PROJECT REPORT ON VIRUDH A FAKE NEWS DETECTION SOLUTION

PREPARED BY
Akarshan Gandotra
(14CSU016)

PREPARED FOR
Department of Computer Science
School of Engineering and Technology
The NorthCap University
Gurugram, Haryana

APRIL 2018

A Project Report
on
VIRUDH
A FAKE NEWS DETECTION SOLUTION

submitted in partial fulfillment of the requirement gor award of the degree

of

Bachelor of Technology
in Computer Science Engineering and Information Technology

by

Akarshan Gandotra (14CSU016)

Under supervision of
Dr. Shilpa Mahajan

Department of CSE & IT
The NorthCap University, Gurugram
May 2018

CERTIFICATE

This is to certify that project entitled **Virudh - a fake new detection solution** submitted by **Akarshan Gandotra** to **The NorthCap University, Gurugram, India** is record of bonafide Project work carried out by him under my supervision and guidance and is worthy of consideration for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering and Information Technology** of the Institute.

Date: Dr. Shilpa Mahajan

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. We are highly indebted to Dr. Shilpa Mahajan for her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would like to express my gratitude towards member of Faculty of University for their kind co-operation and encouragement which help me in completion of this project. We would like to express my special gratitude and thanks to industry persons for giving us such attention and time.

Akarshan Gandotra

ABSTACT

The underlying idea behind **Virudh/विरुध** is to help the people to allow the people to authenticate daily new they read and project them from believing any hoax or misinformation. Virudh is a comprehensive solution to classifies the fake news. Virudh provides APIs which is integrated with the Android Application and performs various checks to ensure authenticity of the news. The APIs can be called from multiple interfaces or can be integrated into other Applications. Using Industrial-Strength Natural Language Processing library, Spacy, micro web framework, Flask, Web Scraping techniques and Machine Learning, Virudh is able to distinguish the news.

Tests/checks associated with detecting fake news are automated and results against returned test. The tests are divided in Soft tests and Hard tests. Soft test modules consist of few elementary checks or tests. Hard tests are more vigorous test that help to identify the source of the news on the web. An Android application that helps user to interact with the Virudh's backend. The backend is hosted on Digital Ocean Droplets. The features or test results are then stored in the database which in near future will be used to apply Machine Learning to get attain accuracy while classification.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

We are living in the second decade of 21st century, and in this era the technology has enabled us to share an information instantly and to a great audience very conveniently. This lead to the issue of increase circulation of the Fake news.

Fake news is a type of yellow journalism or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcast news media or online social media[1]. It has the potential to influence the masses and change election results or spark a communal or cause some other event with severe consequences. The cases of fake news are increasing exponentially as shown in figure 1.

Fake news was not a regular term, but now it is now seen as one of the greatest threats to democracy.

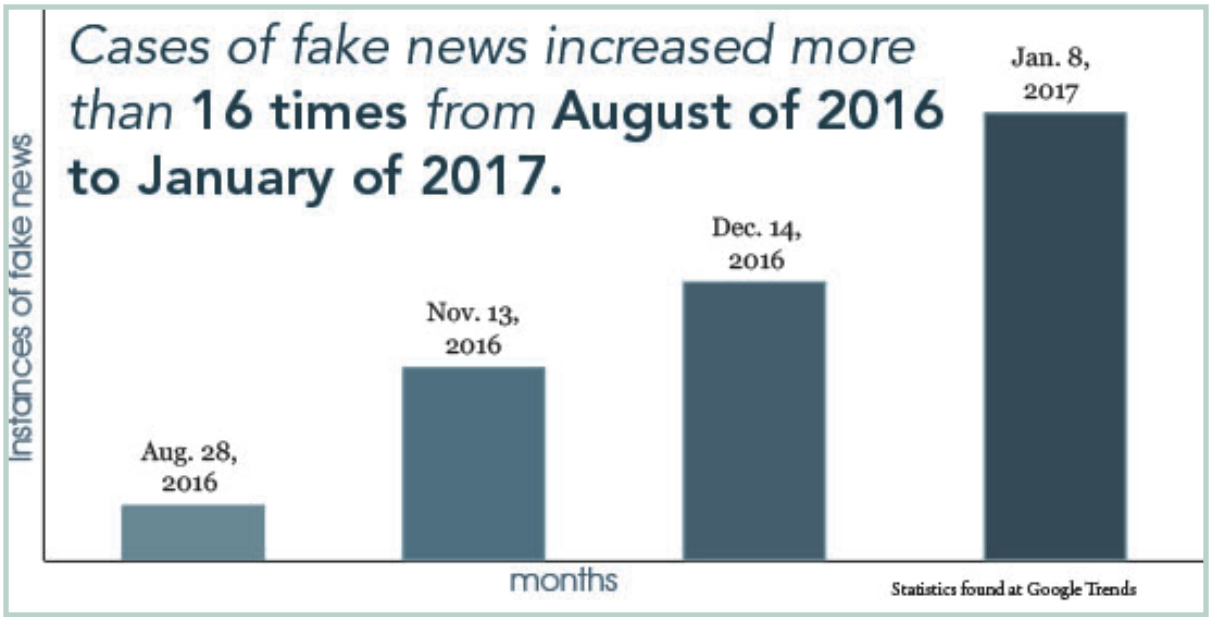


Fig 1.1: Bar-Graph depicting monthly growth of Fake News

Fake news is written and published with the intent to mislead in order to damage an agency, entity, or person, and/or gain financially or politically, often using sensationalist, dishonest, or outright fabricated headlines to increase readership, online sharing, and Internet click revenue.

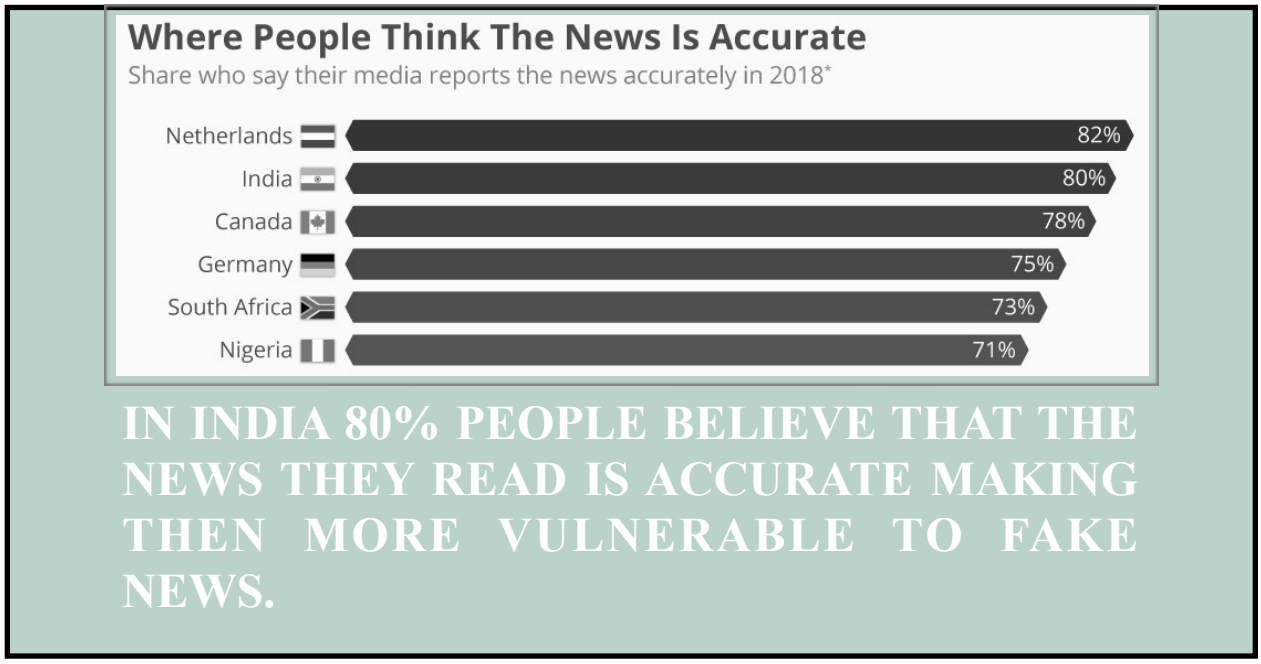


Fig 1.2: Bar chart depicting trust of people in news

The above image depicts a majority of Indians believe the news to be true. The main sources of the fake news these days are social media platforms like Facebook, Whatsapp and other sources. Any misleading news can easily distributed on social media and become viral.

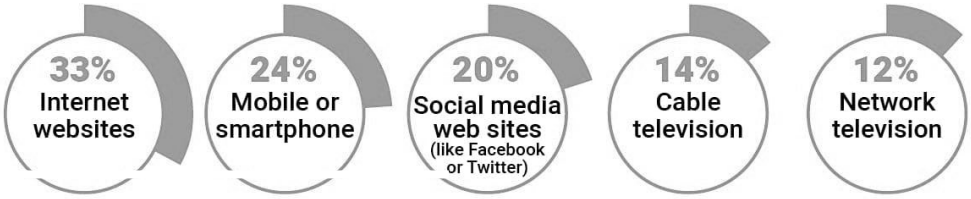


Fig 1.3: Sources of News

The underlying idea behind **Virudh/विरुध** is to help the people to allow the people to authenticate daily new they read and project them from believing any hoax or misinformation. Virudh is a comprehensive solution to classifies the fake news.

Virudh provides APIs which is integrated with the Android Application and performs various checks to ensure authenticity of the news. The APIs can be called from multiple interfaces or can be integrated into other Applications. Using **Industrial-Strength Natural Language Processing library, Spacy, micro web framework, Flask, Web Scraping techniques and Machine Learning,** Virudh is able to distinguish the news.

