Infodemic and Misinformation

Team Name	**First Freshest Freshies**	
Team Members	Aiken Wong	e0543820@u.nus.edu
	Bryan Kwok	bryankwokyf@u.nus.edu
	Chow En Rong	chowenrong@u.nus.edu
	Lee Jae Ho	e0564945@u.nus.edu
	Ryan Kwok	ryankwok@u.nus.edu

General Brief

- Text should be Tahoma font size 11, line spacing of 1.5
- Ensure you have filled up your team name and team members details
- Keep this proposal to a minimum of 5 pages, maximum of 10 pages excluding cover page, content outline and appendix

Content Outline

Significance of Problem

6 in 10 Singaporeans receive fake news about Covid19 from messaging platforms such as WhatsApp, Telegram, and Facebook (Chew, 2020). Yet, many Singaporeans still fall into the trap of believing such news, even though they may originate from untrusted sources.

The root of this problem is not due to the lack of platforms that debunk false information, but the insufficient education amongst Singaporeans on how to determine the reliability of a piece of information. Efforts from the government on educating Singaporeans about identifying false information have also only manifested into posters and infographics (Singapore Government, 2018), and we feel that there is more that can be done.

Singaporeans' difficulty in determining the reliability of information goes beyond just believing fake news. It extends to the inability to objectively understand opinion pieces and develop a balanced perspective on controversial topics. This creates a nation that becomes easily swayed by illegitimate opinions without proper consideration of both sides of the coin.

Solutions Introduction

For most of us, when faced with information of unknown reliability, our natural reaction is to Google it. From the results, we would then determine how legitimate our information is. This process is in line with the government's recommendations to look out for spelling and triggers of strong emotion, as well as to corroborate information with trusted sources. This has also been supported by various agencies such as MediaLiteracyCouncilSG (MediaLiteracyCouncilSG, 2018), TOUCH Community Services (TOUCH Community Services, 2018), and the National Library Board (National Library Board Singapore, 2020). However, this habit is not common among Singaporeans.

Grounded as a Telegram bot, our solution aims to automate this process, and return a recommendation that is accompanied by reasoning to educate users on how to determine reliability in a piece of information. For example, when a user receives a questionable message, he/she can copy the text to the bot, which would then show scores for three criteria: literacy, sentiments, and Google Search corroboration.

Firstly, the piece of text would undergo literacy analysis which consists of a spell checker and a reading level estimator. The spell checker would flag misspelled words and ultimately produce a spelling score. The reading level estimator would analyse the text and determine its complexity. This would produce a reading level score, which would be combined with the spelling score to give a literacy score.

Next, the input text would be analysed with an artificial intelligence (AI) capable of sentiment analysis. The overall objectivity of the text would be determined. Texts that are extremely positive or negative would produce a low sentiment score. High sentiment scores would reflect an objective or balanced article.

Lastly, our program would perform a Google search on this piece of text and collect information (URL, header, synopsis, date) on the results produced, exactly as shown in a normal Google search. An AI would be used to assist in the comparison of results (header and synopsis) against the original text to determine the extent that it corroborates with the user input. The URL and the date of the results would also be processed to determine the reliability of the source and currency of the result. Together, this gives a Google search score.

In the end, the scores for each criterion are returned to the user. Accompanying it would be a brief explanation and recommendation. For example, "This piece of text does not look professionally written and seems to purposefully trigger strong emotion. However, it appears to be sufficiently supported by credible sources. We recommend that you read this article with a pinch of salt and do further research to fully understand all sides of this story."

Impact of Solution

"Give a man a fish, and you feed him for a day. Teach a man to fish, and you feed him for a lifetime." In this day and age, we are inundated with information from various platforms, including WhatsApp, Facebook, and other online news sources. Traditional fact checkers, while useful, can only draw a black and white distinction between real and fake news.

Fortunately, our solution does not aim to remove or debunk false information. Instead, the key is to educate people how to intelligently identify and detect the reliability of information. With Singaporeans being better able to understand the importance of balanced and objective articles when pursuing information, any overload of information regardless of quantity or truth, will never faze discerning Singaporeans.

Such a solution, however, is non-existent in Singapore, until now.

Deep Dive into Solution

The following diagram (Figure 1) outlines the overall process of our solution. Our solution would be further explained in terms of its individual sections.

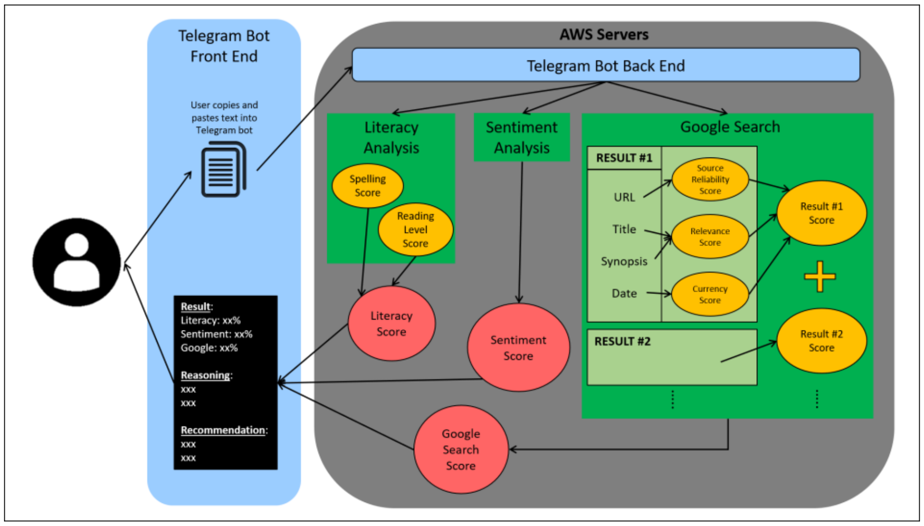


Figure 1: Overall Flow Diagram

- Telegram Platform

The Telegram bot serves as the front-end user interface for our user to input the text they want to be analysed. Users interact with the bot via telegram commands, such as '/start' to input their text or '/help' to view the guidelines for using the bot. The bot will be hosted on AWS Lambda. User input passed in from the front end of the bot gets sent to the Telegram servers.

This is in turn passed to AWS Lambda via a webhook, in which our Python script is executed to compute its reliability. Thereafter, the result will be sent back to the Telegram servers via a HTTP post call and then delivered to users. The scores for each criterion (literacy, sentiment, and Google search corroboration) are shown to educate the user on how to more critically judge information that they come across in future.

Screenshots depicting the Telegram platform can be found in the Appendix (Figure 5).

- Literacy Analysis

Literacy analysis consists of a spell checker and a reading level estimator. Since the Telegram backend is written in Python, we have also opted to use Python implementations of these programs. This code would be hosted on AWS Lambda.

The spell checker uses 'pyspellchecker' to check for spelling errors (Barrus, 2020). This package utilises a Levenshtein Distance algorithm that analyses phrases to determine spelling errors within them. The total number of correctly spelled words is then taken as a percentage of all the words in the original text, producing a spelling score.

The second feature would be a reading level estimator. This would discern credible news articles which generally use more complex vocabulary and sentence structures compared to less credible messages that are usually simplified. We opted to use the Python package 'textstat' that determines the grade level of the text, based on the Flesch Reading Ease formula, Flesch-Kincaid Grade Level, Gunning FOG formula, SMOG index, Automated Readability Index, Coleman-Liau Index, Linsear Write Formula, and the Dale-Chall Readability Score (Shivam Bansal, 2020). The grade output is the estimated United States grade level required to understand the text. The final score is the grade added to 5 to produce the age equivalent, which would be more relevant in Singapore's context.

Screenshots of sample codes as well as their results can be found in the Appendix (Figure 6-9).

- Sentiment Analysis

Journalistic objectivity is one of the standards in news reporting. This means that any article with significantly positive or negative sentiment is less reliable, regardless of its factual truth. We aim to judge this using AWS Comprehend, which is a machine learning, natural language processing service. Accessed via AWS Lambda, the sentiment analysis service would be able to identify the overall sentiment of the text as either positive, negative, neutral, or mixed.

To demonstrate how this would work, we have used AWS Comprehend on a real text message spreading on WhatsApp, as well as a news article from Channel News Asia (CNA). As expected, the CNA article showed a higher neutral/mixed sentiment, which would ultimately translate into a higher sentiment score.

