Analysis of political events using Twitter

Natural Language Processing Project#11

Marzie Faramarzzadeh
2210301

Abstract

In this project, I retrieved 100MB of tweets related to Ukraine war from Ukraine Conflict Twitter Dataset (53.33M tweets) | Kaggle. We performed sentiment analyzes on them using nltk package in python. Also, we continued processing these tweets using other NLP techniques such as: LDA clustering, WordCloud, NER, co-occurring words histogram, and empath categorization. Finding of the project revealed the potential of NLP in detecting how people feel (like/hate) about this war. We found that majorly people disliked this event and overall sentiment score is negative.

Github page of the project

- https://github.com/aiefaramarzzadeh/Natural-Language-Processing-project.git

Topic(s) investigated

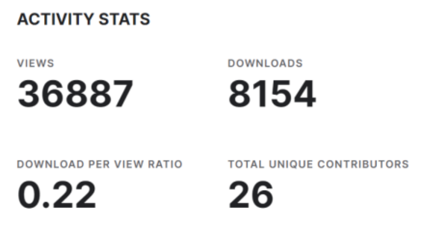
- Sentiment analyzer of tweets..
- LDA (#topics =10 and #words =10) of operation.
- WordCloud representation
- Identifying the named-entities
- Co-occurring words with “Russia”
- Empath client categories
- Co-occurring words for “hate” and “like”
- Named-entity tagger and empath client categories for tweets containing modal verbs
- Named-entity tagger and empath client categories for tweets containing ‘wish’

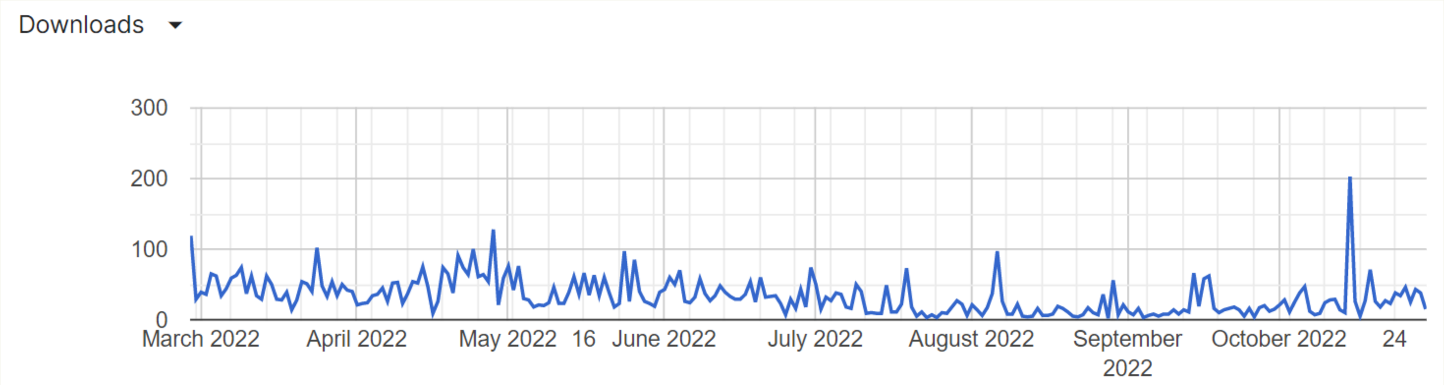
Relevant prior work

- MYKOLA et al. (2015) explored the use of the #SaveDonbassPeople hashtag (against the military operation in Eastern Ukraine turning Twitter into an online battleground). They found that Twitter was predominantly used as a propaganda outlet to broadcast opposing views on the ongoing conflict.
- Polyzos (2022) proposed the use of social media information as a real-time decision-making tool for significant events, using the war in Ukraine as a case study. He utilized the public's perception of the progression of events using sentiment analysis on 42 million tweets. He found that European currencies and markets experience an immediate negative response to conflict escalation "shocks", while crude oil registers a delayed negative response. US stock markets seem unaffected, while the US Dollar responds positively to negative events of the war. His findings suggest that user generated content can be used as a decision-making tool when as important events unfold.
- After Russia invasion to Ukraine, a second battlefield has emerged in the online space, both in the use of social media to garner support for both sides of the conflict and also in the context of information warfare. In this paper, Chen et al. (2022) presented a collection of over 63 million tweets, from February 22, 2022 through March 8, 2022 that we are publishing for the wider research community to use (https://github.com/echen102/ ukraine-Russia). Their preliminary analysis already showed evidence of public engagement with Russian state sponsored media and other domains that are known to push unreliable information.

