Analysis of tweets about Ukraine using text mining techniques "Data is the new oil" - it is only one of many thoughts about data which shows a high value of information. Nowadays we collect many data, much more than in the past and also we have more sources of this data. This project is aimed on collecting, preparing and analysing posts from one of the most popular social networking service - Twitter. Using Twitter's REST API, tweets with query "Ukraine" are collected, then cleaned and analyzed data is used in topic modelling using LDA. In the second project "NER and Sentiment analysis with tweets about Ukraine" the same data is used in Named Entity Recognition and Sentiment analysis. Note: "Data collection" part is presented just for reader to know how the data was collected. In order to execute this code, the reader needs to configure his own keys. However, the project works without executing "Data collection" part. from datetime import date, timedelta import nltk from nltk.tokenize import RegexpTokenizer from nltk.corpus import stopwords from nltk.sentiment.vader import SentimentIntensityAnalyzer analyser = SentimentIntensityAnalyzer() import tweepy from tweepy import OAuthHandler import pandas as pd import numpy as np import matplotlib.pyplot as plt %matplotlib inline import re import warnings warnings.filterwarnings("ignore") Data collection Data are collected using Twitter's REST API. Tweets in english about Ukraine, from 2022-01-27 to 2022-02-04 are selected. To use mentioned API, developer has to know four private keys - "consumer key", "consumer secret", "access token" and "access token secret" which are available after creating account on Twitter's API website. Code below show how connection with API is achieved and how the data is downloaded. More information upon the link: https://developer.twitter.com/en/docs/twitter-api CONSUMER KEY = "xxxx" CONSUMER SECRET = "xxxx" ACCESS TOKEN = "xxxx" ACCESS TOKEN SECRET = "xxxx" auth = tweepy.OAuthHandler(CONSUMER KEY, CONSUMER SECRET) auth.set access token(ACCESS TOKEN, ACCESS TOKEN SECRET) api = tweepy.API(auth) #dates used ['2022-01-27','2022-01-28','2022-01-29','2022-01-30','2022-01-31', # '2022-02-01','2022-02-03','2022-02-04'] df = pd.DataFrame(columns=['Tweets']) start date = date.today() end date = start date - timedelta(days=7) delta = timedelta(days=1) while start date >= end date: start date -= delta data = pd.DataFrame([tweets.text for tweets in api.search tweets(q="Ukraine", lang="en", count=100 , result type='recent',until=start date)] , columns=['Tweets']) #print(start_date.strftime("%Y-%m-%d %H:%M:%S")) df=pd.concat([df,data]) df.reset index(drop=1) #df.to csv('df Ukr aine.csv', index=False, header=True) #df U = df.copy()Data analysis Awareness of data is one of the most important steps for properness in decision making. In this part the data is analysed in order to make properly decisions in further steps. df U = pd.read csv("df Ukraine.csv") df U=df U.rename(columns={"Tweets": "tweet"}) tweet 0 RT @kennardmatt: The Azov Battalion was a neo-... 1 RT @john_sipher: "A free Ukraine will pull Rus... 2 RT @Reuters: President Joe Biden said that he ... RT @DCBMEP: .@BorisJohnson on top form at #PMQ... 4 RT @DonaldJTrumpJr: Americas kids dying in a E... 763 All my hope goes to the brave NATO troops curr... 764 Russia Planned Fake Video to Gain Support for ... 765 I'm going to say it again you can hold a neutr... 766 RT @TimothyDSnyder: 0/6. A thread: Six Things ... 