Screenshots of the results can be found in the Appendix (Figure 10-11).

- Google Search (Web Scraper)

The web scraper is a Python script that takes the user input and inserts it into a Google search query. Since queries are done with keywords not a paragraph, AWS Comprehend would be used to extract keywords the text before sending it into a goodle. This results in more accurate results with less "noise" for the Google algorithm.

As the Google search results page is dynamic, we will use a Python module 'requests_html' to get and fully render the website's HTML via execution of its JavaScript code. From there, we parse the HTML object with another module 'BeautifulSoup' in order to access individual HTML tags and their text content.

As shown in Figure 1, each search result is divided into its URL, header, synopsis, and date published. The URL and date will be analysed to produce the source reliability score and the currency score. The header and synopsis will be passed on to an AI model for comparison and calculation of the relevance score. This Python script would be hosted on AWS Lambda. Screenshots of the code and its output can be found in the Appendix (Figure 12-13).

- Google Search (AI)

In order to determine the relevance of a piece of text, a semantic comparison would be made between the user input and the search results (header and synopsis of news articles) from our web scraper. A high score would suggest that the Google search results corroborates with the user input. To do this, we will be using a state-of-the-art Natural Language Processing (NLP) model known as BERT - Bidirectional Encoder Representations for Transformers (Devlin et al., 2018), which is widely used in NLP tasks today. Due to the high transferability of BERT encodings to other NLP tasks, we will be using a pre-trained base model trained on cased English text (huggingface, 2020).

Our preliminary model that was trained over 2-3 epochs on either the Microsoft Research Paraphrase Corpus (MRPC) or the Semantic Textual Similarity Benchmark (STSB) datasets yielded respectable results of about 84% validation accuracy. If we have the opportunity to, we recommend fine-tuning the model to our use case with a large, manually-generated training set using our web scraper.

Screenshots of sample execution of the model and the results can be found in the Appendix (Figure 14).

- Generation of Reply

The three scores will be compiled to produce a custom message in order to cater to all the unique types of text input. This is because different scores would require different explanations and recommendations to be sent to the user. The tables below outline the variations of both the explanations and the recommendations that will be sent to the user.

In addition to the reply, we would also share one or two links taken from our Google search. This aims to cultivate a habit of conducting further research for credible information in the user. This will be combined together with the explanation and recommendation to form a complete message. A sample of a complete reply can be found in the Appendix (Figure 15).

Score Type	Sample Explanations
Literacy	This piece of text (looks / does not look) very professionally written
Sentiment	(and / but) seems to give a neutral/balanced perspective. (and / but) seems to purposefully trigger strong emotions.
Google Search	(However / Additionally), it appears to be (in)sufficiently supported by credible sources.

Figure 2: Table of Sample Explanations

Score Type	Sample Recommendations
High Overall	This article seems well written and well supported!
High Google Search with Low Literacy/ Sentiment	Therefore, we recommend that you read this article with a pinch of salt and do further research to fully understand all sides of this story.
High Literacy/ Sentiment with Low Google Search	Hence, we recommend that you search up more articles from reputable sources to better understand the truths of this topic.
Low Overall	This article is poorly written and poorly supported, it is very likely that this was created to spread false information. We highly recommend that you seek more articles from reputable sources for more reliable information. You can also send me new texts you find!

Figure 3: Table of Sample Recommendations

Architecture of Solution

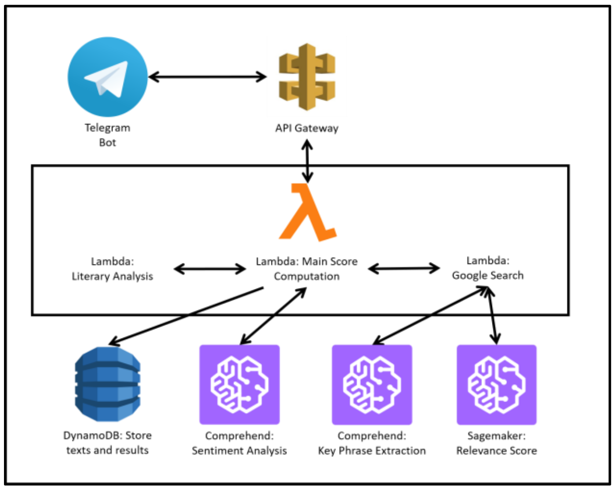

Telegram Bot
API Gateway
Lambda: Literary Analysis
Lambda: Main Score Computation
Lambda: Google Search
DynamoDB: Store texts and results
Comprehend: Sentiment Analysis
Comprehend: Key Phrase Extraction
Sagemaker: Relevance Score

Figure 4: AWS Architecture of Solution

1. Since Telegram is the front end application that the user would be accessing our solution from, AWS API Gateway would help to bridge the users to the process via a webhook.
2. Executing a command in Telegram would trigger the Python script in AWS Lambda. From here, a few processes would be initiated.
3. Firstly, another Python script would be run to execute the literary analysis of the text provided. This returns a score to the main script.
4. Secondly, AWS Comprehend would be triggered to analyse the text's sentiments. Similarly, these scores would be sent back to the main code for computation. It will also be used to extract keywords for web scraping below.
5. Next, the script for the execution of the Google search would be triggered. This includes the query and web scraping of the result site. Results of the web scraping would be sent to AWS SageMaker.
6. A pre-trained machine learning model running on AWS SageMaker would then determine the relevance score by comparing the search results with the original text.
7. The main script would then collate all the scores. The three main scores (literacy, sentiment, and Google search corroboration), along with the original text, would be sent to AWS DynamoDB for logging and possibly future development.
8. The final results would be sent back to the user via the Telegram bot.

Going Further

In discussing future development, we have categorised our ideas into 3 sections, namely commercialisation, scaling, and future improvements. These sections will be explained below.

- Commercialisation

Our solution was not designed for financial profit since the aim is to educate the general public. Hence, to better cater to education, we hope to again integrate more complex AI to better analyse reliability of the information. As of now, we are using basic guidelines to identify factors that might indicate that the information is unreliable. With more time, we can look into research papers and discover key information that can be derived from a text to better determine its reliability. We would then be able to train our AI to measure more of these features from our data set to produce more accurate results.

With a more research-backed foundation, we would then be able to pitch our solution to education institutions. Not only would our solution tackle "fake news", but also aid students in better analysing information and sources when embarking on projects and reports in their schooling life. This will only boost the overall awareness and discernment of Singaporeans.

With all of that said, if commercialisation is a necessity, a potential financial market would be the journalism industry. As they aim to produce more objective and balanced news articles, an AI powered service would be useful for news agencies to progress towards that gold standard.

- Scaling

The first aspect to scaling would be the platform base. Telegram was chosen due to our familiarity with its API and its ease of use. However, with more time and financial resources, we hope to create a similar service for Facebook Messenger and WhatsApp to better target unreliable news that spread on these platforms. This digital scaling allows for our solution to be utilised by a greater pool of users with increasing ease.

Another aspect would be to expand geographically. This would not only mean being able to cater to a greater range of languages, but also being able to handle information from a greater number of sources.

However, the political and social situation in each country is unique, hence there would be different expectations of reliability of information. This also comes with determining the reliability of a greater number of news sources that we might not have even heard of. Scaling would definitely be possible, but only with more time, resources, and research.

- Further Improvements

With the limited time and access to data, we have to rely on open source material available online. While it generally achieves the goal, we would hope to integrate AI technologies to a greater extent. One benefit of a specifically trained AI is its ability to cater to specifically Singaporean characteristics. For example, the spell checker that we are using is a pre-trained model and does not include all words that may be used in a Singaporean context. Terms like temple names and MRT would be flagged out as misspellings.

In addition, more AI capabilities would be able to achieve greater detail and hence more accurate results. If possible, we would like to be able to scrape entire websites rather than just the Google results page. However, different websites have different formatting within their HTML code, making it difficult and complex to do web scraping for entire websites. With AI, we would be able to train a model to accurately identify news headlines and content from HTML pages that represent them in different formats. This would allow us to compare the content of a news article rather than just a snippet, increasing accuracy.

We acknowledge that the accuracy of the Google Search AI is not ideal at 84%. However, Natural Language Processing is still an evolving field in the world of neural networks, and BERT is already the State-of-the-Art encoding for language representation. Most advanced AI comparing semantics would also not reach accuracies above 90%. Hence, our approach would be mitigation, not a replacement.

