A quest to identify Russian propaganda online

Yevhen Barshchevskyi

Background

The ongoing war on the East of Ukraine has created a fertile ground for spreading non-factual information or propaganda about actual situation on the ground. The sole nature of conflict always leads to disinformation[1] campaigns been pursued by sides of a conflict. For example, the Serbian and Croatian governments used then media channels to justify or incite violent actions during the Yugoslav wars in 1990's [1]. The USA propagated a fake claim that Iraqi government possessed weapons of mass destructions as a pre-text for incursion back in 2002 [2]. However, the recent developments in communication means and uncontrolled nature of Internet enormously accelerated the speed with which disinformation reaches out to the target audience. A striking example is fake news that heralded "Pope Francis shocks world, endorses Donald Trump for president" had more than 960k engagements within a couple of hours [3].

While Pope's fake is harmless in a sense, most of the fakes/propaganda connected to the recent Russian expansionism in Europe is far more acute. For instance, the fake news about German girl being allegedly raped by the Syrian refugees in 2016 subsequently led to mass protests against the German security forces. The fake plot was ultimately debunked by German government after several days of investigation. Nevertheless, it was too late since disinformation produced by Russian media proxies in Europe had already exploited the severe social cleavages in the German society, i.e. immigration issue.

Rationale

Given the further developments of communication technologies and Russian information warfare, the quest for automated propaganda identification tool has become more important than ever for CIS countries. As Cohen et al. [4] have suggested computation may hold the key to far more effective and efficient fact-checking. Fully or semi-automated disinformation classifier could serve as an entry point for investigative journalists, who search through tons of information daily in an attempt to rebuke fakes. It could also help citizens to sift information they get from online news or social media.

Hypothesis

Given the recent research of applying machine learning to solve fake news dilemma [4,5,6], there is no coherent mathematically or statistically-driven approach to infer if actual news piece is supported by valid facts or references (the so-called knowledge-based approach). Automated fact-checking requires a repository with already proof checked statements that has a coherent text structure and consistent in the long run [7]. Instead, one of the focus of research lies in text

[1] To omit disambiguation and for the purpose of this study, words "propaganda", "disinformation", "fake news", "fakes" are defined as "printed or online information material aimed at intentionally misleading the reader or creating a distorted image of the recent events''.

structure information retrieval, i.e. words frequency distribution, sentiment analysis, special characters' frequency calculations, etc.

Goal

An overarching goal of this research paper lies in approbating hypothesis that states that fully automated natural language classifier is an effective tool for discerning propaganda narratives in the online media. The author formulates two prior assumptions to support the research objective:

1) Machine learning model could serve as a baseline instrument for searching and revealing propaganda infused news;

2). The aforesaid task could be achieved by constructing model features out of the text patterns.

Data collection

I constructed a labeled dataset of news/articles from the Ukrainian and Russian media outlets. 30% (n=165) of dataset consists of the Russian propagandistic news that were rebuked by the Ukrainian fact-checker "StopFake". The Ukrainian dataset (n=318) consists of news from the reputable Ukrainian media - "Ukrainska Pravda" and "Ukrainian Independent News Agency (UNIAN)". I made a "dummy" assumption that the Ukrainian news are fact-driven (though it is highly disputable among Ukrainian media community). All news was written in Russian language.

Propaganda news was scrapped from "StopFake" web-site, using Python web-scrapper (see "StopFake_web_scrap" iPython notebook for reference). For non-propaganda news, third party applications were used to scrap news with tags "Donbas", "ATO (Anti-Terroristic Operation)", "War" and "Minsk agreements".

To simplify training process, I deliberately limited subject domain to news related to war on the East of Ukraine and Minsk agreements. Since news published by the Ukrainian media already matched in-domain limitations, I had to randomly drop 831 instances to balance dataset[2]. For the Russian propaganda news, I have read all the news pieces, making a judgement call in classifying the latter. Thereby, I got 483 labeled news in total. Moderate skew towards non-propaganda news leads to the imbalanced classification problem with a focus on optimization for true positive scores. A detailed description of the final dataset is provided in Table 1.

