, but I will say you won't see
2	0	This fanciful horror flick has Vincent price playing a mad magician that realizes his vocational talents have been sold to another. He devise ways of avenging all those that have wronged him. His master scheme seems to back fire on him. Price is little below par compared to his master pieces, but is still the only reason to watch this thriller. Supporting cast includes Patrick O'Neal, Mary Murphy, Eva Gabor and Jay Novello.
3	1	Fair drama/love story movie that focuses on the lives of blue collar people finding new life through new love. The acting here is good but the film fails in cinematography, screenplay, directing and editing. The story is only average at best. This film will be enjoyed by Fonda and De Niro fans and by people who love middle age love stories where in the courtship is on a more wiser and cautions level. It would also be interesting for people who are interested on the subject matter regarding illiteracy
4	0	I saw the capsule comment sais "great acting."In my opinion, these are two great actors giving horrible performances, and with zero chemistry with one another, for a great director in his all-time worst effort. Robert De Niro has to be the most ingenious and insightful illiterate of all time. Jane Fonda's performance uncomfortably drifts all over the map as she clearly has no handle on this character, mostly because the character is so poorly written. Molasses-like would be too swift an adjective for this film's excruciating pacing. Although the film's intent is to be an uplifting story of curing illiteracy, watching it is true "bummer". I give it 1 out of 10, truly one of the worst 20 movies for its budget level that I have ever seen.

# Sentimental Analysis using RNN

---

## RESOURCES

Collecting data from kaggle open source, Data sets contains both positive and negative reviews.

- **MOVIE REVIEWS:** Taking both negative and positive feedbacks.

**python3.7** which is a programming language widely used in data science and scikit-learn a very complete and useful library for machine learning containing every techniques and methods.

**Download.py** is to download and extract the data from open source website.

**Imbd** is to read and process the downloaded data to the program.

## PREPROCESSING

It is very important since all the modification that we are going to do during the process will directly impact the classifier's performance. The preprocessing includes cleaning, normalization, transformation, feature extraction and selection. The result of pre-processing will be consistent and uniform data that are workable to maximize the classifier's performance.

### Tokenizer:

Text tokenization is a utility class. This class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf- idf..

We use this method to create tokens of words that can be fed as input. The tokenized words are further processed.

### Creating Embedded vector:

It contains complete code to train word embeddings from scratch on a small dataset, and to visualize these embeddings using the Embedding Projector.

Machine learning models take vectors (arrays of numbers) as input. When working with text, the first thing we must do come up with a strategy to convert strings to numbers (or to "vectorize" the text) before feeding it to the model.

### Padding:

Convolutional layers induce spatial hierarchy. That is, generally speaking, that can reduce the size of our input data for every layer the data passes through – allowing neural networks to learn both very specific and very abstract aspects of our input data.

# Sentimental Analysis using RNN

---

However, sometimes we don't want this to happen: we want the size of our input data to stay the same. In that case, padding can help by adding zeros, constants or different numbers around the reduced input. This way, the size of your input data remains the same.

## DEEP LEARNING

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. It is also known as deep neural learning or deep neural network.

### Keras Model

Keras is an open source and high level neural networks API, written in python and capable of running on top of Tensor Flow, CNTK, or Theano. It is designed to be modular fast and easy to use.

### Sequential model

Sequential is a easiest way to build a model In Keras. It allows you to build a model layer by layer. Each layer has weights that correspond to the layer the follows it. It is defined in the sequential class which is linear stack of layers.

### Sigmoid activation function

It is also called as the logistic function, is traditionally a very popular activation function for neural networks. It exists between 0 and 1 especially used for models where we predict the probability as an output.

---

## 3. COMPUTATIONAL ENVIRONMENT

### SOFTWARE SPECIFICATIONS

□ Operating System	:	Windows/Linux
□ Programming Language	:	Python Environment
□ Browser	:	Mozilla/IE/Chrome
□ Framework	:	TensorFlow, NumPy
□ Deep Learning Module	:	Keras
□ Web Application	:	Flask
□ IDE	:	Spyder
□ Python Version	:	3.8.2

### HARDWARE SPECIFICATIONS

□ RAM Capacity	:	12GB
□ Processor	:	Any 8 <sup>th</sup> Gen Processor
□ Graphics Card	:	4GB
□ Speed	:	2.5GHZ or more

### SOFTWARE FEATURES

Why we use “PYTHON”:

It is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. It is designed with features to facilitate data analysis and visualization. Python's expansive library of open source data analysis tools, web frameworks, and testing instruments make its ecosystem one of the largest out of any programming community. It is an accessible language for new programmers because the community provides many introductory resources. The language is also widely taught in universities and used for working with beginner-friendly devices such as the Raspberry Pi.

Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further makes it easier for you to keep the code base readable and application maintainable. The syntax rules of Python allow you to express concepts without writing additional code.

## Sentimental Analysis using RNN

---

Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming

Even we can use Python interpreters to run the code on specific platforms and tools. It allows you to you to run the same code on multiple platforms without recompilation. The feature makes it easier for you to make changes to the code without increasing development time. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle andCherrypy.

Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. You have to speed up the Python application by making changes to the application code or using custom runtime.

### **About PYTHON:**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

It is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Python can serve as a scripting language for web applications, for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras. Due to Python's user-friendly conventions and easy-to-understand language, it is commonly used as an intro

---



## Sentimental Analysis using RNN

---

language into computing sciences with students. This allows students to easily learn computing theories and concepts and then apply them to other programming languages such as Java.

### **Features:**

#### **Simple:**

Python is a simple and minimalistic language. This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

#### **Easy to Learn:**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax.

#### **Free and Open Source:**

Python is an example of a **FLOSS** (Free/Libre and Open Source Software). It is based on the concept of a community which shares knowledge. It freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.

#### **High-level Language:**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

#### **Portable:**

Due to its open-source nature, Python has been ported to many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on GNU/Linux, Windows, Macintosh, Solaris, Palm OS, Windows CE and PocketPC!

# Sentimental Analysis using RNN

---

## **Interpreted:**

A program written in a compiled language like C or C++ is converted from the source language into binary code i.e.( 0's and 1's) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it.

## **Object Oriented:**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

## **Extensible:**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

## **Embeddable:**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

## **Extensive Libraries:**

It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, HTML, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed.