The first step is ensuring that the AI is analysing what it is intended to analyse to maximise accuracy. Originally, we would send the entire user input as a whole into the Google query, and task the AI to compare the paragraph with the sentence in each result. However, comparing based on a large chunk of text would have inconsistencies especially when the search result is relevant to only specifically one sentence in the user input. Hence, we would have to break up the user input into individual sentence before conducting a Google query, and therefore have a more accurate sentence to sentence comparison.

Nevertheless, splitting the user input into single sentences will result in the loss of context. This is worsened with "meaningless sentences" which cannot be understood without the previous of subsequent sentence, also resulting in irrelevant Google search results. Considering both limitations, we have decided to split the user input into chunks of two sentences with overlap, to maximise AI accuracy and retain as much context as possible. To clarify, this would mean the first Google search would be done on sentence 1 and 2, followed by sentence 2 and 3, then sentence 3 and 4, so on and so forth. Clearly, this is subject to future developments. With improved AI or professional opinion by linguists, we can adjust the number of sentences to query and analyse text more accurately and efficiently.

In addition, another limitation observed is that the AI generally struggles with negation sentences such as "Tom is a human" and "Tom is not a human". This would be very easily resolved with the use of Comprehend to determine the sentiment of the sentences in comparison. If the difference in sentiment exceeds a certain threshold, we can indicate that the likelihood of similarity is slim.

Another tool that can help us mitigate problems with negation is the fact checker. Google Search contains a feature called Google Fact Check, which labels a search result as a "fact check" after meeting certain criteria established by 3rd party fact checkers. An example is shown in the Appendix (Figure 16). Hence, we could make use of this tool in our web scraper to recognise such results in order to produce a different score and send an alert to the user. This alert would be beneficial in notifying users of potential falsehoods.

Works Cited

Barrus, T. (18 February, 2020). *pyspellchecker 0.5.4.* Retrieved from PyPI: https://pypi.org/project/pyspellchecker/

Chew, H. (21 May, 2020). *6 in 10 people in Singapore have received fake COVID-19 news, likely on social media: Survey*. Retrieved from Channel News Asia: https://www.channelnewsasia.com/news/singapore/fake-covid-19-news-study-ncid-messaging-platforms-whatsapp-12756084

huggingface. (2020). *Pretrained models*. Retrieved from huggingface: https://huggingface.co/transformers/pretrained_models.html

jonatasbaldin. (4 February, 2019). *serverless-telegram-bot.* Retrieved from GitHub: https://github.com/jonatasbaldin/serverless-telegram-bot

MediaLiteracyCouncilSG. (2018). *How To Spot Fake News.* Retrieved from Media Literacy Council: https://www.betterinternet.sg/-/media/MLC/Files/SID-2018/Quick-Tips/1_How-to-spot-Fake-News_Tipsheet.pdf

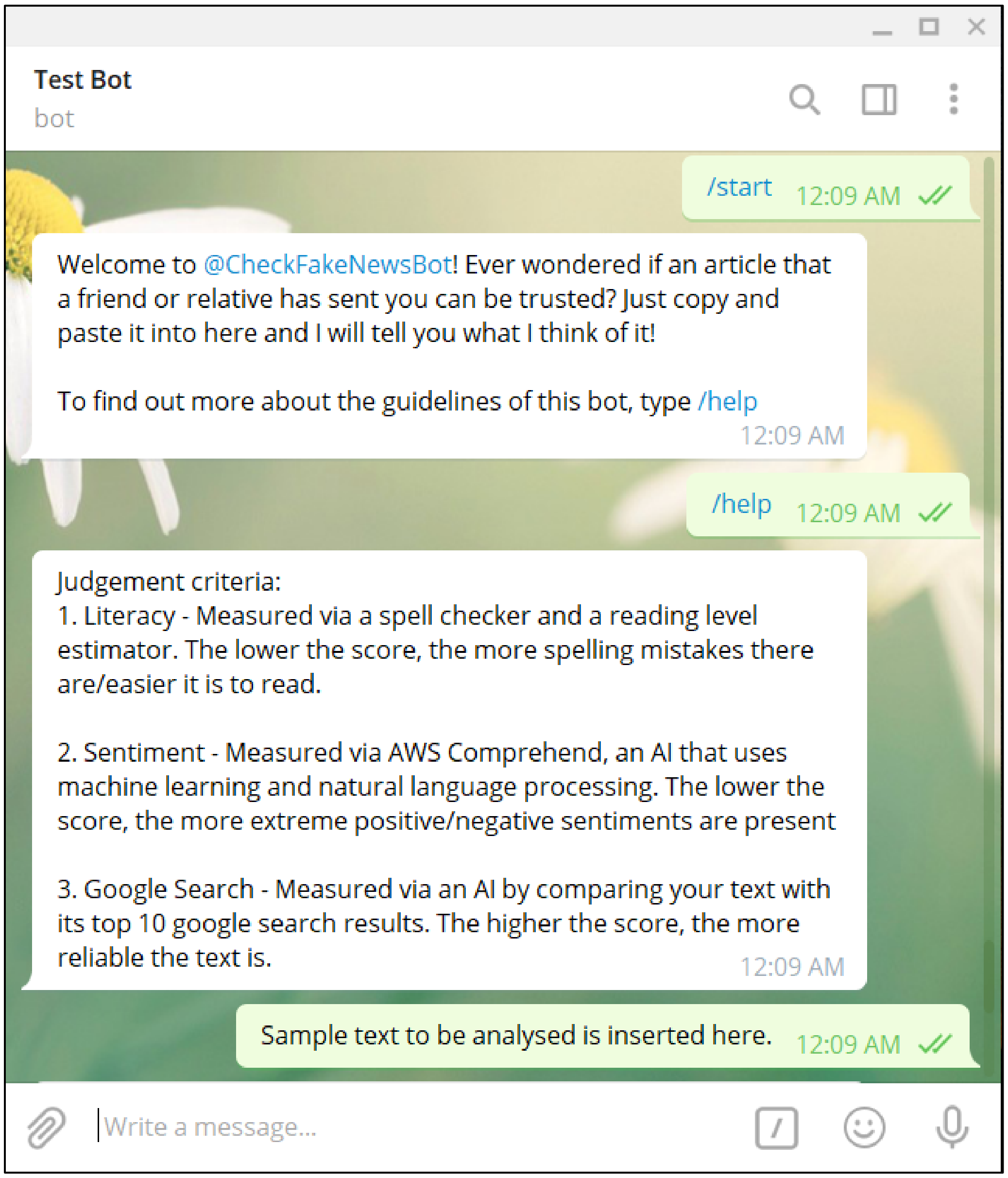
National Library Board Singapore. (06 May, 2020). *How to be S.U.R.E. of your COVID-19 Information.* Retrieved from National Library Board Singapore: https://sure.nlb.gov.sg/blog/fake-news/fn0005

Shivam Bansal, C. A. (24 April, 2020). *textstat 0.6.2.* Retrieved from PyPI: https://pypi.org/project/textstat/

Singapore Government. (22 November, 2018). *Singapore's fight against Fake News: What you can do*. Retrieved from gov.sg: https://www.gov.sg/article/singapores-fight-against-fake-news-what-you-can-do

TOUCH Community Services. (2018). *5 Ways to Spot Fake News.* Retrieved from TOUCH Communinty Services: https://www.touch.org.sg/about-touch/tips-and-resources/details/2018/02/23/5-ways-to-spot-fake-news

Appendix



Test Bot
bot

/start 12:09 AM

Welcome to @CheckFakeNewsBot! Ever wondered if an article that a friend or relative has sent you can be trusted? Just copy and paste it into here and I will tell you what I think of it!