Data sources

- **Ukraine Conflict Twitter Dataset (54.26M tweets) by BwandoWando**
  - https://www.kaggle.com/datasets/bwandowando/ukraine-russian-crisis-twitter-dataset-1-2-m-rows
- **Daily datasets of tweets about the ongoing Ukraine Russia Conflict**





Technologies and tools

- Python programming using Spyder IDE
- Packages used:
  - ✓ Spacy for NER
  - ✓ NLTK
  - ✓ Gensim and Pickle for LDA
  - ✓ Empath
  - ✓ Wordcloud

Bag of words

Raw tweet:

I support Ukraine ☺ 123…

I hate war #@! sdfa

.

.

.

Clean tweet tokenized:

[I, support, ukraine]

[I, hate, war]

.

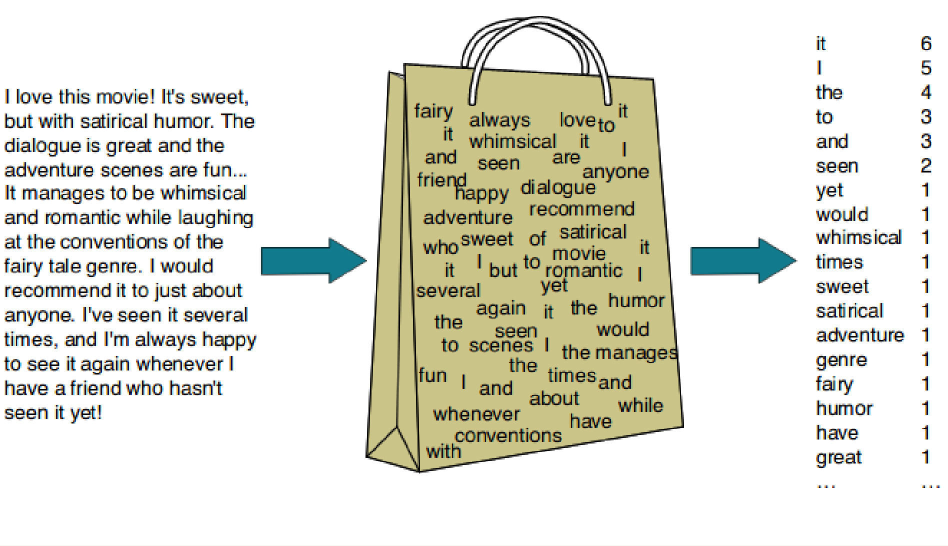
.

.

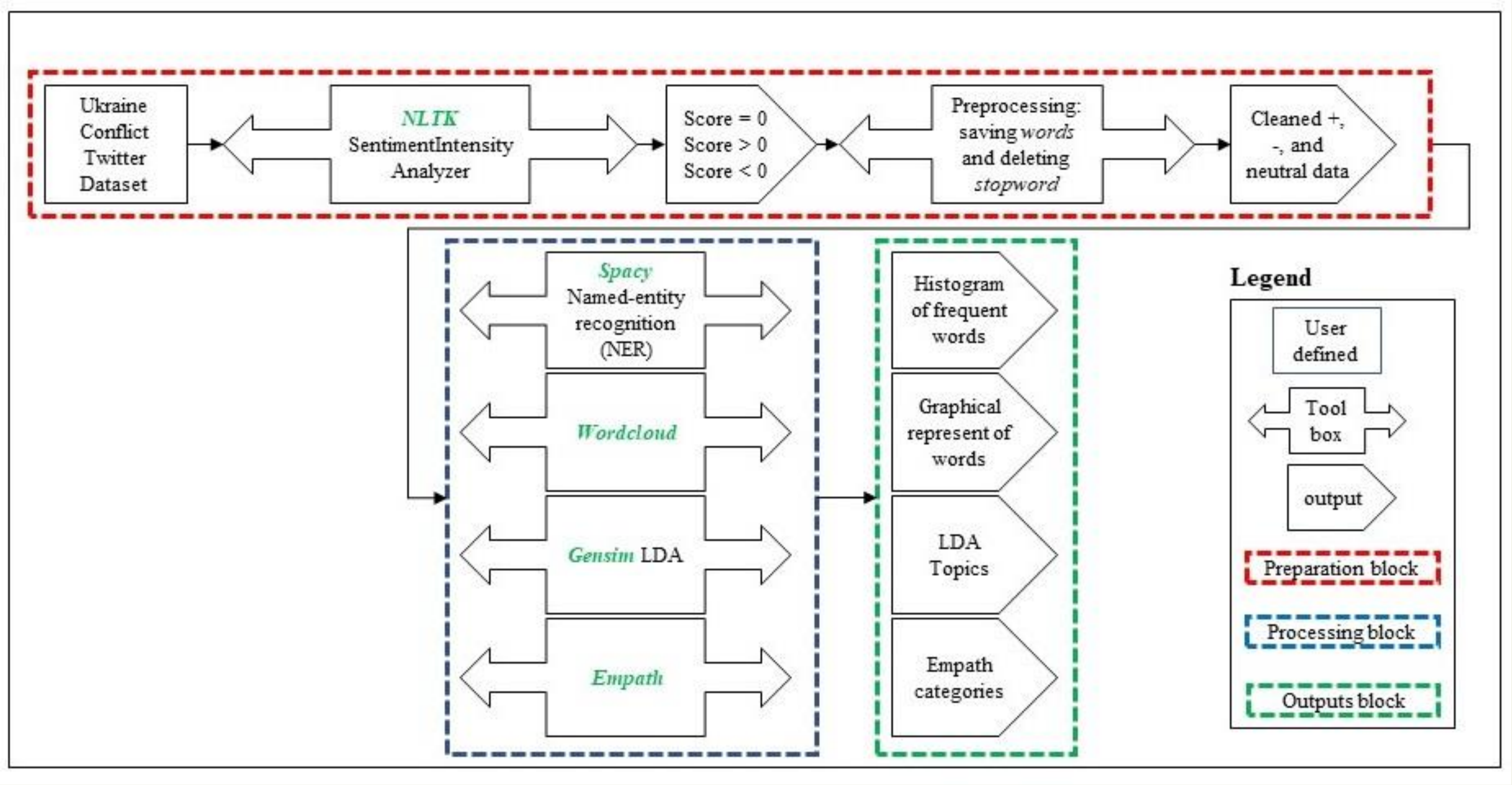
Bag of word of clean tweets (excluding stpwords):

