Extracting news trends using Twitter API

Nikita Alexandrov 583763, Tommi Vehviläinen 439613

April 25, 2018

1 Introduction

Social media has dramatically changed the way how people communicate. This change does not apply only to communication between individuals, but also companies and even states are nowadays expected to provide customer support or maintain public relations via social media. Especially Twitter has established itself as a platform where companies, states, politicians and usual Internet users share the news, publish their opinions and have heated discussions. That's why the analysis of tweets can be useful for extracting current mood, following people's opinions.

For this project our initial plan was to gather tweets related to the 2018 presidential elections in Finland and Russia and compare both the candidates within the countries and the tone of discussion related to politics. However, we found out that the Twitter API we were planning to use does not allow us to download tweets which were published more than a week ago. Twitter has a Developer API without such restrictions, but it would take months of waiting in order to receive access to this API. That's why we decided to change our topic to something very similar that includes both politics and analyzing tweets. For this project we have collected English and Russian tweets related to political topics using a set of search words which we found appropriate.

The report is structured as follows. Section 2 describes how we obtained data and explains the pre-processing steps we have done. We explain what kind of data we got and potential problems related to the methods we have used. In Section 3 we explain how we trained the vector embedding models. Due to our data being highly related to specific topics, we found conventional evaluation methods, such as similarities for common words or country-capital relations to not work well. However, we found the solution to this problem. This approach is discussed in Section 4. Best models according to our scoring were visualized on 2D plane. The received results are impressing: even if we use only 10 words for Twitter search, it is possible to see political trends around the world. Finally, we will summarize our results in Section 5.

2 Data

We collected our data using a freely available Twitter API using specific set of keywords to find tweets that match our specification. The limited throughput of the API resulted in relatively modest amount of tweets. We queried for tweets that have been tweeted on 16-04-2018 or earlier using the specified keywords. We gathered 10000 tweets corresponding to each English keyword. After preprocessing this number was slightly reduced. The amount of tweets in Russian is similar.

2.1 Keyword selection

The first question that we had to solve was deciding what kind of tweets we want to investigate. Although the question seems simple, it is very challenging: dozens of new news are being published every day. Therefore, it is possible to use hundreds of words for search in tweets. However, it was concluded that this way wouldn't be efficient: number of tweets will increase significantly and building the models would be time-consuming. Also, the freely available Twitter API would not allow us to do that within reasonable time. That's why we decided to concentrate on a set of 10 words to be used for search. The rationale of these words is trivial: all words from this list (excluding 'skripal') have been appearing in news titles already years. The list of used **keywords** is:

'syria', 'damascus', 'assad', 'politics', 'putin', 'trump', 'russia', 'usa', 'skripal', 'uk'.

2.2 Preprocessing

When we began downloading the tweets, we noticed that many of them repeats themselves. This problem was easily identified and solved by removing the duplicates. We also noticed that most re-tweets are re-tweets of some single popular tweet. These duplicates would not contribute to our task so we decided to ignore all re-tweets and focus only on tweets that are created by the users themselves.

The next step was text normalization. In case of Russian language, it was very simple: we deleted all non-Cyrillic symbols from tweets, stop-words (gensim has a list of them, but this list was complemented by one of authors after pre-view of normalization) were also excluded.

In Russian tweets removing redundant tokens such as user handles and hyperlinks were taken care of by removing the non-Cyrillic symbols. This process had to be done separately for English tweets. The tweets were first made lowercase and tokenized. Then hyperlinks, user handles and common English stopwords were removed. Since hashtags are commonly used to replace words that are highly related to the topic at hand, we made a deliberate decision not to remove them from the tweets but rather normalize them into words. This decision might introduce some new words in the vocabulary that are derived from hashtags. Finally, the remaining words were lemmatized. Due to no entity recognition, our embedding won't be able to detect entities such as United

Kingdom or Donald Trump but rather consider each string as individual token.

After completing all steps, we joined all words together and calculated MD5 sum for each normalized tweet, so if a tweet has a hash that hasn't appeared in our map, we add this record to the final list of tweets and only after that, we are ready to build models.

3 Methods and experiments

3.1 Pre-trained model

The existing word2vec embedding of our choice was a model that has been pre-trained using Google News corpus of 3 billion running words [1]. Since our main focus is on the political topics, we decided to experiment with our keywords used for querying our training data. The most similar words to the keywords can be seen in Table 1.