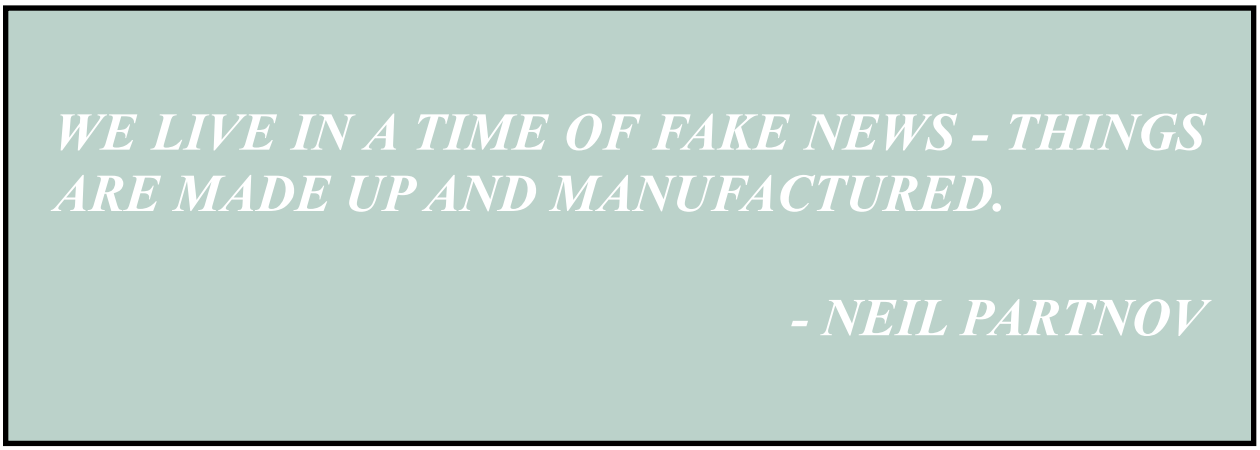


Fig 1.4: A quote on fake news by Neil Partnov

CHAPTER 2

THE APPROACH

Fig 2.1: An illustration of various checks used to determine the fake news

In order to understand the approach we need to discuss how we can spot a fake news. There are several ways to check whether a news is fake is not like checking it's source grammar, abnormalities, date, author, bias, searching for it on the web and asking experts. Some of the checks are picked from [2]. The following are the step help us to spot one.

Check the Source

The source of the news tells a lot about it's authenticity. A credible source ensures real news with accurate facts. If it's source is questionable or unknown, it is difficult to guarantee it's authenticity. And if the news originates from social media platforms puts a lot of question on it's originality and an investigation is a must. Also, checking author can also help to identify.

Grammar Check

Most of the fake news have grammatical errors and spelling mistakes. News having many of such errors or mistakes can be fake.

Check the Date

Reposting old news doesn't mean they are relevant to current events. Date can be a major parameter to determine the authenticity of the news. An irrelevant news from the past can still be damaging. Hence, date of the news also aid in the investigation of the fake news.

Search Engine Results

Simply by searching the news on web can tell a lot about it. If it is published by an authentic source, the results reflects it. In the later section we will discuss more about it. Also clicking on links or supporting sources mention in the news help to determine if information given actually supports the story.

Check the Biases

Fake news are often biased. In recent elections of the USA, the fake news circulated are often biased and this helped to manipulate the election results. Checking if the biased can help us to identify the authenticity of the news.

Past Experiences

Sometimes the news similar to the news we need to check has been circulated in the past later proved to be fake. Past experiences can make us alert and question the authenticity of the news.

Ask the Expert

Consulting the news with friends, experts and other sources can aid in spotting a fake news and also generate awareness [3].

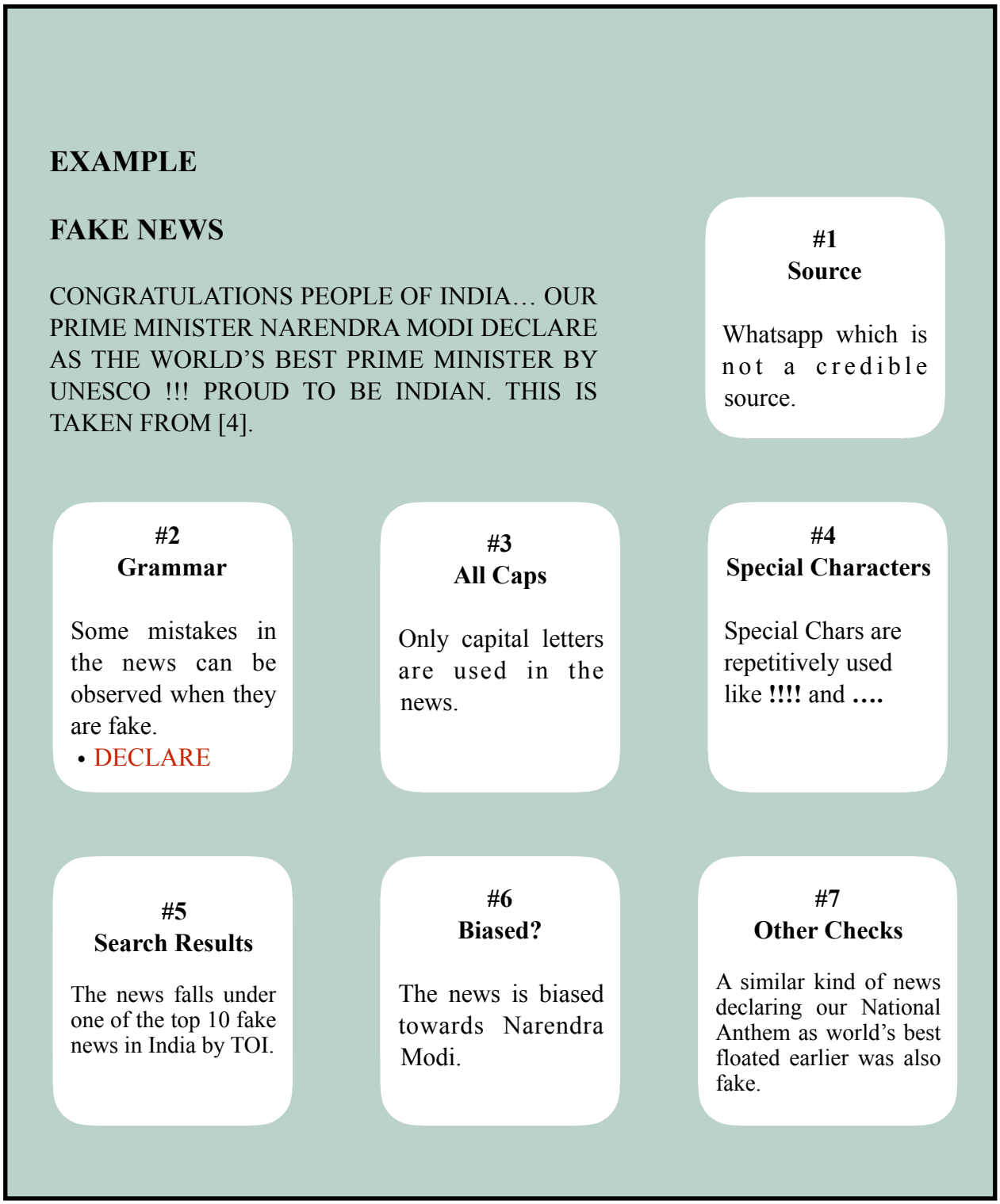


Fig 2.2: An Example of various checks

We can follow these tests/checks to check a news. Virudh automates these tests and checks and provide results against each check just with a single click. These tests/checks are implemented in backend of Virudh. So, let's discuss more about the approach in which Virudh works.

In Virudh we have categorize the tests into two separate categories that are **Soft tests and Hard tests.**

Soft Tests

These tests are some basic level tests that highlight various characteristics associated with the news. Soft tests combine the results from the following:

- Grammar Test
- Important Words Extraction
- Capital Letter Percentage
- Special Character Percentage
- Repetitive Characters

Hard Tests

These are some advance level tests that aim to search the source and other parameters associated. Hard tests involve web scraping and other robust techniques to extract sources, date and the match percentage. Future sections will cover how both these tests are integrated.

The results of these tests are then saved in a database. An analysis on results and user feedbacks are recorded along with the results. Later when we get enough data, machine learning algorithms can be used to give predictive results and even more accurate results. The following figure is a flow of the used approach.

Fig 2.3: The flow of the approach

CHAPTER 3

PROJECT IMPLEMENTATION

3.1 PROJECT INITIALIZATION

Before starting the project, the technologies, tools, techniques and skills that are required to develop Virudh are analyzed. Among various options available a detailed research was conducted in order to make a selection. So, we have decided to go with the following:

BACKEND

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.

MongoDB is a free and open-source cross-platform document-oriented database program. It is NO SQL database

spaCy

spaCy is an open-source software library for advanced **Natural Language Processing**. It is used for **tokenization** and **Part-of-speech tagging**.

LanguageTool to performs the Grammar Checks. It is written in JAVA.

BS4

Beautiful Soup 4 is a Python package for parsing HTML and XML documents. It is used for web scrapping.

Flask is a micro web framework written in Python and based on the **Werkzeug** toolkit and **Jinja2** template engine. It is BSD licensed.

The **IBM Watson Tone Analyze**r service uses linguistic analysis to detect emotional and language tones in written text.

Fig 3.1: List of technologies used in the backend

INTERFACE

Android is a mobile operating system developed by Google. An Android App provides an interface to interact with backend.

*** APIs can be integrated on a number of platforms and interfaces providing access to a larger audience.**

Fig 3.2: Technologies used in interfacing

HOSTING

DigitalOcean calls its cloud servers **Droplets,** each Droplet you create is a new server for the use. Droplets are a scalable compute platform with add-on storage, security, and monitoring capabilities to easily run production applications.

Fig 3.3: Technologies used for hosting

After all the tools, techniques and skills are decided, the project was divided into modules and each module was developed independently. The modules are as follow

#1
SOFT TESTS MODULE
Module dealing with development of Soft Tests.

#2
HARD TESTS MODULE
Module dealing with development of Hard Tests.

#3
API MODULE
Module dealing with the development of APIs in Flask Frame work.

#4
ANROID MODULE
Module dealing with development of Android Application.

Fig 3.4: The modules of the project

Each module mentioned above was developed using an Iterative Waterfall model. A modular system is always convenient to develop.