767 I called it\n\n"The US has accused Russia of c... 768 rows × 1 columns len(df U) In [4]: df U['tweet'].nunique() Out[4]: df U[df U.duplicated(subset=['tweet'])==True] 9 RT @BestForBritain: That's strange. The PM tol... 16 RT @BestForBritain: That's strange. The PM tol... 32 RT @LeftFlankVets: United States got weapons t... 34 RT @Ladyinyellow_: How did my healthy brother... 41 RT @LeftFlankVets: United States got weapons t... 736 RT @RepAdamSchiff: First, Cruz begs forgivenes... 749 RT @27khv: What @anneapplebaum doesn't reveal ... RT @27khv: What @anneapplebaum doesn't reveal ... 760 RT @OlgaNYC1211: Just an observation. If you c... 766 RT @TimothyDSnyder: 0/6. A thread: Six Things ... 149 rows × 1 columns df U['is retweet'] = df_U['tweet'].apply(lambda x: x[:2]=='RT') df U['is retweet'].sum() # number of retweets Out[6]: 588 df_U.loc[df_U['is_retweet']].tweet.nunique() Out[7]: 439 In [8]: # 10 most repeated tweets pd.options.display.max colwidth = 200 df U.groupby(['tweet']).size().reset index(name='counts') \ .sort_values('counts', ascending=False).head(10) tweet counts 523 RT @ryangrim: Biden is in the process of starving and freezing to death millions of people in Afghanistan and there's war fever over Ukrain... 12 234 RT @JackPosobiec: Apparently, Biden was unaware of the talks going on this week in Paris between Germany, France, Russia, and Ukraine 10 532 RT @spaceeong: drop chief keef on the russia ukraine border shit would be handled in a few hours 9 546 RT @thevivafrei: Coward @JustinTrudeau in hiding, worrying about Ukraine's "territorial integrity" while simultaneously violating Canadians... 7 473 RT @leonidragozin: Ukraine is directly accusing the US of spreading disinfo RT @Ladyinyellow_: How did my healthy brother die in Ukraine overnight?\n\nA thread.\n\nThat's my brother, Fadlullah Agboluaje in this 261 6 436 RT @emilyhorne46: This is not true. President Biden said that there is a distinct possibility that the Russians could invade Ukraine in Feb... 6 337 RT @RonnyJacksonTX: Something IS NOT right with the Biden Administration's approach to Ukraine. It's totally illogical. It makes ZERO sense... 307 RT @RWPUSA: Constitutional law 101: Is withholding military aid for Ukraine in order to coerce a criminal investigation of a political oppo... 4 RT @levparnas: Hey @DonaldJTrumpJr, I taught you and your daddy where Ukraine is on a map. This crisis is due to his corrupt quid pro quo t... Dataset with tweets collected using Twitter's REST API is loaded. It contains 768 tweets about Ukraine, where 619 are unique tweets. Number of downloaded tweets is limited by API's setting, which does not allow to download more than 100 tweets in one iteration. In probe 588 are retweets and 439 are unique retweets. In the table are visible 10 most repeated tweets. In [9]: def find retweeted(tweet): '''This function will extract the twitter handles of retweed people''' return re.findall('(?<=RT\s)(@[A-Za-z]+[A-Za-z0-9-]+)', tweet) def find mentioned(tweet): '''This function will extract the twitter handles of people mentioned in the tweet''' return re.findall('(?<!RT\s)(@[A-Za-z]+[A-Za-z0-9-]+)', tweet) def find hashtags(tweet): '''This function will extract hashtags''' return re.findall('(#[A-Za-z]+[A-Za-z0-9-]+)', tweet) df U['retweeted'] = df U.tweet.apply(find retweeted) df U['mentioned'] = df U.tweet.apply(find mentioned) df U['hashtags'] = df U.tweet.apply(find hashtags) pd.set_option('display.min_rows', 20) retweeted mentioned hashtags tweet is retweet RT @kennardmatt: The Azov Battalion was a neo-Nazi paramilitary unit made up of [][][@kennardmatt] True football hooligans. Soon after the "Maidan revolution" in 2... RT @john_sipher: "A free Ukraine will pull Russia westward if it isn't brought to 1 [][]True [@john_sipher] heel." https://t.co/NjGViJWzHc RT @Reuters: President Joe Biden said that he would consider imposing direct 2 [] [][@Reuters] True sanctions on President Vladimir Putin if Russia invades Ukrain... RT @DCBMEP: .