Syria	**Damascus**	**Assad**	**Politics**	**Putin**
Syrian	Syria	President_Bashar_Assad	partisan_politics	Medvedev
Syrians	Syrian	al_Assad	Politics	Vladimir_Putin
Iran	Beirut	Mubarak	political	President_Vladimir_Putin
Damascus	Tehran	Bashar_Assad	politcs	Prime_Minister_Vladimir_Putin
Hezbollah	Assad	Syrian	poltics	Kremlin
Hizbullah	Teheran	President_Bashar	Lisa_Vorderbrueggen_covers	Lukashenko
Egypt	Cairo	Syria	partisanship	Nazarbayev
Lebanon	Syrians	Siniora	politicians	Saakashvili
Israel	Lebanon	Damascus	politician	Lukashenka
Hizbollah	Amman	Abbas	politicking	Yushchenko
Trump	**Russia**	**USA**	**Skripal**	**UK**
Donald_Trump	Ukraine	lifts_Squaw_Valley	Sergei_Skripal	Britain
impersonator_entertained	Moscow	Mobility_NASDAQ_USMO	Gennady_Vasilenko	United_Kingdom
Ivanka_Trump	Russian	lifts_Sugarloaf	Gennady_Vasilenko_former	UKs
Ivanka	Belarus	proudly_proclaims_Made	Zaporozhsky	British
mogul_Donald_Trump	Kremlin	World_Premiere_Narrative	Herman_Simm	Britains
Trump_Tower	Kazakhstan	Mobility_Sets_Date	businessman_Andrei_Lugovoi	Wiley_Chichester
Kepcher	Russians	MONTVALE_NJ_Mercedes_Benz	Zaporozhsky_Igor_Sutyagin_Gennady	Hassan_Mirza_Gay.com
billionaire_Donald_Trump	Biologist_Anatoly_Kochnev	Heavy_Duty_Waterproof_Flashlight	Alexander_Zaporozhsky	Britian
Trumpster	Azerbaijan	U.S.A.	exchanged_Sergei_Skripal	Rob_Burman_IGN
tycoon_Donald_Trump	Putin	L._Kasuske_Seattle	Gennady_Vasilenko_Sergei_Skripal	Great_Britain

Table 1: Words that are most similar to our keywords according to pre-trained Google News embedding

The model seems to map our keywords so that most of the most similar words are semantically meaningful. For example, the similar words for countries are institutions in that specific country or other countries that are somehow similar to this country. Similarly presidents seem to have multiple synonyms and they are related to their countries. However, most of the words do have a similar word that is not meaningful such as Lisa_Voerderbruggen_covers, Heavy_Duty_Waterproof_Flashlight, lifts_Sugarloaf or Hassan_Mirza_Gay.com. It is not immediately clear why these words appear as similar words but it might be due to the nature of the training data. In addition to that, it is clearly seen that model was trained on old data and it doesn't reflect the current situation (Trump isn't a president yet, Syria has more geographical context and Putin is still mentioned as a Prime Minister).

However, the kind of semantic meaningfulness captured by these embeddings is not what we are looking for. Our goal is to extract relations within a broader semantic window. The inability of the pre-trained model to do this is mainly due to the training data that is not suited for this specific task but also large amount of data and minimal pre-processing might play a role.

3.2 Learning embeddings from tweets

The task of building our own word2vec models [2] consisted of using the pre-processed English and Russian tweets and experimenting with different parameters. Our aim was to try out a reasonable number of parameters, see what is the effect of changing the values and find a good but not necessarily optimal embedding.

For the English tweets we implemented a grid to find all possible values in the defined parameter space. The possible parameter values in this experiment were output dimensions 50, 100 and 200, window sizes 2, 5 and 8 for CBOW and 5, 10 and 15 for skip-gram model and minimum word counts 5, 50 and 100. We used these parameters for both CBOW and skip-gram models which results in 54 different embeddings in total. We also briefly experimented with downsampling parameters of 1e-5 and 1e-3 but they consistently decreased the performance.

For Russian language model we tried a little bit more models:

- Output dimension: [10, 50, 100, 200, 500, 1000];
- Window size: [1, 2, 3, 5, 7];
- Minimal word count: [0, 25, 50, 75, 100];
- Two different algorithms: skip-gram and CBOW.

We ran each configuration 4 times in order to calculate the average model score and reduce the effect of outliers.

4 Results

In the evaluation phase our aim was to select a suitable embedding from the set of candidate embeddings. We will then further experiment with these embeddings to better understand how they work.

Our initial conception was that since the tweets focus on rather limited area of topics, the embeddings would not do well in those tasks that are commonly used to evaluate more general models. Brief experimentation with common nouns agreed with this conception: for example, dog and cat have many unrelated neighbors whereas apple is closer to iphone than any other fruit.