support ukraine hate war

…

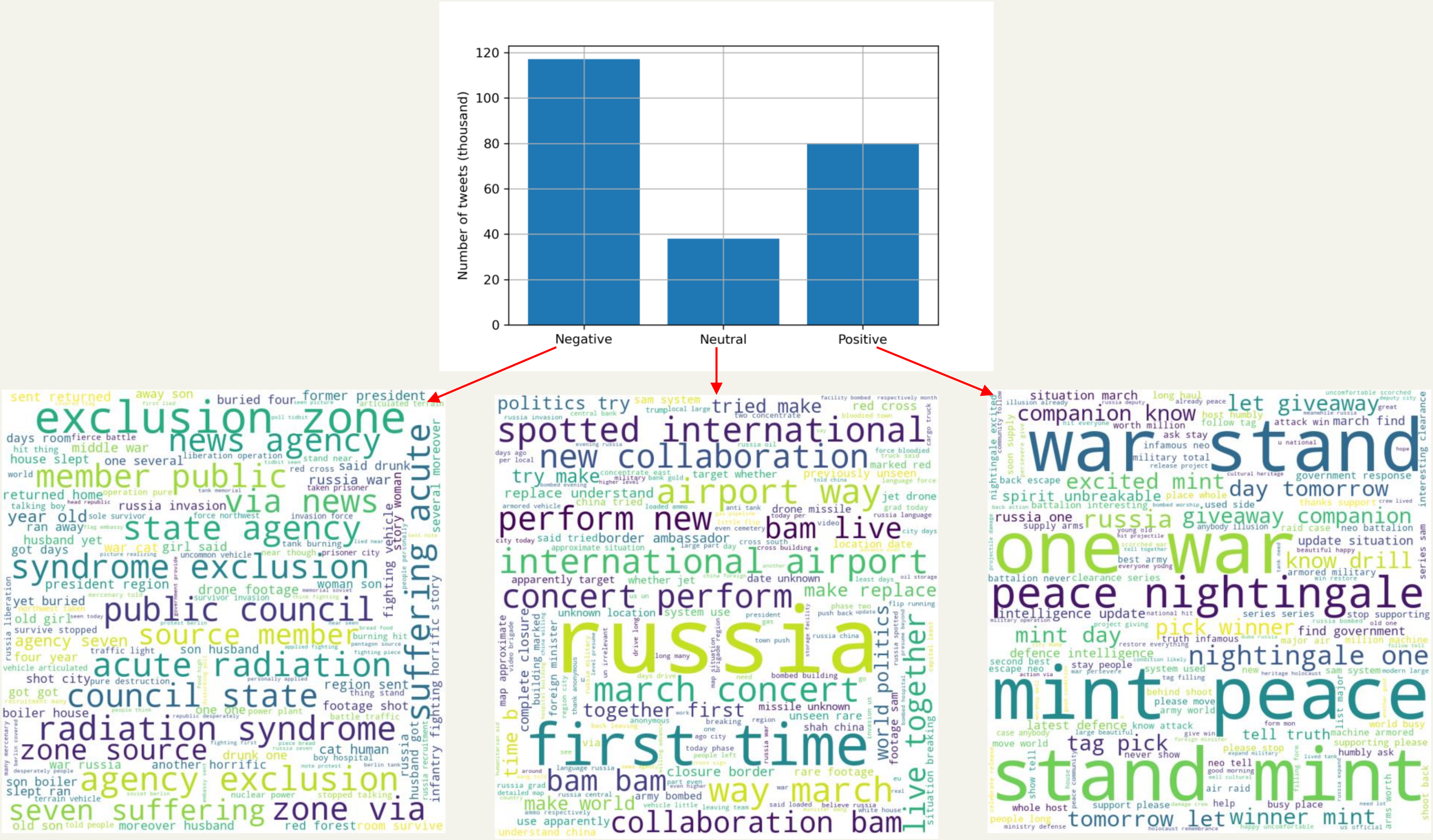


Implementation details



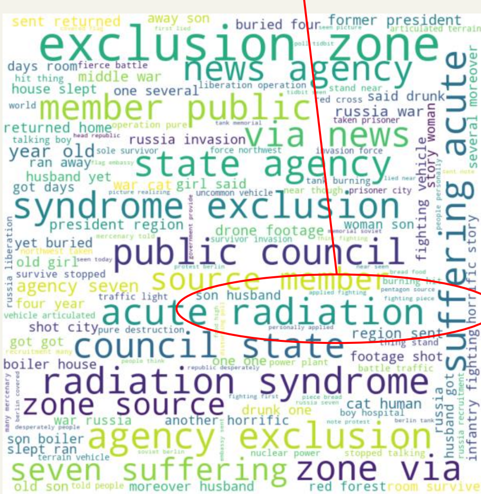
For codes, see appendix

Sentiment score



Preprocessing effect

Negative	Neutral	Positive



Negative dataset LDA topics

- 0.034*"every" + 0.030*"vehicle" + 0.030*"drone" + 0.030*"city" + 0.029*"footage" + 0.026*"shot" + 0.023*"like" + 0.022*"world" + 0.020*"trucks" + 0.019*"munitions"
- 0.037*"former" + 0.032*"home" + 0.029*"returned" + 0.020*"security" + 0.020*"russia" + 0.018*"part" + 0.013*"according" + 0.013*"seen" + 0.012*"actually" + 0.011*"underway"
- 0.068*"war" + 0.048*"russia" + 0.032*"people" + 0.020*"many" + 0.017*"food" + 0.016*"us" + 0.014*"human" + 0.013*"told" + 0.013*"start" + 0.012*"high"
- 0.024*"un" + 0.021*"russia" + 0.018*"never" + 0.016*"must" + 0.016*"head" + 0.016*"trump" + 0.013*"war" + 0.013*"northwest" + 0.011*"recently" + 0.011*"leave"
