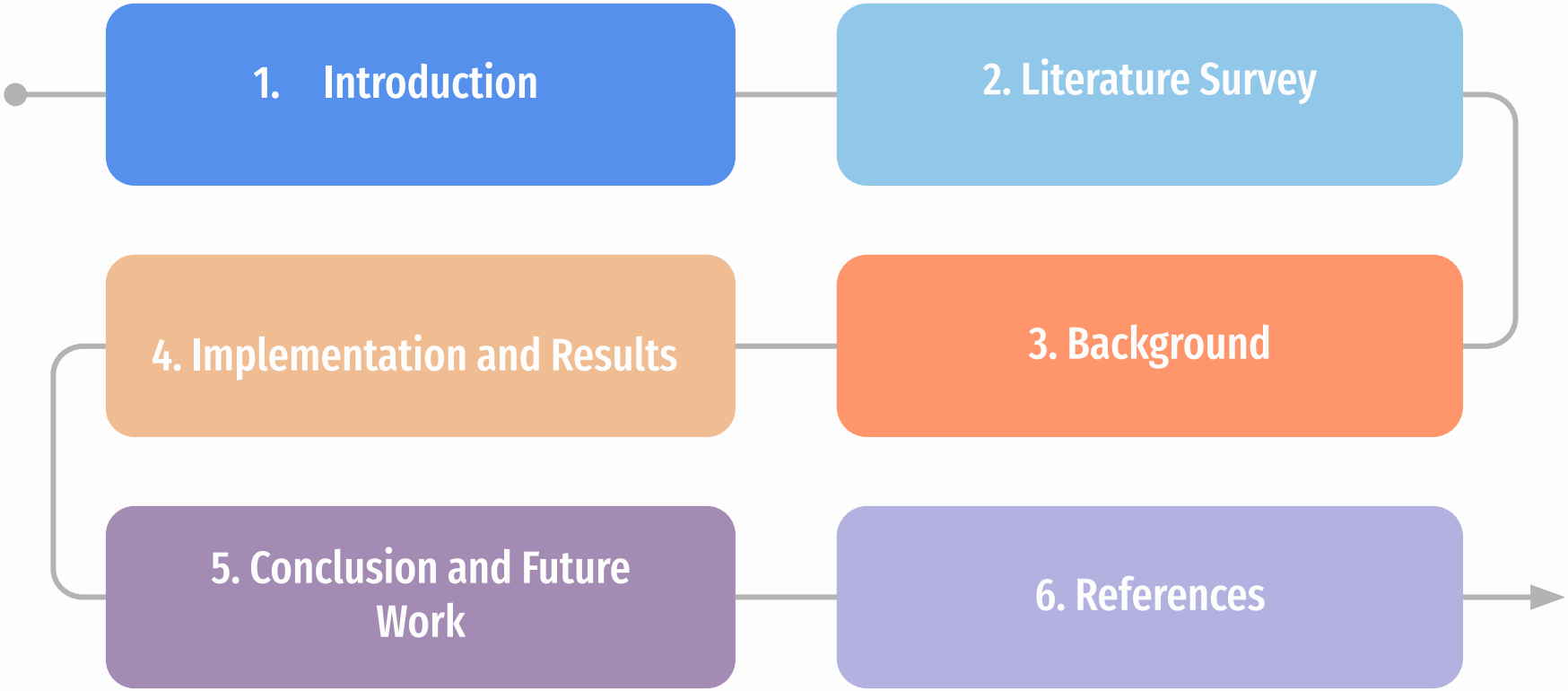
Sentiment Analysis on Covid Dataset

Created By: Yamini Chitikela, Rishika Devaragatla, Chaitanya Yadavally, Sravanthi Shyamreddy , Saidi Reddy Chennu

Contents

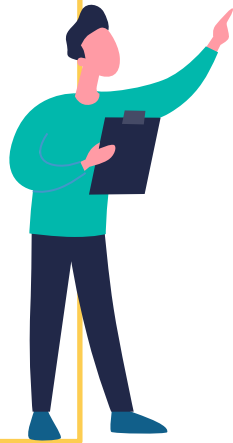

1. Introduction
2. Literature Survey
3. Background
4. Implementation and Results
5. Conclusion and Future Work
6. References