3.2 PROJECT ARCHITECTURE

An architecture for the project is designed to have a clearer picture of interaction between different modules. Virudh's architecture is as follow:

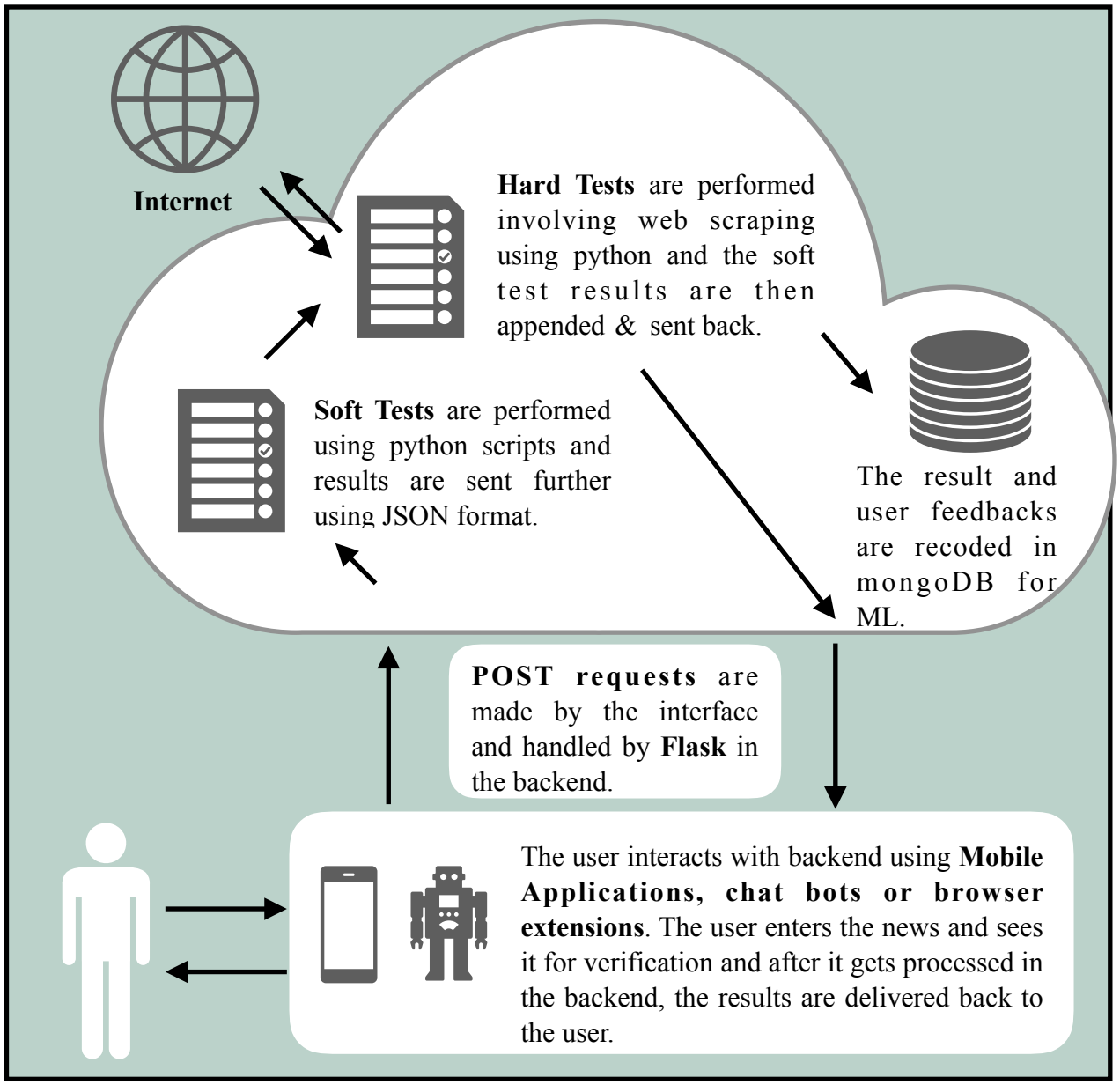


Fig 3.6: The Project Architecture

The user initially enters the news in the Android Application and sends to the server on a click of the button. A post request sending user input in JSON format is made (or an API is called). Once the server receives the data and methods related to tests or checks are called and results a re accumulated in JSON format and sent back to the from where the requests are made. The results are parsed and shown to the user. This is how Virudh works.

Let's discuss how the modules were developed.

3.3 SOFT TEST MODULE

Soft test modules consist of few elementary checks or tests. The scripts performing soft tests is written in python, The following shows the function called during soft tests associated with each news.

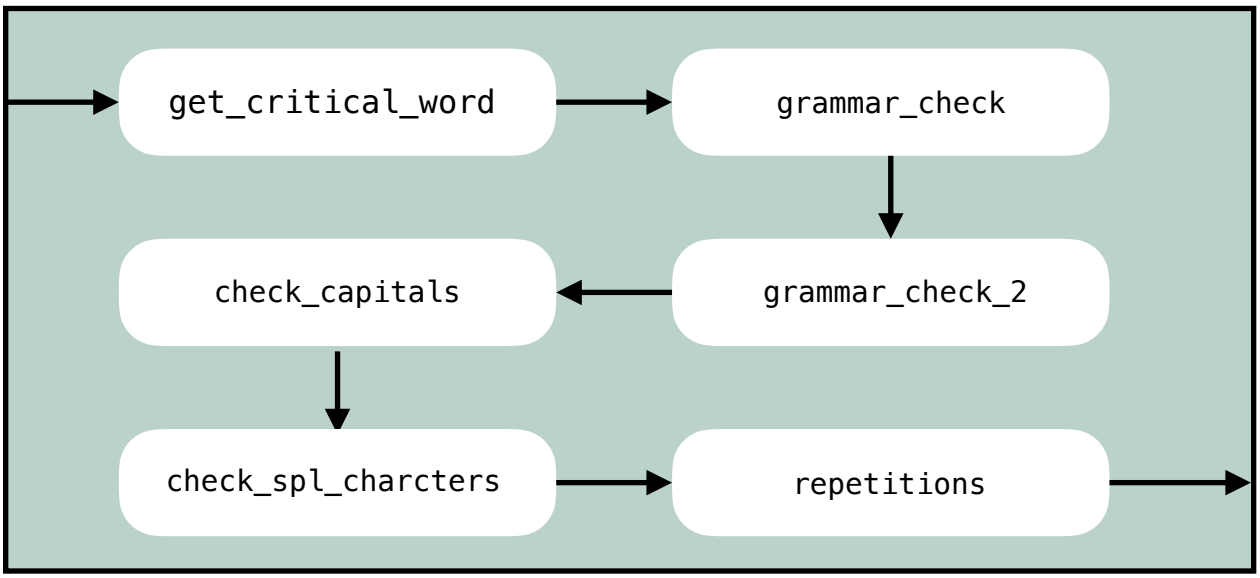


Fig 3.7: Functions of the Soft Tests Module

Every function is given an input and the output OR results are consolidated in the form of JSON. There are two functions for the checking the grammar. Both use different tools for the same. The thought behind it is that different tools have different rules of grammar, if one tool fails to notice an error, it can be picked up by another. Let's discuss and analyze each function one by one.

3.3.1 Critical Words Extraction

Spacy is an open-source software library for advanced **Natural Language Processing**. It is used for **tokenization** and **Part-of-speech tagging**.

Spacy is mainly used in this function. The news in String format is passed as the parameter. The space tokenizes the news and then **deep learning** algorithms perform **Parts of Speech Tagging**. After space loads en model, the tokenized words are iterated and **proper nouns** are extracted and appended in a list. The list is then sent back to the main.

The critical words have it's own importance. It tells the user about the context of the news or what the news is talking about. The maintained database of fake news also store critical words against each and every news it's store. Later when classification of the news is done, we can consolidate the critical words of all the stated news and find the most common word occurring in the fake news.

A notification can be later sent making people aware of news containing those words. In a dataset from **Kraggle**, we have found that during 2017 America's election, J Donald Trump and Hilary Clinton were most commonly occurring words or nouns.

Hence, critical words can tell a lot about the news and track what are the most fake news is all about and get_critical_word function helps to get them.

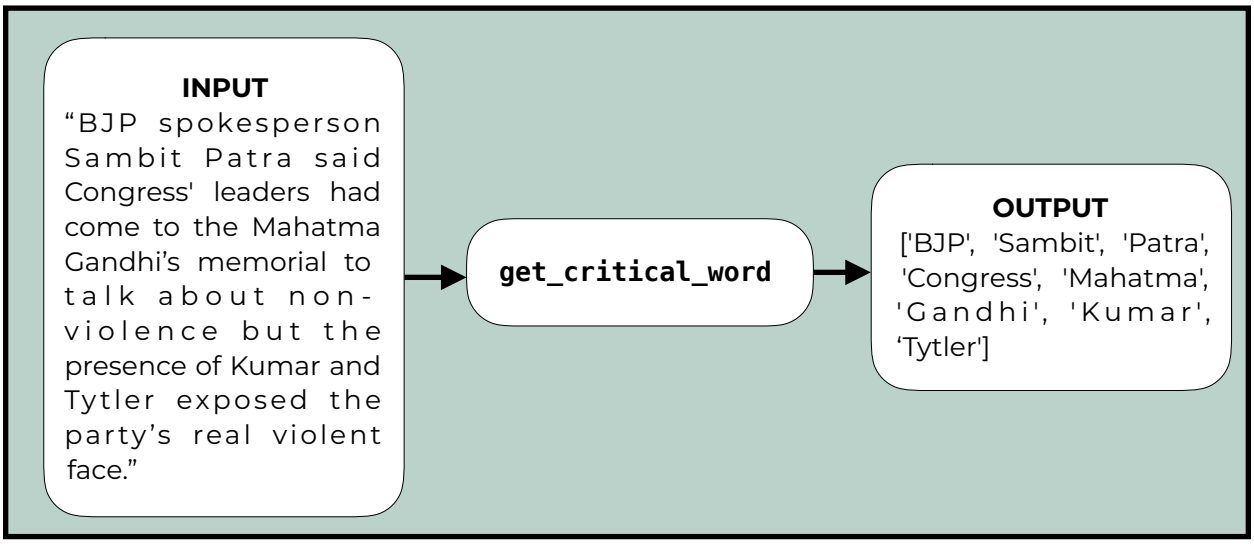


Fig 3.8: get_critical_word method

3.3.2 Grammar or Language check

A news from a genuine source or a known media house is always free of grammatical errors and spelling mistakes as the news is checked a number of times by highly professional and experienced people before getting published. A fake news distributor is generally more prone to make grammatical errors and spelling mistakes than a reputed media house.

This stage has two functions, **grammar_check** and **grammar_check_2.** For checking grammar, language spelling mistakes we need a **Grammarly** like tool. Unfortunately no library in python supports it. In JAVA, **LanguageTool** is available for similar kind of functionality. So, we developed a **python wrapper** for the same. For this we needed to setup JDK and JRE. A python wrapper is a binding to a JAVA library so that developers can use JAVA library using python code.

The other function **grammar_check_2,** on the other hand calls languagetool.org's check api for the same. A GET request is made (https://languagetool.org/api/v2/check?text=enterthetext&language=en-GB&enabledOnly=false) and response is generated by the server. The response returns incorrect results for the same. The results are in JSON. The JSON is the parsed and the results are extracted. The results are mapped and returned to the main.

The no. of grammar mistakes and spelling mistakes can be a powerful feature to classify fake news.

