Social Media Analysis using Natural Language Processing Techniques

Jyotika Singh[‡*]

Abstract—Social media is very popularly used every day with daily content viewing and/or posting that in turn influences people around this world in a variety of ways. Social media platforms, such as YouTube, have a lot of activity that goes on every day in terms of video posting, watching and commenting. While we can open the YouTube app on our phones and look at videos and what people are commenting, it only gives us a limited view as to kind of things others around us care about and what is trending amongst other consumers of our favorite topics or videos. Crawling some of this raw data and performing analysis on it using Natural Language Processing (NLP) can be tricky given the different styles of language usage by people in today's world. This effort highlights the YouTube's open Data API and how to use it in python to get the raw data, data cleaning using NLP tricks and Machine Learning in python for social media interactions, and extraction of trends and key influential factors from this data in an automated fashion. All these steps towards trend analysis are discussed and demonstrated with examples that use different open-source python tools.

Index Terms—nlp, natural language processing, social media data, youtube, named entitity recognition, ner, keyphrase extraction

Introduction

Social media has large amounts of activity every second across the globe. Analyzing text similar to text coming from a social media data source can be tricky due to the absence of writing style rules and norms. Since this kind of data entails user written text from a diverse set of locations, writing styles, languages and topics, it is difficult to normalize data cleaning, extraction, and Natural Language Processing (NLP) methods.

Social media data can be extracted using some official and open APIs. Examples of such APIs include YouTube Data API and Twitter API. One important thing to note would be to ensure one's use case fits within compliance of API guidelines. In this effort, the YouTube Data API will be discussed along with common gotchas and useful tools that can be leveraged to access data.

One can perform NLP if the text data type is available for analysis. The nature of noise seen in text from social media sources will be discussed and presented. Cleaning of the noisy text using python techniques and open-source packages will be further analyzed. Social media data additionally entails statistics of content popularity, likes, dislikes and more. Analysis on statistical and text extracted from YouTube API will be discussed and evaluated.

Finally, trend analysis will be performed using open-source python tools, social media data, statistics, NLP techniques for data cleaning and named entity recognition (NER) for a story-telling analytics piece.

* *Corresponding author: singhjyotika811@gmail.com*
‡ *ICX Media, Inc.*

Natural Language Processing

Natural language processing (NLP) is the computer manipulation of natural language. Natural language refers to language coming from a human, either written or spoken. [Wik21] defined NLP as follows: NLP is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. At one extreme, it could be as simple as counting word frequencies to compare different writing styles. [BKL09] mentions, "At the other extreme, NLP involves "understanding" complete human utterances, at least to the extent of being able to give useful responses to them. NLP is challenging because Natural language is messy. There are few rules and human language is ever evolving.".

Some of the common NLP tasks on text data include the following.

1) Named entity recognition

   Named-entity recognition (NER) (also known as (named) entity identification, entity chunking, and entity extraction) is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into predefined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. Some popular Python libraries that can be leveraged to perform named entity recognition for a variety of different entities include SpaCy [HMVLB20] and NLTK [BKL09].

2) Keyphrase extraction

   Keyphrase extraction is the task of automatically selecting a small set of phrases that best describe a given free text document. [BSMH+18] Some popular tools that can be used for keyphrase extraction as

mentioned in this article[1] include Gensim [RS11] and RAKE-NLTK[#]_. Another way keyphrase extraction can be performed is using NLTK [BKL09] methods. This implementation is included in the pyYouTubeAnalysis [Sin21] library.

3) Unigrams/Bigrams/Trigrams analysis

Breaking down text into single words, a pair of consecutive written words or three consecutively written words and analyzing occurrence patterns.

4) Custom classifier building (public dataset -> features -> ML models)

If out-of-box solutions do not exist for one's NLP task, building custom models to help solve for the problem is an option with the help of available data, NLP libraries (such as NLTK[2], SpaCy[3], and gensim[4]), and Machine Learning libraries (scikit-learn[5]).

5) Others

Tokenization, Part-of-speech tagging, Lemmatization & Stemming, Word Sense Disambiguation, Topic modeling, Sentiment Analysis and Text summarization are some other popularly used NLP tasks. This list is not all inclusive.

A human can only see N number of text samples a day to learn, whereas a machine can analyze a lot greater than N. Leveraging machines for NLP tasks along with several processing solutions available with Python, such as multiprocessing[6], can help analyze large amounts of data in a reasonable time-frame.