To find out more about the guidelines of this bot, type /help
12:09 AM

/help 12:09 AM

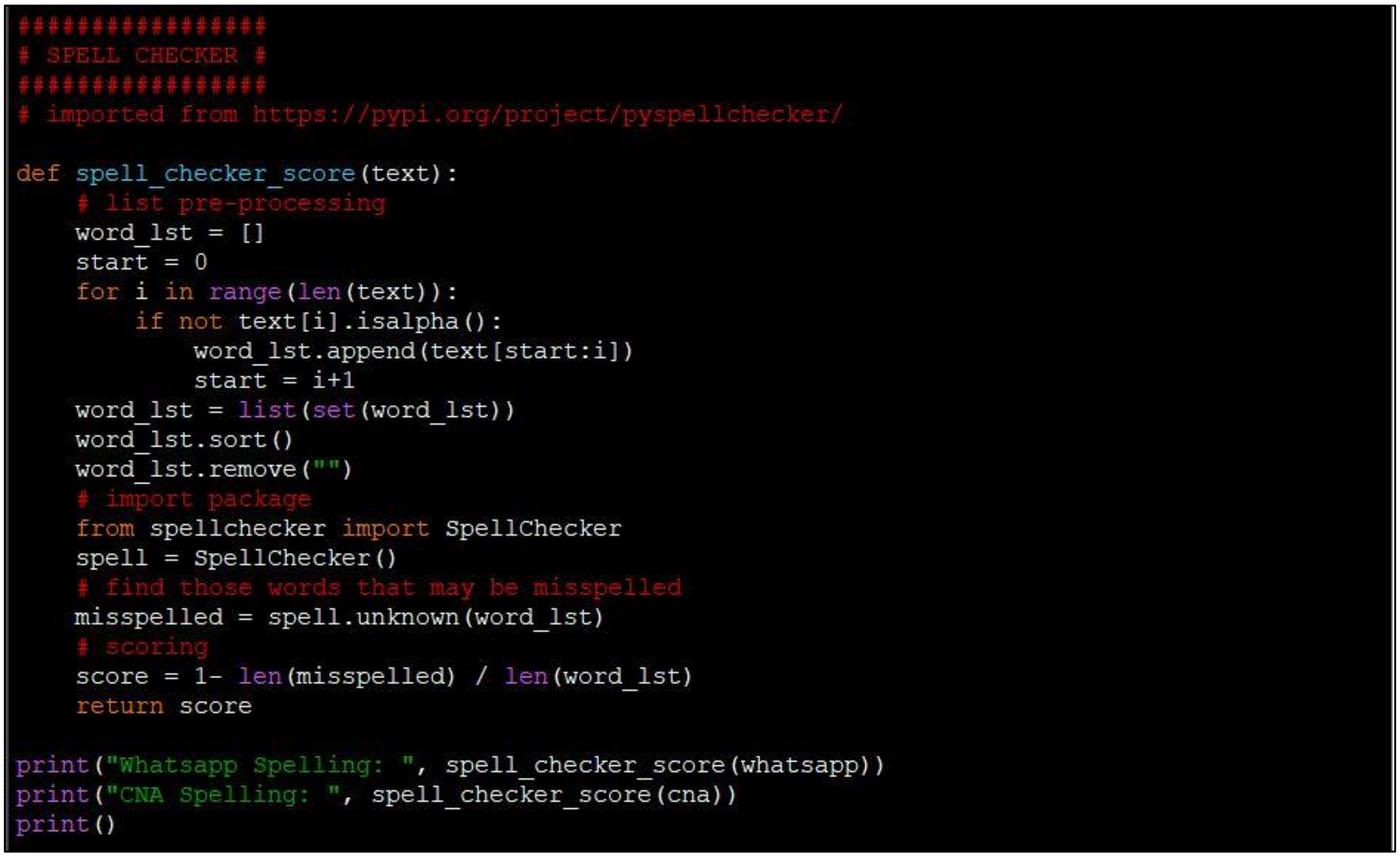
Judgement criteria:
1. Literacy - Measured via a spell checker and a reading level estimator. The lower the score, the more spelling mistakes there are/easier it is to read.

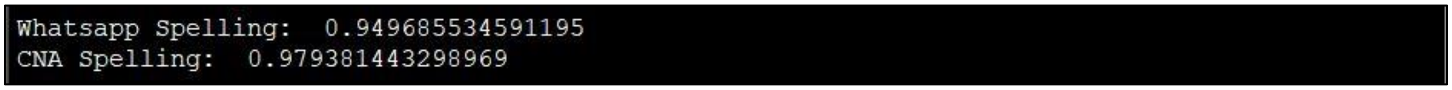
2. Sentiment - Measured via AWS Comprehend, an AI that uses machine learning and natural language processing. The lower the score, the more extreme positive/negative sentiments are present

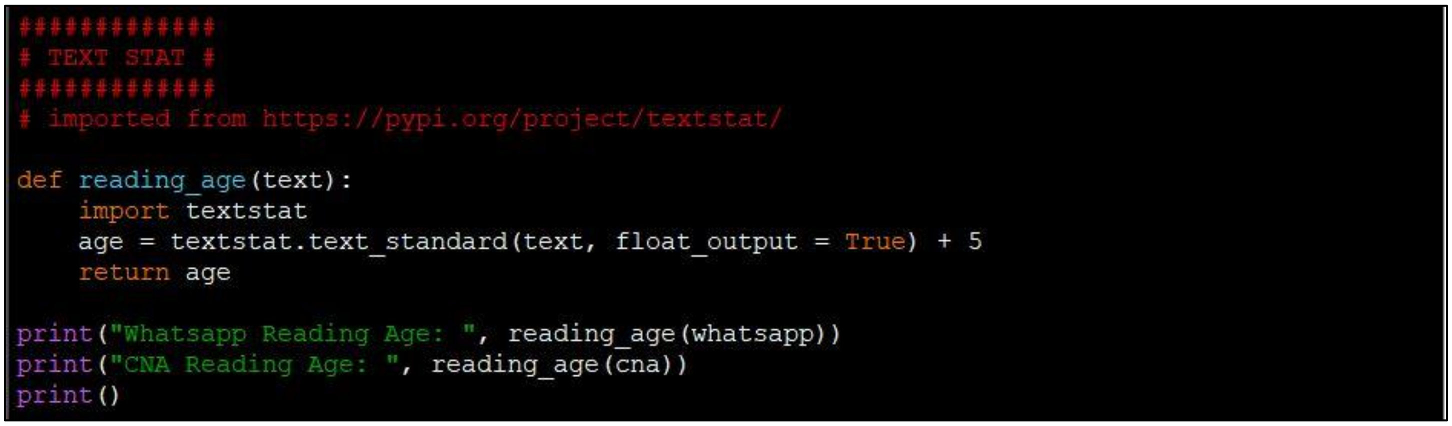
3. Google Search - Measured via an AI by comparing your text with its top 10 google search results. The higher the score, the more reliable the text is.
12:09 AM

Sample text to be analysed is inserted here. 12:09 AM

Figure 5: Screenshot of Telegram Interface

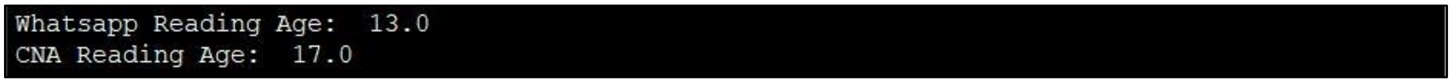

```
##################
# SPELL CHECKER #
##################
# imported from https://pypi.org/project/pyspellchecker/

def spell_checker_score(text):
    # list pre-processing
    word_lst = []
    start = 0
    for i in range(len(text)):
        if not text[i].isalpha():
            word_lst.append(text[start:i])
            start = i+1
    word_lst = list(set(word_lst))
    word_lst.sort()
    word_lst.remove("")
    # import package
    from spellchecker import SpellChecker
    spell = SpellChecker()
    # find those words that may be misspelled
    misspelled = spell.unknown(word_lst)
    # scoring
    score = 1- len(misspelled) / len(word_lst)
    return score

print("Whatsapp Spelling: ", spell_checker_score(whatsapp))
print("CNA Spelling: ", spell_checker_score(cna))
print()
```
Figure 6: Spell Checker Sample Code


```
Whatsapp Spelling:  0.949685534591195
CNA Spelling:  0.979381443298969
```
Figure 7: Spell Checker Sample Results