Despite the specialties of our training data, we evaluated the model for English tweets using the set of analogical reasoning task provided in the course folder. These tasks consist of finding a relation between a known object and an unknown target when similar relation is given as an example. Such task could be, for example, finding an embedding that is to Finland what Stockholm is to Sweden. The list of these tasks include such country-capital relations, relations between countries and their currencies and multiple kind of grammatical reasoning tasks. In total there were 19544 such tasks. We evaluated the models by giving score 1 for each task where the correct word was mentioned in the list of 10 closest embeddings and 0 otherwise. The architecture of the best and the worst model are described in Table 2 along with the corresponding scores.

architecture	output dimension	window size	min count	score
skip-gram	50	5	5	230
CBOW	200	8	100	57

Table 2: The best and worst model according to analogical reasoning task

We also came up with our own analogical reasoning task which we expect to be more suitable given the nature of our training data. Rest of this section will focus on this evaluation metric and the results we got.

4.1 Evaluation metric for our data

Since the tweets are related to politics, we were not expected the embeddings to perform well on general reasoning task. In fact, performing well on such general task would likely conflict with our goal of relating words to those contexts in which they are used. Therefore we came up with our own evaluation metric which is similar to the analogical reasoning tasks described before but the relations are between countries and presidents. We chose the countries so that they are relevant to the keywords and we expect them to occur in the tweets. In other words, we expect a good embedding to be able to perform well in this reasoning task. The list of pairs is shown in Table 3. Similar set of pairs was used to evaluate tweets in Russian.

country	president
russia	putin
america	trump
syria	assad
turkey	erdogan
japan	abe
france	macron
britain	may
germany	merkel

Table 3: Pairs of countries and presidents that are used in our evaluation metric

From each two pairs we constructed four different reasoning task. In total, this resulted in 112 reasoning tasks. To measure the performance on these tasks we consider 10 most similar embeddings and give the model score of $\frac{1}{k}$ if the kth embedding corresponds to the target word. This scoring scheme is somewhat arbitrary since it doesn't consider the absolute distances but rather just the rank. However, as we will see, the models selected using this metric perform reasonably well.

4.2 Evaluating the embeddings

To evaluate the embeddings we used the metric we just described. The best-performing model configurations, average scores and number of solved equations for English embeddings are provided in Table 4. The similar information for Russian embeddings is described in Table 5. For comparison we evaluated the pretrained model using this evaluation metric and it received a score of 37.8.

vector dim	window size	min count	algorithm	average score	# solved equations
50	15	50	skip-gram	40.3	63.25
50	10	50	skip-gram	35.0	62.75
50	15	5	skip-gram	28.3	38.75

Table 4: Best-performing models on English tweets

vector dim	window size	min count	algorithm	average score	# solved equations
10	1	50	skip-gram	22.412	51.5
200	7	50	skip-gram	21.6	39.75
100	7	50	skip-gram	18.819	36

Table 5: Best-performing models on Russian tweets

As can be seen, the best models were generated using skip-gram algorithm. Also, it seems to be important to remove rare words by using an adequate min count value. This can be seen in the tables: only the third embedding in Table 4 has low min count value. Surprisingly the dimensions of the embeddings vary a lot. For English words-embeddings 50 dimensions seem to work well, but for Russian embeddings dimensions 10 and 200 result in almost the same score. In Russian embeddings we can see that window size increases as the dimensionality grows. Our reasoning is that when more components are being to the vectors, more context must be provided to fill these vectors with meaningful information. Also, skip-gram algorithm seems to work much better for our data. Not only are all the best embeddings using skip-gram, but also most of the worst embeddings were using CBOW algorithm. We believe that this is due to CBOW performing worse on infrequent words. In addition, skip-gram performs better on small datasets such as ours. Due to the size of our dataset, considering the performance is not really necessary.

To verify the quality of the embeddings and to see how well our evaluation metric works, we have also chosen the some of the best and the worst models according to our evaluation metric and experimented with word similarities. We chose to find words that are similar to the keywords used to query the training data since there should be enough data to learn meaningful embeddings for these words. The ten most similar words to the keywords for the best three models can be seen in Table 6 and the similar words for the worst three models can be seen in Table 7. Similar words for the best and worst Russian word-embeddings can be seen in Tables 8 and 9.