Table 1. Dataset description

Number of news instances	Average number of tokens per instance	Class	URLs
165	2440	Propaganda	http://ukraina.ru http://ren.tv https://www.sovsekretno.ru

[2] This sample was used to test classifier performance in the "real-life" scenario.

			http://rian.com.ua http://izvestia.ru http://tvzvezda.ru https://lenta.ru https://ria.ru http://www.mk.ru https://www.vesti.ru http://www.politonline.ru http://dnr24.com http://www.politnavigator.net http://www.vz.ru http://tass.ru http://informator.lg.ua http://rusvesna.su http://www.kp.ru https://www.gazeta.ru http://itar-tass.com http://top.rbc.ru https://www.1tv.ru http://www.bk55.ru
318	1175	Non-propaganda	http://www.pravda.com.ua/ https://www.unian.ua/

Feature extraction

Terms frequency: I used Python scikit-learn library to build terms frequency matrix (TF) that contains 1796 features (including unigrams, bigrams and trigrams). TF matrix includes only words that are lemmatized using Snowball stemmer algorithm for Russian language. I also excluded words that do not bring real content value (the so-called "stopwords").

Initial idea lay in creating TF-IDF (terms-frequency inverse document frequency) matrix that proved to be a state-of-the-art model for text classification. After training a few test models, I understood that simple terms frequency on average perform better because news comes from the same domain. Thus, main features were constructed as simple terms frequency.

N-grams: Given the enormous predictive power of N-grams[3] in classifying Jihadist messages in Twitter [8], I created several variations of word n-grams, particularly unigrams, bigrams, and trigrams. I have ended up with using all of them since it is proved to be useful in catching words sentiment differences.

Propaganda sensitive words: Propaganda generally exploits different word clichés that are intent to make users believe in fake story or narrative. In the context of war in Donbass, one can immediately see patterns if a similar word or word combinations occur in a range of propagandistic news. On

[3] N-grams - In the fields of computational linguistics and probability, an **n-gram** is a contiguous sequence of **n** items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application.

the other hand, subject affected by propaganda tends to develop a similar counter-propaganda vocabulary. This strategy ultimately leads to words polarization that could be easily retrieved from text. For example, the Russian media always report about "rebels fighting for the autonomy against Kyiv regime", whereas Ukrainian media says that "terrorists in the ATO zone continue firing from weapons banned by Minsk agreements". Therefore, I tried to come up with vocabulary of propaganda-sensitive words. One created a dummy features' vectors for each of the word, i.e. if a word is present in news instance, the value equals 1, else 0. I ended up having 18 features in total (10 for propaganda and 8 for non-propaganda).

Digits and special number frequencies: According to several research findings of fake news classification [7], manipulative or hoax news tends to have few statistical or number references. I also hypothesized that fact-driven articles are written by professional journalists and, thus, grammatically are more complex. Therefore, news of this kind should have higher frequency of special characters and digits than propaganda news. Following this assumption, I have constructed two vectors, where vector values represent normalized scores for *n*-numbers of digits and special characters in each news instance. Because averaged tokens value will be biased due to different news length, I have built 10-point logarithmic scale. One constructs logarithm which base is 10-th root for total number tokens in each news instance, and which parameter is averaged value of *n*-tokens (digits or special characters). Below, one can find a formalized definition of normalization model.

1. $b = \sqrt[10]{\sum_{i=1}^{n} x_i}$, where each $\boldsymbol{x_i}$ is token in news instance;

2. $\log_b(\sum_{i=1}^{n} y_i)$, where each $\mathbf{y}_i$ is special characters' token in news instance[4].

Sentiment: TextBlob library was used to build a sentiment scoring. Since TextBlob does not have support for the Russian language, I had to translate each instance into English using library's in-built Google translate API. Each sentence was given a polarity score ranging from "-1" (negative) to "+1" (positive). As a result, a vector of average polarity scores for each instance was created.