Introduction

- One of the most pressing challenges related to COVID-19 is finding effective treatments during the initial stage of outbreak.. Since the virus has began spreading, numerous papers have been published regarding different treatment options.
- In this project, we have created a sentiment model that takes in the names of popular COVID-19 treatments and returns the sentiment of different papers that are about these treatments.
- We used the VADER sentiment model to test the polarity of each paper. Each paper's sentiment is scored from -1 to +1.
- The major steps involved in text mining are gathering unstructured data from several sources that are available in various document formats, such as plain text, web pages, PDF files, etc.
- Pre-processing and data purification techniques are used to find and eliminate discrepancies in the data. The data purification step makes sure that the original text is recorded in order to prevent word stemming.
- The data gathering, processing, and controlling processes are used to review and further clean the data.
- Pattern analysis is employed by the Management Information System.

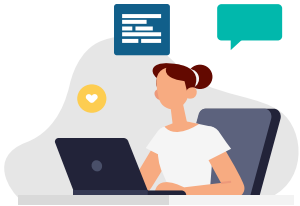
Literature Survey

1. **A Survey on Japanese population thoughts on coronavirus disease 2019 related discrimination by tweets. []**
   - Data collection and subsequent analysis was performed using R Version 4.0.2.
   - Tweets were retrieved from search query using keyword “health care providers and discrimination” and “carona and rural area”
2. **Research lines on the impact of the COVID-19 pandemic on business. A text mining analysis.[]**
   - goal in this paper was to discover current research lines established around COVID-19 and their influence on the corporate environment.
   - They employed text mining methods and statistical software 'R' to do this.
3. **Survey on Individual opinions on the COVID-19 vaccine hesitancy from twitter via Application Programming Interface. .[]**
   - 42 different models were built to determine the best model for classifying the dataset from twitter into positive, negative or neutral
4. **Text mining approaches for dealing with the rapidly expanding literature on COVID-19.[]**
   - Proposed a system that interacts with the user in the form of question-answer and filters out data based on the input.



Background

Data Cleaning - is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting or a data quality firewall.



Data Extraction - Data extraction is the process of collecting or retrieving disparate types of data from a variety of sources, many of which may be poorly organized or completely unstructured. Data extraction makes it possible to consolidate, process, and refine data so that it can be stored in a centralized location in order to be transformed. These locations may be on-site, cloud-based, or a hybrid of the two.

VADER Sentiment Analysis

VADER Sentiment Analysis - VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. Introduced in 2014, VADER text sentiment analysis uses a human-centric approach, combining qualitative analysis and empirical validation by using human raters and the wisdom of the crowd.

```
# import SentimentIntensityAnalyzer class
# from vaderSentiment.vaderSentiment module.
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# function to print sentiments
# of the sentence.
def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    print("Overall sentiment dictionary is : ", sentiment_dict)
    print("sentence was rated as ", sentiment_dict['neg']*100, "% Negative")
    print("sentence was rated as ", sentiment_dict['neu']*100, "% Neutral")
    print("sentence was rated as ", sentiment_dict['pos']*100, "% Positive")

    print("Sentence Overall Rated As", end = " ")

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        print("Positive")

    elif sentiment_dict['compound'] <= - 0.05 :
        print("Negative")

    else :
        print("Neutral")



# Driver code
if __name__ == "__main__" :

    print("\n1st statement :")
    sentence = "Geeks For Geeks is the best portal for \
# Driver code
if __name__ == "__main__" :
    print("\n1st statement :")
    sentence = "Geeks For Geeks is the best portal for \
                the computer science engineering students."

    # function calling
    sentiment_scores(sentence)

    print("\n2nd Statement :")
    sentence = "study is going on as usual"
    sentiment_scores(sentence)

    print("\n3rd Statement :")
    sentence = "I am very sad today."
    sentiment_scores(sentence)
```

Dataset

- The White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 1,000,000 scholarly articles, including over 400,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses.
- Few of the columns are:
  - **Cord_uid:** unique identification for each research article.
  - **Source_x**: Source from which the paper is brought from
  - **Abstract:** abstract of the paper
  - **Authors:** Name of the authors
  - **Title:** title of the paper
  - **Url:** Link to the research paper.