Potential use cases include the following.

1) Analytics, intelligence and trends

Analyzing patterns in text based on word occurrences, language, combining text occurrences with other available data, topics, sentiment information, NLP method outputs, or combinations thereof.

2) Story telling

Analyzing text using the various NLP techniques along with other statistical and other available data aids in converting raw data to an informative story piece that helps uncover and understand the patterns that exist within the data. Depending on the data available, a time-window analysis can help study patterns as they change with respect to time in terms of word usages, topics, text lengths, or combinations thereof.

1. https://towardsdatascience.com/extracting-keyphrases-from-text-rake-and-gensim-in-python-eefd0fad582f
2. https://pypi.org/project/rake-nltk/
3. https://scikit-learn.org/
4. https://www.nltk.org/
5. https://spacy.io/
6. https://radimrehurek.com/gensim/
7. https://scikit-learn.org/
8. https://docs.python.org/3/library/multiprocessing.html

Social Media APIs

There are several social media platforms that let you programmatically collect publicly available data and/or your own published data via APIs. Whatever you intend to do with this data, it is important to ensure that you use the data in compliance with the API's guidelines and terms and services.

Some types of available requests on YouTube include search, video, channel and comments.

YouTube Data API documentation[7] is a great resource to learn more and get started. At a high level, the getting started[8] steps include registering a project, enabling the project and using the API key generated. With this key, the user can start making requests to the API to crawl data.

Gotchas

There are a few items to keep in mind when using the YouTube Data API. Some of the gotchas while using the api include the following.

1) Rate limits

The API key registered to you comes with a daily quota. The quota-spend depends on the kind of requests you make. API does not warn you in API request response if you are about to finish your daily quota but does throw that error once you have exceeded the daily quota. It is important to know how your application will behave if you hit the quota to avoid unexpected behavior and premature script termination.

2) Error handling

If trying to query for a video, comment or channel that is set to private by the owner, the API throws an error. Your code could end prematurely if you are querying in a loop and one or a few of the requests have that issue. Error handling could help automate one's process better on such expected errors.

Interacting with the YouTube Data API

There are several ways to interact with the YouTube Data API. Some of them are as follows.

1) Use the API web explorer's "Try this API" section[9]
2) Build your own code using API documentation examples[10]
3) Open-source tools

1. Wrappers of YouTube Data API[11] : Libraries that act as wrappers and provide a way to use YouTube Data API V3.

2. pyYouTubeAnalysis :cite *pyYouTubeAnalysis*[12] : This library allows the user to run searches, collect videos and comments, and define search params (search keywords, timeframe, and type). Furthermore, the project includes error handling that allows code execution to continue and not stop due to unforeseen errors while interacting with YouTube data API. Additional features included in pyYouTubeAnalysis are NLP methods for social media text pre-processing mentioned in a later section *Data Cleaning Techniques*, NLTK

9. https://developers.google.com/youtube/v3/docs
10. https://developers.google.com/youtube/v3/getting-started

based keyphrase extraction and SpaCy based Named Entity Recognition (NER) that runs entity extraction on text.

Social Media / YouTube Data Noise

Text fields are available within several places on YouTube, including video title, description, tags, comments, channel title and channel description. Video title, description, tags, and channel title and description are filled by the content/channel owner. Comments on the other hand are made by individuals reacting to a video using words and language.

The challenges in such a data source arise due to writing style diversity, language diversity and topic diversity. Figure 1 shows a few examples of language diversity. On social media, people use abbreviations, and sometimes these abbreviations may not be the most popular ones. Other than the non-traditional abbreviation usage, different languages, different text lengths, and emojis used by commenters are observed.

Data Cleaning Techniques

Based on some noise seen on YouTube and other social media platforms, the following data cleaning techniques have been found to be helpful cleaning methods.

