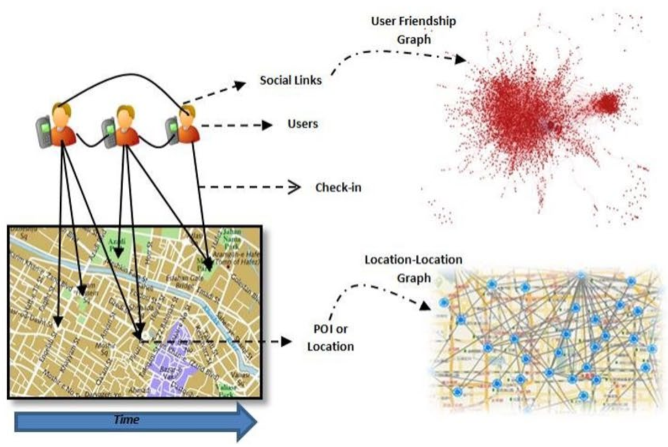
Micro-module 2: Location-based Social Network (LBSN) and Text Mining

We are in an era of ever-increasing user-generated content, patterns of human activity can br revealed from online social media platforms. Obtaining meaningful information from these sources represents both a challenge and an opportunity. Specifically, digital data sources provide researchers with a new approach to the study of urban phenomena. To this content, analyzing the city through data retrieved from Location Based Social Networks (LBSNs) has received considerable attention as a promising method for applied research.
This module will cover the basic concepts and examples of LBSN data and its application in urban analytics, and it will provide a step-by-step tutorial on sentiment analysis, using social media posts as an example.

1. Location Based Social Network (LBSN) Data in Urban Analytics

- LBSN Data

Location Based Social Network (LBSN) Data



There are several definitions for “geosocial network” or “location-based social network”: the first formal definition was given by Quercia et al. in 2010, who defined it as “a type of social networking in which geographic services and capabilities such as geocoding and geotagging are used to enable additional social dynamics”. One year later, Zheng refined this definition by stating that “a location-based social network (LBSN) does not only mean adding a location to an existing social network so that people in the social structure can share location embedded information but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and texts”.
Nowadays, people are exposed to different social media platform, that generate different types of LBSN data, for instance, google map and foursquares are more based on location sharing, we can found people’s reviews and comments on different location/point of interests on it. Another set of popular social media platform can be represented by x (formally known as twitter), Facebook, and isograms. These are more opinion-based platform, people can share their opinions by texts and photos at varied location. Data from Airbnb represent urban economy vibrancy and the accessibility to public services to some extent.

- **Google Place**

  Information on Google Places listings —classified by category and sub-category— reveals clusters of economic activities as well as quantity, diversity, and complexity in the spatial distribution of these activities and places of interest. The regrouping of categories into much fewer and more general categories is helpful not only for making easier reading and interpretation of cartographies. For studying specialization of economic activities. For instance, previous experiences have proven that the recategorization of places into the Land Based Classification Standards, enables the identification of location patterns and spatial distribution of economic activities at different scales and granularity.

  Here we can see google map platform provides a set of google places, for each place, we can check their population time, and people's attitudes to this place are collected by reviews.

- **Social Media Data**

  We'd like to give another example more on location-based text mining and analysis. Nowadays, people are highly involved in social media, and they are motivated to share their emotions and thoughts online, leaving a large and continuously updated user generated content. Studying sentiment level from users' posts may eliminate the social desirability effect that traditional self reports bring, due to participants' inaccurate and dishonest evaluation of emotional bonding. Hence, Twitter data has been widely used by researchers and accessed through Twitter's public API.

  Unfortunately, now x has closed their free Api, if you are going to use the data from x, you should subscribe their advanced plan or use some scrapers.

The recognition of certain activities, opinions, ideas and trending topics that are predominant in a given place and at a given time, and it can be detected by using the information related to the tweet content —text, hashtags— and sentiment analysis. Also the collected public perceptions can be used to evaluate the urban facilities and policies, and activated evidence based urban intervention.

Public sentiment provides an important social reference for urban management and planning. we can do research on the correlation of the entire urban environment with public sentiment.