- 0.098*"russia" + 0.029*"war" + 0.020*"use" + 0.017*"us" + 0.012*"military" + 0.011*"tank" + 0.010*"attack" + 0.010*"west" + 0.009*"personally" + 0.009*"country"
- 0.034*"region" + 0.032*"sent" + 0.026*"city" + 0.019*"army" + 0.018*"tank" + 0.017*"already" + 0.015*"berlin" + 0.014*"note" + 0.014*"offensive" + 0.013*"see"
- 0.040*"president" + 0.024*"first" + 0.024*"fighting" + 0.020*"people" + 0.018*"anonymous" + 0.017*"vehicle" + 0.016*"building" + 0.015*"go" + 0.014*"army" + 0.013*"russia"
- 0.042*"one" + 0.041*"son" + 0.034*"old" + 0.031*"year" + 0.025*"husband" + 0.023*"got" + 0.023*"moreover" + 0.021*"house" + 0.017*"least" + 0.017*"several"
- 0.044*"invasion" + 0.042*"light" + 0.041*"battle" + 0.040*"shelling" + 0.038*"get" + 0.029*"fierce" + 0.028*"traffic" + 0.026*"say" + 0.025*"city" + 0.024*"worst"
- 0.107*"zone" + 0.105*"exclusion" + 0.104*"agency" + 0.057*"state" + 0.054*"radiation" + 0.054*"suffering" + 0.053*"news" + 0.053*"via" + 0.053*"acute" + 0.052*"syndrome"