1) Removing URLs

Social media text data comes with a lot of URLs. Depending on the task at hand, removing the urls have been observed to come in handy for cleaning the text. Remove the URLs prior to passing text through keyphrase or NER extractions has been found to return cleaner results. This implementation is also contained in pyYouTubeAnalysis.

```
import re

URL_PATTERN = re.compile(
    r"https?://\S+|www\.\S+",
    re.X
)

def remove_urls(txt):
    """
    Remove urls from input text
    """
    clean_txt = URL_PATTERN.sub(" ", txt)
    return clean_txt
```

2) Removing emojis

Emojis are widely used across social media by users to express emotions. Emoijis provide benefit in some NLP tasks, such as certain sentiment analysis implementations that rely on emoji based detections. On the contrary, for many other NLP tasks, removing emojis from text can be a useful cleaning method that improves the quality of the processed outcome. For named-entity recognition and keyphrase extraction, certain emojis are observed getting falsely detected as locations or nouns of the type NN or NNP. This impacts the quality of the NLP methods. Removing the emojis prior to passing such text through named-entity recognition or keyphrase extractions has been found to return cleaner results. This implementation is also contained in pyYouTubeAnalysis.

```
import re

EMOJI_PATTERN = re.compile(
    "[\U00010000-\U0010ffff]",
    flags=re.UNICODE
)

def remove_emojis(txt):
    """
    Remove emojis from input text
    """
    clean_txt = EMOJI_PATTERN.sub(" ", txt)
    return clean_txt
```

3) Spelling / typo corrections

Some NLP models tend to do very well for a particular style of language and word usage. On social media, the language seen can be accompanied with various incorrectly spelled words, also known as typos. PySpellChecker [LT16][13], Autocorrect[14] and Textblob [Lor18] are examples of open-source tools that can be used for spelling corrections.

4) Language detection and translations

Developing NLP methods on different languages is a challenging and popular problem. Often when one has developed NLP methods for english language text, detection of a foreign language and translation to english serves as a good solution and allows one to keep their NLP methods fixed. Such tasks introduce other challenges such as the quality of language detection and translation. Nonetheless, detection and translation is a popular technique while dealing with multiple different languages. Some examples of Python libraries that can be used for language detection include langdetect [Shu10], Pycld2[15], Textblob [Lor18], and Googletrans[16]. Translate[17] and Googletrans can be used for language translations.

Trend Analysis Case Study

In the year 2020, COVID hit us all hard. The world went through a lot of changes in the matter of no time to reduce the spread of the virus. One such impact was observed massively in the travel and hospitality industry. Figure 2[18] shows the flight search trends between February and November 2020 for domestic and international flight searches from the US using Kayak. Right before lockdown and restrictions were enforced starting in March across different places across the globe, a big spike can be seen in flight searches, correlating with the activity of people trying to fly back home if they were elsewhere before restrictions disabled them to do so.

11. https://developers.google.com/youtube/v3/docs/search/list
12. https://developers.google.com/youtube/v3/quickstart/python
13. https://github.com/rohitkhatri/youtube-python, https://github.com/sns-sdks/python-youtube
14. https://github.com/jsingh811/pyYouTubeAnalysis
15. https://pypi.org/project/pyspellchecker/
16. https://pypi.org/project/autocorrect/
17. https://pypi.org/project/pycld2/
18. https://pypi.org/project/googletrans/
19. https://pypi.org/project/translate/

Fig. 1: *Random sample of YouTube comments representing writing style diversity.*