Location-based social network (LBSN) data, such as that of Twitter or Facebook, can overcome the limitation and provide a tangible vision to present “invisible” public sentiments in nearly real-time. The common feeling can be observed and aggregated through texts, emoticons and specific behavior (e.g. by giving a “like” or forwarding). Therefore, sentiment analysis via location-based social network (LBSN) data has been a popular topic in urban studies on subjects such as work stress, the sentiment of railway passengers.

- **Application of Street View Images in Urban Analytics**

The recognition of certain activities, opinions, ideas and trending topics that are predominant in a given place and at a given time, and it can be detected by using the information related to the tweet content —text, hashtags— and sentiment analysis. Also the collected public perceptions can be used to evaluate the urban facilities and policies, and activated evidence based urban intervention.

Public sentiment provides an important social reference for urban management and planning. we can do research on the correlation of the entire urban environment with public sentiment.

Location-based social network (LBSN) data, such as that of Twitter or Facebook, can overcome the limitation and provide a tangible vision to present “invisible” public sentiments in nearly real-time. The common feeling can be observed and aggregated through texts, emoticons and specific behavior (e.g. by giving a “like” or forwarding). Therefore, sentiment analysis via location-based social network (LBSN) data has been a popular topic in urban studies on subjects such as work stress, the sentiment of railway passengers.

2. Natural Language Processing and Text Mining

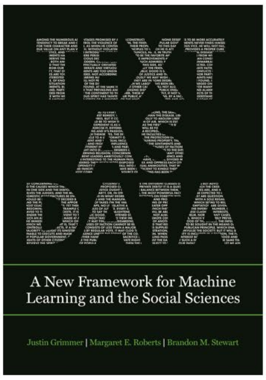
- **Natural Language Processing**

Natural language processing is an artificial intelligence application.

> *"NLP combines computational linguistics – rule-based modeling of human language – with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment."*
>
> –IBM
>
> https://www.ibm.com/topics/natural-language-processing

NLP applications have become commonplace in daily life. The voice assistant function on your smartphone or home assistant is an example of natural language processing. NLP is used in language translation apps, digital transcription services, and autofill or predictive text functions. The very popular example, is chatgpt, is a very successful application of nlp.

- **Text as Data**



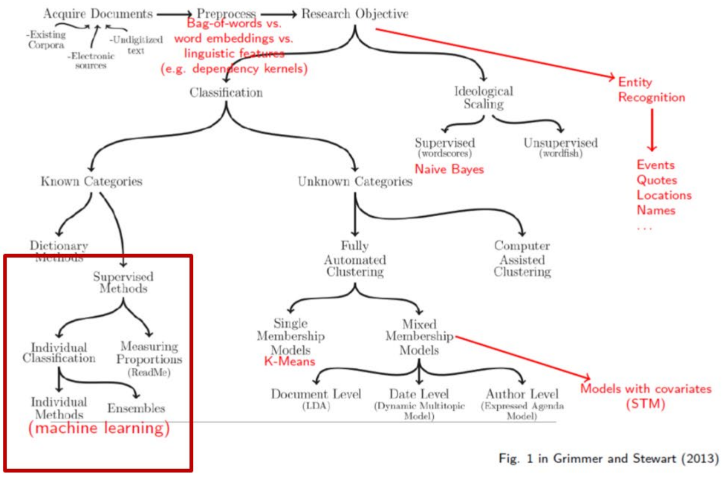


Fig. 1 in Grimmer and Stewart (2013)

In our case, we focus on text data, instead of voice data. we borrow the flow chart for the famous book called ' text as data' to give you an overall understanding of the framework of text mining when we deal with different task of research objectives. First, the input is the acquire document, it can be text archives, electronic sources, and in our case, we will deal with text from the content of social media posts.

After we get the text data, we should convert it into different types are suitable for different algorithms, for instance, bag of words, or word embedding. (we will introduce more on these terminologies in the later session). After that, the key issue is to be clear about the research objectives, the main purpose is the do classification for different types of text. under the task of classification, again, it can be divided into known categories and unknown categories.

We can show you two examples, for instance, if your research purpose is to categorize different post by different emotions we use positive, negative, and neutral as types for categorization. This is a very typical know categories task. we are sure about the specific post will be aligned to positive or negative.