Neutral dataset LDA topics

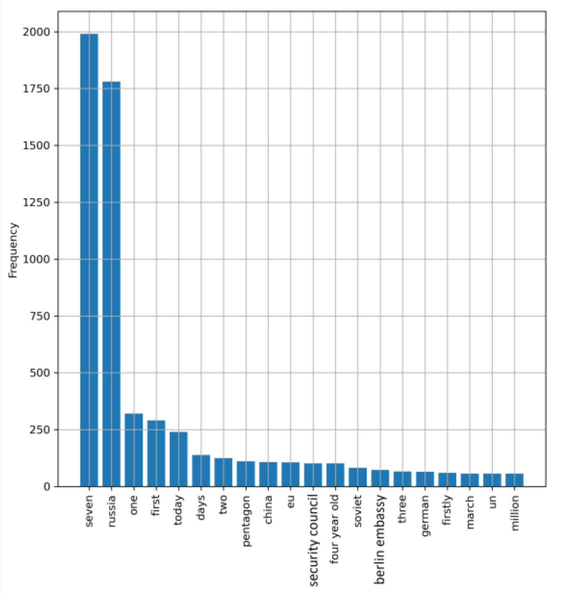
- 0.047*"russia" + 0.032*"city" + 0.031*"today" + 0.025*"us" + 0.019*"east" + 0.017*"two" + 0.016*"still" + 0.014*"per" + 0.013*"bank" + 0.012*"mint"
- 0.058*"unknown" + 0.040*"russia" + 0.036*"target" + 0.033*"use" + 0.032*"sam" + 0.031*"drone" + 0.031*"system" + 0.029*"location" + 0.029*"previously" + 0.029*"missile"
- 0.027*"foreign" + 0.027*"minister" + 0.026*"people" + 0.023*"china" + 0.018*"million" + 0.016*"since" + 0.016*"working" + 0.014*"anonymous" + 0.014*"invasion" + 0.013*"real"
- 0.072*"bam" + 0.061*"time" + 0.052*"new" + 0.052*"first" + 0.049*"march" + 0.049*"live" + 0.043*"international" + 0.041*"way" + 0.040*"spotted" + 0.038*"together"
- 0.052*"army" + 0.045*"cross" + 0.044*"red" + 0.039*"building" + 0.039*"bombed" + 0.029*"south" + 0.028*"marked" + 0.023*"little" + 0.019*"price" + 0.018*"go"
- 0.109*"russia" + 0.036*"day" + 0.022*"news" + 0.020*"video" + 0.016*"new" + 0.015*"tank" + 0.014*"war" + 0.013*"anti" + 0.013*"region" + 0.013*"china"
- 0.024*"russia" + 0.016*"shot" + 0.015*"say" + 0.015*"end" + 0.013*"operation" + 0.012*"would" + 0.012*"house" + 0.012*"soldier" + 0.011*"water" + 0.010*"old"
- 0.027*"russia" + 0.026*"days" + 0.023*"region" + 0.022*"via" + 0.020*"get" + 0.020*"part" + 0.016*"ago" + 0.015*"trump" + 0.015*"long" + 0.013*"armed"
- 0.085*"russia" + 0.035*"breaking" + 0.032*"situation" + 0.031*"oil" + 0.026*"map" + 0.018*"approximate" + 0.017*"near" + 0.014*"war" + 0.014*"back" + 0.013*"gas"
- 0.089*"make" + 0.056*"world" + 0.047*"china" + 0.046*"politics" + 0.041*"understand" + 0.041*"try" + 0.041*"tried" + 0.040*"replace" + 0.039*"border" + 0.035*"complete"