Evaluation

I performed two class classification using two supervised machine learning methods, including Multinomial Naïve Bayes Classifier (MultinomialNB) and Support Vector Machine (SVM). During the initial learning phase, SVM had the best accuracy in general. It was also prone to the dataset deviations, i.e. SVM got pretty consistent results for different sample sizes. Thus, SVM with linear kernel was used as the main machine learning algorithm for this task. All parameters were set to default.

The whole evaluation process was divided into several stages, starting from training model on TF matrix only and ending up with mixed features. I used 5-fold cross validation for the first stage,

[4] Special characters' tokens include symbols like comma, dot, semi-column, column, question tag, etc.

where features were built out of TF matrix only. Cross-validation indicated that results for the model were consistent, with average mean of F1-score – 98%. In all other cases, I have done a train/test split (70/30). Area under the curve (AUC) was chosen as the scoring metric for validating model performance. Model real-life testing was done using a dataset of 830 non-propaganda that was deliberately dropped at the data collection stage. Table 2 shows a detailed review of evaluation techniques used in this research.

Table 2. Model performance

Feature space	**Features**	**Accuracy**	**AUC (Area under the curve)**	**Test dataset accuracy, n=831**
TF matrix	1796	~98%	~94%	~78%
TF matrix + digits and special characters frequencies	1798	~98%	~97%	~75%
Propaganda sensitive words+ digits and special characters frequencies	19	~86%	~84%	~100%
TF matrix + propaganda sensitive words	1814	~98%	~98%	~79%

- The sole use of TF matrix achieves great results – 98% accuracy and 94% AUC score for train/test split. (see “Methodology I. TF matrix” iPython notebook). However, the model performance immediately drops to 77% on the previously unseen dataset. I hypothesized that this behavior is partially explained with the fact that n-grams heavily memorize text structure and is thus biased to the stylistic patterns of news sources used for training model. This outcome is similar to the Jihadist twitter classification task [8] where n-gram features greatly outweigh other variables.

- Next, I have added two vectors of digits and special characters’ frequencies (see “Methodology II. TF matrix + special characters and digits” iPython notebook). Contrary to the results achieved in classifying U.S. propaganda and fake news [7], these features led to slight dropdown in accuracy for the testing set. However, both features’ means have statistical difference with $p< .001$. Therefore, I have decided to add these features for the next model.

- I have also run a model using only 18 features constructed from propaganda sensitive words, e.g. *terrorists vs. rebels, self-proclaimed authorities vs. people’s government,* adding two vectors from previous model (see “Methodology III. Propaganda-sensitive words” iPython notebook). An astounding 100% accuracy was achieved for the test set. Surprisingly enough, with only 20 features in hand, the model performance is a way beyond the SVM with 1796 words' frequencies features.

- Last but not least, I have combined TF matrix with propaganda sensitive features to see if this could improve first model generalizability (see "Methodology IV. TF matrix + propaganda-sensitive words" iPython notebook). Indeed, there was a minor uptick in accuracy for the test dataset, but nothing similar to the third model's performance.

I have never included sentiment score in the model because Ukrainian news appeared to have more negative sentiment than fake news. This finding contradicts other research findings that propaganda/fake news appeal to emotions, thus, having stronger negative sentiments [7,8].

Challenge

Since the trained classifier in the first and fourth example primarily relies on the terms frequencies' features, I was interested in measuring what impact different text stylistic and patterns could have on model's predicting power (see "Challenge" iPython notebook). Simultaneously, I was interested to see if the robust performance of the second model will "survive" new dataset. Therefore, I have collected a set of in-domain news from other Ukrainian media outlet – ZIK newspaper (n=18). Ideally, one needs to run model on the new fake set as well, but I could not retrieve more fake texts. Similarly, to the evaluation stage, three different models were tested.