Implementation & Results

Data Preprocessing

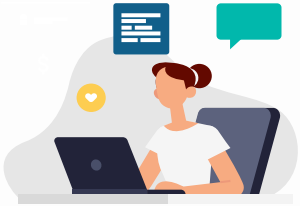
- Import the metadata.csv file
- Separated authors with a semicolon.
- Listed down synonyms of Covid 19
- Search the synonyms in abstract and title
- Created extra column in Dataframe to represent true or false if synonym is found
- If synonyms found then for that particular paper tag column is updated as true

```
COVID19_SYNONYMS = [
                    'covid',
                    'coronavirus disease 19',
                    'sars cov 2', # Note that search function replaces '-' wit
h ' '
                    '2019 ncov',
                    '2019ncov',
                    r'2019 n cov\b',
                    r'2019n cov\b',
                    'ncov 2019',
                    r'\bn cov 2019',
                    'coronavirus 2019',
                    'wuhan pneumonia',
                    'wuhan virus',
                    'wuhan coronavirus',
                    r'coronavirus 2\b'
]
```

_pmc_xml_parse	full_text_file	url	tag_disease_covid19
e	custom_license	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1...	False
e	custom_license	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1...	False
e	custom_license	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1...	False

```
# df = pd.read_csv("data/metadata.csv")
df = pd.read_csv(f'{datadir}/{metadata}', na_filter= False)
corona, covid19_counts = count_and_tag(df, COVID19_SYNONYMS, 'disease_covid19')
df
```

Code Implementation



Selected only those with tag_disease_covid19 are true.

```
corona_filtered = corona[corona.loc[:,"tag_disease_covid19"] == True]
corona_filtered
```

```
covid19_counts.sort_values(ascending=False)
```

Out[6]:

covid	2817
sars cov 2	1043
2019 ncov	641
coronavirus 2\b	275
coronavirus 2019	74
coronavirus disease 19	32
ncov 2019	13
wuhan coronavirus	12
wuhan pneumonia	7
2019ncov	7
wuhan virus	4
2019 n cov\b	3
2019n cov\b	2
\bn cov 2019	0

dtype: int64

Code implementation

```
treatment_names = [
'remdesivir',
'kaletra',
'actemra',
'kevzara',
'convalescent plasma',
'avigan',
'favilavir',
'tjm2',
'medicago',
'at-100',
'tzls-501',
'oya1',
'bpi-002',
'ino-4800',
'np-120',
'ifenprodil',
'mrna-1273',
'brilacidin',
'bcx4430',
'regn3048',
'regn3051',
'interferon-β',
'oseltamivir phosphate',
'tamiflu',
'zanamivir',
'relenza',
'peramivir',
```

Then added a list of famous Covid 19 treatments which were popular during the initial stages of outbreak.

Code implementation

Filtered the dataset with papers that mention the COVID-19 treatments

```
treatments, treatments_counts = count_and_tag(corona_filtered, treatment_names, '
treatments_covid19')
```

Again filter only the title and abstract related to covid-19

```
treatments_filtered = treatments[treatments.loc[:,"tag_treatments_covid19"] == Tr
ue]
treatments_filtered2 = treatments_filtered[treatments_filtered['abstract'].notna(
)]
# titles = treatments_filtered.loc[:,"title"]
titles = treatments_filtered2.loc[:,"abstract"]
```

Code implementation

```
sid = SentimentIntensityAnalyzer()
```

```
treatments_filtered2["scores"] = treatments_filtered2["abstract"].apply(lambda re
view: sid.polarity_scores(review))
treatments_filtered2["compound"] = treatments_filtered2["scores"].apply(lambda d:
d["compound"])
treatments_filtered2["comp_score"] = treatments_filtered2["compound"].apply(lambd
a score: 'pos' if  score >=0 else 'neg')
```