## 4.FEASIBILITY STUDY

A feasibility study is a preliminary study which investigates the information of prospective users and determine the resources requirements, costs, benefits and feasibility of a proposed system. A feasibility study takes account into various constraints within which the system should be implemented and operated. In this stage, the resource needed for the implementation such as computing equipment manpower and costs are estimated. The estimated are compared with available resources and cost benefit analysis of the system is made. The feasibility analysis activity involves the analysis of a problem and collection of all the relevant information relating to the project. The main objectives of feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility or not. It is to make sure that the input data which are required for the project are available. Thus we evaluated the feasibility of the system in terms of following categories:

- Technical feasibility
- Operational feasibility
- Economic feasibility

### TECHNICAL FEASIBILITY

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs et, A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system. Before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the system. Is the required technology available? Our system “SENTIMENTAL ANALYSIS” is technically feasible sine all the required tools are easily available. Python and flask python module, Spyder software can be easily handled. Although all tools seem to be easily available there are challenges too.

## **OPERATIONAL FEASIBILITY**

Proposed project is beneficial only if it can be turned into information system that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to implementation?

The proposed was to make a simplified web application. It is simpler to operate and can be used in any web pages. It is free and not costly to operate.

## **ECONOMIC FEASIBILITY**

Economic system attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information.

---

## 5.SYSTEM ANALYSIS

### EXISTING SYSTEM

There are many traditional methods which provide the benefit of having knowledge and update on latest technology in medical field. Some of the methods are discussed below:

#### Surveys and Questionnaires

A set of questions with a choice of answers, devised for healthcare surveys helps a lot to gain knowledge on effectiveness of the treatment. There are various surveys such as patient experience survey which captures every patient's voice using email, phone and mail to provide a deeper data and a clearer picture of patient perception for the entire care. Employee and physician surveys are of equal importance to improve the quality of care.

#### Interviews

Watching interviews of doctors, patients and drug developers helps in being update with latest technology and effectiveness of the treatments available.

#### Feedback

Now—a—days many pharmaceutical companies are asking for consumer feedback. These pharmaceutical companies make record of both negative and positive feedback and use these records to come up with a better optimized drug.

However, the existing traditional methods are time consuming and as they are performed manually, their efficiency decreases with an increase in the frequency. Another important problem is that a patient may not convey information properly in a formal setting when compared to an informal medium like social media.

### PROPOSED SYSTEM

The Proposed System is on Sentiment Analysis using Recurrent Neural networks in Keras with tensorflow as backend.

#### Algorithm for Proposed System:

- First we need to convert the raw text-words into so-called tokens which are integer values.
- Then we convert these integer-tokens into so-called embeddings which are real-valued vectors, whose mapping will be trained along with the neural network, so as to map words with similar meanings to similar embedding-vectors.
- Then we input these embedding-vectors to a Recurrent Neural Network which can take sequences of arbitrary length as input and output a kind of summary of what it has seen in the input.
- Output from the RNN is squashed by an activation function. (Sigmoid in this case)
- Output is between 0 and 1. { 0: highly negative, 1 : highly positive }

## 6.SYSTEM DESIGN

The System Design includes the management of the workers information, their login details, authorization, details of the messages with respect to the user logged into and their accessibility information in the database server.

The design also needed to provide the communication channel to the user to communicate with the website to know better details about the system. Authorization and authentication services are performed most for this purpose.

### 6.1 UML DIAGRAMS

What is UML

The Unified Modeling Language (UML) is a standard language for specifying, Visualizing, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML use mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

#### Goals:

The primary goals in the design of the UML are as follows:

1. Be independent of particular programming languages and development processes.
2. Provide a formal basis for understanding the modeling language.
3. Encourage the growth of the OO tools market.
4. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
5. Integrate best practices.

## CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

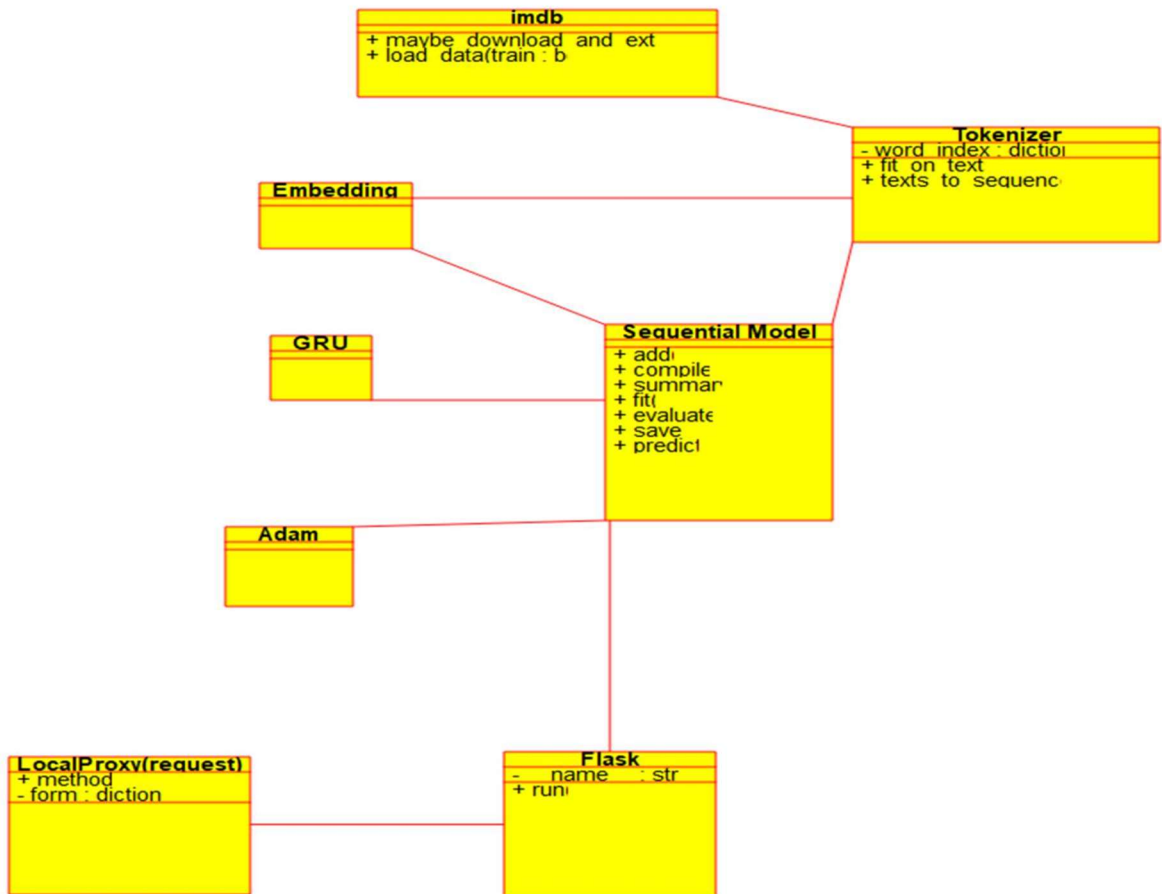


Fig 6.1 Class Diagram

# Sentimental Analysis using RNN

## USE CASE DIAGRAM:

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted as shown in the Fig:6.2 Use Case Diagram.