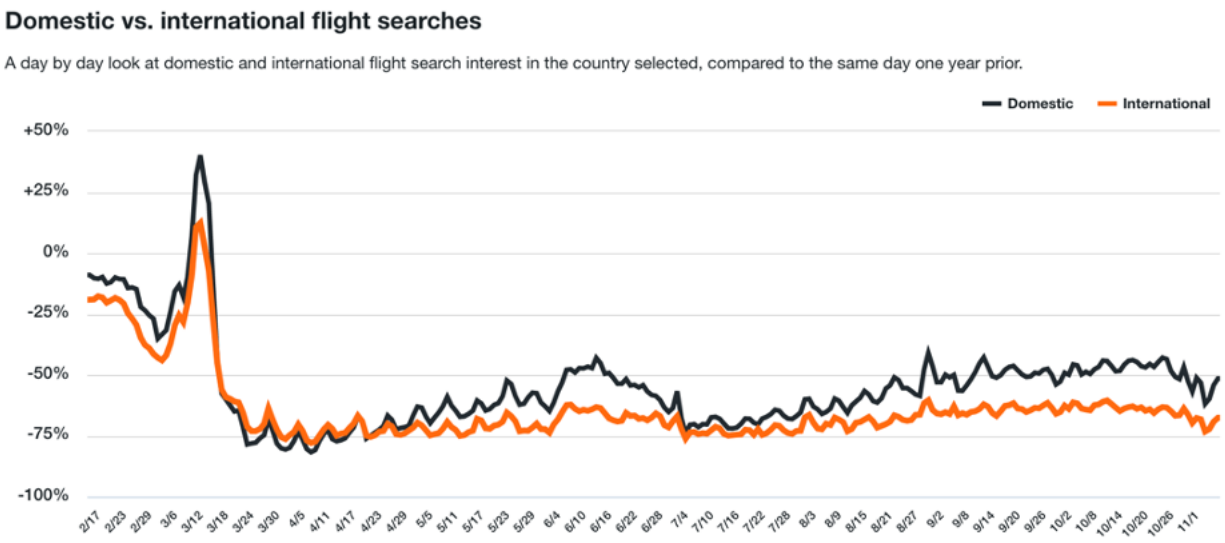

Fig. 2: *Domestic and international flight search patterns in 2020.*

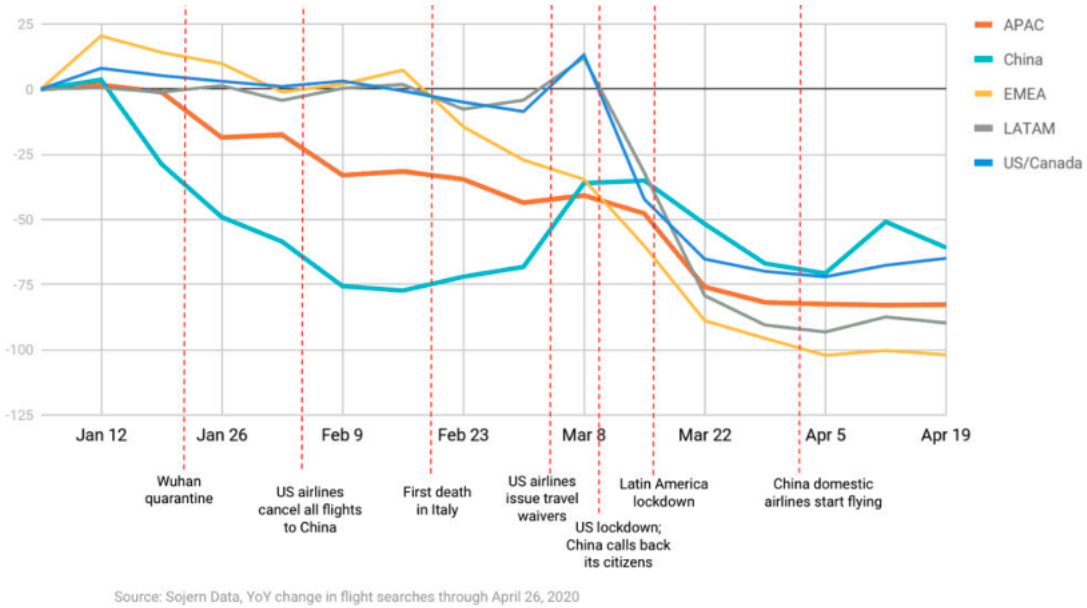

Fig. 3: *Global flight search patterns in 2020.*

A massive reduction in flight searches can further be seen in figure 3[19] showing the impact at a global level. Timeline beyond January of 2020 for China, and beyond March of 2020 for most other locations, faced the most impact as travel was reduced due to COVID imposed events and restrictions.

Aligning with reduced flight searches, reduced hotel search were also reported from March onwards as can be seen in figure 4[20].

Let's try to correlate these findings and understand content consumption within those time periods on YouTube.

First, a search was performed to gather videos about "travel vlogs" using the pyYouTubeAnalysis library. Travel vlogs are a popular content genre on YouTube where a lot of people are able to find reviews, advice and sneak peaks of different destinations that wows them and inspires travel plans. Such videos typically consist of people traveling to different locations and recording themselves at different spots.

Statistically, it can be seen from figures 5, 6 and 7 that travel vlog has been a growing topic of interest and has been growing along with online content consumption over the years up till 2019. A downward trend was seen in average views, comments, and likes on travel vlog videos in 2020, where the views went down by 50% compared to the year before.