@BorisJohnson on top form at #PMQs - Labour look sunken behind 3 True [@DCBMEP] [@BorisJohnson] [#PMQs] Starmer; who has missed his target: 'a lawyer not a Leader'. No... RT @DonaldJTrumpJr: Americas kids dying in a European land war with Russia is a 4 [] []True [@DonaldJTrumpJr] small price to pay for Hunter Biden and fam to get rich AF... RT @WriterJackWhite: As @nato meets Russia under Putin's blackmail of 100,000 5 [@WriterJackWhite] [][@nato] True troops to invade Ukraine. In 2018 after NATO meet after Salis... RT @JuliaDavisNews: The tactic of terrorizing American audiences with the 6 [] [][@JuliaDavisNews] True possibility of nuclear war, in order to undermine U.S. support fo... RT @BestForBritain: That's strange. The PM told Parliament he is the one leading 7 [] [][@BestForBritain] True Europe on Ukraine negotiations. He's not even in the room.... @davnrussell @ASBMilitary All this could be resolved if Ukraine remains neutral it's [@davnrussell, 8 []False @ASBMilitary] really simple. No need for a N... https://t.co/nTRWngCO11 RT @BestForBritain: That's strange. The PM told Parliament he is the one leading 9 [] []True [@BestForBritain] Europe on Ukraine negotiations. He's not even in the room.... ••• RT @VentureBeat: Cyberattack was attempted against a western government 758 [][@VentureBeat] [] True 'entity' in Ukraine, researchers say https://t.co/6LTuSDIYdv RT @djrothkopf: So today we have:\n--Growing consensus that Biden team [] 759 True [@djrothkopf] []handling of Ukraine crisis has been very strong as pieces continue to... RT @OlgaNYC1211: Just an observation. If you can't locate Ukraine, name cities in 760 [] [] True [@OlgaNYC1211] Ukraine, have no clue what has been happening there over... RT @trussliz: My statement in response to US intelligence that Russia plans to 761 [] [][@trussliz] True fabricate an attack to justify military action against Ukrai... RT @willevans: Very casually publishing the best writers from from Pakistan—Uzma 762 [@willevans] [@AKurkov] []True Aslam Khan—& Dikraine—@AKurkov's GREY BEES translated by @B... All my hope goes to the brave NATO troops currently deployed on the border of 763 [][] []False Ukraine and Eastern Europe. While I d... https://t.co/hkjKoyvxUB Russia Planned Fake Video to Gain Support for Ukraine Invasion | 764 False [@Newsmax] []https://t.co/RzqOdGseQG https://t.co/GZwCZwpbiQ via @Newsmax I'm going to say it again you can hold a neutral position on Russia and US what I 765 [] []False have no time for is the whatabout... https://t.co/2Sk4ynuGFq RT @TimothyDSnyder: 0/6. A thread: Six Things Progressives Know About Ukraine [@USProgressives, [@TimothyDSnyder] 766 [#Ukraine] and Russia. @USProgressives @AOC #Ukraine I called it\n\n"The US has accused Russia of creating a propaganda video featuring [] [] 767 []False crisis actors, staged fake explosi... https://t.co/AAfjJxf7V6 768 rows × 5 columns # popular hashtags hashtags list df = df U.loc[df U.hashtags.apply(lambda hashtags list: hashtags list !