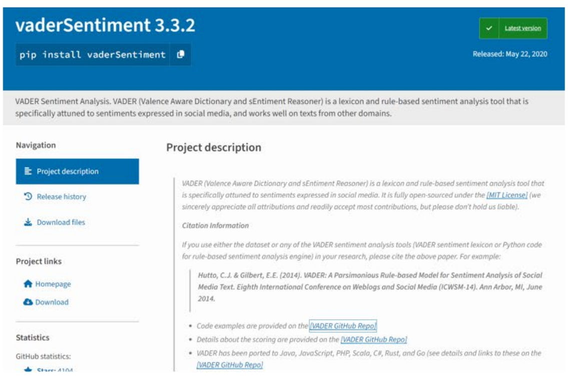
Under this dimension, we can use different method to deal with it. dictionary method is more straight forward, it based on sepecific rules to categorize content. for instance, if this post have more than 50 proportion of words are pre-defined as positive words, it will be assigned as a postive post. Another method is to use supervised machine learning, for instance, we manually label 1o percent of the whole dataset, and munnual label them by positive and negative, and the machine learning algorithms will extract latent variables and predict the sentiment score for the rest posts in the dataset. So that's how supervised method works. we will give you more details in the afternoon session.

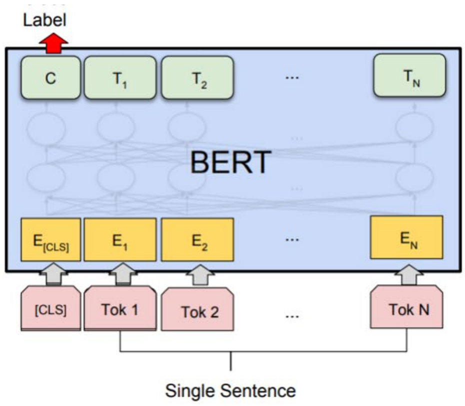
Apart from that, what's unknown categories? You can imagine that if you are going to extract several topics for a set of content or speeches. You will have no idea what topic will be generated. So that is unknown categories. Two set of very commonly used tasks are included in this dimension, one is clustering, another one is topic modeling, In this workshop, we will not touch upon these two tasks. If you are interested in them, you can search more on this book the text as data.

- **Sentiment Analysis**

  Sentiment Analysis is a type of text analytics that examines a body of text and assesses its overall sentiment (the author's attitude or emotional state) or its overall opinion (what the author believes about something). As we mentioned, it is under the task of categorized text data with known categories. And it can be dealt with by two methods. One is lexicon-based method, so co dictionary-based method. The lexicon-based method contains a list of words that are predefined in terms of sentiment polarity and sentiment strength. It requires high quality of lexical resources for good performance. Based on dictionaries, the machine sentiment classifier searches for the matched emotion expression in texts and assigns sentiment scores to them. They applied heavy human inspection in the production of the lexicon to improve the quality. the algorithm of the lexicon-based method is easier to understand and adopt. Here, we will use a commonly used package, called Vader.

  Another method is supervised machine learning, here we use the state-of-art example, BERT model





- **VADER**

  VADER means Valence Aware Dictionary and Sentiment Reasoner. It's a lexicon and rule-based feeling analysis instrument. VADER utilizes a mix of lexical highlights (e.g., words) that are, for the most part, marked by their semantic direction as one or the other positive or negative. Thus, VADER not only tells about the Polarity score yet, in addition, it tells us concerning how positive or negative a conclusion is.

In this methodology, every word in the vocabulary is appraised with respect to whether it is positive or negative, and, how +ve or -ve. Here you can see an extract from VADER's vocabulary, where more positive words have higher positive evaluations and more adverse words have lower negative grades.


VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

C.J. Hutto Eric Gilbert

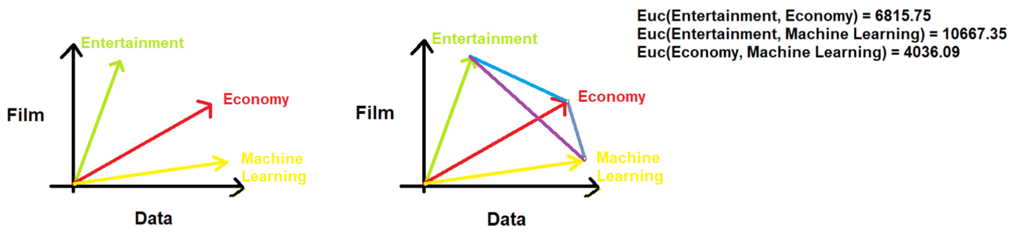
Word	Sentiment rating
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

- **Vector Space Model and Word Embedding**

  The most common disadvantage of the lexicon based model is that they can not reflect the in-text content, the content of the sentence actually is very complex. Here we show two example, you can see rely on the vocabulary of single words may miss the inter-word relationship, which is important to understand human language for instance to to predict emotions.