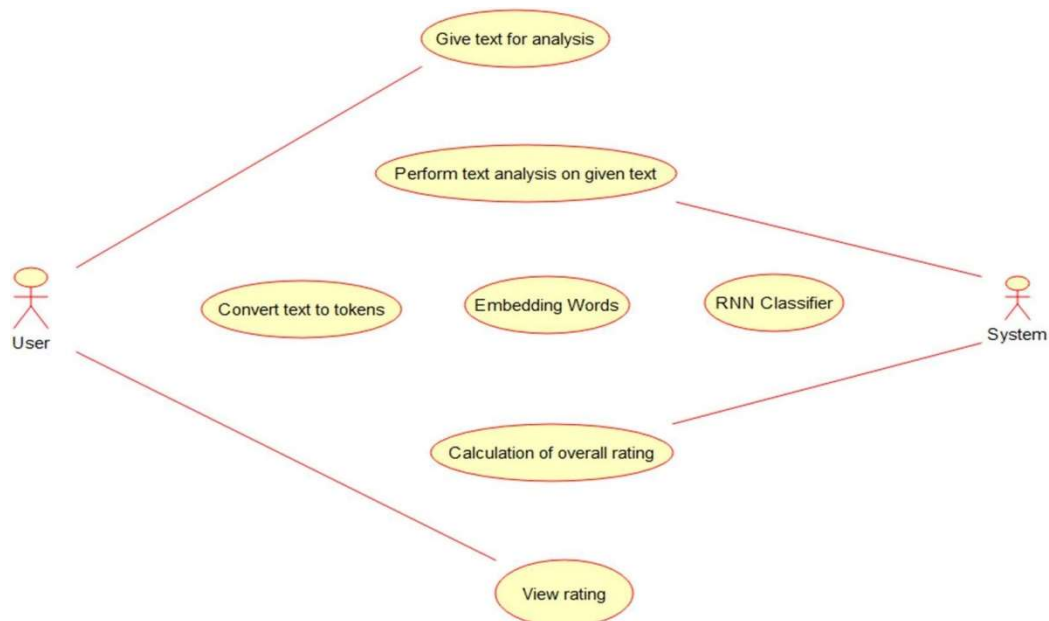


Fig:6.2 Use Case Diagram



## SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

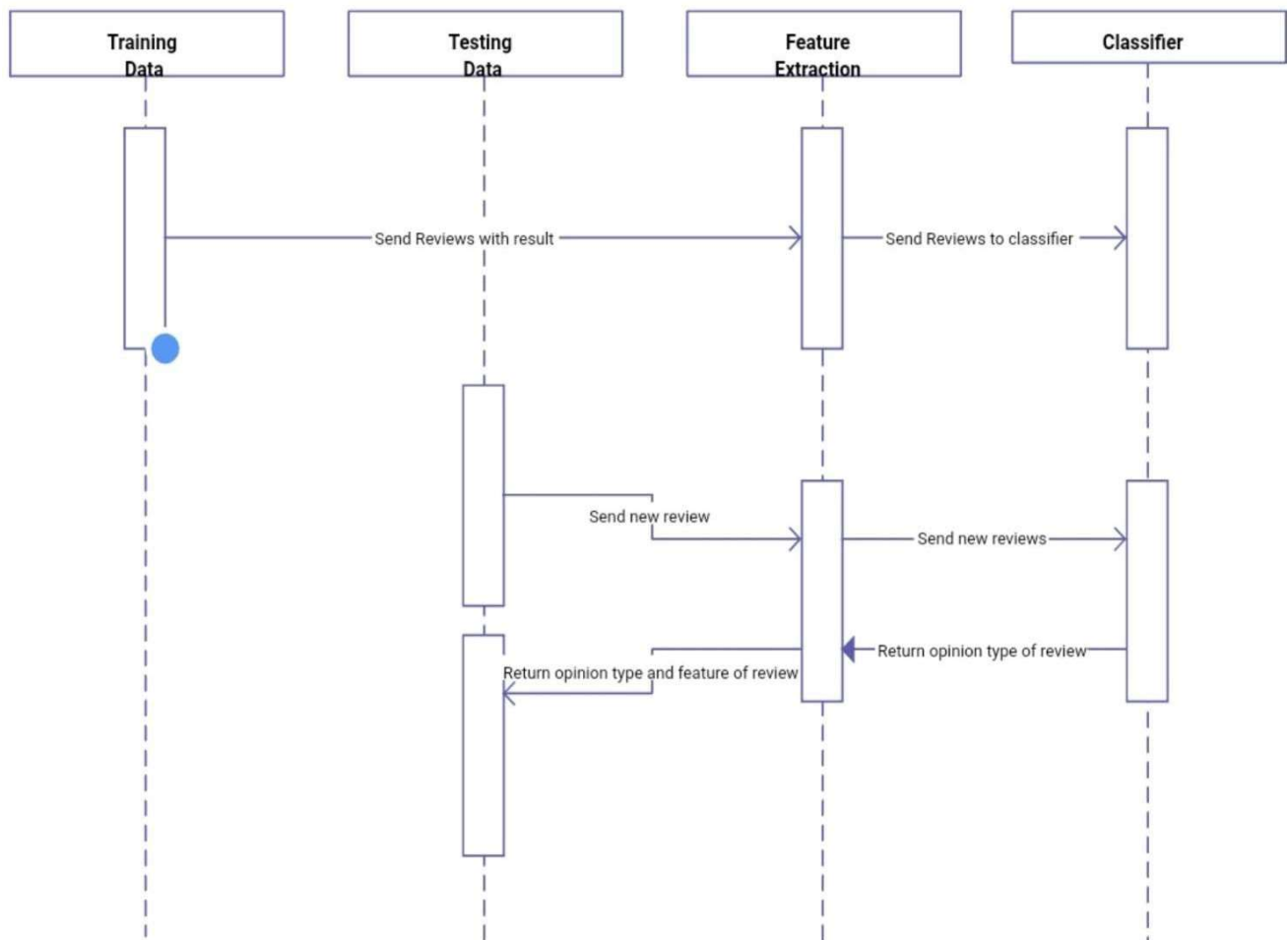


Fig 6.3 Sequence Diagram

### ACTIVITY DIAGRAM

Activity diagram is basically a flow chart to represent the flow from one activity to other activity. they are graphical representations of workflow of stepwise activities and support for choice, iteration and concurrency. In the Unified Modeling Language, are intended to model both computational and organizational processes(i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