=[]),['hashtags']] flattened_hashtags_df = pd.DataFrame([hashtag for hashtags list in hashtags list df.hashtags for hashtag in hashtags_list], columns=['hashtag']) popular hashtags = flattened hashtags df.groupby('hashtag').size()\ .reset index(name='counts') \ .sort values('counts', ascending=False) \ .reset index(drop=True) popular hashtags.head(10) hashtag counts 0 #Ukraine 25 1 #Russia 11 2 #BREAKING 4 3 #NATO 4 #MeganRapinoe 5 #TigrayGenocide 2 6 #China 7 **#UNCharter** #JohnsonTheCorru #AffiliateMarketing counts = flattened_hashtags_df.groupby(['hashtag']).size()\ .reset index(name='counts')\ .counts plt.figure() plt.hist(counts,color='deepskyblue') plt.xlabels = np.arange(1, counts.max()+1, 1)plt.xlabel('Hashtag number of appearances') plt.ylabel('Frequency',) plt.yscale('log', nonposy='clip') plt.show() Frequency 101 10° 10 15 20 25 Hashtag number of appearances Three definied function are used to get information about who is being retweeted, who is being tweeted at/mentioned and what hashtags are being used. #Ukraine is the most often used hashtag, what is not a surprise, because tweets about Ukraine were selected. Also second most often used hashtag #Russia has been expected. Other hashtags are used only in few tweets what is well visible on the hashtag frequency plot. In [14]: # take hashtags which appear at least this amount of times min appearance = 4 # find popular hashtags - make into python set for efficiency popular hashtags set = set(popular hashtags[popular hashtags.counts>=min appearance]['hashtag']) # make a new column with only the popular hashtags hashtags list df['popular hashtags'] = hashtags list df.hashtags.apply(lambda hashtag_list: [hashtag for hashtag in hashtag_list if hashtag in popular_hashtags_set]) # drop rows without popular hashtag popular_hashtags_list_df = hashtags_list_df.loc[hashtags list df.popular hashtags.apply(lambda hashtag list: hashtag list !=[])] # make new dataframe hashtag vector df = popular hashtags list df.loc[:, ['popular hashtags']] for hashtag in popular hashtags set: # make columns to encode presence of hashtags hashtag vector df['{}'.format(hashtag)] = hashtag vector df.popular hashtags.apply(lambda hashtag list: int(hashtag in hashtag list)) print(hashtag_vector_df) popular hashtags #BREAKING #MeganRapinoe #Ukraine \ [#Russia] 0 0 50 [#Ukraine] 0 0 1 0 63 [#Ukraine, #Ukraine] 0 1 0 0 [#Ukraine] 103 1 192 [#BREAKING] 0 0 202 [#Ukraine, #Russia] 0 275 [#Russia] 326 0 [#Ukraine] [#NATO, #Ukraine] 327 [#Russia, #Ukraine] 330 0 0 0 0 345 [#NATO, #Ukraine] 348 [#Russia] 351 [#BREAKING] [#Russia, #Ukraine] 352 0 376 [#Ukraine] 402 [#Ukraine, #Russia] 0 439 0 [#BREAKING] 1 454 [#Ukraine] 0 0 468 [#Russia, #Ukraine, #Ukraine] 1 0 472 0 [#NATO] 482 0 [#NATO] 0 518 [#Ukraine] 1 0 526 [#Ukraine] 542 [#Russia, #Ukraine] 0 [#BREAKING] 565 [#Russia, #Ukraine] 0 0 585 [#Ukraine] 0 0 655 [#Russia, #Ukraine] 707 [#Ukraine] 712 [#Ukraine] 0 721 [#Ukraine] 746 [#MeganRapinoe, #MeganRapinoe] 1 757 [#MeganRapinoe, #MeganRapinoe] 766 #Russia #NATO 37 1 50 0 0 63 103 192 202 1 275 1 326 0 327 0 1 330 0 1 345 348 1 351 1 352 0 0 376 1 0 402 439 0 454 0 0 1 468 472 482 0 518 0 526 542 1 564 1 565 0 585 655 1 707 712 0 0 721 0 746 0 0 757 0 0 766 hashtag matrix = hashtag vector df.drop('popular hashtags', axis=1) # calculate the correlation matrix correlations = hashtag_matrix.corr() print(correlations) #BREAKING #MeganRapinoe #Ukraine #Russia #BREAKING #BREAKING 1.000000 -0.091287 -0.528004 -0.252523 -0.133333 #MeganRapinoe -0.091287 1.000000 -0.361499 -0.172891 -0.091287 #Ukraine -0.528004 -0.361499 1.000000 0.075099 -0.137740 #Russia -0.252523 -0.172891 0.075099 1.000000 -0.252523 -0.133333 -0.091287 -0.137740 -0.252523 1.000000 #NATO The