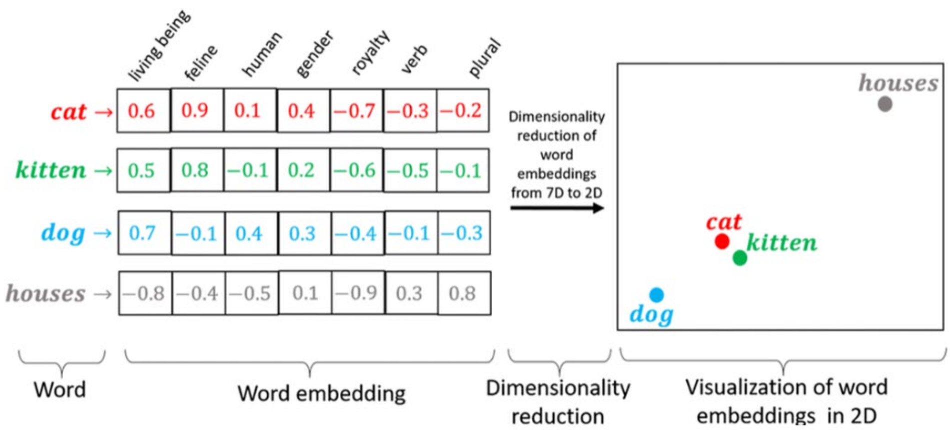
Vector space models are algebraic models that are often used to represent text (although they can represent any object) as a vector of identifiers. With these models, we are able to identify whether various texts are similar in meaning, regardless of whether they share the same words. Vector space model has a wide range of applications in NLP(Natural Language Processing).

for instance, here is an example, word can be converted to vectors. and represented by their spatial position and distance.



Word embeddings are vector representations of words that capture their semantic meaning. They are used in natural language processing tasks to represent words as numerical values that can be fed into machine learning algorithms. BERT generates word embeddings by taking into account the context in which a word appears, making its embeddings more accurate and useful than traditional methods such as bag-of-words or TF-IDF.

through word embedding and dimensional reduction, a set of words can be plot on a 2d form.

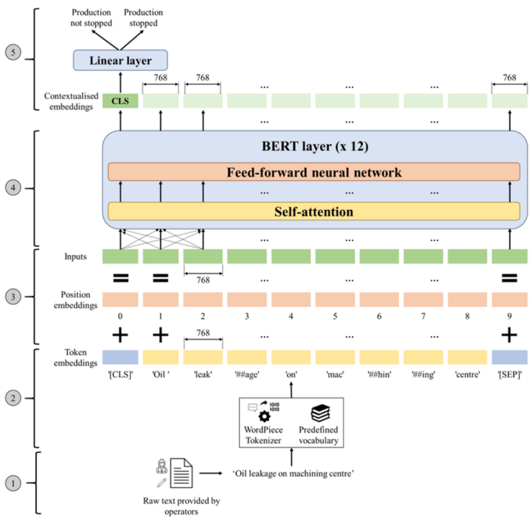


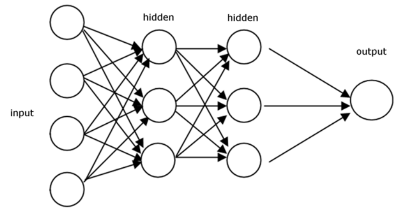
- **BERT Model**

  BERT, or Bidirectional Encoder Representations from Transformers, is a powerful language model developed by Google. It has been widely used in natural language processing tasks such as sentiment analysis, text classification, and named entity recognition. One of the key features of BERT is its ability to generate word embeddings, which are numerical representations of words that capture their meaning and relationships with other words.