Initialized the Vader sentiment model it aims to measure the attitude, sentiments, evaluations of the subjectivity(polarity i.e, positive or negative opinion) for title and abstract

Code implementation

```
treatments_filtered2
```

	tag_disease_covid19	tag_treatments_covid19	scores	compound	comp_score
vww.ncbi.nlm.nih.gov/pmc/articles/PMC7...	True	True	{'neg': 0.047, 'neu': 0.827, 'pos': 0.126, 'co...	0.5106	pos
vww.ncbi.nlm.nih.gov/pmc/articles/PMC7...	True	True	{'neg': 0.026, 'neu': 0.915, 'pos': 0.058, 'co...	0.7037	pos
doi.org/10.1002/jmv.25707	True	True	{'neg': 0.06, 'neu': 0.847, 'pos': 0.093, 'com...	0.6124	pos
doi.org/10.1074/jbc.ac120.013056	True	True	{'neg': 0.08, 'neu': 0.844, 'pos': 0.076, 'com...	0.4517	pos
doi.org/10.3760/cma.j.cn112147-2...	True	True	{'neg': 0.068, 'neu': 0.802, 'pos': 0.13, 'com...	0.8360	pos
doi.org/10.3760/cma.j.cn501120-2...	True	True	{'neg': 0.091, 'neu': 0.864, 'pos': 0.045, 'co...	-0.6258	neg

Here the VADERS sentimentIntensityAnalyzer() takes in a string and returns in a dictionary of scores in positive or negative.

Code implementation

In [14]:

```
abstracts = treatments_filtered2.sort_values("compound", ascending = False).loc[:,"abstract"]
for ab in abstracts:
    found_words = []
    for word in treatment_names:
        if word in ab:
            found_words.append(word)
    print(found_words)
```

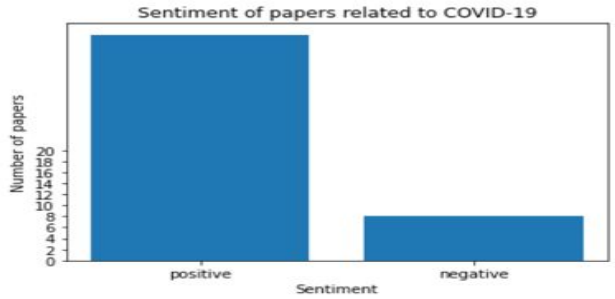
```
['convalescent plasma']
['convalescent plasma']
[]
['remdesivir']
['remdesivir']
['convalescent plasma']
['remdesivir']
['remdesivir']
['remdesivir']
[]
['remdesivir']
[]
[]
['remdesivir']
['remdesivir']
['convalescent plasma']
['remdesivir']
```

It scans all the abstracts and displays the treatments which are in the papers with the highest compound polarity score. Compound score is the sum of the positive, negative and neutral scores and then it normalises the score between +1 and -1.

Code implementation

```
bars = treatments_filtered2.loc[:,"comp_score"]
pos_height = len(treatments_filtered2[treatments_filtered2.loc[:,"comp_score"] == "pos"])
neg_height = len(treatments_filtered2[treatments_filtered2.loc[:,"comp_score"] == "neg"])
plt.bar(["positive", "negative"], [pos_height, neg_height])
# Add some text for labels, title and custom x-axis tick labels, etc.
plt.ylabel('Number of papers')
plt.yticks(np.arange(0,22, 2))
plt.xlabel('Sentiment')
plt.title('Sentiment of papers related to COVID-19')
```

```
Text(0.5, 1.0, 'Sentiment of papers related to COVID-19')
```