syria	**damascus**	**assad**	**politics**	**putin**	**trump**	**russia**	**usa**	**skripal**	**uk**
				dim = 50, window size = 15, min count = 50, algorithm = skip-gram					
west	capital	syrian	religion	russia	republican	russian	99	bz	france
syriaairstrikes	rubble	punish	political	buddy	one	putin	mexico	swiss	britain
syriastrike	raw	syria	age	again	impeachment	kremlin	8	salisbury	theresamay
punish	square	civilian	relevant	russian	potus	u	com	poisoning	accuses
assad	yesterday	weapon	idea	vlad	president	coming	free	produced	k
u	night	chemical	platform	puppet	mueller	syria	hot	motive	fr
syrian	14	basically	character	vladimir	democrat	diplomat	spain	poisoned	skripal
chemicalweapons	precise	attack	mix	kremlin	even	prepared	worldwide	uk	boris
vow	explosion	regime	college	would	him	prepare	drama	evidence	salisbury
global	playlist	butcher	topic	west	gop	ending	sport	agent	embassy
				dim = 50, window size = 10, min count = 50, algorithm = skip-gram					
syriaairstrikes	missionaccomplished	syrian	relevant	russia	him	putin	free	bz	france
syriastrike	capital	syria	religion	again	impeachment	russian	mexico	swiss	fr
assad	syriaairstrikes	weapon	nasty	buddy	he	kremlin	spain	salisbury	britain
west	yesterday	punish	ignorance	vladimir	crooked	u	x	motive	theresamay
u	raw	nikkihaley	always	vlad	comey	syria	steel	poisoned	spain
syriastrikes	rubble	civilian	character	russian	office	diplomat	italy	lavrov	skripal
punish	syriastrike	regime	culture	mess	crook	attack	amazon	poisoning	accuses
rouhani	night	basically	political	intention	rant	coming	drama	recent	k
chemicalweapons	14	absolutely	reading	puppet	realdonaldtrump	prepare	99	produced	foiled
vow	saa	genocide	aside	warn	even	possible	mix	agent	poisoned
				dim = 50, window size = 15, min count = 5, algorithm = skip-gram					
assad	capital	syria	gender	russia	appointee	russian	cinema	skripals	france
unitedstates	syriabombing	syrian	religion	buddy	impeachtrumpnow	putin	tshirt	novichok	itvnews
foresees	reduces	retaliating	political	hmmmm	muellerinvestigation	diplomat	sofary	bz	fra
basnews	hamza	regime	appreciate	dangerously	notmypresident	kremlin	hiphop	salisbury	fr
vladimirputin	syriastrike	inhumane	practiced	again	loudobbs	rf	tab	toxin	pmqs
syriaairstrikes	missionaccomplished	supplying	race	russian	muellertime	reutersus	freestyle	swiss	bbcqt
basharalassad	decry	unaffected	prosperity	kremlin	one	glove	mvp	substance	votelabour
foreignpolicy	rubble	fortunately	lecture	glove	sank	provoking	newyorkcity	yulia	fukus
déjà	syriastrikes	chemical	disappointing	teheran	resister	promising	disco	spiez	marilynlavala
vía	assadmustgo	insist	bigotry	poo	impeachtrumppence	vodka	beach	poisoning	reacts

Table 6: Words that are most similar to the keywords according to the three best-performing English models

syria	**damascus**	**assad**	**politics**	**putin**	**trump**	**russia**	**usa**	**skripal**	**uk**
coalition	wall	ability	liberal	russian	realdonaldtrump	israel	home	produced	britain
diplomacy	explosion	change	folk	russia	losing	russian	mexico	nerve	k
planned	city	isi	respect	monster	obama	hezbollah	free	agent	england
condemn	empty	iranian	leftist	relationship	j	turkey	australia	deadly	ally
following	capital	islam	business	donald	elected	nuke	car	poisoning	germany
tn	building	however	crazy	enemy	graham	north	canada	alleged	london
response	area	brutal	game	ally	idiot	korea	service	suspected	intelligence
dropping	yesterday	him	amazing	israel	potus	economic	africa	confirmed	push
allied	ghouta	continue	ur	iran	him	phone	driver	cw	british
empty	future	dictator	political	master	joe	saudi	eu	purpose	n
diplomacy	explosion	ability	liberal	russian	realdonaldtrump	israel	mexico	nerve	britain
coalition	yesterday	change	folk	monster	losing	russian	car	produced	k
ally	city	dictator	sad	behavior	idiot	hezbollah	australia	agent	england
planned	capital	iranian	political	enemy	obama	turkey	free	deadly	ally
condemn	coalition	anyone	crazy	everything	graham	saudi	africa	alleged	germany
humanitarian	ghouta	hard	cut	russia	bos	directly	canada	suspected	british
recent	syrianstrikes	brutal	mouth	consequence	j	phone	driver	sarin	intelligence
response	empty	him	lmao	how	resign	diplomacy	india	cw	n
basically	wall	saddam	amazing	saddam	vladimir	corbyn	travel	hedge	push
siria	ruin	rebel	game	iran	joe	arabia	service	confirmed	israeli
coalition	explosion	ability	folk	monster	realdonaldtrump	israel	mexico	nerve	britain
planned	saa	change	liberal	how	losing	turkey	service	agent	k
following	area	basically	mouth	graham	graham	directly	car	deadly	england
condemn	city	continue	piece	iran	potus	hezbollah	australia	alleged	british
diplomacy	yesterday	rebel	history	trouble	bernie	saudi	canada	produced	germany
saa	ruin	civilian	short	who	him	syriastrike	home	suspected	intelligence
tn	capital	brutal	business	erdogan	idiot	arabia	art	cw	ally
recent	turned	iranian	attention	mob	obama	russian	india	hedge	n
syriastrike	coalition	others	racism	resign	bos	korea	free	chemicalweapons	london
wwiii	syrianstrikes	there	more	israel	resign	germany	eu	poisoning	israeli