hashtags with at least 4 appearances are considered as popular hashtags. The table above shows correlation between these hashtags. The highest correlation value is between #Ukraine and #Breaking which is reasonable taking into account the current situation. Text preprocessing Tweets are not prepared for modelling. Therefore, implementing text preprocessing techniques is needed to achieve clean tweets which may be used in topic modelling. def remove links(tweet): '''Takes a string and removes web links from it''' tweet = re.sub(r'http\S+', '', tweet) tweet = re.sub(r'bit.ly/\S+', '', tweet) tweet = tweet.strip('[link]') return tweet def remove users(tweet): '''Takes a string and removes retweet and @user information''' tweet = re.sub('(RT\s@[A-Za-z]+[A-Za-z0-9-]+)', '', tweet) tweet = re.sub('(@[A-Za-z]+[A-Za-z0-9-]+)', '', tweet)return tweet my stopwords = nltk.corpus.stopwords.words('english') word rooter = nltk.stem.snowball.PorterStemmer(ignore stopwords=False).stem $\label{eq:my_punctuation} my_punctuation = '!"$%&\'()*+,-./:;<=>?[\\]^_`{|}~~``''...@#'$ # cleaning master function def clean_tweet(tweet, bigrams=False): tweet = remove users(tweet) tweet = remove_links(tweet) tweet = tweet.lower() tweet = re.sub('['+my_punctuation + ']+', ' ', tweet) tweet = $re.sub('\s+', '', tweet)$ tweet = re.sub('([0-9]+)', '', tweet)tweet = re.sub(' \(\frac{1}{2} \) BREAKING | \(\frac{1}{2} \) breaking \(\frac{1}{2} \) ', 'breaking', tweet) tweet_token_list = [word for word in tweet.split(' ') if word not in my_stopwords] tweet_token_list = [word_rooter(word) if '#' not in word else word for word in tweet_token_list] if bigrams: tweet_token_list = tweet_token_list[i]+'_'+tweet_token_list[i+1] for i in range(len(tweet token list)-1)] tweet = ' '.join(tweet_token_list) return tweet df U['clean tweet'] = df U.tweet.apply(clean tweet) #df U.to csv('df U.csv', index=False, header=True) #saving dataset for second project mentioned hashtags clean_tweet tweet is retweet retweeted battalion neo nazi paramilitari RT @kennardmatt: The Azov Battalion was a neo-Nazi paramilitary unit made unit made up of football hooligans. Soon after the "Maidan [@kennardmatt] 0 True footbal revolution" in 2... hooligan soon revolut free ukrain pull russia RT @john_sipher: "A free Ukraine will pull Russia westward if it isn't 1 [@john_sipher] True westward brought to heel." https://t.co/NjGViJWzHc brought heel presid joe biden said would consid RT @Reuters: President Joe Biden said that he would consider impos direct 2 imposing direct sanctions on President Vladimir Putin if Russia True [@Reuters] sanction invades Ukrain... presid vladimir putin russia invad ukrain top form pmq labour look sunken RT @DCBMEP: .@BorisJohnson on top form at #PMQs - Labour look behind sunken behind Starmer; who has missed his target: 'a lawyer not a [@BorisJohnson] [#PMQs] True [@DCBMEP] starmer miss Leader'. No... target 'a lawyer leader america kid european RT @DonaldJTrumpJr: Americas kids dying in a European land war land war with Russia is a small price to pay for Hunter Biden and fam to get [@DonaldJTrumpJr] russia small rich AF... price pay hunter biden fam get rich af meet russia putin RT @WriterJackWhite: As @nato meets Russia under Putin's blackmail blackmail of 100,000 troops to invade Ukraine. In 2018 after NATO [@WriterJackWhite] [@nato] troop invad meet after Salis... ukrain nato meet sali tactic terror american audienc RT @JuliaDavisNews: The tactic of terrorizing American audiences possibl 6 [@JuliaDavisNews] with the possibility of nuclear war, in