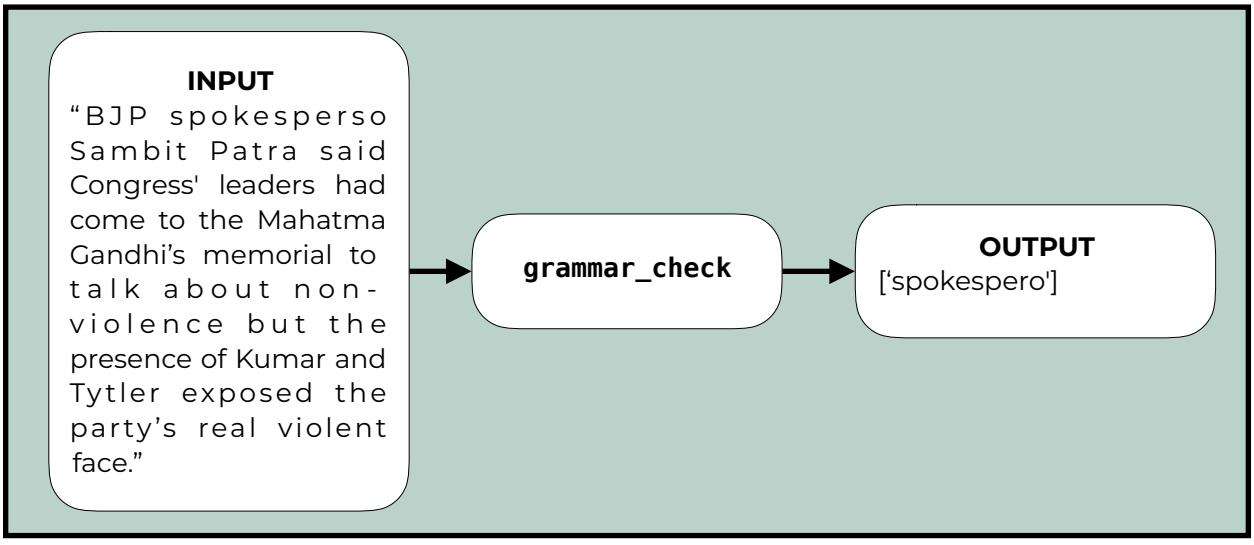


Fig 3.9: grammar_check method

3.3.3 Capital Letter check

Capital letters can be considered a third form of emphasis, among Italics and Bold text. They are used to denote a louder, almost shouting pronunciation. For example,
THIS TEXT HAS MORE IMPACT THAN this text.

Clearly, all caps part of the sentence is more impactful than the small characters and the fake news try to take advantage of this. Most fake news use only capital letters to give a more emphasis to the news or the message they are sending. The fake news often carry high percentage of all capital letters while a genuine news use them wherever required. This can turn to be a useful to detect the fake news.

A simple algorithm in python using **Regex** is written to detect the number of capital letter present in the news and gives it's percentage by dividing by total number characters multiplied by 100. It is one of the simplest to find but is very important feature to tell authenticity of the news. This

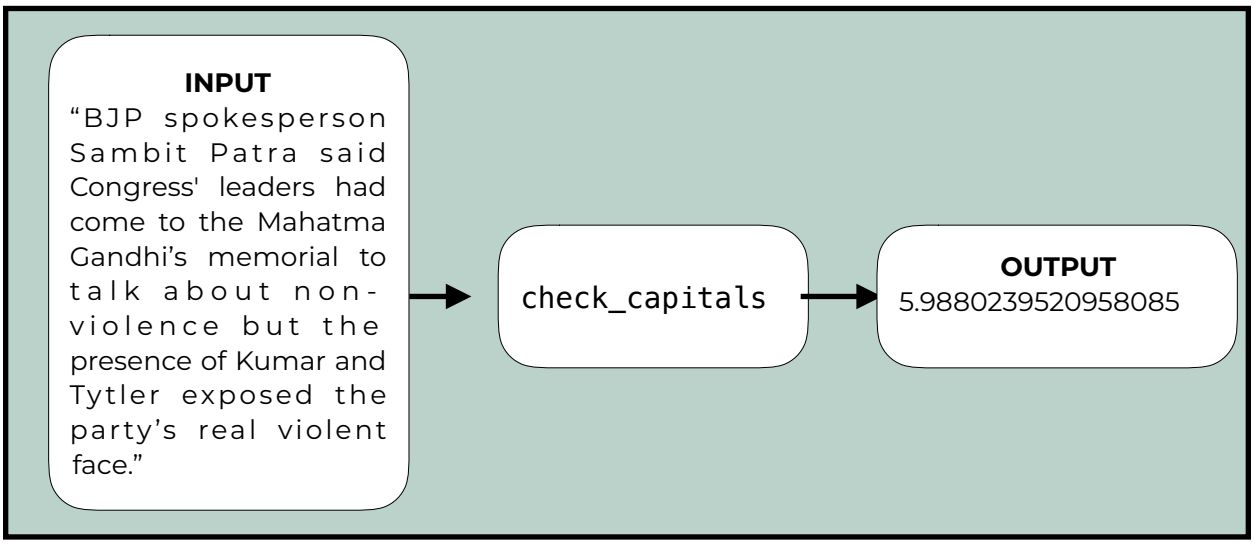


Fig 3.10: captial_check method

percentage then sent back to main and consolidated in JSON.

Along with Grammar and other checks, this feature can prove to be a important decider to check whether the news is fake or not.

3.3.4 Special Character check

A fake news uses a number of special characters without any significance. As fake news is mostly written by unprofessional or not so educated people, they end up using a lot of special characters. The reason is similar to the use of capital letters, to create emphasis or impact.

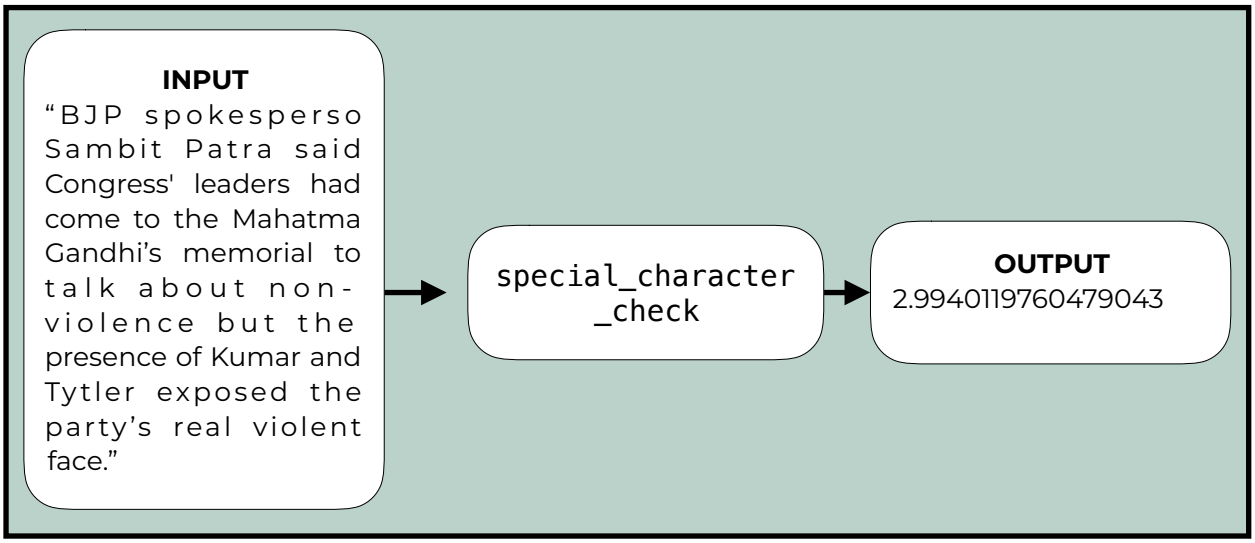


Fig 3.11 special_character_check method

The method associated with this test is similar to that of capital check. We used a simple algorithm in python using **Regex** is written to detect the number of special characters present in the news and gives it’s percentage by dividing by total number characters multiplied by 100. It is one of the simplest to find but is very important feature to tell authenticity of the news. This percentage then sent back to main and consolidated in JSON.

Similar to capital letter check, this check is also used can prove as a important feature to distinguish the fake news.

3.3.5 Repetition check

This is an extension of special_character_check method. Most of the fake news don’t only use special characters but they tend to have them in repetitions. For example the use of **!!!!, …., $$$$** is common.

So, we have created a method that can figure such findings. A simple **Regex** helped to get subsequent repetitions of any character, if this is greater than 2 then it is counted.

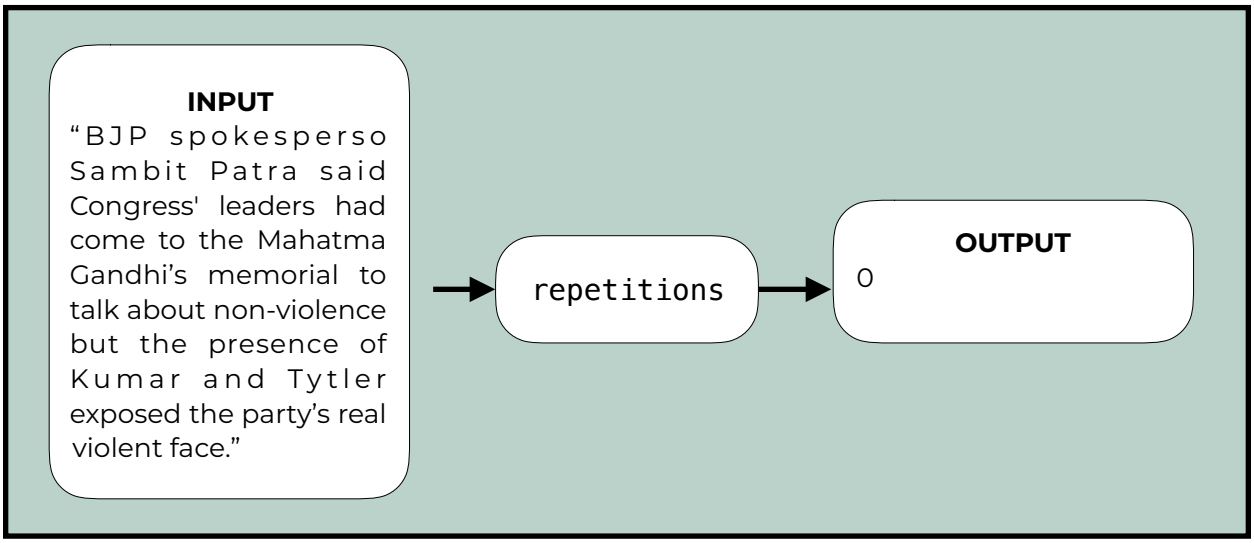


Fig 3.12 repetitions method

3.3.5 Soft Tests Result Consolidation

In computing, JavaScript Object Notation or **JSON** is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types.
When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects. This way we can work with the data as JavaScript objects, with no complicated parsing and translations.

Once all checks or tests are done, the results are consolidate in **JSON** format and will be returned as follow:

```
{
    "critical_words": [
        "Congress",
        "Rahul",
        "Gandhi",
        "sunday",
        "China",
        "India"
    ],
    "incorrect_text_1": [],
```

```
"capital_percentage": 2.564102564102564,
"repetitive_characters": 0,
"special_percentage": 1.0256410256410255 }
```

This can enable us to share results easily with the applications or used by developers for API integration purposes.