Table 7: Words that are most similar to the keywords according to the three worst-performing English models

The most similar words certainly are semantically similar to the keywords. For example in the best model Syria is similar to Assad, Trump is similar to republican and president and Russia is similar to both Putin and Kremlin. As we predicted earlier, the model has introduced new words that correspond to hashtags such as syriaairstrikes, realdonaldtrump, missionaccomplished and impreachtrumppence.

Also the worst models seem to have some level of semantic connection to the keywords but the most common words are somewhat vague and are not necessary the best candidates for the most similar words. However, there are some exception such as britain which is the most similar word to uk in all of worst models but is not included in any of the best models. The evaluation metric we have used tested only one type of relation so it was expected that the best models according to our metric might not give us the globally best embeddings.

syria	**damascus**	**assad**	**politics**	**putin**	**trump**	**russia**	**usa**	**skripal**	**uk**
dim = 10, window size = 1, min count = 50, algorithm = skip-gram									
strike	homs	bashar	main	hitler	donald	continue	ready	sergei	great britain
sara	al	pentagon	development	mister	personally	attempt	jointly	daughter	explain
massive	building	high	terrorism	solovyev	order	go	intend	post	usa
join	shoot	estimate	situation	personally	medvedev	deripaska	britain	lavrov	france
coalition	tomahawk	hit	political	insist	mister	enter	immediately	sister	jonson
air strike	shairat	french	fight	earn	decide	revolution	join	groom	nato
bombing	bomb out	also	sport	komi	cooperate	cyber-espionage	france	father	immediately
reflection	province	again	uzbekistan	armenian	helper	end	plan	skripalyi	jointly
fall into	photo	cooperate	growth	compare	want	kremlin	great britain	ha	intend
warn	vvs	match/equal	behave	oh	poroshenko	background	begin	employee	membership
dim = 200, window size = 7, min count = 50, algorithm = skip-gram									
massive	bombing	bashar	political	vladimir	donald	rf	russianspring	skrifall	great britain
coalition	shoot	destroy	complicated	solovyev	personally	open	america	daughter	england
ally	al	child	important	tsar	want	win	earlier	victoria	london
air strike	building	syria	comment	phone	match/equal	supply	american	julia	distribute
russianspring	homs	syrian	politician	listen	punish	open	newsyandex	sister	clue
syrian	massive	dictator	avoid	evening	america	car	jointly	father	british
saturday	bomb out	camp	step	national	recently	building	massive	groom	brit
reflection	syrian	infrastructure	west	rule	see	frame	threaten	appeal	again
sara	guta	peacemaker	important	suffice	president	actual	further	gru	grounds
air shoot	saa	recovery	main	choose	again	radio	connect	scotland	ostensibly

Table 8: Words that are most similar to the keywords according to the three best-performing Russian models

Looking at the table, it is possible to claim that model with configuration (200, 7, 50, skip-gram) works better, since it is able to connect first name to family name for presidents and also find "great britain" (it is one word in Russian), "england" for "uk", "america" for "usa" and "rf" (shorten name of Russian Federation) for "russia".

As a comparison, we can output the worst model according to our metric.

syria	**damascus**	**assad**	**politics**	**putin**	**trump**	**russia**	**usa**	**skripal**	**uk**
dim = 1000, window size = 7, min count = 0, algorithm = cbow									
rocket	bastion	daesh	lnr	modelling	gundyaev	ukriya	crushing	bias	great britain
strike	reflect	kpg	donbass	x	beast	embargo	damage	op	nebenzya
modest	goltz	target	region	extrusion	nikko	calendar	iran	interesnya	terroristic
be applied	firing	waddington	tugushevo	adhere	announce	humanity	sharpen	examine	tkrrorist
satisfaction	antiaircraft	airstrike	wrecking	disappoint	reach	input	broadcasting	skri(bust)	sara

Table 9: Words that are most similar to the keywords according to the three worst-performing Russian models

As it was predicted, removing a limit for minimal count of word occurence and applying CBOW produces very rare words (which are difficult to translate into English :)) and doesn't make any sense.