order to undermine U.S. True nuclear war support fo... order undermin u support fo strang pm told parliament RT @BestForBritain: That's strange. The PM told Parliament he is the one lead [@BestForBritain] [] one leading Europe on Ukraine negotiations. He's not even in the True europ room.... ukrain negoti even room could resolv ukrain @davnrussell @ASBMilitary All this could be resolved if Ukraine [@davnrussell, remain 8 remains neutral it's really simple. No need for a N... False @ASBMilitary] neutral realli https://t.co/nTRWngCO11 simpl need strang pm told parliament RT @BestForBritain: That's strange. The PM told Parliament he is the one lead [@BestForBritain] one leading Europe on Ukraine negotiations. He's not even in the True europ room.... ukrain negoti even room cyberattack attempt RT @VentureBeat: Cyberattack was attempted against a western western 758 [] government 'entity' in Ukraine, researchers say True [@VentureBeat] govern https://t.co/6LTuSDIYdv entiti ukrain research say today grow consensu RT @djrothkopf: So today we have:\n--Growing consensus that biden team Biden team handling of Ukraine crisis has been very strong as pieces True [@djrothkopf] handl ukrain continue to... crisi strong piec continu observ locat RT @OlgaNYC1211: Just an observation. If you can't locate Ukraine, ukrain name [@OlgaNYC1211] 760 name cities in Ukraine, have no clue what has been happening there True citi ukrain clue happen statement respons us intellig RT @trussliz: My statement in response to US intelligence that Russia russia plan True [@trussliz] fabric attack plans to fabricate an attack to justify military action against Ukrai... justifi militari action ukrai casual publish best writer RT @willevans: Very casually publishing the best writers from from pakistan— 762 Pakistan—Uzma Aslam Khan—& Dikraine—@AKurkov's GREY True [@willevans] [@AKurkov] uzma aslam BEES translated by @B... khan— amp ukraine grey bee translat b hope goe brave nato troop All my hope goes to the brave NATO troops currently deployed on current 763 the border of Ukraine and Eastern Europe. While I d... False [] deploy https://t.co/hkjKoyvxUB border ukrain eastern europ russia plan fake video Russia Planned Fake Video to Gain Support for Ukraine Invasion | gain False [@Newsmax] https://t.co/RzqOdGseQG https://t.co/GZwCZwpbiQ via @Newsmax support ukrain invas via go say hold I'm going to say it again you can hold a neutral position on Russia neutral posit 765 russia us and US what I have no time for is the whatabout... False https://t.co/2Sk4ynuGFq time whatabout thread six thing RT @TimothyDSnyder: 0/6. A thread: Six Things Progressives Know [@USProgressives, 766 True [@TimothyDSnyder] [#Ukraine] progress @AOC] About Ukraine and Russia. @USProgressives @AOC #Ukraine know ukrain russia ukrain call us accus russia creat I called it\n\n"The US has accused Russia of creating a propaganda propaganda 767 video featuring crisis actors, staged fake explosi... False video featur https://t.co/AAfjJxf7V6 crisi actor stage fake 768 rows \times 6 columns In order to achieve clean tweets are applied functions which are based on removing web links, retweet and @user information, lowering the text, striping punctuation, removing double spacing, stopwords, numbers and after it is applied word rooter. The results are visible in table above. In [19]: from wordcloud import WordCloud #plt.figure(figsize=(10, 10)) long string = ','.join(list(df U['clean tweet'].values)) wordcloud = WordCloud(background color="black", max words=100, contour