3.4 HARD TEST MODULE

Hard tests are more vigorous test that help to identify the source of the news on the web. Another test it does is sentiment analysis. These test results contain more information and tells more if the news is fake or not. The following shows the function called during hard tests.

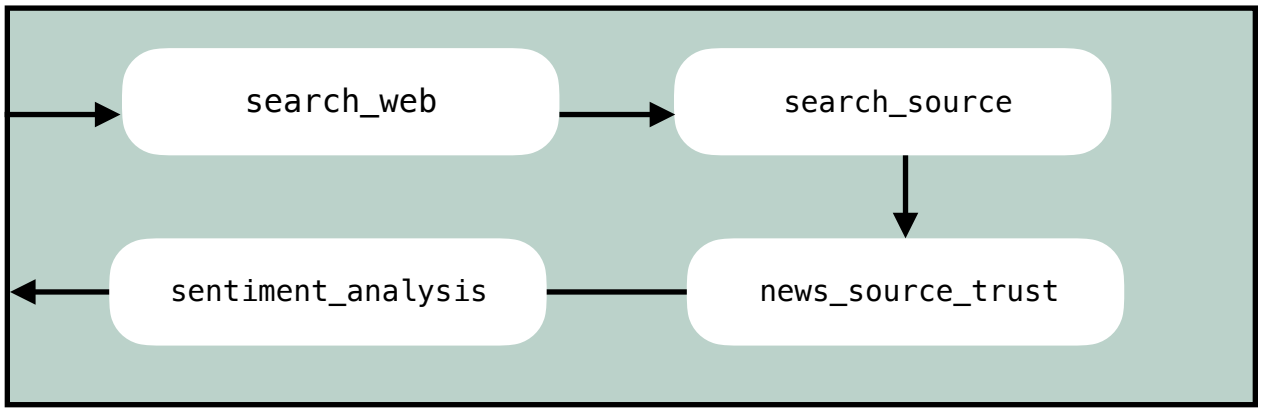


Fig 3.13: Functions of the Hard Tests Module

Every function is given an input and the output OR results are consolidated in the form of JSON. The search_web and search_source are methods to search the news and scan the source. While sentiment_analysis tells about the tone of the news. Let's discuss and analyze each function one by one.

3.4.1 Search the Source

Today most of the news companies have their presence over the internet. As the smartphone users increased most of them today use internet as their news feed and the users of the newspaper are declined sharply, it is depicted in figure 12. The news is on the internet as soon as it breaks. News sites share them on their websites, social media platforms and other places. This is taken as an advantage .
A genuine new will be published by a number of media sources while fake news only spread on not so popular websites, Facebook or Whatsapp or any other platform.

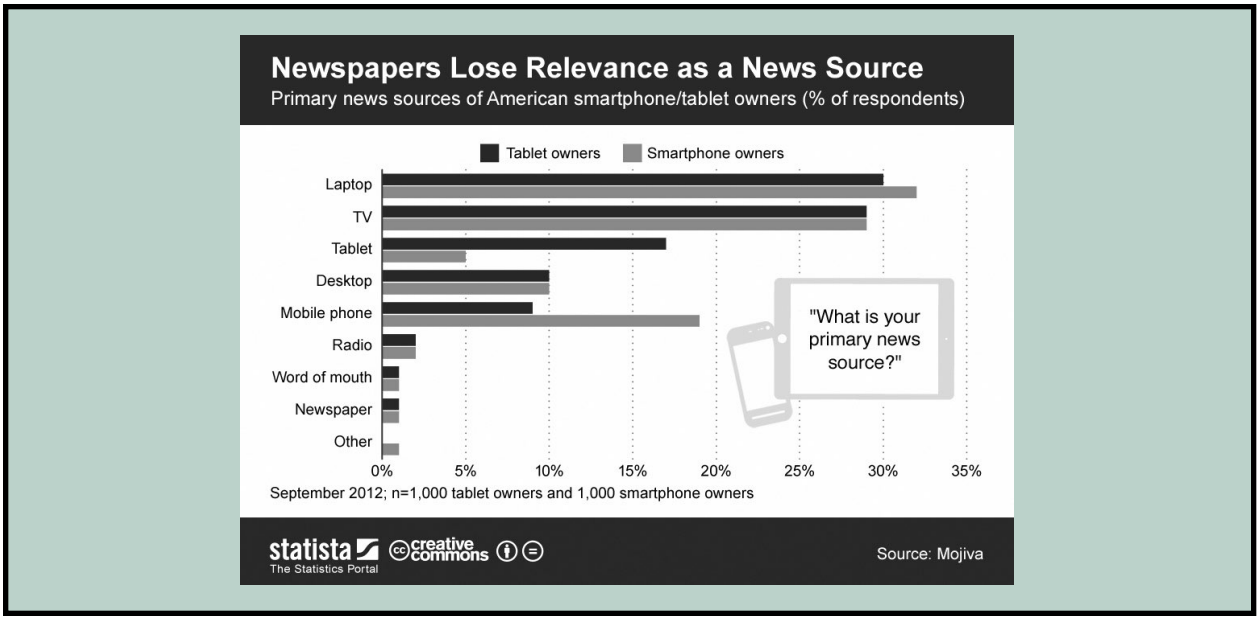


Fig 3.14: primary sources of news

In this method **web scrapping** is done on the search results. Initially is search query is sent to Google. The search query is initially **unicode encoded** as Google accepts unicode string as search query. For example

- **top 10 traveling places in Europe** gets converted into **top+10+traveling+places +in+Europe**
- **(Hello) ! &** gets converted into **%28Hello%29+%21+%26**

This is done by replacing the character of string with their equivalent unicodes.

Once a request is made we get a response. The response is in HTML. Now the html is parsed using **Beautiful Soup** library in python. BS4 is a Python package for parsing HTML and XML documents. After the parsing is done we extract the "a tags". The a tags contains the link to the of best matches or results of the search query which in this case in news. Once the links are extracted, they can't be used further. Some links are associated with AMP while others with Google cache pages and can't be used further. For this we cleaned the links by splitting that parts in the link. The links are consolidated in a list and sent for further process.

This is how links are scrapped using BS4.

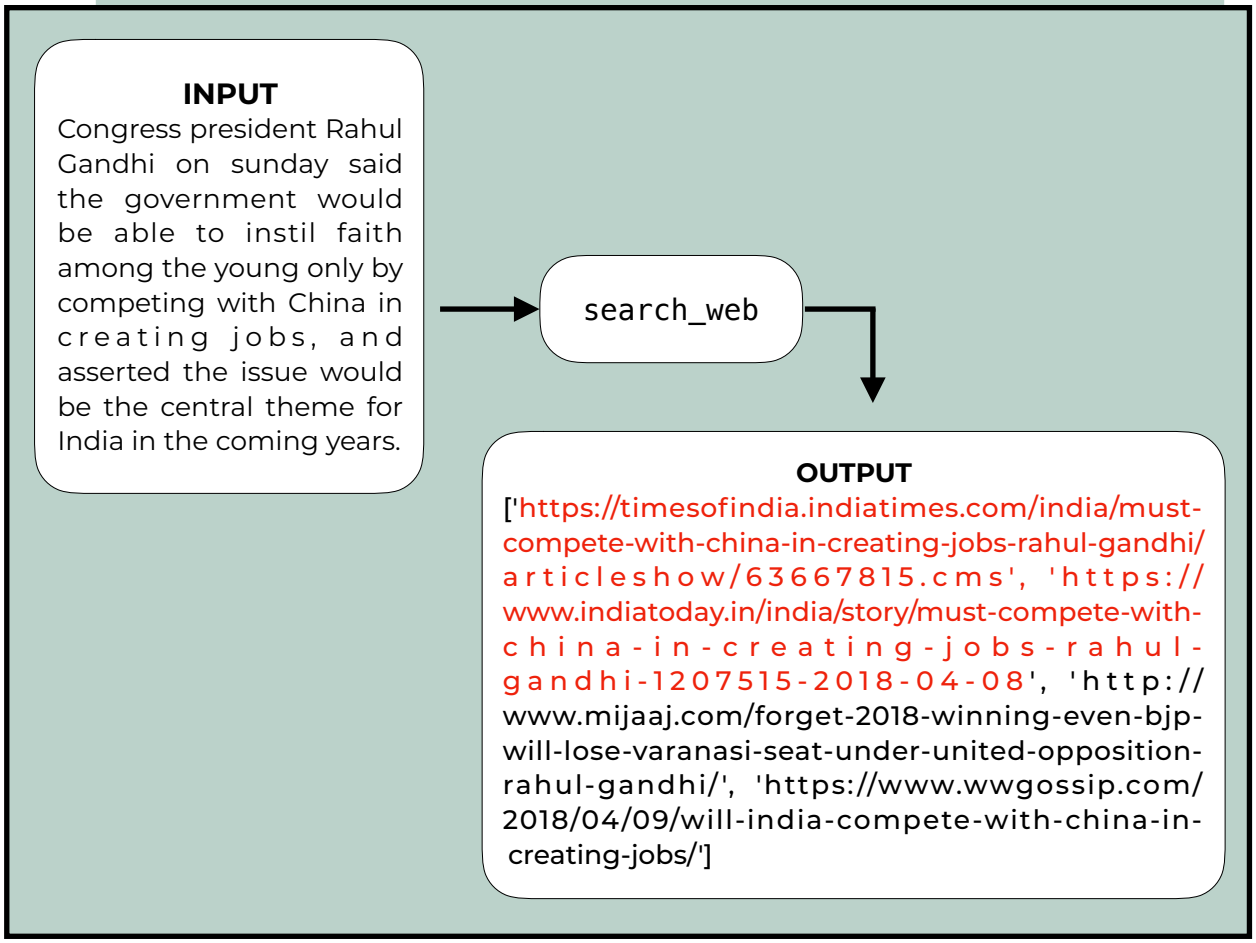


Fig 3.15 search_web function

3.4.2 Scanning the Source

Once we get relevant links of the sources we then call each link and see the percentage match of the the news with the source. The is to check the each how much the source is relevant and by what percent it's similar to the news the user wants to check. From the list each link is passed through search_source method and the results are embedded with link for later purpose.

The search_source takes the news text and the link as parameters. Using **Beautiful Soup** we scrape the website. A request is made on the source link and a response is received. This response is HTML and parsed using BS4. Before going further the soup (object of BS4) is cleaned and the the news data is searched. The match is recorded and converted into percentage. Once we get the match percentage, it is stored with the link. The same process is repeated with all the links so in the end we have a list of tuples of links with the match percentage.

This list is sorted in descending order and only top 3 match hits are kept in the list while the rest are removed.