4.3 Visualizing the embeddings

After finding the best embedding models, we have projected the embeddings into two-dimensional plane to visualize them. First we have normalized the word vectors into unit vectors. After that, we reduced the dimensionality of the vectors to 2 using t-SNE (t-Distributed Stochastic Neighbor Embedding). We first projected the high-dimensional embeddings into 15 dimensions using PCA and then applied t-SNE to get the final two-dimensional visualization.

Visualization helps to understand how good the model is. If we form some clusters (words related to one subtopic), they should be grouped together on 2D plane. To evaluate how our embeddings handle such clusters, we create the following clusters and plot them:

- Russian cluster: ‘putin’, ‘vladimir’, ‘russia’, ‘rf’, ‘russian’, ‘kremlin’, ‘oligarch’, ‘medvedev’, ’moscow’;
- American cluster: ‘trump’, ‘usa’, ‘donald’, ’state’, ‘america’, ‘american’, ‘nato’;
- Syrian cluster: ‘assad’, ‘bashar’, ‘attack’, ‘chem’, ‘weapon’, ‘rocket’, ‘explosion’, ‘victim’, ‘raid’, ‘die’, ‘force’, ‘homs’, ‘duma’, ‘shoot’, ‘target’, ‘bomb’;
- Skripal’s case: ‘solsbery’, ‘skripal’, ‘novichok’, ‘poisonning’, ‘poison’, ‘agent’, ’substance’;
- European cluster: ’may’, ‘teresa’, ‘macron’, ‘france’, ‘germany’, ‘merkel’, ‘eu’, ‘britain’, ‘uk’, ’europe’.

However, it will be also interesting to analyze other topics and trends that we didn’t look for before:

- Economic cluster: ‘money’, ‘rouble’, ‘dollar’, ‘economic’, ‘business’, ‘market’, ‘financial’, ‘money’, ‘corruption’, ‘bank’, ‘finances’, ‘oil’, ’товар’, ‘currency’;
- Blocking of Telegram: ‘durov’, ‘telegram’, ‘blocking’, ‘rkn’, ‘roskomnadzor’, ‘block’, ‘attempt’, ‘internet’;
- North Korean cluster: ‘kim’, ‘jong’, ‘un’, ‘korea’, ‘north’, ‘dprk’, ‘nuclear’;
- Ukrainian cluster: ‘ukraine’, ‘crimea’, ‘kiev’, ‘donbass’, ‘poroshenko’;
- Protests in Armenia: ‘armenia’, ‘sargsyan’, ‘protest’, ‘street’, ‘yerevan’, ‘revolution’, ‘action’;
- Positive words: ’strong’, ‘good’, ‘great’, ‘win’, ‘friend’, ‘clever’, ‘truth’, ‘help’, ‘defend’;
- Negative words: ‘weak’, ‘bad’, ‘shameful’, ‘destroy’, ‘enemy’, ‘stupid’, ‘lie’, ‘strange’, ‘aggressor’, ‘propaganda’.