To understand the differences between the travel vlog content consumed in 2019 versus 2020 in further detail, a monthly data crawl was performed. Figures 8, 9 and 10 show a month over month comparison between 2019 and 2020 to analyze average audience engagement patterns. The viewership trends reflect the reduction from March onwards when COVID hit most locations across the globe. Figure 11 further shows engagement shift between 2019 and 2020. The trend slopes upwards until March hits, which is when a lot of locations imposed stay at home orders and lockdowns. The trend slopes downwards, picks up a little July onwards, which correlates with the time Europe lifted a lot of the travel restrictions. The chart representing "travel vlog" content engagement largely correlates with the flight search trend as shown in figure 2. It can be seen however, people were still creating travel vlogs and commenting on such videos. Between June and September 2020, amidst a much-reduced travel, what were these videos, what content was getting created, who was creating it, and what were the commenters talking about?

Figure 12 shows a word cloud representation of what these videos talked about generated using keyphrase extraction implementation in pyYouTubeAnalysis, where the text passes through data cleaning techniques prior to keyphrase extraction that is inbuilt within the implementation. Application of these techniques prior to extracting keyphrases eliminated the noisy samples and improved the overall results quality. Additionally, wordcloud [OMIB11][21] was used for creating the visualization. Word cloud is a form of term occurrence visualization where the size of the appearance of a term in the word cloud is directly proportional to its occurrence count. Travel that would entail easier implementation of social distance was seen popping up in 2020, such as hiking, beach trips and road traveling. Location names such as Italy, France and Spain were also seen showing up in the videos.

While we have seen what content gained the most engagement, let's look into who the creators of such content were that drove the most comments and engagement. With the help of engagement statistics and videos read for the 2020 time frame, the YouTube influencer channels that drove high engagement during summer and fall of 2020 include the following.

1) 4K Walk[22] – YouTube channel creating videos about walking tours all over Europe and America.
2) BeachTuber[23] – YouTube channel creating vlogs from different beaches all over Europe.
3) Beach Walk[24] – YouTube channel posting about different beaches all over Europe and America.
4) DesiGirl Traveller[25] – YouTube channel creating videos about India travel.
5) Euro Trotter[26] – YouTube channel creating videos about Europe travel.

A few examples of comments that were being left by audiences of such videos are as follows.

> "i'm going to sorrento in 10 days and i'm so excited. i've been watching tonnes of sorrento and italy vlogs and yours are so lush X) <3"
>
> "Did they require you to have a prior covid test?"
>
> "I loved the tour looked like you guys had fun. im going there next week, how long ago were you there and were there lots of restrictions and closing due to covid"
>
> "Great video man, this place looks amazing. I have never been to Iceland, would love to visit some day. Honestly can't wait for the lockdown to be lifted so I can start travelling again. Thanks for sharing your experience. :)"

It was seen that people expressed interest in inquiring about the lifting of the travel ban due to COVID, pre-travel COVID test requirements, along with the sentiments around being able to travel again. People were seen mentioning a lot of location names in their comments. With the help of named-entity recognition implementation in pyYouTubeAnalysis, location extractions were performed. The underlying process passed the comments through URLs and emojis removal prior to location extraction, which led to cleaner results and reduced manual filtering. Figure 13 shows the location popularly mentioned by commenters in a word cloud representation. One can see European locations, along with some Asian and American locations which correlate with travel restriction reductions in some of the places.

This analysis, including data collection from social media, keyphrase extraction, and NER, was performed using pyYouTube-Analysis library [Sin21][27]. Similar analysis for content other than "travel vlogs" can be performed for custom time windows using similar tools and the other NLP libraries mentioned in this effort.

Conclusion

User content creations and interactions via text on social media platforms contain mixed writing styles, topics, languages, typing

20. https://www.kayak.com/news/category/travel-trends/
21. https://www.sojern.com/blog/covid-19-insights-on-travel-impact-hotel-agency/
22. https://www.sojern.com/blog/covid-19-insights-on-travel-impact-hotel-agency/
23. https://pypi.org/project/wordcloud/, https://www.wordclouds.com/
24. https://youtube.com/c/4KWALK
25. https://youtube.com/c/BeachTuber
26. https://youtube.com/c/BeachWalk
27. https://youtube.com/c/DesiGirlTraveller
28. https://youtube.com/c/EuroTrotter
29. https://github.com/jsingh811/pyYouTubeAnalysis