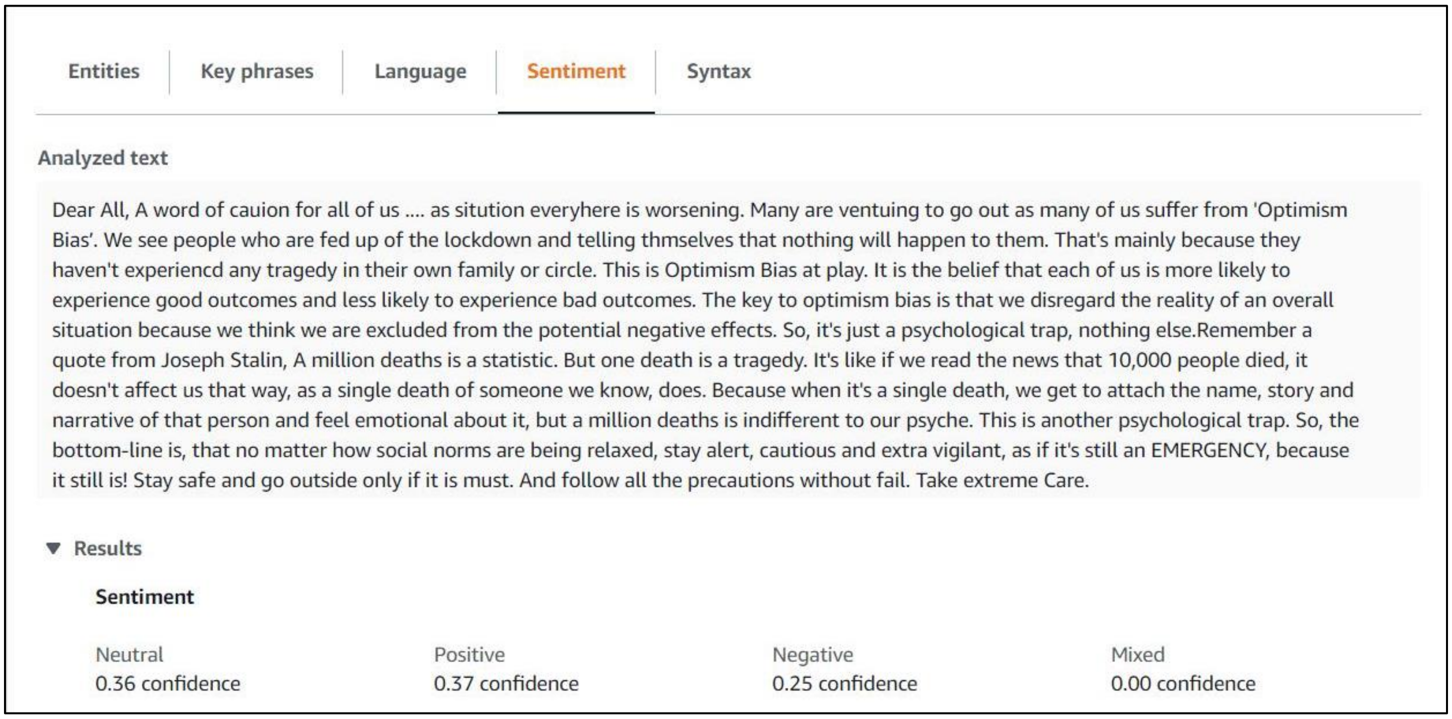
```
##############
# TEXT STAT #
##############
# imported from https://pypi.org/project/textstat/

def reading_age(text):
    import textstat
    age = textstat.text_standard(text, float_output = True) + 5
    return age

print("Whatsapp Reading Age: ", reading_age(whatsapp))
print("CNA Reading Age: ", reading_age(cna))
print()
```

Figure 8: Reading Level Sample Code



```
Whatsapp Reading Age:  13.0
CNA Reading Age:  17.0
```

Figure 9: Reading Level Sample Results



Analyzed text

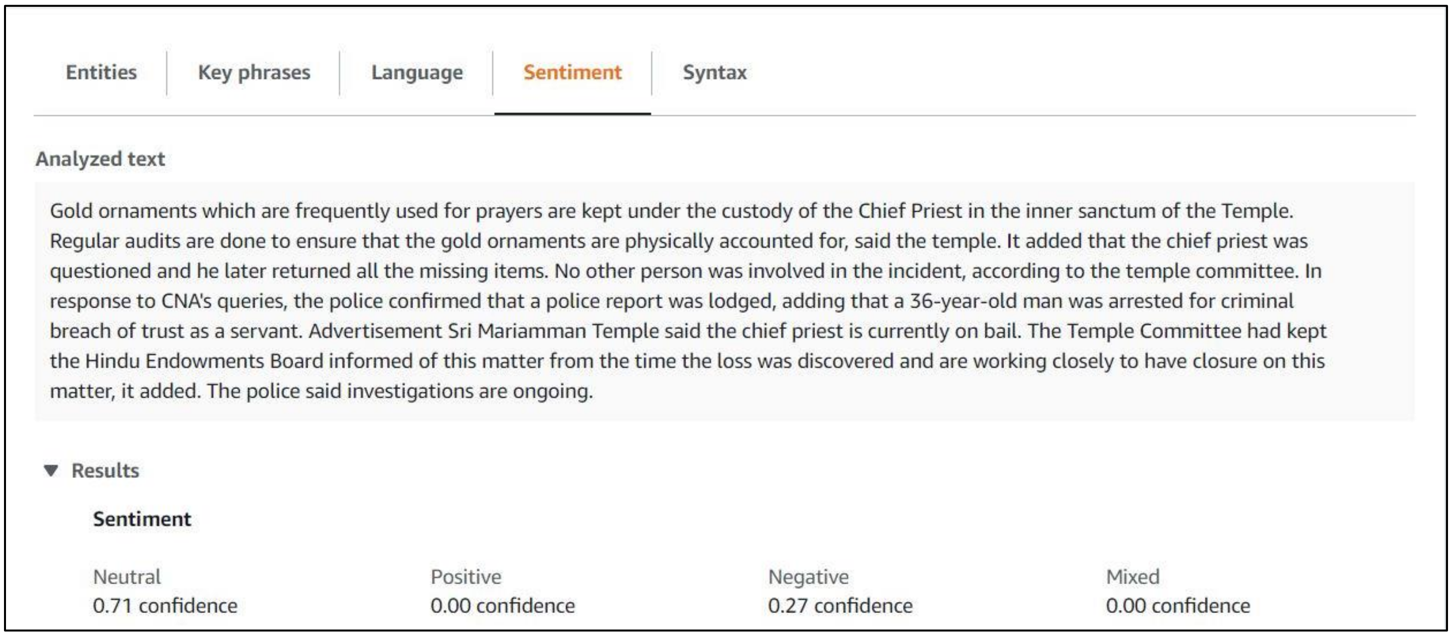
Dear All, A word of cauion for all of us .... as sitution everyhere is worsening. Many are ventuing to go out as many of us suffer from 'Optimism Bias'. We see people who are fed up of the lockdown and telling thmselves that nothing will happen to them. That's mainly because they haven't experiencd any tragedy in their own family or circle. This is Optimism Bias at play. It is the belief that each of us is more likely to experience good outcomes and less likely to experience bad outcomes. The key to optimism bias is that we disregard the reality of an overall situation because we think we are excluded from the potential negative effects. So, it's just a psychological trap, nothing else.Remember a quote from Joseph Stalin, A million deaths is a statistic. But one death is a tragedy. It's like if we read the news that 10,000 people died, it doesn't affect us that way, as a single death of someone we know, does. Because when it's a single death, we get to attach the name, story and narrative of that person and feel emotional about it, but a million deaths is indifferent to our psyche. This is another psychological trap. So, the bottom-line is, that no matter how social norms are being relaxed, stay alert, cautious and extra vigilant, as if it's still an EMERGENCY, because it still is! Stay safe and go outside only if it is must. And follow all the precautions without fail. Take extreme Care.