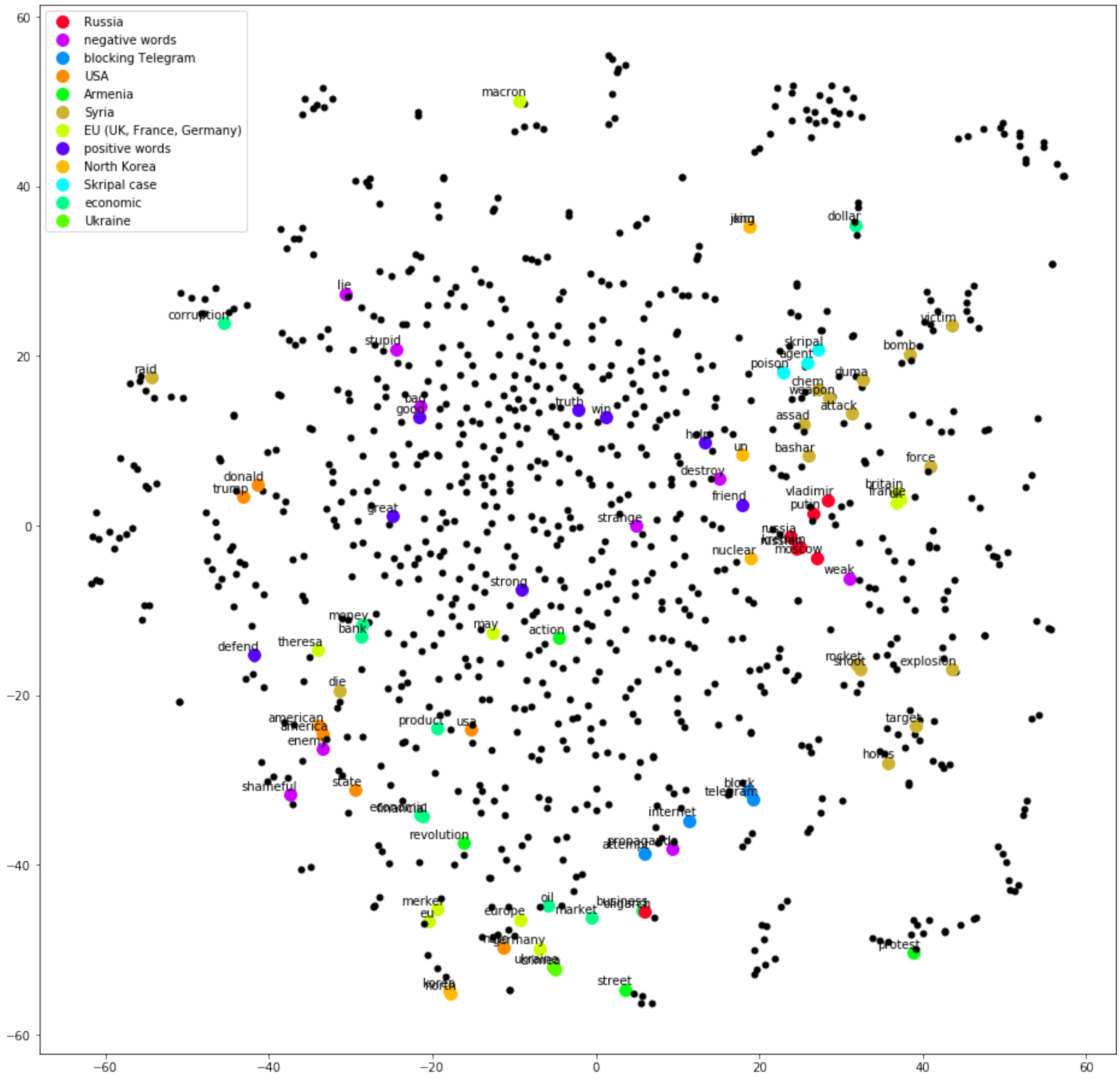

Figure 4.1: 2D visualizing of the best model (English language)