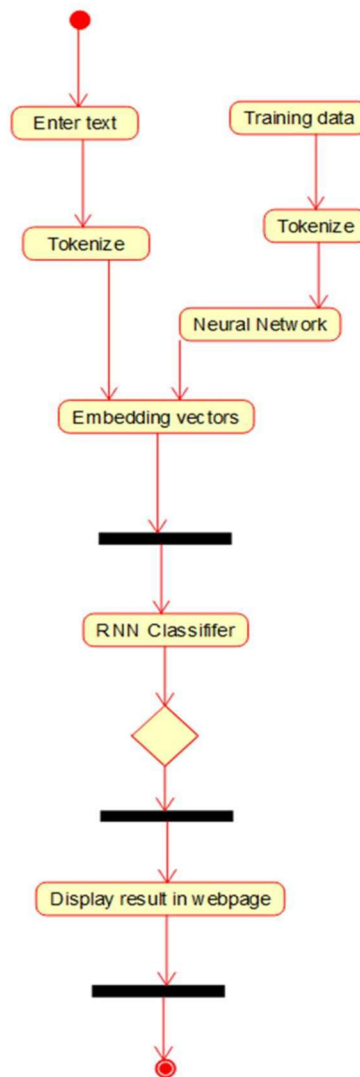


Fig 6.4 Activity Diagram

---

### COMPONENT DIAGRAM:

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executable, libraries etc. So, the purpose of this diagram is different. Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.

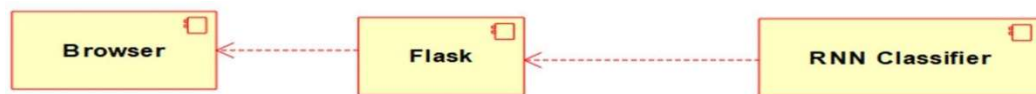


Fig 6.5 Component Diagram

---

### DEPLOYMENT DIAGRAM

The Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.



Fig 6.6 Deployment Diagram

## 7. SYSTEM IMPLEMENTATION

### MODULES

There are five modules in this application. They are:

- Download
- imdb
- Tokenizer
- Sequential model
- Flask

#### **Modules:**

The system uses following modules to provide required functionality.

#### **Download:**

This module is used to extract text reviews or text from the given url.

#### **Imdb:**

This module provides functionality to load data. This also has methods to load training data and test data.

#### **Tokenizer:**

Since computer cannot understand text, first we need to convert the raw text-words into so-called tokens which are integer values. Tokenizer provides the required functionality.

#### **Sequential model:**

This model is used to create Recurrent Neural Network. It provides methods to add different types of network layers sequentially, each performing unique operation on the tokenized text data to learn the mapping and understanding the context. It provides method to predict output of unseen input.

#### **Flask:**

This is a python framework used to create web pages. This helps us to take the corresponding input from the user and to render the output of our model through web page in browser.

## 8. TESTING

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

### UNIT TESTING:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Each module can be tested in the following two Strategies:

#### Black box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has following categories: been uses to find errors in the following categories:

- i. Incorrect or missing functions Interface errors
- ii. Errors in data structure or external database access
- iii. In this testing only, the output is checked for correctness.
- iv. The logical flow of the data is not checked.
- v. Performance errors Initialization and termination errors.

#### White box Testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- i. Execute all logical decisions on their true and false sides.
- ii. Guarantee that all independent paths have been executed.
- iii. Execute all loops at their boundaries and within their operational bounds

## **INTEGRATION TESTING:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## **FUNCTIONAL TESTING:**

Functional tests provide systematic demonstration that functions tested are available specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current test is determined.

## **SYSTEM TESTING:**

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

### **Testing Approach:**

Testing can be done in two ways:

- ☐ Bottom up approach
- ☐ Top down approach

### **Bottom up Approach:**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and embedded within it provides the needed data so that the module is asked to perform the way it will when used within the larger system. When bottom level modules are tested attention turns to those on the next use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

### **Top down Approach:**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

## **ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.



---

## 9. SAMPLE SOURCE CODE

### TRAINING\_FILE.py

```
get_ipython().run_line_magic('matplotlib', 'inline')
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from scipy.spatial.distance import cdist
# from tf.keras.models import Sequential # This does not work!
from keras.models import Sequential
from keras.layers import Dense, GRU, Embedding
from keras.optimizers import Adam
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

import imdb # this is helper package to download and load the imdb dataset

imdb.maybe_download_and_extract() #Downloading and Extracting the dataset

x_train_text, y_train = imdb.load_data(train=True) #loading train data
x_test_text, y_test = imdb.load_data(train=False) # loading test data

print("Train-set size: ", len(x_train_text))
print("Test-set size: ", len(x_test_text))

data_text = x_train_text + x_test_text

x_train_text[100] # looking at an example text
```

## Sentimental Analysis using RNN

---

```
y_train[100]

num_words = 10000

tokenizer = Tokenizer(num_words=num_words)

get_ipython().run_cell_magic('time', '', 'tokenizer.fit_on_texts(data_text)')

tokenizer.word_index
# This is ordered by the number of occurrences of the words in the data-set.
# These integer-numbers are called word indices or "tokens" because they uniquely
# identify each word in the vocabulary.

x_train_tokens = tokenizer.texts_to_sequences(x_train_text) # converting all the text in training data to
tokens

x_train_text[1] # actual text without tokens

np.array(x_train_tokens[1]) # text after tokenizing

x_test_tokens = tokenizer.texts_to_sequences(x_test_text) # converting text data into tokens

# # Padding
#
# The Recurrent Neural Network can take sequences of arbitrary length as input, but in order to use a
# whole batch of data, the sequences need to have the same length.
# But we can't take the length of longest review and pad that many zeros to the shorter reviews because
# it will take lot of memory so we have to figure out a particular length that will be sufficient for most of
# our data

num_tokens = np.array([len(tokens) for tokens in x_train_tokens + x_test_tokens])
#making a list to store the lengths of tokenized reviews in both training and test data set
```