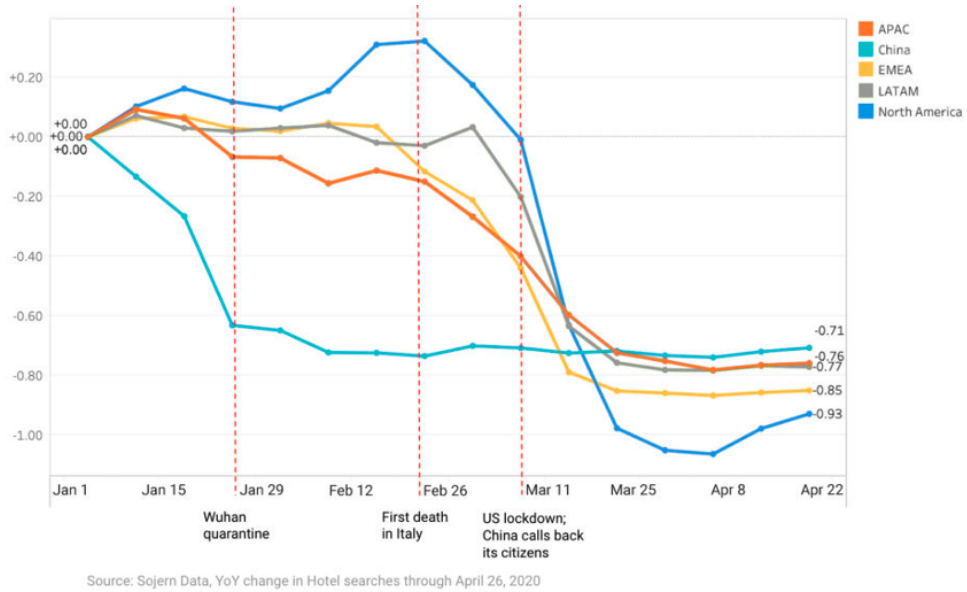

Fig. 4: *Hotel booking search patterns in 2020.*

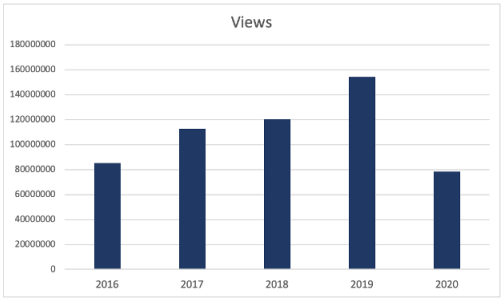

Fig. 5: *Yearly video views.*

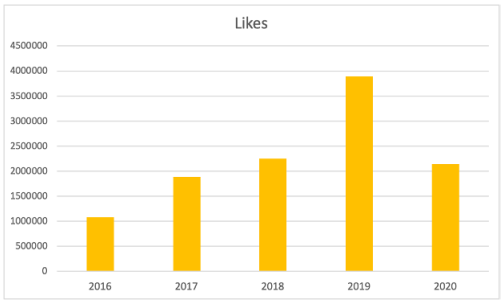

Fig. 6: *Yearly video likes.*

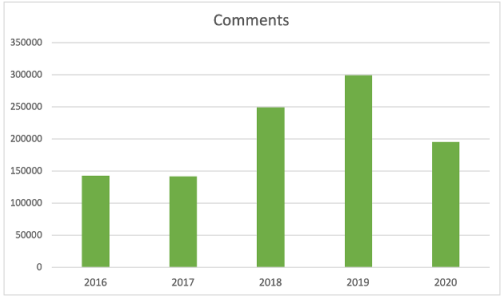

Fig. 7: *Yearly video comments.*

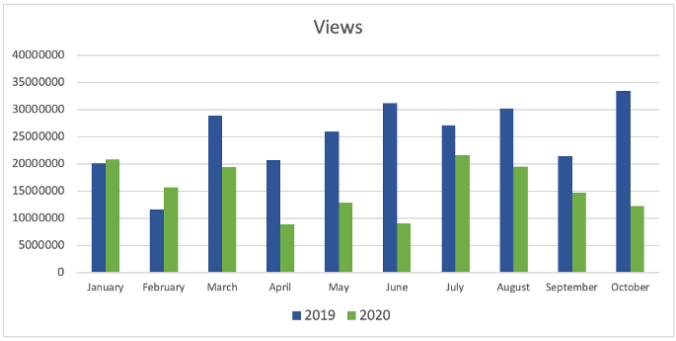

Fig. 8: *Monthly video views for 2019 and 2020.*

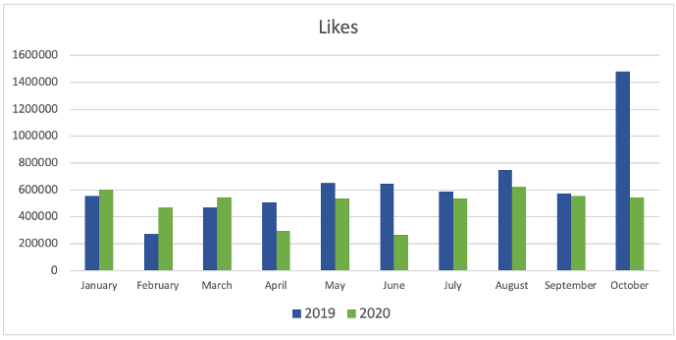

Fig. 9: *Monthly video likes for 2019 and 2020.*

errors, freeform emojis and abbreviations. This diversity of content and language makes it harder to perform NLP tasks on data coming from social media. Described cleaning techniques such as emoji removal, hyperlink removal, language detection and translations, and typo corrections have been found useful in priming and pre-processing language of such nature. Subjecting the text through these methods prior to other Natural Language Processing (NLP) methods such as keyphrase extraction and named-entity recognition result in cleaner output.

Social media data contain statistics in addition to text data

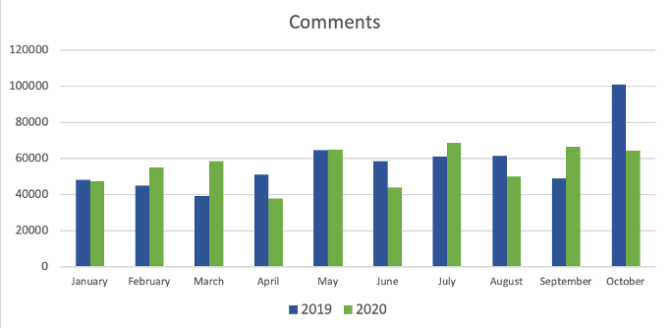

Fig. 10: *Monthly video comments for 2019 and 2020.*

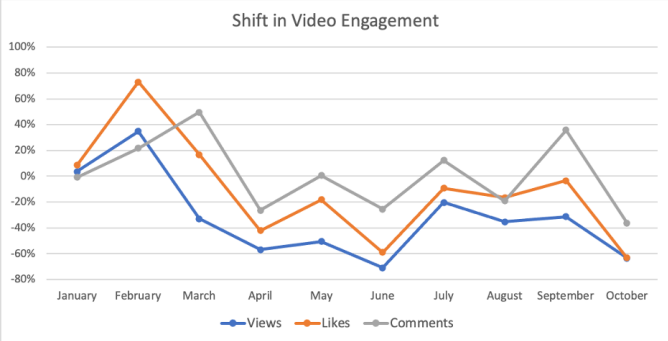

Fig. 11: *Difference in video engagements between 2019 and 2020.*

Fig. 12: *Word cloud of video topics.*

Fig. 13: *Word cloud of location names used in comments.*