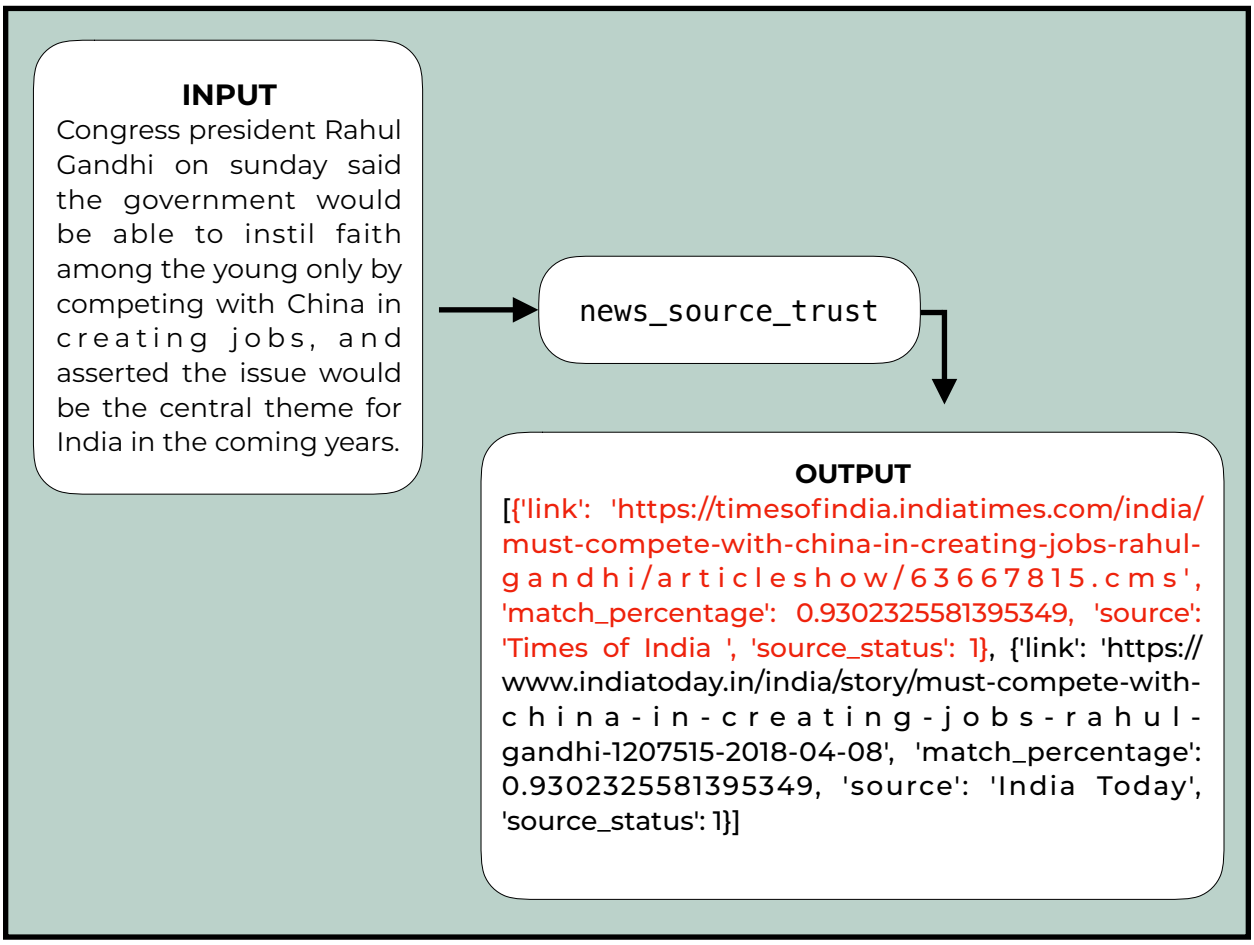
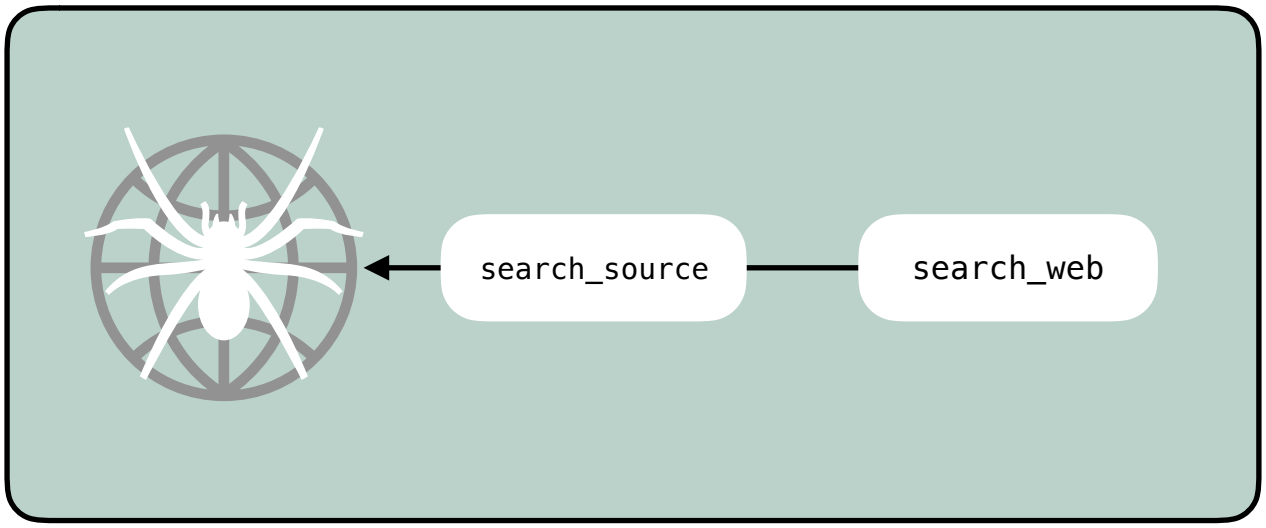


Fig 3.16: search_web + search_source + news_source_trust function

3.4.4 Web Spiders

Along with the methods **search_web** and **search_source** we made a web spider. A **Web crawler**, sometimes called a **spider**, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing (*web spidering*).

Web search engines and some other sites use Web crawling or spidering software to update their web content or indices of others sites' web content. Web crawlers copy pages for processing by a search engine which indexes the downloaded pages so users can search more efficiently. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses.



3.4.5 Source Trust Score

Another script is written python which is mainly responsible for scanning the most credible news sources over the internet. Few blogs update them weekly so the script runs every week ensuring we get a list of latest most credible source over the web.

The script again uses BS4 to scrape these sites and store results in JSON file along with their links.

After we get the top 3 links along with match percentage, **news_source_trust** method is called the presence of the links is searched through the maintained JSON file. If the link is present 1 is returned stating link as a credible source else 0 is returned stating link or source not credible. As we are able to detect fake news through Machine Learning we can also find those sources which are spreading fake news and alert the people through notifications.

3.4.6 Sentiment or Tone Analysis

Fig 3.17: IBM Watson Tone Analyze Logo

The **IBM Watson Tone Analyzer** service uses linguistic analysis to detect emotional and language tones in written text. The service can analyze tone at both the document and sentence levels. A tone can be anger, fear, joy, sadness, and disgust. News Tones are very useful feature to asses the sentiment of the news and later for fake news classification. Fake news generally carry negative tone as shown in a result in figure 24.

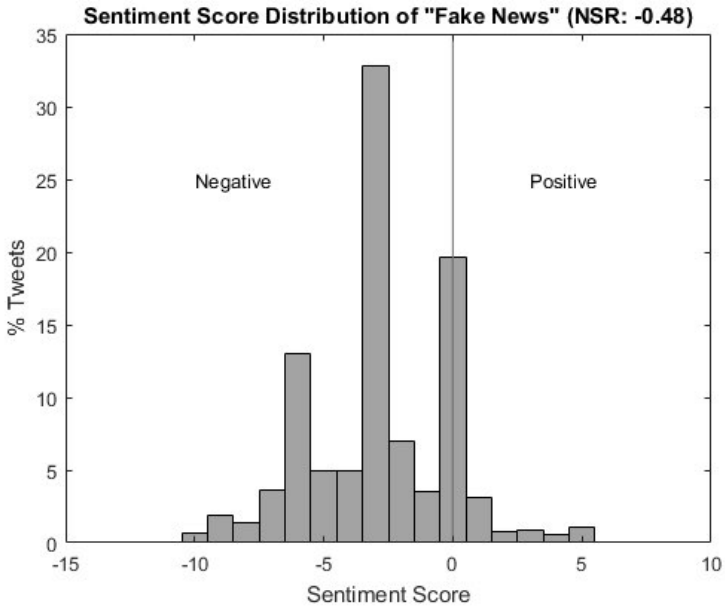


Fig 3.18: Results depicting fake new have negative sentiments

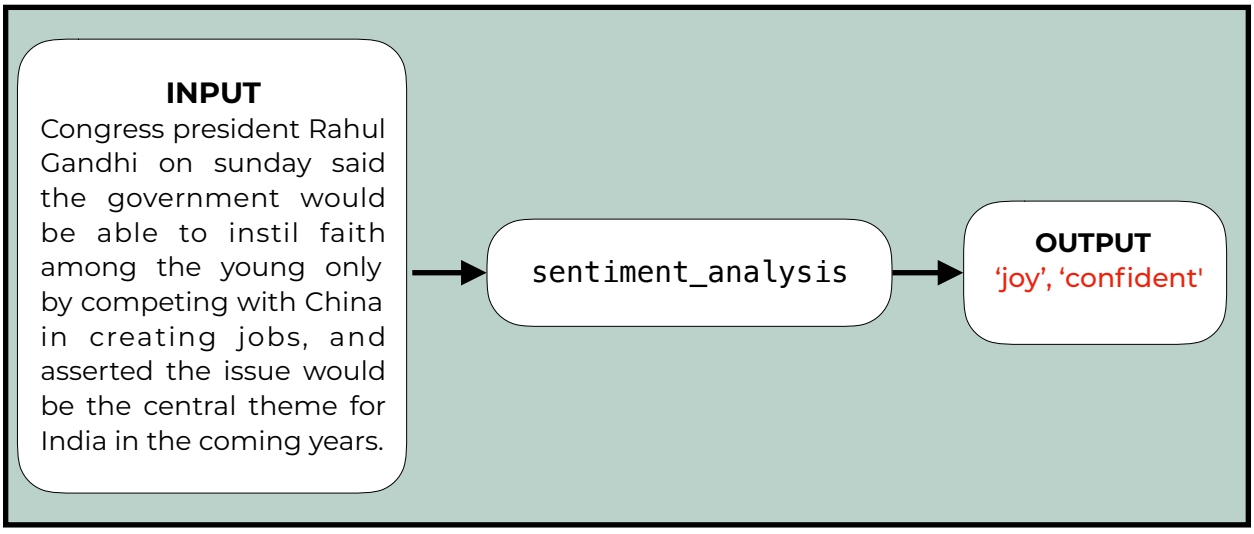


Fig 3.19: sentiment_analysis function

After other hard tests are performed Tone Analyzer api is called to get the tone of the news. For this a **bluemix** account was made and credential are also sent along with the api.