## Sentimental Analysis using RNN

---

```
np.mean(num_tokens) #calculating average length

np.max(num_tokens) # maximum length of any tokenized review

np.min(num_tokens)# minimum

# ## Visualizing Token Lengths

import plotly.plotly as py
import plotly.graph_objs as go

trace0 = go.Box(
    y=num_tokens
)
data=[trace0]
py.iplot(data)

# #### We can see in the above box plot that most of the token lengths are between 0 and 500 but also
# we have some outliers which go upto 2000+

max_tokens = np.mean(num_tokens) + 2 * np.std(num_tokens)
max_tokens = int(max_tokens)
max_tokens

#The max number of tokens we will allow is set to the average plus 2 standard deviations.

# We have already seen that most of our data is between 0 and 500 length but we can verify it

str(np.sum(num_tokens<max_tokens) / len(num_tokens) * 100) +' %'
```

## Sentimental Analysis using RNN

---

```
# ## When we pad data we need to decide where to pad the data, whether pad in the beginning or in the
end
#
# ##### - If we pad at the end then there might be a chance that RNN might get confused seeing lot of
zeroes after processing some data
#
# ##### - So we need to pad in beginning

pad = 'pre'

x_train_pad = pad_sequences(x_train_tokens, maxlen=max_tokens,
                           padding=pad, truncating=pad)

x_test_pad = pad_sequences(x_test_tokens, maxlen=max_tokens,
                           padding=pad, truncating=pad)

x_train_pad.shape

x_test_pad.shape

np.array(x_train_tokens[1]) # before padding

np.array(x_train_pad[1]) # After Padding

num_tokens_pad = np.array([len(tokens) for tokens in x_train_pad + x_test_pad])

import plotly.plotly as py
import plotly.graph_objs as go

trace0 = go.Box(
```

## Sentimental Analysis using RNN

---

```
y=num_tokens_pad
)
data=[trace0]
py.iplot(data)

## We can see in the above plot that now all the data have same length (544)

# ## Alternatives to padding
#
# ### There are various options if you don't want to do padding:
#
# ##### 1) Make your batch size equal to 1 i.e feed data one by one into RNN
#
# ##### 2) Grouping sequences of same lengths i.e all the sequences of particular lengths like 100 or
# 500 together
#
# ## Creating the Recurrent Neural Network using Keras
#
model = Sequential()

# The first layer in the RNN is a so-called Embedding-layer which converts each integer-token into a
vector of values
# Tokenized data is huge from 0 to vocabulary length (10000 in this case) and value of tokens(integer
values) does not make any sense so tokens are converted into embedded vector which is a vector that
maps words with similar semantic meanings and of length usually 100-300
#
embedding_size = 8

model.add(Embedding(input_dim=num_words,
output_dim=embedding_size,
```

## Sentimental Analysis using RNN

---

```
input_length=max_tokens,  
    ))
```

```
model.add(GRU(units=16, return_sequences=True))#layer below will be processing sequences so thats  
why return_sequences=True
```

```
model.add(GRU(units=8, return_sequences=True))#layer below will be processing sequences so thats  
why return_sequences=True
```

```
model.add(GRU(units=4))# now we don't need sequences because in the next layer we will predict the  
output
```

```
model.add(Dense(1, activation='sigmoid'))# fully connected layer with output =1 since we will predict  
either positve or negative
```

```
optimizer = Adam(lr=1e-3)
```

```
model.compile(loss='binary_crossentropy',  
              optimizer=optimizer,  
              metrics=['accuracy'])
```

```
model.summary()
```

```
get_ipython().run_cell_magic('time', '', 'history=model.fit(x_train_pad, y_train,\nvalidation_split=0.05, epochs=3, batch_size=50)')
```

```
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
epochs = range(1, len(acc) + 1)
```

## Sentimental Analysis using RNN

---

```
trace0 = go.Scatter(  
    x = list(epochs),  
    y = list(acc),  
    mode = 'lines',  
    name = 'training accuracy'  
)  
  
trace1= go.Scatter(  
    x = list(epochs),  
    y = list(val_acc),  
    mode = 'lines',  
    name = 'Validation accuracy'  
)  
  
layout = go.Layout( title='Training and Validation Accuracy')  
  
data = [trace0,trace1]  
fig = go.Figure(data=data, layout=layout)  
  
py.iplot(fig)  
  
trace0 = go.Scatter(  
    x = list(epochs),  
    y = list(loss),  
    mode = 'lines',  
    name = 'training loss'  
)  
  
trace1= go.Scatter(  
    x = list(epochs),  
    y = list(val_loss),
```

## Sentimental Analysis using RNN

---

```
mode = 'lines',
name = 'Validation loss'
)

layout = go.Layout( title='Training and Validation loss')

data = [trace0,trace1]
fig = go.Figure(data=data, layout=layout)

py.iplot(fig)

get_ipython().run_cell_magic('time', '', 'result = model.evaluate(x_test_pad, y_test)')
print("Accuracy: {0:.2%}".format(result[1]))
```

### **APPLICATION\_FILE.py**

```
from keras.models import load_model
model = load_model('output.h5')
model.summary()

from flask import Flask, request
app = Flask(__name__)

#%matplotlib inline
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from scipy.spatial.distance import cdist
# from tf.keras.models import Sequential # This does not work!
```



## Sentimental Analysis using RNN

---

```
from keras.models import Sequential
from keras.layers import Dense, GRU, Embedding
from keras.optimizers import Adam
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import download
import imdb
imdb.maybe_download_and_extract()

x_train_text, y_train = imdb.load_data(train=True) #loading train data
x_test_text, y_test = imdb.load_data(train=False) # loading test data
print("Train-set size: ", len(x_train_text))
print("Test-set size: ", len(x_test_text))

data_text = x_train_text + x_test_text
num_words = 10000
tokenizer = Tokenizer(num_words=num_words)
tokenizer.fit_on_texts(data_text)
tokenizer.word_index

x_train_tokens = tokenizer.texts_to_sequences(x_train_text)
x_test_tokens = tokenizer.texts_to_sequences(x_test_text) # converting text data into tokens
#print(x_train_text[2])
#print(np.array(x_train_tokens[0]))
num_tokens = np.array([len(tokens) for tokens in x_train_tokens + x_test_tokens])
print(np.mean(num_tokens))
print(np.max(num_tokens))
print(np.min(num_tokens))

max_tokens = np.mean(num_tokens) + 2 * np.std(num_tokens)
max_tokens = int(max_tokens)
print(max_tokens)
```