width=5, contour color='steelblue', width=600, height=400, min word length=2) wordcloud.generate(long string) wordcloud.to image() Out[19]: invas ukrain freez death process from imageio import imread twitter mask = imread('./twitter mask.png') wordcloud = WordCloud(background color='white', width=800, height=500, mask=twitter mask, contour color='deepskyblue', min word length=2,colormap='Blues r').generate(long string) plt.figure(figsize=(10, 10)) plt.imshow(wordcloud) plt.axis("off") #plt.savefig('./my twitter wordcloud.png', dpi=1000) plt.show() weapon SaV tension ರಾtrump first 2. biden administr border one johnson > ™troop Sendeide ukrainian vladimir putin talk go germani franc diplomat ukrain crisi anid anid anid In order to visualize results, wordclouds are used. Wordclouds are visual representations of words that give greater prominence to words that appear more frequently. The most visible words on the plots above are "ukrain" and "russia". Topic modelling Previous steps were necessary for applying topic modelling. Latent Dirichlet Allocation is used to indicate the hidden topics presented in the tweets. from sklearn.feature extraction.text import CountVectorizer # the vectorizer object will be used to transform text to vector form $vectorizer = CountVectorizer (max df=0.9, min df=10, token pattern='\w+|\$[\d\.]+|\S+')$ # apply transformation tf = vectorizer.fit transform(df U['clean tweet']).toarray() # tf feature names tells us what word each column in the matric represents tf feature names = vectorizer.get feature names() tf feature names[:20] Out[21]: ['accus', 'administr', 'afghanistan', 'also', 'america', 'american', 'amid', 'amp', 'anoth', 'appar', 'approach', 'ask', 'attack', 'biden', 'border', 'break', 'britain', 'call', 'carlson', 'chief'] from sklearn.decomposition import LatentDirichletAllocation number of topics = 5 model = LatentDirichletAllocation(n components=number of topics, random state=0) model.fit(tf) Out[22]: LatentDirichletAllocation(n components=5, random state=0) def display topics(model, feature names, no top words): topic dict = {} for topic idx, topic in enumerate(model.components): topic dict["Topic %d words" % (topic idx)]= ['{}'.format(feature names[i]) for i in topic.argsort()[:-no top words - 1:-1]] topic dict["Topic %d weights" % (topic idx)]= ['{:.1f}'.format(topic[i]) for i in topic.argsort()[:-no top words - 1:-1]] return pd.DataFrame(topic dict) In [24]: no_top_words = 10 topics = display topics(model, tf feature names, no top words) topics Out[24]: Topic 0 Topic 0 Topic 1 Topic 1 Topic 2 Topic 2 Topic 3 Topic 3 Topic 4 **Topic 4** words weights words weights words weights words weights words weights 0 ukrain 96.8 ukrain 137.0 ukrain 76.2 ukrain 127.0 ukrain 157.0 1 biden 64.2 russia 56.5 russia 43.7 russian 74.9 russia 96.5 2 border 59.4 putin 27.8 31.9 invas 40.3 62.7 qo 3 presid 44.9 23.2 26.2 35.0 51.3 troop support germani russia 4 36.5 16.4 biden 24.7 33.5 34.1 putin invad war say 5 35.9 16.2 talk 22.9 militari 29.2 nato 31.1 send ask 6 34.2 ukrainian 16.2 border 20.0 20.7 crisi 29.2 peopl plan 7 want 23.5 secur 15.9 would 19.9 need 20.2 28.8 war 8 million 21.2 14.0 17.8 18.3 biden 26.0 europ new us 17.2 13.2 week 15.2 16.7 25.1 joe meet amp for i in range(0,len(topics.columns),2): topic zip = dict(zip(topics.iloc[:,i],pd.to numeric(topics.iloc[:,i+1]))) wordcloud = WordCloud(width=600, height=400).fit words(topic zip) plt.figure(figsize=(5,5)) plt.imshow(wordcloud) plt.axis("off") plt.tight layout() plt.title('Topic '+str(int(i/2)), size=20) plt.show()