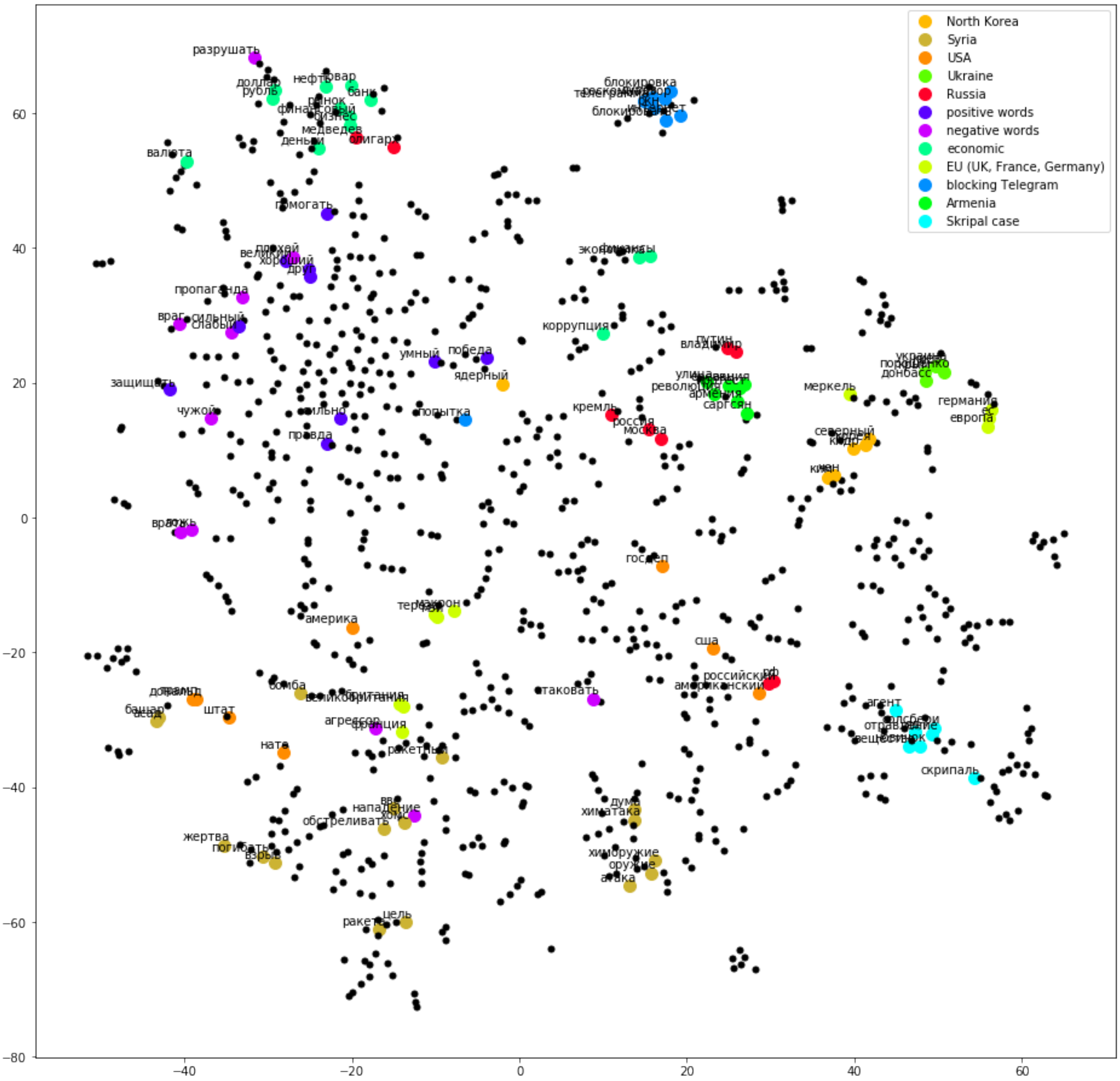

Figure 4.2: 2D visualizing of the best model (Russian language)

On the example of Russian language model, it is possible to see that even with the limited number of keywords, we can find clusters that weren't related to keywords. At the same time, English language model extracts cluster a bit worse. We suggested that English language tweets are less "political" than Russian twitter: it is clearly can be seen, if we check the most similar words for 'america', we will see that they don't have anything common with politics. The absence of some clusters for English language model can be explained by saying that these clusters aren't so relevant for English-speaking audience and they aren't discussed in Twitter.

5 Conclusion and comparison

In this project we have gathered English and Russian tweets and pre-processed them so that they are suitable for building an embedding model. We then experimented with different parameters and model architectures and evaluated these embeddings. As a result, best architecture for word2vec model is skip-gram. Also, it was shown the importance of extracting very rare words. For evaluation we tried using more general approaches, but decided to create our own evaluation metric which we expected to be more in line with our training data. Finally we projected the word-embeddings into two-dimensional space for visualization and they seem to form clusters that correspond to certain topics.

6 Appendix

6.1 Code

All code materials can be found here:
https://github.com/alexandrov-nikita/StatisticalNLP/tree/master

6.2 Sample examples

9,"@seanhannity There's no difference between Joy Behar , Larry or Curly or Shemp for that matter . Why doesn't anyone even give her the time of day . Stupid comments by a stupid woman . Same as Rachel Madcow theorizing that Trump bombed Syria to divert attention from the Stormy Daniels scandal."

10,@Flowerstoall Syria accomplished little. Hang in there! New week tomorrow

11,"@exoticgamora @DanaScottLO @mcspocky @wesley_jordan @tizzywoman @TrinityResists @WomanResistorNC @AynRandPaulRyan @JCTheResistance @anti_ orange1 @TaggartRehnn Syria is a deeply vexing problem with no easy answers for any real POTUS who actually cares about outcomes for those in distress, and America's standing in the world. For Drumph, it's just part of his reality show."

12,French president brags that he convinced Trump to strike Syria as the US was poised to pull out https://t.co/7JpuIXFk7c

13,"""Shit's been going down in Syria for 7 years. Who has the answer to this conflict? This man himself. Ernesto, 41 years old, teaches sociology and has a YouTube channel. Smokes weed and lives with his mom."" https://t.co/BK6yArGpaI"

14,U.S. troops not leaving Syria until goals accomplished: Haley https://t.co/9lkLYTEeYP

15,"Because when you think of cutting edge cancer research and drug creation, you naturally think Syria!!! It would have been better if they stuck to the baby milk manufacturing plant.... https://t.co/bkIdXDTdhq"

16,"""if i was putin i would sell some 8 regiments of s300 to syria, iran and nk, just to watch the. . . "" — Claudio Cadalço https://t.co/hYBFBp2DNB"

References

[1] Google. *word2vec*. 2013. URL: https://code.google.com/archive/p/word2vec/.

[2] Mikolov Tomas et al. "Distributed Representations of Words and Phrases and their Compositionality". In: (). URL: https://arxiv.org/pdf/1310.4546.pdf.