  As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it’s non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

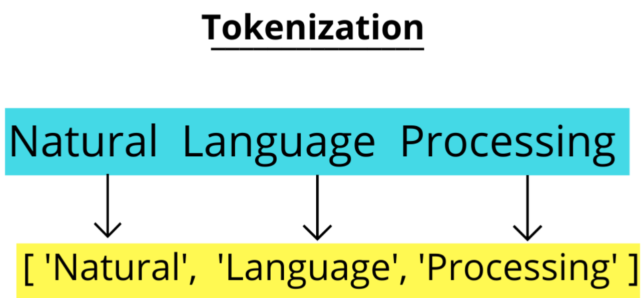
  the input variable of Bert model is the result after dimensional reduction of word-vectors. and it included a 12-layer feed-forward neural network to generate hidden latent variables, and generate the final predictable variables for different word embedding.





- **Data Preprocessing**

  Tokenization



Also called word segmentation, tokenization is one of the simplest and most important techniques involved in NLP.

It's a crucial preprocessing step in which a long string of text is broken down into smaller units called tokens. Tokens include words, characters, and subwords. They are the building blocks of natural language processing, and most NLP models process raw text on the token level.

Lowercase

Raw	Lowercased
Canada CanadA CANADA	canada
TOMCAT Tomcat toMcat	tomcat

Stemming, Lemmatization

Stemming is a natural language processing (NLP) technique that involves reducing words to their base or root form, known as the "stem." The stem is a shorter representation of a word that encompasses its core meaning. Stemming aims to remove prefixes, suffixes, and inflections from words, allowing variations of a word to be treated as the same root word. This process helps in reducing vocabulary size, normalizing text, and improving text analysis tasks such as information retrieval, text classification, and sentiment analysis. Stemming algorithms apply a set of rules or heuristics to identify and remove affixes, resulting in the stem form of a word. However, stemming does not always produce valid or meaningful words, as it focuses on linguistic reduction rather than semantic accuracy.

Please note that different stemming algorithms may produce different stems for the same word, and some stems may not be actual valid words in the language. Stemming aims to reduce words to their core form for analysis purposes, sacrificing semantic accuracy for linguistic normalization.

Lemmatization is a natural language processing (NLP) technique that involves reducing words to their base or dictionary form, known as the "lemma." Unlike stemming, which simply removes affixes from words to derive a root form, lemmatization takes into account the word's part of speech and context to determine its canonical form. The resulting lemma represents the base meaning of the word and is a valid word found in a dictionary. Lemmatization aims to normalize words while preserving their semantic integrity, making it a more linguistically accurate approach compared to stemming. By transforming words into their lemmas, lemmatization helps improve text analysis, information retrieval, and language understanding tasks in NLP.

In lemmatization, the resulting lemma represents the canonical or base form of a word, considering its part of speech and context. Lemmatization provides linguistically valid and meaningful lemmas, which can enhance the accuracy of text analysis and language processing tasks.

Stop words removal

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Stop words removal is another preprocessing step of NLP that removes filler words to allow the AI to focus on words that hold meaning. This includes conjunctions such as "and" and "because," as well as prepositions such as "under" and "in."

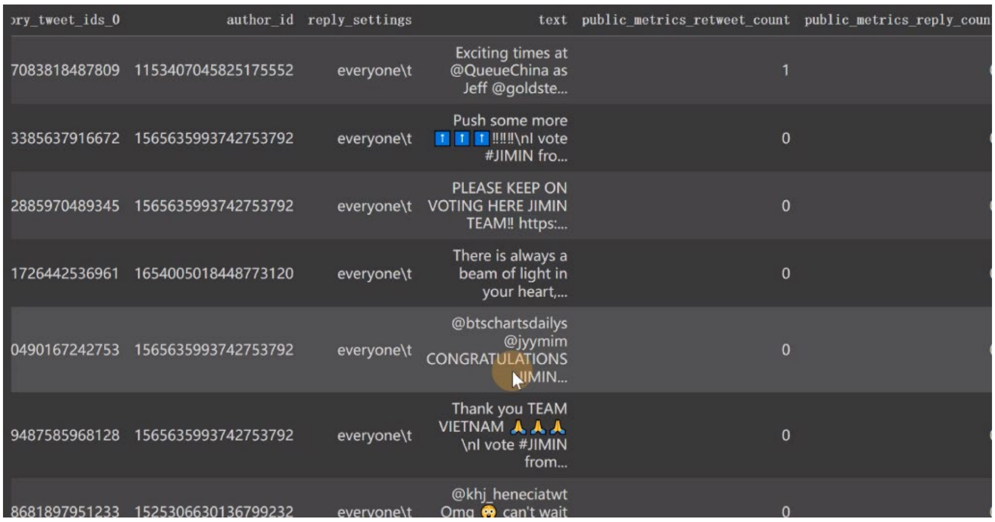
By removing these unhelpful words, NLP systems are left with less data to process, allowing them to work more efficiently. It isn't a necessary step of every NLP use case, but it can help with things such as text classification.