▼ Results

Sentiment

Neutral	Positive	Negative	Mixed
0.36 confidence	0.37 confidence	0.25 confidence	0.00 confidence

Figure 10: AWS Comprehend Results of a WhatsApp Message



Analyzed text

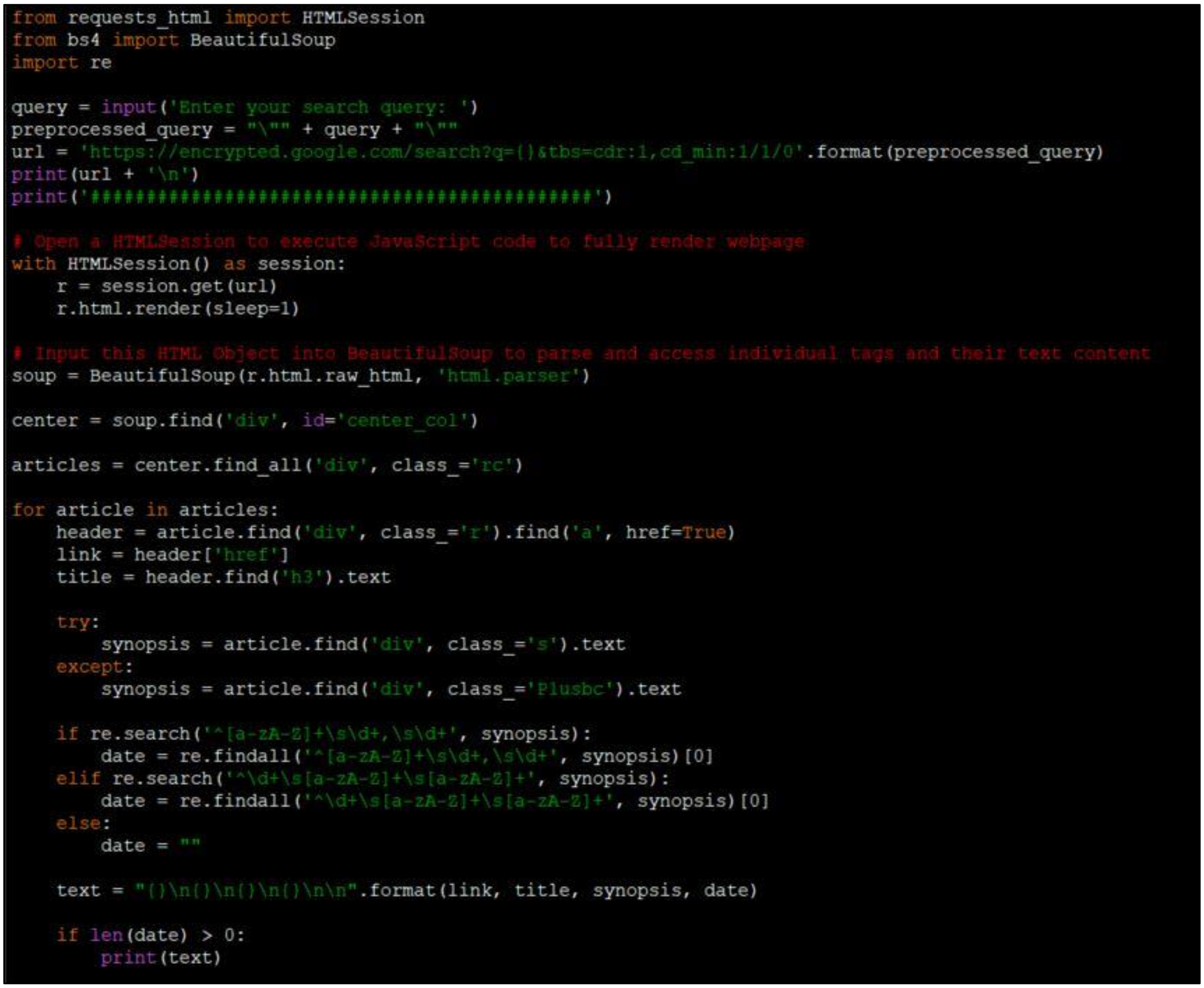
Gold ornaments which are frequently used for prayers are kept under the custody of the Chief Priest in the inner sanctum of the Temple. Regular audits are done to ensure that the gold ornaments are physically accounted for, said the temple. It added that the chief priest was questioned and he later returned all the missing items. No other person was involved in the incident, according to the temple committee. In response to CNA's queries, the police confirmed that a police report was lodged, adding that a 36-year-old man was arrested for criminal breach of trust as a servant. Advertisement Sri Mariamman Temple said the chief priest is currently on bail. The Temple Committee had kept the Hindu Endowments Board informed of this matter from the time the loss was discovered and are working closely to have closure on this matter, it added. The police said investigations are ongoing.

▼ Results

Sentiment

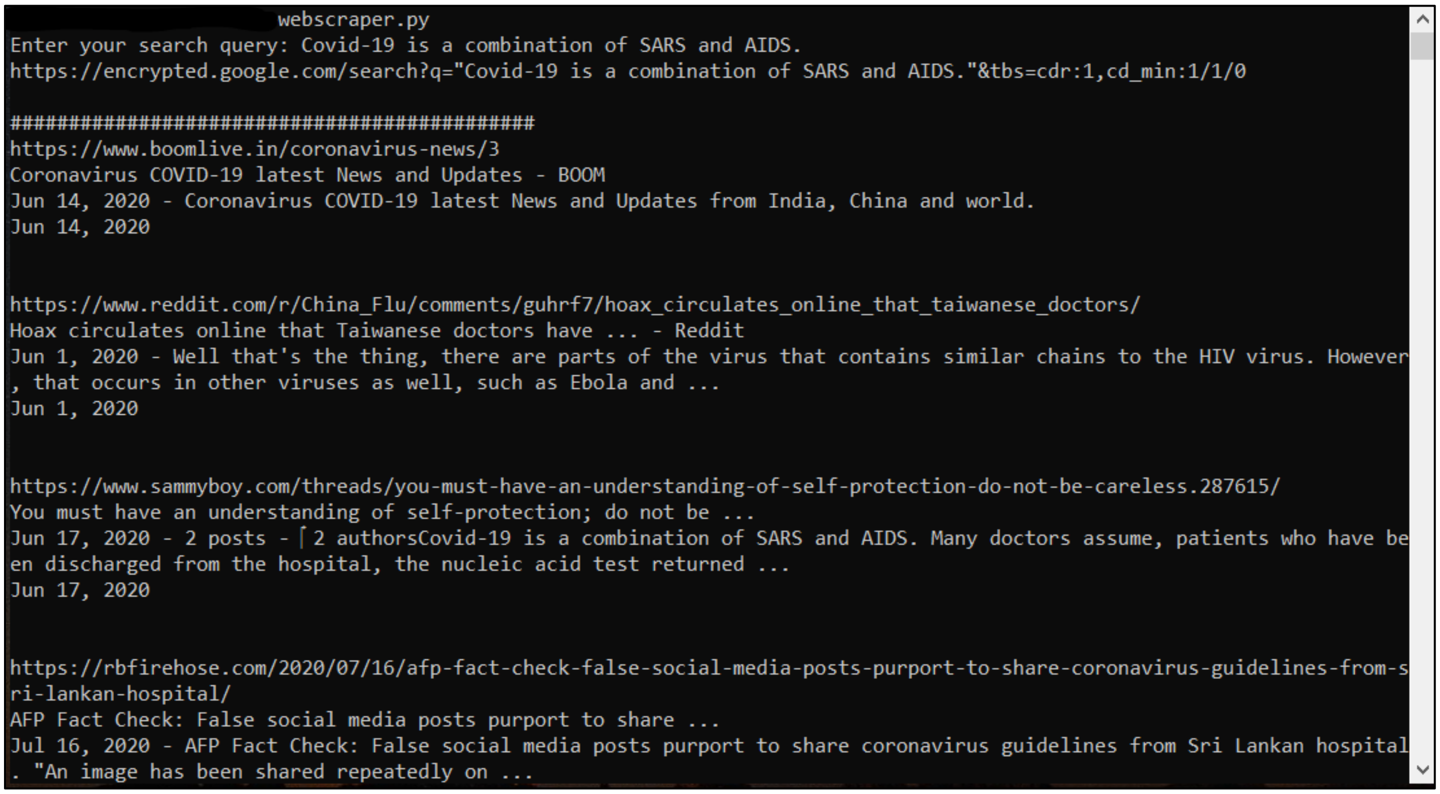
Neutral	Positive	Negative	Mixed
0.71 confidence	0.00 confidence	0.27 confidence	0.00 confidence