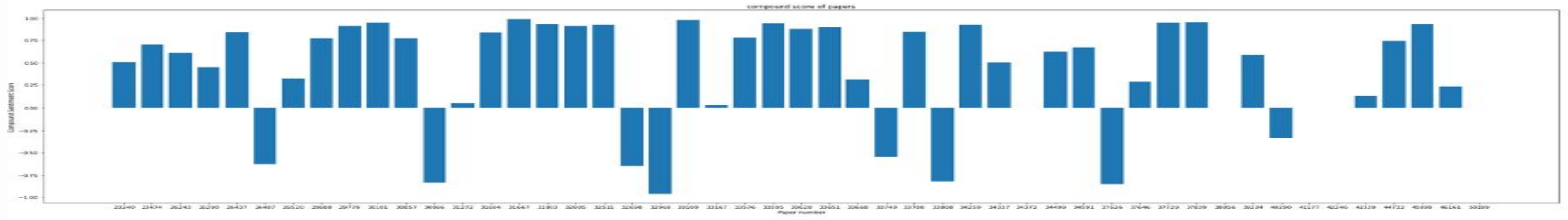


The bar graph shows the polarity of the sentiment vs number of papers.

Code implementation

```
paper_numbers = treatments_filtered2.index
x = np.array(paper_numbers)
paper_numbers_string = []
for i in x:
    paper_numbers_string.append(str(i))
scores = treatments_filtered2.loc[:,"compound"]
figure(figsize=(40,10))
plt.bar(paper_numbers_string, scores)
plt.ylabel('Compound Sentiment Score')
plt.xlabel('Paper number')
plt.title('compound score of papers')
```

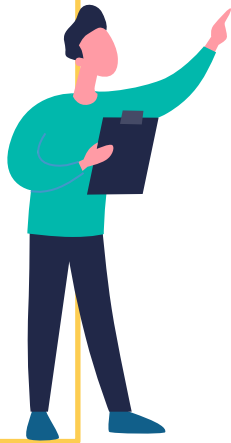
```
Text(0.5, 1.0, 'compound score of papers')
```



Compound Sentiment score vs the paper id

Conclusion and Future Work

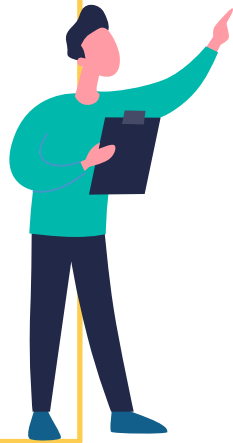
- The goal of this study is to check the validity of different Covid 19 treatments that were suggested during the initial stage of outbreak by scanning through research papers related to Covid 19
- This study was done using sentiment analysis by determining the tone of the paper and conclude whether the treatment was success or a failure
- remdesivir, convalescent plasma, interferon-β, and zanamivir had a positive sentiment.
- Our study is limited to the title and abstract only
- In future we would like to consider the entire text for further analysis by doing so it would give deeper classification to check if the treatment prescribed was efficient or not.



References

Papers Cited:
[1]. Research lines on the impact of the COVID-19 pandemic on business. A text mining analysis
[2]. Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset.
[3]. COVID-19 related discrimination in Japan
A preliminary analysis utilizing text-mining
[4]. Text mining approaches for dealing with the rapidly
expanding literature on COVID-19.
https://www.geeksforgeeks.org

- .Research lines on the impact of the COVID-19 pandemic on business. A text mining analysis Patricia Carracedo a,*, Rosa Puertas b, Luisa Marti ba Universidad Internacional de Valencia, Area ´ de empresa, c/Pintor Sorolla, 21, Valencia 46022, Spain b Universitat Polit`ecnica de Val`encia, Departamento de Economía y Ciencias Sociales, Camino de Vera, s/n 46002 Valencia Spain
- .Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset
- .COVID-19 related discrimination in Japan A preliminary analysis utilizing text-mining
- Text mining approaches for dealing with the rapidly expanding literature on COVID-19
- https://www.investopedia.com/terms/d/datamining.asp
- https://blog.hubspot.com/website/data-mining
- https://www.geeksforgeeks.org/text-mining-in-data-mining/
- https://www.simplilearn.com/what-is-text-mining-in-data-mining-article
- https://www.youtube.com/watch?v=JB8khWIKtV0



Thank You!!