3. Sentiment Analysis

- **Data cleaning**

First, we should import several useful packages for our task.

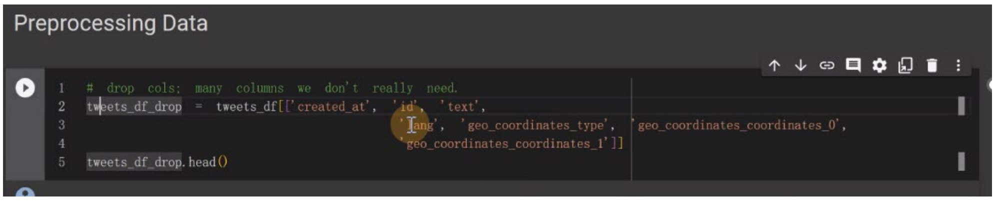
After that, you can upload the sample file to the default folder in google colab. Noting that the uploaded file will not be stored permanently. You can choose to mount your own google drive paths and upload file through google drive.



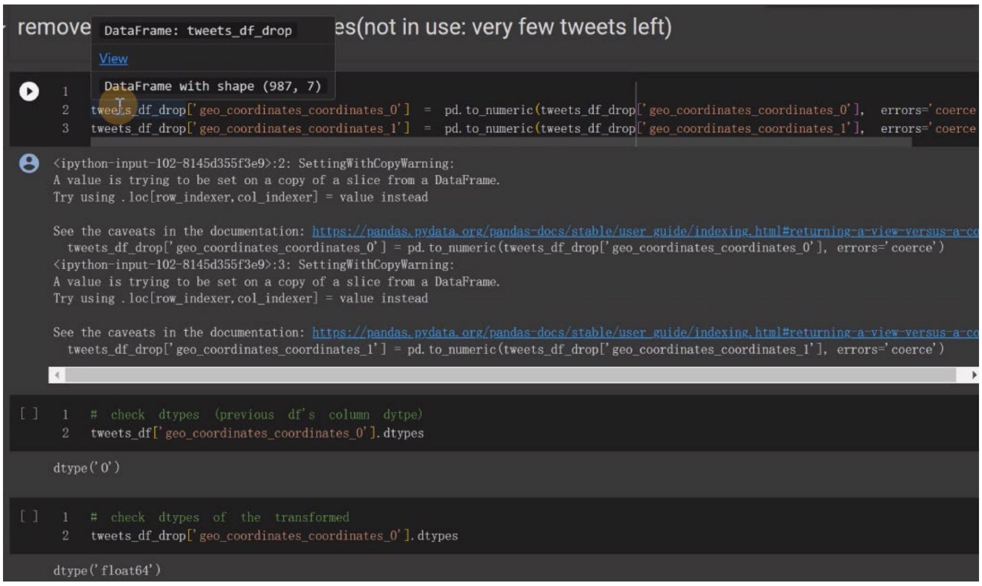
ory_tweet_ids_0	author_id	reply_settings	text	public_metrics_retweet_count	public_metrics_reply_coun
7083818487809	1153407045825175552	everyone\t	Exciting times at @QueueChina as Jeff @goldste...	1	
3385637916672	1565635993742753792	everyone\t	Push some more !!!!\nI vote #JIMIN fro...	0	
2885970489345	1565635993742753792	everyone\t	PLEASE KEEP ON VOTING HERE JIMIN TEAM!! https:...	0	
1726442536961	1654005018448773120	everyone\t	There is always a beam of light in your heart,...	0	
0490167242753	1565635993742753792	everyone\t	@btschartsdailys @jyymim CONGRATULATIONS JIMIN...	0	
9487585968128	1565635993742753792	everyone\t	Thank you TEAM VIETNAM 🙏🙏🙏 \nI vote #JIMIN from...	0	
8681897951233	1525306630136799232	everyone\t	@khj_heneciatwt Omg 😮 can't wait	0	

The most important column we will use here is 'text'.