Figure 11: AWS Comprehend Results of a CNA Article



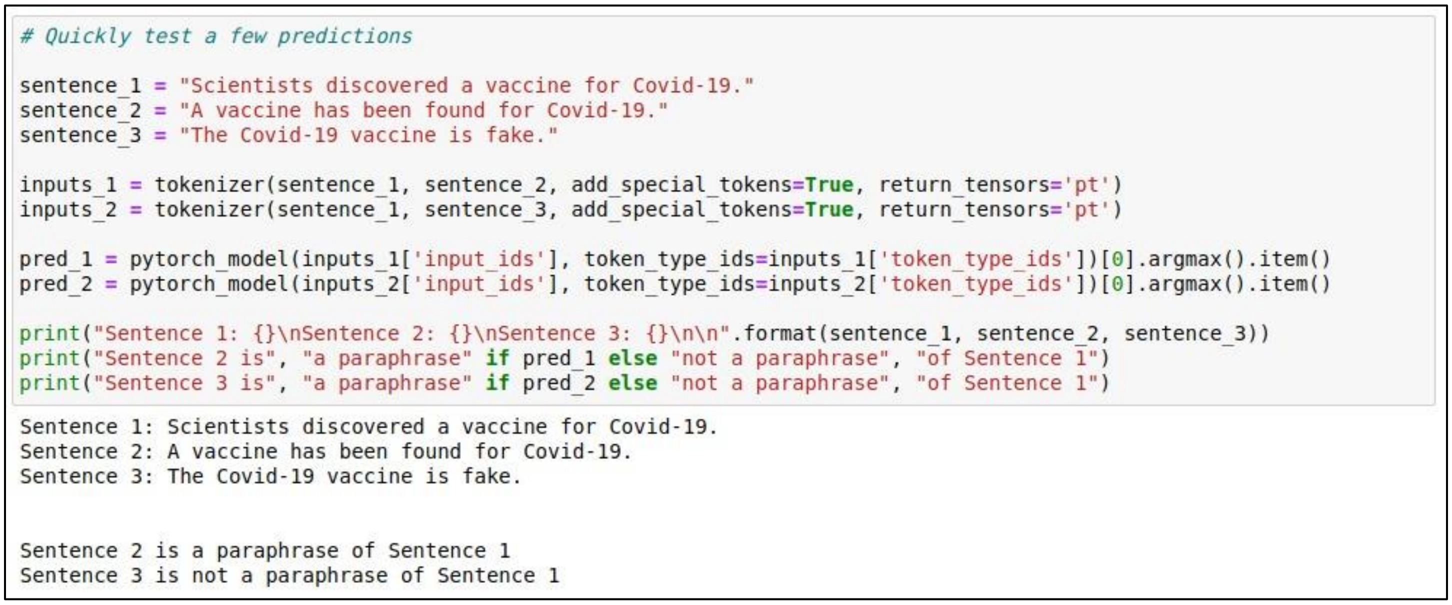
```
from requests_html import HTMLSession
from bs4 import BeautifulSoup
import re

query = input('Enter your search query: ')
preprocessed_query = "\"" + query + "\""
url = 'https://encrypted.google.com/search?q={}&tbs=cdr:1,cd_min:1/1/0'.format(preprocessed_query)
print(url + '\n')
print('#############################################')

# Open a HTMLSession to execute JavaScript code to fully render webpage
with HTMLSession() as session:
    r = session.get(url)
    r.html.render(sleep=1)

# Input this HTML Object into BeautifulSoup to parse and access individual tags and their text content
soup = BeautifulSoup(r.html.raw_html, 'html.parser')

center = soup.find('div', id='center_col')

articles = center.find_all('div', class_='rc')

for article in articles:
    header = article.find('div', class_='r').find('a', href=True)
    link = header['href']
    title = header.find('h3').text

    try:
        synopsis = article.find('div', class_='s').text
    except:
        synopsis = article.find('div', class_='Plusbc').text

    if re.search('^[a-zA-Z]+\s\d+,\s\d+', synopsis):
        date = re.findall('^[a-zA-Z]+\s\d+,\s\d+', synopsis)[0]
    elif re.search('^\d+\s[a-zA-Z]+\s[a-zA-Z]+', synopsis):
        date = re.findall('^\d+\s[a-zA-Z]+\s[a-zA-Z]+', synopsis)[0]
    else:
        date = ""

    text = "{}\n{}\n{}\n{}\n\n".format(link, title, synopsis, date)

    if len(date) > 0:
        print(text)
```

Figure 12: Web Scraper Sample Code



```
[redacted] webscraper.py
Enter your search query: Covid-19 is a combination of SARS and AIDS.
https://encrypted.google.com/search?q="Covid-19 is a combination of SARS and AIDS."&tbs=cdr:1,cd_min:1/1/0

#############################################
https://www.boomlive.in/coronavirus-news/3
Coronavirus COVID-19 latest News and Updates - BOOM
Jun 14, 2020 - Coronavirus COVID-19 latest News and Updates from India, China and world.
Jun 14, 2020


https://www.reddit.com/r/China_Flu/comments/guhrf7/hoax_circulates_online_that_taiwanese_doctors/
Hoax circulates online that Taiwanese doctors have ... - Reddit
Jun 1, 2020 - Well that's the thing, there are parts of the virus that contains similar chains to the HIV virus. However
, that occurs in other viruses as well, such as Ebola and ...
Jun 1, 2020


https://www.sammyboy.com/threads/you-must-have-an-understanding-of-self-protection-do-not-be-careless.287615/
You must have an understanding of self-protection; do not be ...
Jun 17, 2020 - 2 posts - ⌈2 authorsCovid-19 is a combination of SARS and AIDS. Many doctors assume, patients who have be
en discharged from the hospital, the nucleic acid test returned ...
Jun 17, 2020


https://rbfirehose.com/2020/07/16/afp-fact-check-false-social-media-posts-purport-to-share-coronavirus-guidelines-from-s
ri-lankan-hospital/
AFP Fact Check: False social media posts purport to share ...
Jul 16, 2020 - AFP Fact Check: False social media posts purport to share coronavirus guidelines from Sri Lankan hospital
. "An image has been shared repeatedly on ...
```

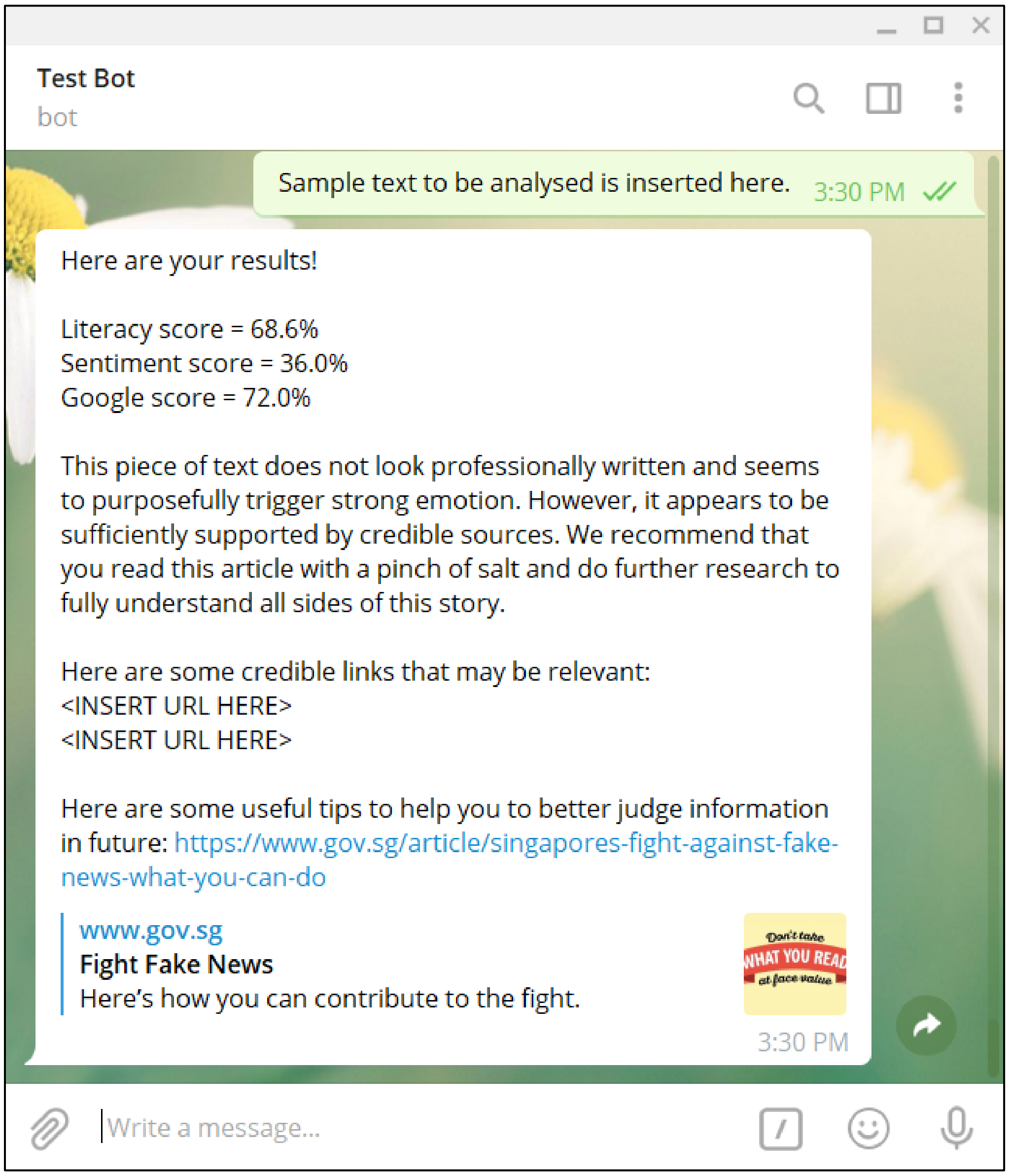
Figure 13: Web Scraper Sample Results



```python
# Quickly test a few predictions

sentence_1 = "Scientists discovered a vaccine for Covid-19."
sentence_2 = "A vaccine has been found for Covid-19."
sentence_3 = "The Covid-19 vaccine is fake."

inputs_1 = tokenizer(sentence_1, sentence_2, add_special_tokens=True, return_tensors='pt')
inputs_2 = tokenizer(sentence_1, sentence_3, add_special_tokens=True, return_tensors='pt')

pred_1 = pytorch_model(inputs_1['input_ids'], token_type_ids=inputs_1['token_type_ids'])[0].argmax().item()
pred_2 = pytorch_model(inputs_2['input_ids'], token_type_ids=inputs_2['token_type_ids'])[0].argmax().item()

print("Sentence 1: {}\nSentence 2: {}\nSentence 3: {}\n\n".format(sentence_1, sentence_2, sentence_3))
print("Sentence 2 is", "a paraphrase" if pred_1 else "not a paraphrase", "of Sentence 1")
print("Sentence 3 is", "a paraphrase" if pred_2 else "not a paraphrase", "of Sentence 1")
```

```
Sentence 1: Scientists discovered a vaccine for Covid-19.
Sentence 2: A vaccine has been found for Covid-19.
Sentence 3: The Covid-19 vaccine is fake.


Sentence 2 is a paraphrase of Sentence 1
Sentence 3 is not a paraphrase of Sentence 1
```