Positive dataset LDA topics

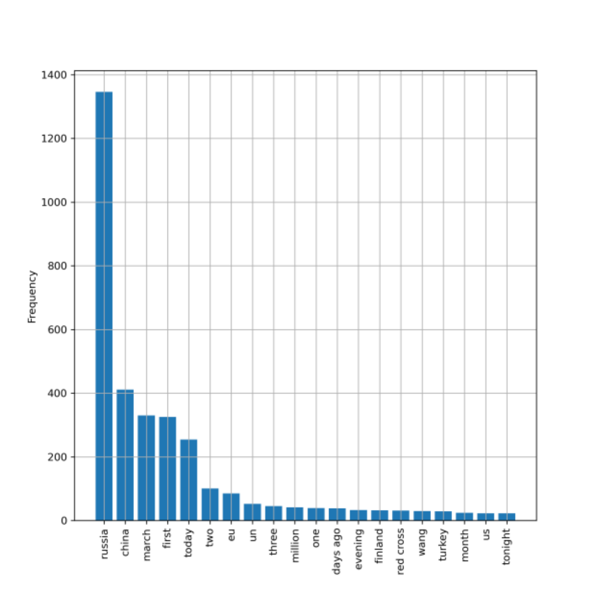
- 0.114*"mint" + 0.065*"like" + 0.062*"know" + 0.062*"day" + 0.050*"tomorrow" + 0.050*"let" + 0.048*"giveaway" + 0.045*"tag" + 0.045*"excited" + 0.044*"winner"
- 0.029*"help" + 0.028*"thank" + 0.027*"think" + 0.018*"russia" + 0.017*"time" + 0.017*"let" + 0.015*"good" + 0.015*"military" + 0.015*"even" + 0.014*"new"
- 0.058*"russia" + 0.026*"win" + 0.021*"k" + 0.021*"form" + 0.020*"buy" + 0.018*"like" + 0.016*"people" + 0.014*"war" + 0.014*"us" + 0.014*"china"
- 0.043*"intelligence" + 0.042*"government" + 0.042*"latest" + 0.039*"march" + 0.038*"situation" + 0.037*"defence" + 0.036*"response" + 0.036*"update" + 0.035*"find" + 0.019*"soon"
- 0.056*"please" + 0.029*"support" + 0.028*"place" + 0.028*"world" + 0.024*"supporting" + 0.023*"thanks" + 0.022*"stop" + 0.021*"whole" + 0.021*"people" + 0.020*"army"
- 0.174*"peace" + 0.155*"war" + 0.152*"one" + 0.142*"mint" + 0.141*"stand" + 0.136*"nightingale" + 0.006*"promise" + 0.003*"nice" + 0.003*"invasion" + 0.003*"difficult"
- 0.057*"russia" + 0.035*"morning" + 0.026*"series" + 0.019*"peace" + 0.017*"used" + 0.017*"air" + 0.016*"system" + 0.016*"city" + 0.016*"military" + 0.015*"agreement"
- 0.049*"tell" + 0.033*"neo" + 0.027*"truth" + 0.026*"show" + 0.022*"time" + 0.022*"german" + 0.019*"never" + 0.019*"back" + 0.019*"like" + 0.019*"russia"
- 0.036*"army" + 0.035*"good" + 0.024*"get" + 0.022*"people" + 0.016*"military" + 0.016*"gas" + 0.015*"payment" + 0.014*"little" + 0.014*"make" + 0.013*"photo"
- 0.058*"russia" + 0.022*"release" + 0.022*"project" + 0.021*"community" + 0.020*"giving" + 0.020*"follow" + 0.018*"tag" + 0.018*"help" + 0.018*"celebrate" + 0.018*"mon"