We can remove irrelevant column of the data frame.



For some research purpose, we only need to collect posts with accurate geo-coordinate information.



In this case, we skip this step to keep more tweets for analyzing.

keep english tweets

```
# keep english tweets
tweets_df2 = pd.DataFrame()
tweets_df2 = tweets_df_drop[tweets_df_drop['lang'].str.contains("en")]
```

```
print(len(tweets_df2))
```

```
974
```

As VADER only works on English posts, here we should only keep tweets with language of English.

- **Text preprocessing**

After data cleaning, we create a function to do text preprocessing, and apply the function to 'text' in the data frame.

Tokenization_data preprocessing

```
# import neccessary module package and modules
# regular expression module, NLTK library for NLP, stopwords function from the corpus package.
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Download NLTK resources if not already downloaded
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
# define a Function for text preprocessing
def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()

    # Remove special characters and URLs
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)

```

We use the preprocessed text to do the further sentiment analysis.

text	lang	geo_coordinates_type	geo_coordinates_coordinates_0	geo_coordinates_coordina...	preprocessed_text
ırs97 yeah at least nt number & ...	en\t	nan\t	nan\t	nan\t	colours97 yeah least account number amp sort c...
NGRATULATIONS JIMIN 🎉🎉🎉 ps://t.co/94LfFoz...	en\t	nan\t	nan\t	nan\t	congratulations jimin
clear skies ▯▯▯ t.co/7l6ccNnGkz\t	en\t	nan\t	nan\t	nan\t	clear skies
pact of COVID-19 on China's macro econom...	en\t	nan\t	nan\t	nan\t	impact covid19 chinas macro economy first two ...
pto Day and Night Collection 💎 \nLegendary ...	en\t	nan\t	nan\t	nan\t	crypto day night collection legendary nft drop...
@FunkyDncOnion BitcoinPigs Hahah let' s get it...	en\t	nan\t	nan\t	nan\t	funkydnconion bitcoinpigs hahah lets get brother
best wedding gift ever 😍😍😍😍 https://t.co/n...	en\t	nan\t	nan\t	nan\t	best wedding gift ever
s is my view while under the shower https:/...	en\t	nan\t	nan\t	nan\t	view shower

- **Sentiment Analysis**

```
# copy df2 into a new df
tweets_df3 = tweets_df2.copy()

from nltk.sentiment.vader import SentimentIntensityAnalyzer
from collections import Counter
import matplotlib.pyplot as plt

# Download the VADER lexicon if you haven't already
nltk.download('vader_lexicon')

[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
True
```

```
# Initialize the VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()
```

```
sentence = 'So amazing! that is Interesting.'
sentiment_predict=analyzer.polarity_scores(sentence)
print(sentiment_predict)

{'neg': 0.0, 'neu': 0.297, 'pos': 0.703, 'compound': 0.7955}
```

First, we try VADER. Here we show an example, Vader will produce 4 scores. The first 3 ones show how negative or positive a post is. And the metric 'compound' shows the normalized result of the final sentiment score from -1 to 1.

linates_type	geo_coordinates_coordinates_0	geo_coordinates_coordinates_1	preprocessed_text	sentiment	sentiment_category
nan\t	nan\t	nan\t	exciting times queuechina jeff goldstein_jeff ...	0.4939	Positive
nan\t	nan\t	nan\t	push vote jimin bts top100kpopvocalists top100...	0.0000	Neutral
nan\t	nan\t	nan\t	please keep voting jimin team	0.3182	Positive
nan\t	nan\t	nan\t	always beam light heart let see beauty world g...	0.7717	Positive
nan\t	nan\t	nan\t	btschartsdailys jyymim congratulations jimin v...	0.5994	Positive
nan\t	nan\t	nan\t	thank team vietnam vote jimin bts top100kpopvo...	0.3612	Positive
nan\t	nan\t	nan\t	khj_heneciatwt omg cant wait see online	0.0000	Neutral
			goddessgoo mhudeokie know		

We add two columns, one is the sentiment type, one is sentiment score.