that measures human engagement and interest in different types of content. Combining these statistics with inferences from NLP techniques such as named-entity recognition (NER) and keyphrase extraction are found to be helpful in trend analysis, analytics, and observing correlations and affinities of user engagement with social media.

REFERENCES

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 01 2009.

[BSMH+18] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. 10 2018. doi:10.18653/v1/K18-1022.

[HMVLB20] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL: https://doi.org/10.5281/zenodo.1212303, doi:10.5281/zenodo.1212303.

[Lor18] Steven Loria. textblob documentation. *Release 0.15*, 2, 2018.

[LT16] Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA), 2016. URL: http://www.lrec-conf.org/proceedings/lrec2016/summaries/947.html.

[OMIB11] Layla Oesper, Daniele Merico, Ruth Isserlin, and Gary Bader. Wordcloud: A cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, 6:7, 04 2011. doi:10.1186/1751-0473-6-7.

[RS11] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[Shu10] Nakatani Shuyo. Language detection library for java. 2010. URL: http://code.google.com/p/language-detection/.

[Sin21] Jyotika Singh. jsingh811/pyyoutubeanalysis: Youtube data requests and natural language processing on text, 2021. URL: https://zenodo.org/record/5044556, doi:10.5281/ZENODO.5044556.

[Wik21] Wikipedia contributors. Natural language processing — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Natural_language_processing&oldid=1030186679, 2021. [Online; accessed 25-June-2021].