3.5 API MODULE

Once the functions are written and Soft and Hard Tests Modules are tested we required to develop APIs so that the backend can be utilized via interfaces like applications and bots. The API's were created using Flask. Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine.

The API's are designed specifically keeping in mind the integration with the application, other interfaces and 3rd party developers. The modularity of the system is also reflected into APIs. APIs can be also used to perform specific pool of tests individually.

There are 5 APIs. They are as follows:

3.5.1 /soft_tests

Methods: POST, GET
Parameters: news: <news user wants analyze>
Response:


```
{
   "critical_words": [
      "Congress",
      "Rahul",
      "Gandhi",
      "sunday",
      "China",
      "India"
   ],
   "incorrect_text_1": [],
   "capital_percentage": 2.564102564102564,
   "repetitive_characters": 0,
   "special_percentage": 1.0256410256410255
}
```


Description: Returns the result from Soft Tests.

3.5.2 /hard_tests

Methods: POST, GET
Parameters: news: <news user wants analyze>

Response:


```
[
    {
         "link": "https://timesofindia.indiatimes.com/india/must-compete-with-
china-in-creating-jobs-rahul-gandhi/articleshow/63667815.cms",
        "match_percentage": 0.9302325581395349,
        "source": "Times of India ",
        "source_status": 1
    },
    {
            "link": "https://www.indiatoday.in/india/story/must-compete-with-
china-in-creating-jobs-rahul-gandhi-1207515-2018-04-08",
        "match_percentage": 0.9302325581395349,
        "source": "India Today",
        "source_status": 1
    },
    {
        "link": "http://www.mijaaj.com/forget-2018-winning-even-bjp-will-lose-
varanasi-seat-under-united-opposition-rahul-gandhi/",
        "match_percentage": 0.9302325581395349,
          "source": "http://www.mijaaj.com/forget-2018-winning-even-bjp-will-
lose-varanasi-seat-under-united-opposition-rahul-gandhi/",
        "source_status": 0
    },
 ]
```


Description: Returns the result from Hard Tests.

3.5.3 /both_tests

Methods: POST, GET

Parameters: news: <news user wants analyze>

Response:


```
{
    "critical_words": [
        "Congress",
        "Rahul",
        "Gandhi",
        "sunday",
        "China",
        "India"
    ],
    "incorrect_text_1": [],
    "capital_percentage": 2.564102564102564,
    "repetitive_characters": 0,
    "special_percentage": 1.0256410256410255,
    "links": [
        {
```

```
              "link": "https://timesofindia.indiatimes.com/india/must-compete-
with-china-in-creating-jobs-rahul-gandhi/articleshow/63667815.cms",
            "match_percentage": 0.9302325581395349,
            "source": "Times of India ",
            "source_status": 1
        },
        {
             "link": "https://www.indiatoday.in/india/story/must-compete-with-
china-in-creating-jobs-rahul-gandhi-1207515-2018-04-08",
            "match_percentage": 0.9302325581395349,
            "source": "India Today",
            "source_status": 1
        },
        {
             "link": "http://www.mijaaj.com/forget-2018-winning-even-bjp-will-
lose-varanasi-seat-under-united-opposition-rahul-gandhi/",
            "match_percentage": 0.9302325581395349,
               "source": "http://www.mijaaj.com/forget-2018-winning-even-bjp-
will-lose-varanasi-seat-under-united-opposition-rahul-gandhi/",
            "source_status": 0
        },
        {
              "link": "https://www.wwgossip.com/2018/04/09/will-india-compete-
with-china-in-creating-jobs/",
            "match_percentage": 0.9302325581395349,
            "source": "https://www.wwgossip.com/2018/04/09/will-india-compete-
with-china-in-creating-jobs/",
            "source_status": 0
        }
    ],
    "sentiments": [
        {
            "score": 0.671911,
            "tone_id": "joy",
            "tone_name": "Joy"
        },
        {
            "score": 0.608441,
            "tone_id": "confident",
            "tone_name": "Confident"
        }
    ]
}
```


Description: Returns the result from Soft Tests and Hard Tests.

3.5.4 /store_results

Method: POST

Parameters: results: <results of the analysis in son format>
feedback: User feedback

news: <news on analysis is done>

Response:

```
{
      “Status” : “Success”
}
```

Description: Records user response and send and store it along with the other results.

3.6 ANDROID MODULE

Fig 4.1: A shot of Vidhur running on Android

Once the backend was done, the Android Module we started developing the Android Module. An Android application that helps user to interact with the Virudh’s backend.

We started with the logo designing. The logo was designed using **GIMP.** To give a nice UI and UX, a research of colors combinations was done and dark gray and sea green color were selected as the primary colors. Color scheme of the Application is #48D1CC, #3D4C5F and #40E0D0. A

mockup of each screen is initially designed and after some improvisations the UI was faired in the Virudh App and made using XML.

There are mainly 3 screens: Verify, Trending and Result screen.

3.6.1 Verify Screen

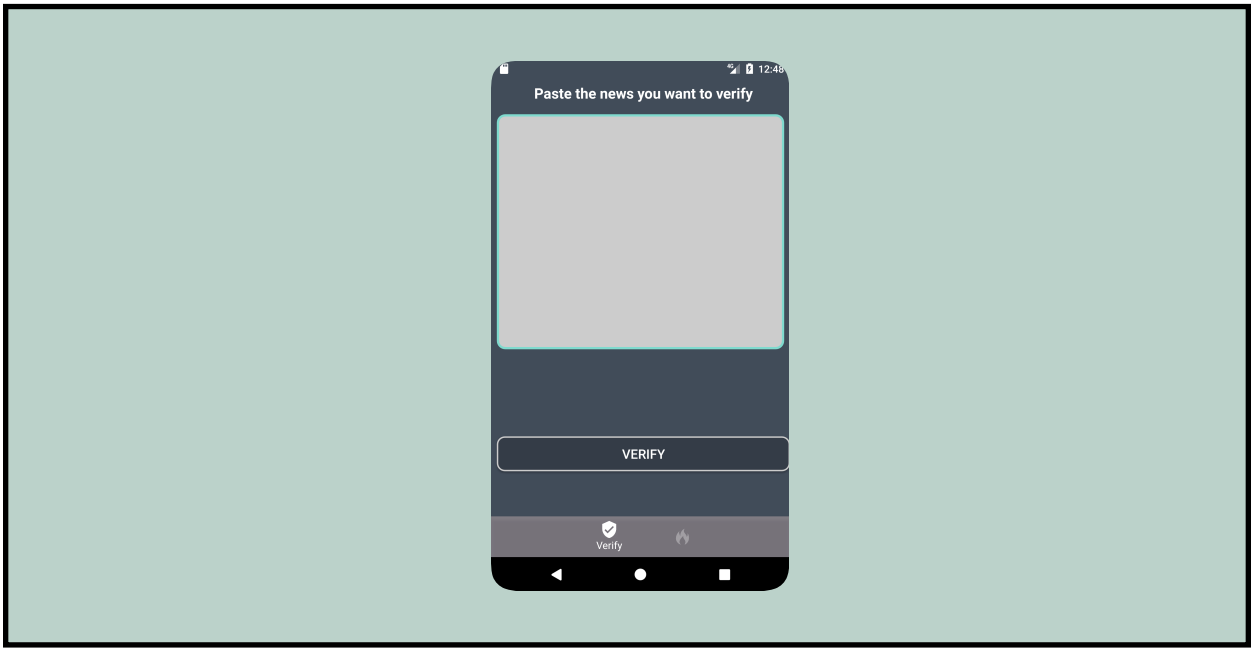


Fig 4.2: Verify Screen

In this screen the user can paste the news he or she wants to verify. There is a edit-box in the screen where user can paste the news he/she wants to verify. The screen also comprises of a button and a bar that helps the user to switch in between verify and trending tabs.
Fragments are also used while building the application. A **fragment** is usually used as part of an activity's user interface and contributes its own layout to the activity. To provide a layout for a **fragment**, you must implement the onCreateView() callback method, which the **Android** system calls when it's time for the **fragment** to draw its layout.

3.6.2 Result Screen

The screen is responsible to send a GET request of **/both_tests** api. This api is called in a Thread. In the Result Screen Activity, communication with the server goes in parallel which result in inoperable UI if both these functions are performed in the main thread. This would severely degrade the user experience. In order to avoid this, we simply used **AsyncTask**. AsyncTasks are used to perform communications with the device in the background thread and update UI when

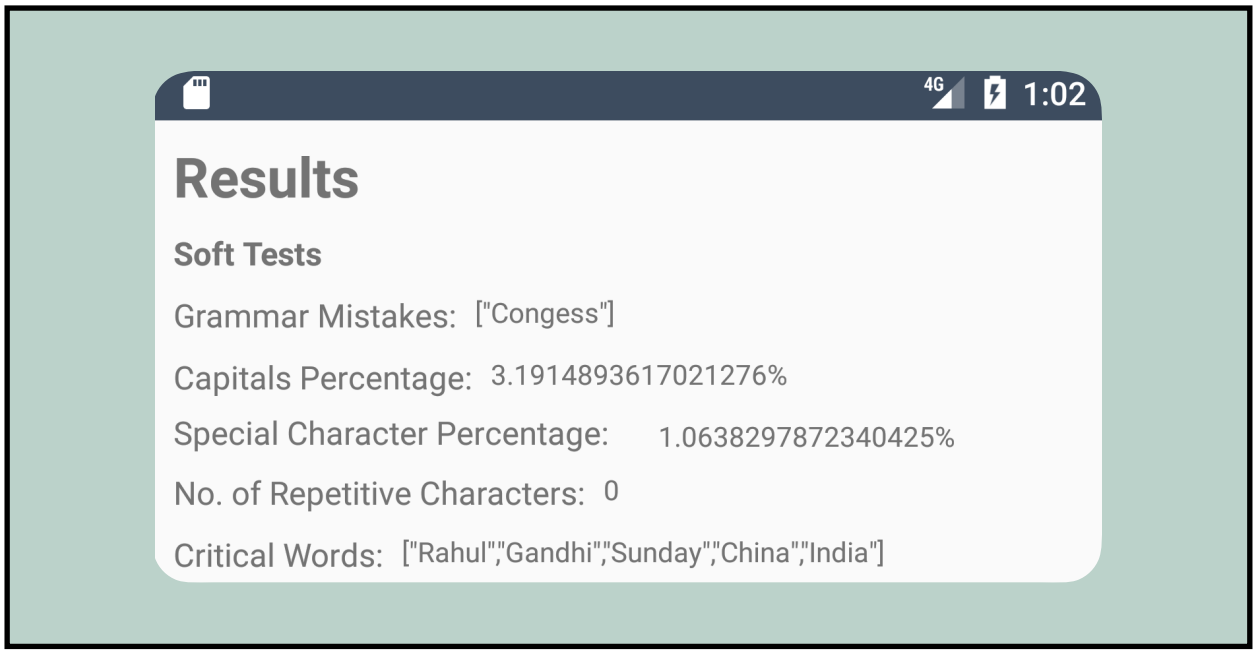


Fig 4.3: Result Screen

the task in background thread completes. AsyncTask thus solves the problem of making UI super laggy while performing certain time-consuming functions. The UI remains responsive throughout. AsyncTask is an abstract Android class which helps the Android applications to handle the Main UI thread in a more efficient way. AsyncTask class allows to perform long lasting background operations and update the results in UI thread without affecting the main thread.