## Sentimental Analysis using RNN

---

```
print(str(np.sum(num_tokens<max_tokens) / len(num_tokens) * 100) +' %')
```

```
pad = 'pre'
```

```
x_train_pad = pad_sequences(x_train_tokens, maxlen=max_tokens,  
                             padding=pad, truncating=pad)
```

```
x_test_pad = pad_sequences(x_test_tokens, maxlen=max_tokens,  
                             padding=pad, truncating=pad)
```

```
num_tokens_pad = np.array([len(tokens) for tokens in x_train_pad + x_test_pad])
```

```
result = model.evaluate(x_test_pad, y_test)
```

```
@app.route("/", methods=["GET", "POST"])
```

```
def adder_page():
```

```
    errors = ""
```

```
    if request.method == "POST":
```

```
        try:
```

```
positive_review = (request.form["input"])
```

```
    except:
```

```
        errors += "<p>{!r} is not a number.</p>\n".format(request.form["input"])
```

```
    if positive_review is not None:
```

```
        text=[positive_review]
```

```
        tokens = tokenizer.texts_to_sequences(text)
```

```
tokens_pad = pad_sequences(tokens, maxlen=max_tokens,
```

```
                             padding=pad, truncating='pre')
```

```
    a=model.predict(tokens_pad)
```

```
    return ""
```

```
<html>
```

```
    <body style="background-color: #EAF0F1; text-align: center; margin: 15%;">
```

```
        <div>
```

```
            <h2>Positive Review with a score of {} %</h2>
```

```
                <a href="/">Click here to calculate again</a>
```

```
        </div>
```

```
</body>
```

## Sentimental Analysis using RNN

---

```
</html>

    """.format(a[0]*100)
    return ""

<html>
<body style="background-color: #EAF0F1; text-align: center; margin: 12%;">
<div class="content">
<div class="heading">
<h1 style="text-align :center">
SentimentAnalysis-RNN
</h1>
</div>
<form method="post" action=".">
<p>Enter your text: </p>
<textarea name="input" rows="4" cols="40"></textarea><br><br>
<input type="submit" value="Do calculation"
        style="background-color:#2ecc72;
                border: none;
color: white;
                padding: 15px 32px;
                text-align: center;
                text-decoration: none;
                display: inline-block;
                font-size: 16px;
                box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19)" />
</form>
</div>
</body>
</html>

    """.format(errors=errors)

if __name__ == '__main__':
    app.run()
```

## Sentimental Analysis using RNN

---

### IMBD.py

```
#####  
#  
# Functions for downloading the IMDB Review data-set from the internet  
# and loading it into memory.  
#  
# Usage:  
# 1) Set the variable data_dir with the desired storage directory.  
# 2) Call maybe_download_and_extract() to download the data-set  
#    if it is not already located in the given data_dir.  
# 3) Call load_data(train=True) to load the training-set.  
# 4) Call load_data(train=False) to load the test-set.  
# 5) Use the returned data in your own program.  
#  
# Format:  
# The IMDB Review data-set consists of 50000 reviews of movies  
# that are split into 25000 reviews for the training- and test-set,  
# and each of those is split into 12500 positive and 12500 negative reviews.  
# These are returned as lists of strings by the load_data() function.  
#  
#####  
import os  
import download  
import glob  
#####  
  
# Directory where you want to download and save the data-set.  
# Set this before you start calling any of the functions below.  
data_dir = "data/"  
  
# URL for the data-set on the internet.
```

## Sentimental Analysis using RNN

---

```
data_url = "http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz"

#####

# Private helper-functions.

def _read_text_file(path):
    """
    Read and return all the contents of the text-file with the given path.
    It is returned as a single string where all lines are concatenated.
    """
    with open(path, 'rt', encoding='utf-8') as file:
        # Read a list of strings.
        lines = file.readlines()

        # Concatenate to a single string.
        text = " ".join(lines)
    return text

#####

# Public functions that you may call to download the data-set from
# the internet and load the data into memory.

def maybe_download_and_extract():
    """
    Download and extract the IMDB Review data-set if it doesn't already exist
    in data_dir (set this variable first to the desired directory).
    """
    download.maybe_download_and_extract(url=data_url, download_dir=data_dir)

def load_data(train=True):
    """
    Load all the data from the IMDB Review data-set for sentiment analysis.
    :param train: Boolean whether to load the training-set (True)
```

## Sentimental Analysis using RNN

---

or the test-set (False).

```
:return:    A list of all the reviews as text-strings,
            and a list of the corresponding sentiments
            where 1.0 is positive and 0.0 is negative.
"""

# Part of the path-name for either training or test-set.
train_test_path = "train" if train else "test"

# Base-directory where the extracted data is located.
dir_base = os.path.join(data_dir, "aclImdb", train_test_path)

# Filename-patterns for the data-files.
path_pattern_pos = os.path.join(dir_base, "pos", "*.txt")
path_pattern_neg = os.path.join(dir_base, "neg", "*.txt")

# Get lists of all the file-paths for the data.
paths_pos = glob.glob(path_pattern_pos)
paths_neg = glob.glob(path_pattern_neg)

# Read all the text-files.
data_pos = [_read_text_file(path) for path in paths_pos]
data_neg = [_read_text_file(path) for path in paths_neg]

# Concatenate the positive and negative data.
x = data_pos + data_neg

# Create a list of the sentiments for the text-data.
# 1.0 is a positive sentiment, 0.0 is a negative sentiment.
y = [1.0] * len(data_pos) + [0.0] * len(data_neg)

return x, y
```

```
#####
```