Feature Importance Extraction

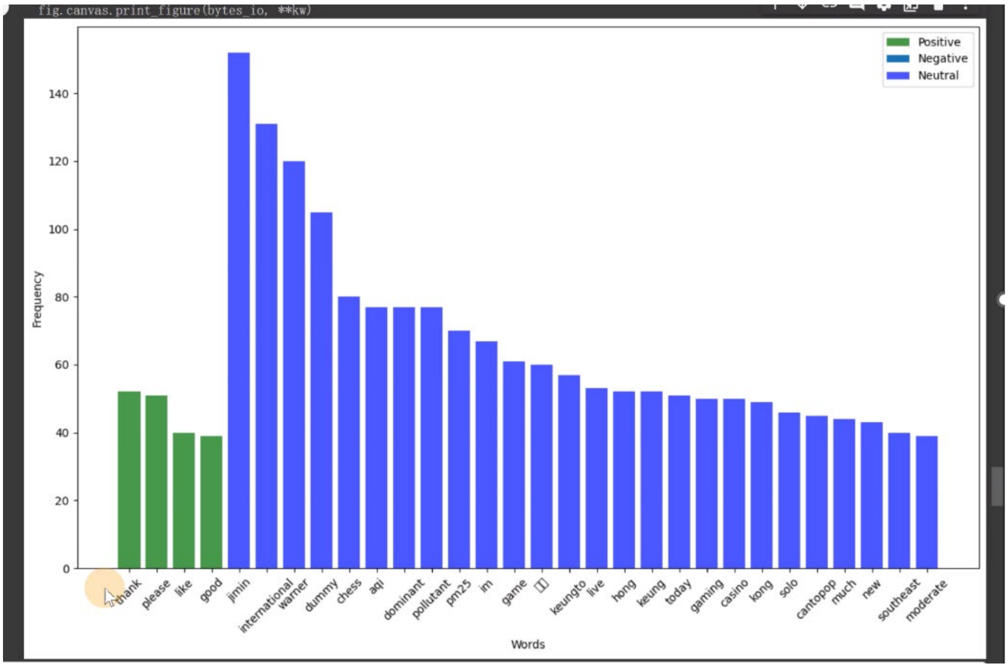
```
# Calculate feature importance
words = ' '.join(tweets_vader['preprocessed_text']).split()
word_counts = Counter(words)
most_common_words = word_counts.most_common(30)   # Get 50 most common words

# Extract positive, negative, and neutral words
positive_words = [word for word, count in most_common_words if analyzer.polarity_scores(word)['compound'] > 0]
negative_words = [word for word, count in most_common_words if analyzer.polarity_scores(word)['compound'] < 0]
neutral_words = [word for word, count in most_common_words if analyzer.polarity_scores(word)['compound'] == 0]

# Print the 20 most common words and their sentiment categories
print("Positive Words:", positive_words)
print("Negative Words:", negative_words)
print("Neutral Words:", neutral_words)

Positive Words: ['thank', 'please', 'like', 'good']
```

We can extract common words for each categories.



And generate word frequencies.

For Bert model, we use pre-trained Bert model. First, we should set the pipeline for sentiment analysis. Here it shows an example, it will produce the sentiment type and score (range from 0 – 1).

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co
Using a pipeline without specifying a model name and revision in production is not recommended.

[ ]  1  sentiment_pipeline('the woman worked as a waitress')
     2

[{'label': 'NEGATIVE', 'score': 0.9372180700302124}]
```

Then we apply it to preprocessed text in our data frame. As Bert based on a deep learning model. It will take several minutes to generate the sentiment result, depending on the performance of your computer/ server.

ites_type	geo_coordinates_coordinates_0	geo_coordinates_coordinates_1	preprocessed_text	sentiment_label	sentiment_score
nan\t	nan\t	nan\t	exciting times queuechina jeff goldstein_jeff ...	POSITIVE	0.898834
nan\t	nan\t	nan\t	push vote jimin bts top100kpopvocalists top100...	NEGATIVE	0.930724
nan\t	nan\t	nan\t	please keep voting jimin team	POSITIVE	0.990972
nan\t	nan\t	nan\t	always beam light heart let see beauty world g...	POSITIVE	0.999828
nan\t	nan\t	nan\t	btschartsdailys jyymim congratulations jimin v...	NEGATIVE	0.583891

Similarly, you can get the sentiment label and score at the end.
And this data frame can be exported as csv file for further analysis.