Figure 14: Google Search AI Sample Execution and Results


Test Bot
bot

Sample text to be analysed is inserted here. 3:30 PM

Here are your results!

Literacy score = 68.6%
Sentiment score = 36.0%
Google score = 72.0%

This piece of text does not look professionally written and seems to purposefully trigger strong emotion. However, it appears to be sufficiently supported by credible sources. We recommend that you read this article with a pinch of salt and do further research to fully understand all sides of this story.

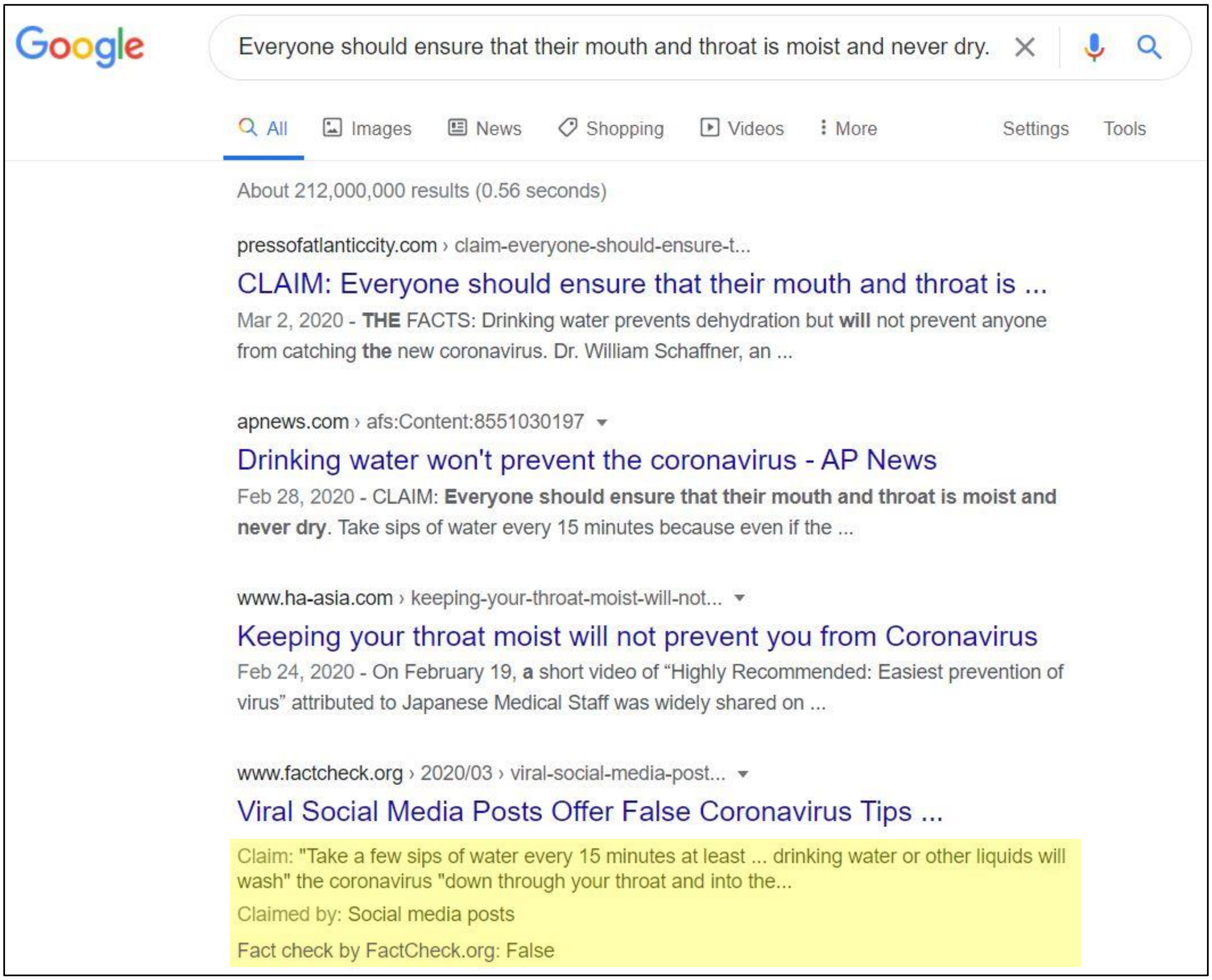
Here are some credible links that may be relevant:
<INSERT URL HERE>
<INSERT URL HERE>

Here are some useful tips to help you to better judge information in future: https://www.gov.sg/article/singapores-fight-against-fake-news-what-you-can-do

www.gov.sg
Fight Fake News
Here's how you can contribute to the fight.

3:30 PM

Figure 15: Sample Reply on Telegram



Everyone should ensure that their mouth and throat is moist and never dry.

About 212,000,000 results (0.56 seconds)

pressofatlanticcity.com › claim-everyone-should-ensure-t...
CLAIM: Everyone should ensure that their mouth and throat is ...
Mar 2, 2020 - **THE** FACTS: Drinking water prevents dehydration but **will** not prevent anyone from catching **the** new coronavirus. Dr. William Schaffner, an ...

apnews.com › afs:Content:8551030197
Drinking water won't prevent the coronavirus - AP News
Feb 28, 2020 - CLAIM: **Everyone should ensure that their mouth and throat is moist and never dry**. Take sips of water every 15 minutes because even if the ...

www.ha-asia.com › keeping-your-throat-moist-will-not...
Keeping your throat moist will not prevent you from Coronavirus
Feb 24, 2020 - On February 19, **a** short video of "Highly Recommended: Easiest prevention of virus" attributed to Japanese Medical Staff was widely shared on ...

www.factcheck.org › 2020/03 › viral-social-media-post...
Viral Social Media Posts Offer False Coronavirus Tips ...
Claim: "Take a few sips of water every 15 minutes at least ... drinking water or other liquids will wash" the coronavirus "down through your throat and into the...
Claimed by: Social media posts
Fact check by FactCheck.org: False

Figure 16: Fact Checking Function from Google