## Sentimental Analysis using RNN

---

### DOWNLOAD.py

```
#####  
#  
# Functions for downloading and extracting data-files from the internet.  
#  
# Implemented in Python 3.5  
#  
#####  
  
import sys  
import os  
import urllib.request  
import tarfile  
import zipfile  
  
#####  
  
def _print_download_progress(count, block_size, total_size):  
    """  
    Function used for printing the download progress.  
    Used as a call-back function in maybe_download_and_extract().  
    """  
  
    # Percentage completion.  
    pct_complete = float(count * block_size) / total_size  
  
    # Limit it because rounding errors may cause it to exceed 100%.  
    pct_complete = min(1.0, pct_complete)  
  
    # Status-message. Note the \r which means the line should overwrite itself.
```

## Sentimental Analysis using RNN

---

```
msg = "\r- Download progress: {0:.1%}".format(pct_complete)

# Print it.
sys.stdout.write(msg)
sys.stdout.flush()

#####

def download(base_url, filename, download_dir):
    """
    Download the given file if it does not already exist in the download_dir.

    :param base_url: The internet URL without the filename.
    :param filename: The filename that will be added to the base_url.
    :param download_dir: Local directory for storing the file.
    :return: Nothing.
    """

    # Path for local file.
    save_path = os.path.join(download_dir, filename)

    # Check if the file already exists, otherwise we need to download it now.
    if not os.path.exists(save_path):
        # Check if the download directory exists, otherwise create it.
        if not os.path.exists(download_dir):
            os.makedirs(download_dir)

    print("Downloading", filename, "...")

    # Download the file from the internet.
    url = base_url + filename
```



## Sentimental Analysis using RNN

---

```
file_path, _ = urllib.request.urlretrieve(url=url,
                                          filename=save_path,
                                          reporthook=_print_download_progress)

print(" Done!")

def maybe_download_and_extract(url, download_dir):
    """
    Download and extract the data if it doesn't already exist.
    Assumes the url is a tar-ball file.

    :param url:
        Internet URL for the tar-file to download.
        Example: "https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz"

    :param download_dir:
        Directory where the downloaded file is saved.
        Example: "data/CIFAR-10/"

    :return:
        Nothing.
    """

    # Filename for saving the file downloaded from the internet.
    # Use the filename from the URL and add it to the download_dir.
    filename = url.split('/')[-1]
    file_path = os.path.join(download_dir, filename)

    # Check if the file already exists.
    # If it exists then we assume it has also been extracted,
    # otherwise we need to download and extract it now.
```

## Sentimental Analysis using RNN

---

```
if not os.path.exists(file_path):
    # Check if the download directory exists, otherwise create it.
    if not os.path.exists(download_dir):
os.makedirs(download_dir)

    # Download the file from the internet.
file_path, _ = urllib.request.urlretrieve(url=url,
                                          filename=file_path,
                                          reporthook=_print_download_progress)

print()
print("Download finished. Extracting files.")

    if file_path.endswith(".zip"):
        # Unpack the zip-file.
zipfile.ZipFile(file=file_path, mode="r").extractall(download_dir)
    elif file_path.endswith(("tar.gz", ".tgz")):
        # Unpack the tar-ball.
tarfile.open(name=file_path, mode="r:gz").extractall(download_dir)

    print("Done.")
else:
print("Data has apparently already been downloaded and unpacked.")

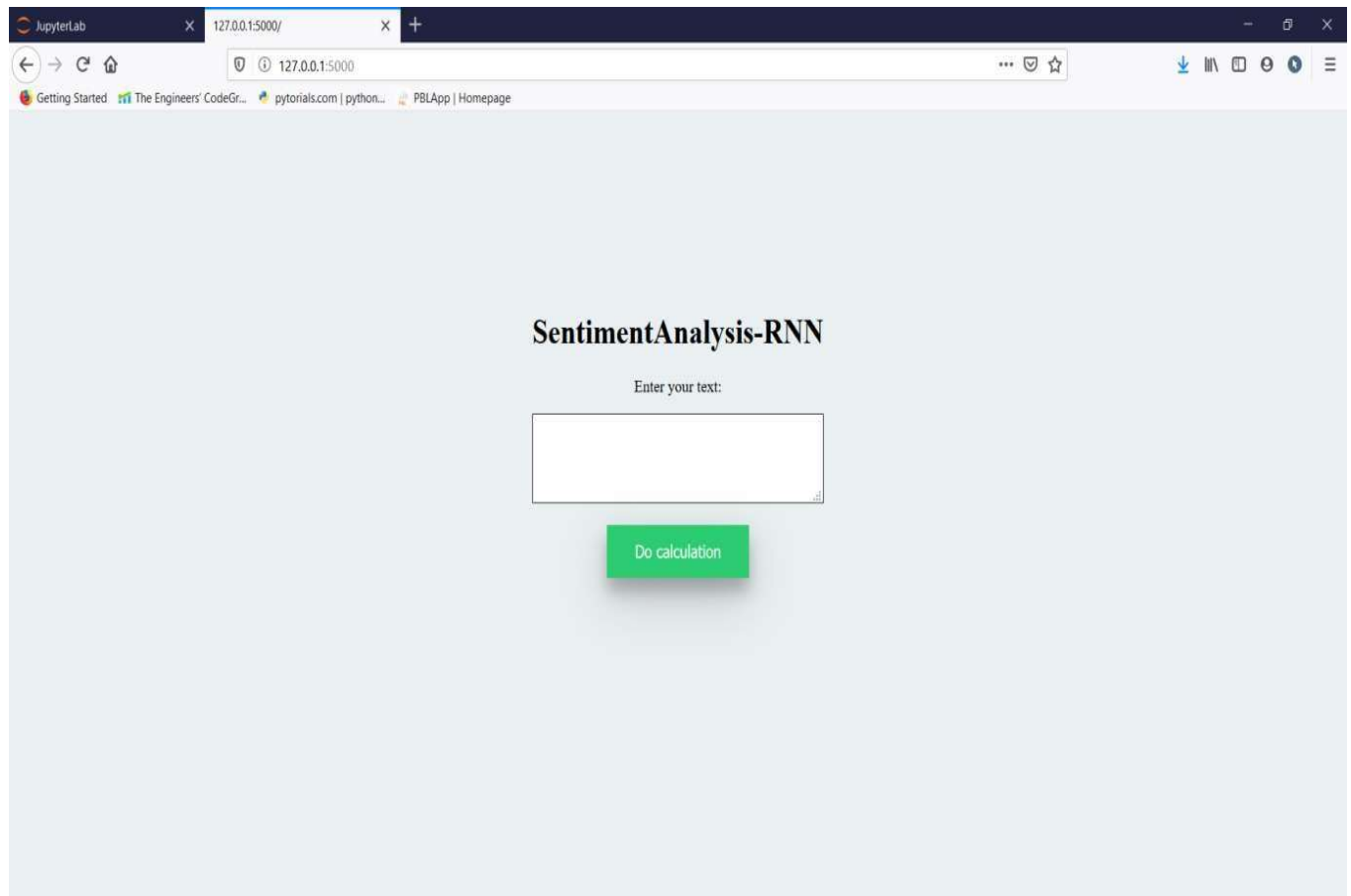
#####
```

## 10. SCREEN LAYOUTS

### Screen Layout 1: Home Page

#### Description:

User has to enter the review in the text box available and click on submit button to process the review.