Fig 4.4: User feedback dialog box

After the GET request is made a JSON response is obtained which is parse and displayed to the user on the result screen. After the results when user clicks on proceed, a dialog box appears asking the user for his or her feedback. The answers is sent with result to the server using / store_results api where it is stored in mongo db.

CHAPTER 4

TESTING AND HOSTING

4.1 TESTING

We have followed **unit testing** approach initially where each module and the software components of each module were tested individually. With the development and integration of the different modules we performed **integration testing**. Finally, as we reach to the final phase of the Development, **Application and the Backed** was tested as a complete system. The testing was done keeping in mind all possible scenarios and cases. Errors are also handled carefully taking in account all possible errors that can take place while execution.

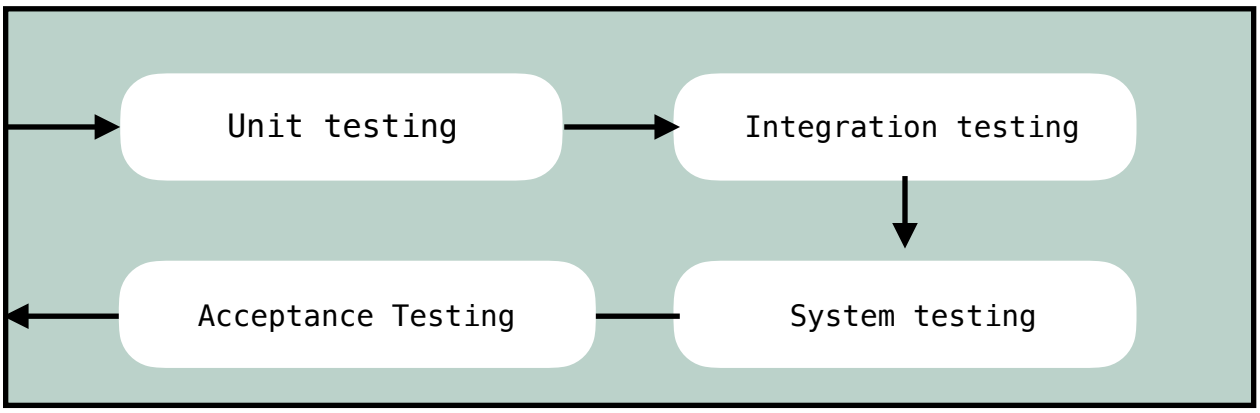


Fig 4.1: Flow of testing

4.2 HOSTING

The backend is hosted on Digital Ocean Droplets. DigitalOcean calls its cloud servers Droplets, each Droplet you create is a new server for the use. Droplets are a scalable compute platform with add-on storage, security, and monitoring capabilities to easily run production applications.

A virtual environment is created on the server in which all the add on libraries and modules are added from requirements.txt (a file listing all the project requirements). All the scripts then copied to the server and we ensured that they were run perfectly. Beside setting up python, JDK was also installed for executing **LanguageTool**.

Flask is a development server, that can’t process requests synchronously. To make the Flask a development server, **Gunicorn** is used. Gunicorn 'Green Unicorn' is a Python WSGI HTTP Server for UNIX. It's a pre-fork worker model. The **Gunicorn** server is broadly compatible with various web frameworks, simply implemented, light on server resources, and fairly speedy.

Fig 4.2: Gunicorn logo

CHAPTER 5

FUTURE WORKS

5.1 MACHINE LEARNING

Machine learning is the science of getting computers to act without being explicitly programmed. Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed. Machine Learning is coming into its own, with a growing recognition that ML can play a key role in a wide range of critical applications, such as data mining, natural language processing, image recognition, and expert systems. This is also followed in [5]

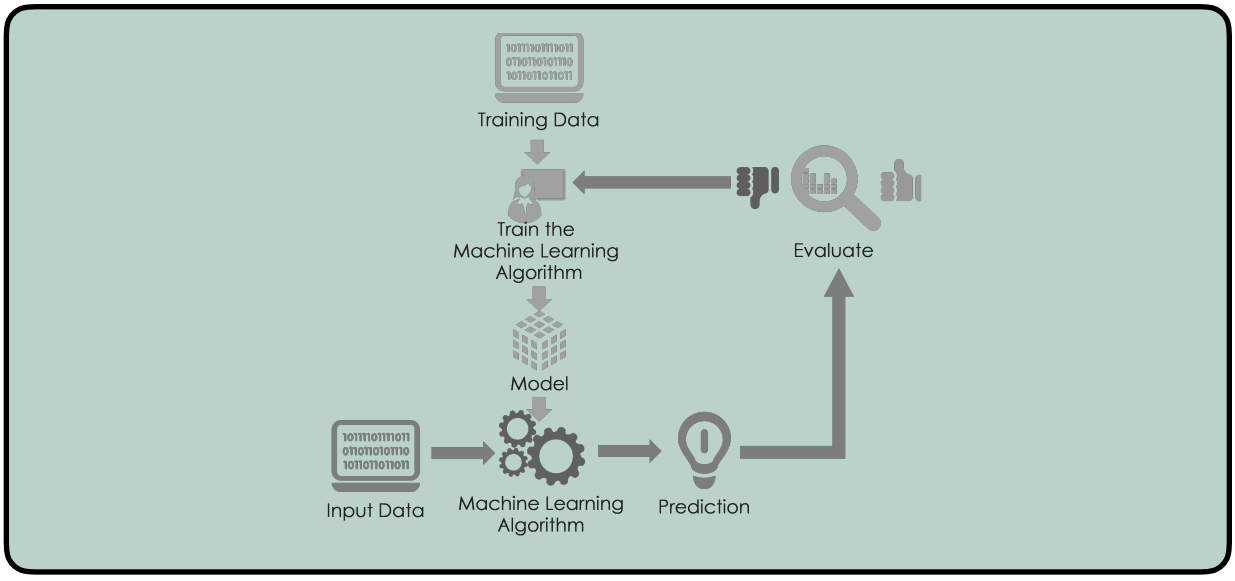


Fig 5.3: Flow of machine learning

The main struggle regarding fake news classification is that the absence of dataset of fake news in India. Without dataset Machine Learning predictions can't be made as we can't train the model.

Virudh attempts to start from very starting and help to build that dataset so that in near future it can use it to make even mow accurate predictions and get better results.
This is how the data is stored in MongoDB on the server.

```
{
    "critical_words": [
        "Congress",
        "Rahul",
        "Gandhi",
        "sunday",
        "China",
        "India"
    ],
    "incorrect_text_1": [],
    "capital_percentage": 2.564102564102564,
    "repetitive_characters": 0,
    "special_percentage": 1.0256410256410255,
    "links": [
        {
             "link": "https://timesofindia.indiatimes.com/india/must-compete-
with-china-in-creating-jobs-rahul-gandhi/articleshow/63667815.cms",
            "match_percentage": 0.9302325581395349,
            "source": "Times of India ",
            "source_status": 1
        },
        {
             "link": "https://www.indiatoday.in/india/story/must-compete-with-
china-in-creating-jobs-rahul-gandhi-1207515-2018-04-08",
            "match_percentage": 0.9302325581395349,
            "source": "India Today",
            "source_status": 1
        },
        {
             "link": "http://www.mijaaj.com/forget-2018-winning-even-bjp-will-
lose-varanasi-seat-under-united-opposition-rahul-gandhi/",
            "match_percentage": 0.9302325581395349,
               "source": "http://www.mijaaj.com/forget-2018-winning-even-bjp-
will-lose-varanasi-seat-under-united-opposition-rahul-gandhi/",
            "source_status": 0
        },
    ],
    "sentiments": [
        {
            "score": 0.671911,
            "tone_id": "joy",
            "tone_name": "Joy"
        },
        {
            "score": 0.608441,
            "tone_id": "confident",
            "tone_name": "Confident"
        }
    ]
“status” : “genuine” }
```

Once enough data to train the model is there, we can apply Machine Learning algorithms to train the dataset.

5.2 TELEGRAM CHAT BOTS

Bots are simply Telegram accounts operated by software – not people and they'll often have AI features. They can do anything – teach, play, search, broadcast, remind, connect, integrate with other services, or even pass commands to the Internet of Things. Telegram apps makes interacting with bots super-easy. In most cases you won't even have to type anything, because bots will provide you with a set of custom buttons.

Imagine interacting with Virudh's backend using a chatbot. Just send a news via bot and the results are delivered back.

5.3 OPEN SOURCE

The project is on GitHub under MIT license. Soon it will be under an open source, **Yogdan**. We are expecting more contributors to the project.

CHAPTER 6
CONCLUSION

We are now able give an idea to user about which news can be possibly fake. A range of tests are performed to successfully extract relevant features of the news that are used to check its authenticity. These features are then stored in the database which in near future will be used to apply Machine Learning to get attain accuracy while classification.

The user can use Android Application or call APIs to get results. The API are optimized working efficiently and deliver the required results. Though we have tried to make a perfect system, still there is scope of improvements.

REFERENCES

[1] Title of web page: “Fake_news”, Available [“https://en.wikipedia.org/wiki/Fake_news”]
[2] Damian Mrowca, “Stance Detection for Fake News Identification”, Stanford Report
[3] Title of web page: “How To Spot Fake News”, Available [”https://www.ifla.org/publications/node/11174"]
[4] Title of web page: “UNESCO declares Modi best Prime Minister: Top 10 fake news that we (almost) believed in 2016”, Available [“https://www.indiatoday.in/india/story/top-ten-fake-news-that-we-almost-believed-in-2016-modi-best-pm-declared-unesco-359619-2016-12-26”]
[5] Title of web page: “I trained fake news detection AI with >95% accuracy, and almost went crazy”, Available [“https://towardsdatascience.com/i-trained-fake-news-detection-ai-with-95-accuracy-and-almost-went-crazy-d10589aa57c"]