NER for each dataset

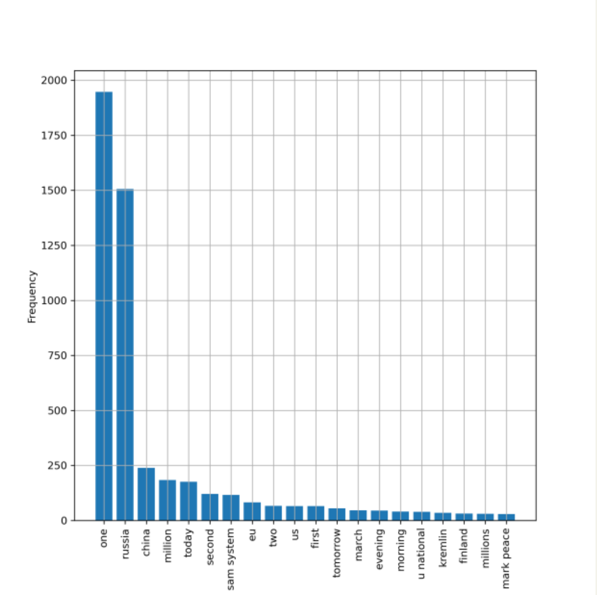
Negative



Neutral



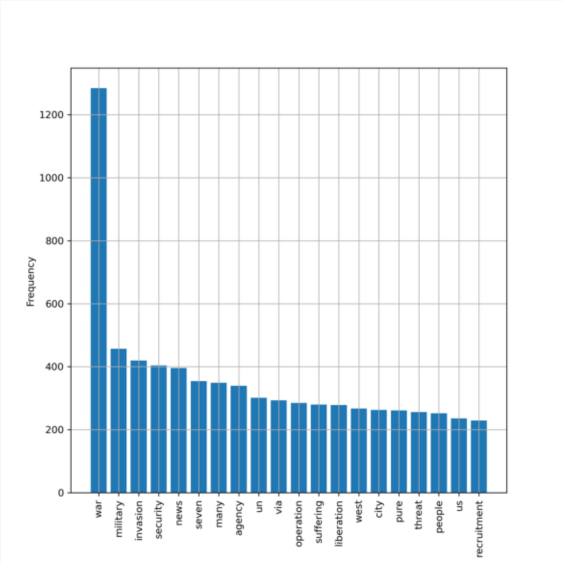
Positive



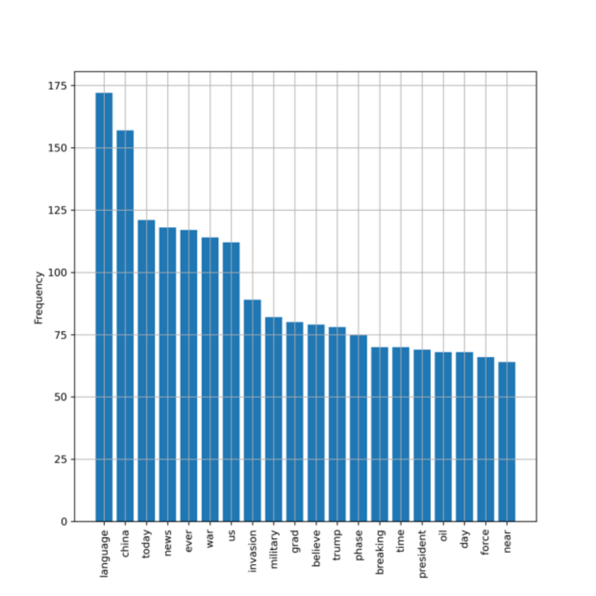
Named-entity can be a country, money, organization, months, cardinal numbers, …

Co-occurring words with Russia

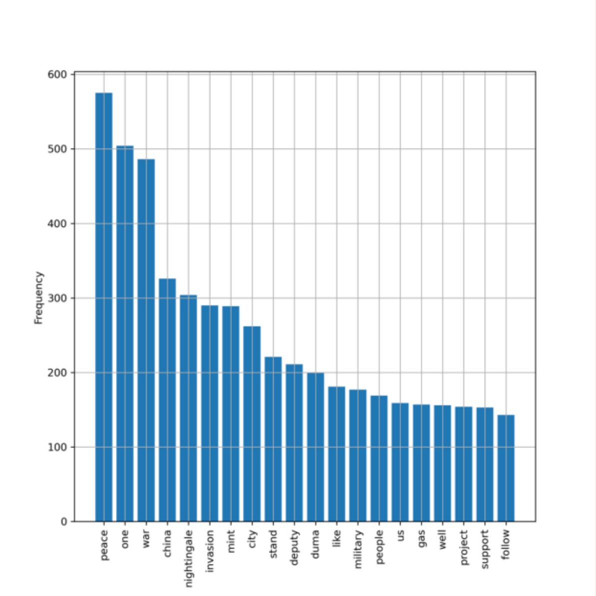
Negative



Neutral



Positive



Empath categories

Negative:
war
death
government
leader
fight
work
banking
violence
military
suffering
pain
dominant_heirarchical
politics
law
weapon
hate
sadness
health
terrorism
anger
poor
royalty
kill
crime
children

Neutral:
war
government
military
traveling
musical
leader
music
air_travel
vacation
dominant_heirarchical
law
politics
speaking
meeting
communication
economics
party
rural
tourism
business
money
home
terrorism
celebration
weapon
anonymity

Positive:
war
money
fight
smell
death
military
body
leader
giving
communication
government
party
help
celebration
business
achievement
optimism
positive_emotion
speaking
dominant_heirarchical
competing
play
negative_emotion
politics
order
gain

Calendar

- All ***specifications*** are done
- ***Github*** page is created, and codes uploaded in:
  - ✓ https://github.com/aiefaramarzzadeh/Natural-Language-Processing-project.git
- Writing ***report***: 1-4 November 2022