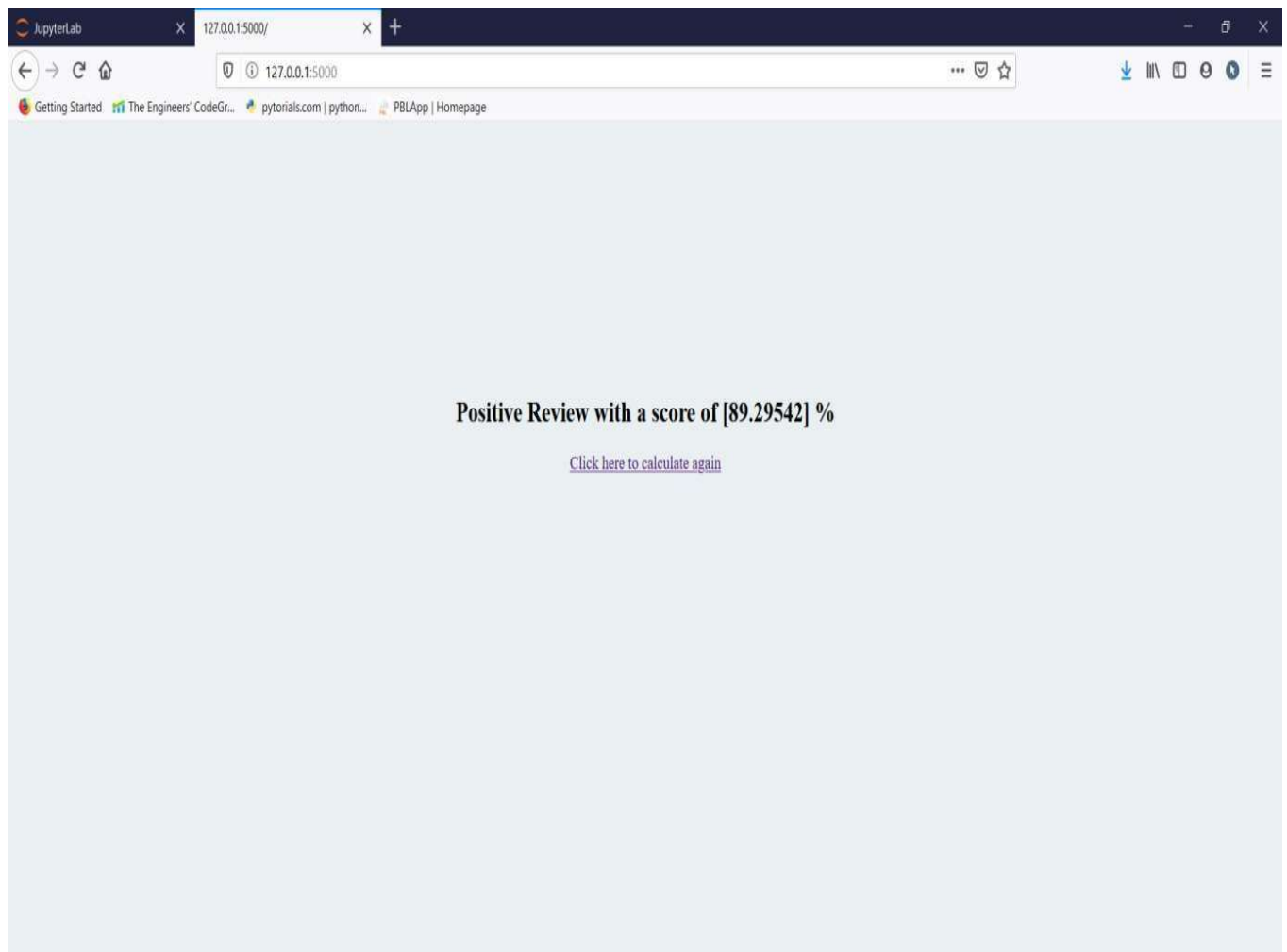


# Sentimental Analysis using RNN

## Screen Layout 2: Result Page

### Description:

This page displays the evaluated results of the sentiment of the review with score of it.



## 11. CONCLUSION AND FUTURE ENHANCEMENTS

It is difficult to read the whole review and predict the final sentiment when it comes to a very long review. Using this we can reduce the time and effort in prediction. It just takes a matter of seconds to predict the type of review and it is very easy to use.

First, we convert the review into tokens and add padding to the tokens to make all of them equal in terms of scaling. Then we train the padded tokens into RNN using sequential model with sigmoid activation, which tells result in true or false.

### **Future Enhancements:**

We plan to make this type of analysis for all the domains. Improved User-Interface in future shall increase user experience and this shall improve the way they read and respond to their reviewers or feedbacks.

## 12.BIBILOGRAPHY

### Web Resources:

- [1] <https://www.machinecurve.com/index.php/2020/02/08/how-to-use-padding-with-keras/>
- [2] <https://keras.io/getting-started/sequential-model-guide/>
- [3] "[http://ai.stanford.edu/~amaas/data/sentiment/aclImdb\\_v1.tar.gz](http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz)"
- [4] <https://blog.keras.io/keras-as-a-simplified-interface-to-tensorflow-tutorial.html>
- [5] <https://machinelearningmastery.com/data-preparation-variable-length-input-sequences-sequence-prediction/>

### Book Resources:

- [1] Laura Auria, Rouslan A Moro, "Support Vector Machines", 2008, web, 20 August 2014.
- [2] H Kim, P Howland, and H Park "Dimension reduction in text classification with support vector machines", Journal of Machine Learning Research, 2005.
- [3] Pang- Ning Tan, Michael Steinbach, Vipin Kumar,"Introduction to Data Mining", Pearson publications.