- SVM-trained classifier, using only TF matrix as feature space, correctly classified 11 news that constitutes 61% of sample. Given the binary classification task, this result is only 11% better in comparison to simple guessing – a sad point for any machine learning endeavor;
- Next, I have used model trained on propaganda-sensitive words, including two special marks' and digits vectors, which gave roughly 89% accuracy result. I found this result really fascinating. A humble dictionary-based approach, which includes vocabulary of propaganda-sensitive words, could largely outperform computationally expensive TF models;
- Finally, combined model of TF matrix and propaganda-sensitive words indicated drop in predicting power to the same 61% - a meager accuracy for binary classification problem.

Conclusions

The initial findings of this research indicate low feasibility of training fully automated fake/propaganda classification model, using text information retrieval (essentially, terms frequency calculations) as model input features. Based on the trained model performance, one can underscore several limitations:

- Machine learning models that rely solely on tokens' and n-grams frequencies are prone to heavily memorizing text structures and stylistic. These models are showing good performance when classifying news coming from same sources, but immediately falter whenever news source is changed;

- Contrary to the U.S. fakes spread during 2016 election campaign, Russian propaganda is written more professionally with complex grammatical structure. Therefore, some features that worked in the context of the U.S. fakes, e.g. digits, special characters, sentiment scoring, have weak or zero added value in this case;
- Models trained for multiple domains will be computationally expensive, "data-hungry" and will require frequent feature updates. This research focused only on two topics of propaganda – war in the East of Ukraine and Minsk agreements – and already faced an issue with collecting more fakes to test classifier. By expanding the propaganda sample space, the number of features required for updating will grow exponentially.

Possibly, the most prominent result of the research is solid performance of classifier with propaganda-sensitive words as its input. Even with a small number of features, the model showed good results at the testing stage. What is more important, it was prone to the change of news sources. This outcome leads to conclusions that, with frequently updated vocabulary of domain sensitive words, semi-automated classification of propaganda news could be real. However, human expertise is required when it comes to labelling different propaganda-sensitive words. Therefore, one of the promising future directions of research might be development of versatile vocabulary that will incorporate propaganda-sensitive words from multiple domains.

Acknowledgment

This research has become possible due to enormous work made by the Ukrainian NGO "StopFake". Since 2014, StopFake has found, processed and published rebuttal of fake/propaganda news spread by Russian online outlets. All the fake/propaganda news used to train classifier were scrapped from - http://www.stopfake.org.

I am also very grateful to a friend of mine – Dr. Denys Zhernovyi – who contributed to collecting a set of non-fake news.

References

1. M. Pesic, *Serbian Propaganda: A Closer Look.* National Public Radio, 1999. - http://www.bu.edu/globalbeat/pubs/Pesic041299.html

2. H. Kozlowska, *Who Was Right About W.M.D.s in Iraq?* The New York Times journal, 2014. - https://op-talk.blogs.nytimes.com/2014/10/17/who-was-right-about-w-m-d-s-in-iraq/?mcubz=0

3. H. Richie, *Fake News Stories of 2016.* CNBC news, 2016. - https://www.cnbc.com/2016/12/30/read-all-about-it-the-biggest-fake-news-stories-of-2016.html

4. S. Cohen, C. Li, J. Yang, and C.Yu. *Computational journalism: A call to arms to database researchers*. - http://web.eecs.umich.edu/~congy/work/cidr11.pdf

5. S. Bajaj, *The Pope has a new baby. Fake news detection using deep learning.* Stanford University press, 2017. - https://web.stanford.edu/class/cs224n/reports/2710385.pdf

6. H. Rashkin, E. Choi et al. *Truth of varying shades: analyzing language in fake news and political fact-checking.* University of Washington, 2017. - https://homes.cs.washington.edu/~hrashkin/publications/factcheck_emnlp17.pdf

7. C. Fan, *Classifying fake news,* University of Illinois, 2017. - http://www.conniefan.com/wp-content/uploads/2017/03/classifying-fake-news.pdf

8. E. Omer, *Using machine learning to identify jihadist messages on Twitter.* Uppsalla University, 2015.