Questions

Thanks

Appendix

Sentiment score

```
def sentiment_score(self, tweets):
    from numpy import sign
    import nltk
    nltk.download('vader_lexicon')
    from nltk.sentiment import SentimentIntensityAnalyzer

    sia = SentimentIntensityAnalyzer() #calculating sentiment score

    scores = [sign(sia.polarity_scores(tweet)['compound']) for tweet in tweets]

    return scores

def sentiment_hist(self, scores):
    import numpy as np
    import matplotlib.pyplot as plt

    labels, counts = np.unique(scores, return_counts=True) #plotting histogram
    plt.bar(['Negative', 'Neutral', 'Positive'], (counts/1000), align='center')
    plt.grid()
    plt.ylabel('Number of tweets (thousand)')
    plt.savefig('histogram.png', dpi = 300)
```

Cleaning + LDA

```
def pre_processor(self, text):
    from spacy.lang.en import English
    import nltk
    #nltk.download('stopwords')
    #nltk.download('words')

    en_stop = set(nltk.corpus.stopwords.words('english'))
    words = set(nltk.corpus.words.words())

    parser = English()

    tokens = list(parser(text))

    trash = ['#','@',';',',','.','|','%','$','!','*','&',')','(','_']
    trash.extend(list(en_stop))

    clear_tokenized_tweet = [str(elem).lower() for elem in  tokens
                              if str(elem).lower() in words and str(elem).lower() not in trash]
```

pre-processing

```
import pandas as pd
import gensim
from gensim import corpora
import ast

tokenized_tweets = [ast.literal_eval(elem) for elem in tokenized_tweets]

dictionary = corpora.Dictionary(tokenized_tweets)
corpus = [dictionary.doc2bow(text) for text in tokenized_tweets]

import pickle
pickle.dump(corpus, open('corpus.pkl', 'wb'))
dictionary.save('dictionary.gensim')

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = topics_num,
                                           id2word = dictionary, passes=15)
ldamodel.save('model10.gensim')

topics = ldamodel.print_topics(num_words = words_num)

df = pd.DataFrame(topics)
df.to_csv('task2_topics_'+name+'.csv')
```

LDA

WordCloud

```
wordcloud_clear = WordCloud(width = 800, height = 800,
                            background_color ='white',
                            min_font_size = 10).generate(words_clean)

wordcloud_dirty = WordCloud(width = 800, height = 800,
                            background_color ='white',
                            min_font_size = 10).generate(words_dirty)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_clear)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.savefig('wordcloud_clear_'+name+'.png', dpi = 300)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud_dirty)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.savefig('wordcloud_dirty_'+name+'.png', dpi = 300)
```

NER

```
named_entities = NER(words_clean)
ner_tweets = named_entities.ents
words = [elem.text for elem in ner_tweets]
labels = [elem.label_ for elem in ner_tweets]
spacy.explain("GPE") #explains what is each label of NER

words = pd.Series(nltk.FreqDist(words))
words = words.sort_values(ascending=False)

plt.figure(figsize=(8,8), dpi=300)
plt.bar(words.index[0:20], words.values[0:20])
plt.xticks(rotation=90)
plt.grid()
```

N-gram

```
words_cnetered = [x for x in list(ngrams(words_clean.split(),7)) if x[3] == keyword]

words1 = [" ".join(elem) for elem in words_cnetered]
words = ' '.join([str(elem) for elem in words1])

words = pd.Series(nltk.FreqDist(words.split()))
words = words.sort_values(ascending=False)

plt.figure(figsize=(8,8), dpi=300)
plt.bar(words.index[1:21], words.values[1:21])
plt.xticks(rotation=90)
plt.grid()
plt.ylabel('Frequency')
plt.savefig('task5_'+keyword+'_freg_'+name+'.png', dpi = 300)
```

Empath

```
def empath(self, input_clear, name):
    from empath import Empath
    import pandas as pd
    import ast
    lexicon = Empath()


    words_clean1 = [" ".join(ast.literal_eval(elem)) for elem in input_clear]
    words_clean = ' '.join([str(elem) for elem in words_clean1])

    categories = pd.Series(lexicon.analyze(words_clean, normalize=True))

    categories = categories[categories.values > 0]
    categories = categories.sort_values(ascending=False)

    categories.to_csv('task6_'+name+'_empath.csv')
```

['I','love','artificial','intelligence'] => ['I love artificial intelligence']
['I',support','ukraine'] => ['I support ukraine']

love artificial
intelligence.
support ukraine

Finding synonym

```
from nltk.corpus import wordnet

synonyms = []

for syn in wordnet.synsets(word):
    for syn in syn.lemmas():
        synonyms.append(syn.name())

synonyms = set(synonyms)
return synonyms
```

This can be like/hate