Tracking Bias in News Sources Using Social Media: the Russia-Ukraine Maidan Crisis of 2013–2014

Peter Potash, Alexey Romanov, Anna Rumshisky
Department of Computer Science
University of Massachusetts Lowell
{ppotash,aromanov,arum}@cs.uml.edu

Mikhail Gronas
Department of Russian
Dartmouth College
mikhail.gronas@dartmouth.edu

Abstract

This paper addresses the task of identifying the bias in news articles published during a political or social conflict. We create a silver-standard corpus based on the actions of users in social media. Specifically, we reconceptualize bias in terms of how likely a given article is to be shared or liked by each of the opposing sides. We apply our methodology to a dataset of links collected in relation to the Russia-Ukraine Maidan crisis from 2013-2014. We show that on the task of predicting which side is likely to prefer a given article, a Naive Bayes classifier can record 90.3% accuracy looking only at domain names of the news sources. The best accuracy of 93.5% is achieved by a feed forward neural network. We also apply our methodology to gold-labeled set of articles annotated for bias, where the aforementioned Naive Bayes classifier records 82.6% accuracy and a feed-forward neural networks records 85.6% accuracy.

1 Introduction

The proliferation of online information sources and the dissolution of the centralized news delivery system creates a situation where news no longer comes from a restricted set of reputable (or not-so-reputable) news organizations, but rather from a collection of multiple distributed sources such as blogs, political columns, and social media posts. In times of social or political conflict, or when contentious issues are involved, such sources may present biased opinions or outright propaganda, which an unprepared reader is often not equipped to detect. News aggregators (such as Google News) present the news organized by topics and popularity. But an adequate understanding of a news story or a blog post requires weeding out the "spin" or "framing", which reflects the source's position on the spectrum of conflicting opinions. In short, we need to know not only the *content* of the story, but also the *intent* behind it.

Many supervised approaches to bias detection rely on text analysis (Recasens et al., 2013; Iyyer et al., 2014), effectively detecting words, phrases, and memes characteristic of an ideology or a political position. All such methods can be characterized as *language-based methods* of bias detection. In contrast, the methods that we term *reaction-based* use human response to a news source in order to identify its bias. Such response is registered, for example, in social media when users post links to news sources, or like the posts that contain such links. We observe that with respect to divisive issues, users tend to split into cohesive groups based on their *like* streams: people from conflicting groups will like and pass around sources and links that express the opinions and the sentiment common only within their group. Put simply, reaction-based methods determine the bias of a source by how the communities of politically like-minded users react to it, based on the amount of liking, reposting, retweeting, etc., the text gets from the opposing groups. Such methods have recently been used with success in the context of liberal/conservative biases in US politics (Conover et al., 2011; Zhou et al., 2011; Gamon et al., 2008).

We believe the language-based and reaction-based methods are complementary and should be combined to supplement each other. Much work in bias detection relies on pre-existing annotated corpora of texts with known conservative and liberal biases. Such corpora obviously do not exist for most ideologies and biases found outside of American or Western discourse. In this work, we propose to use a reaction-based analysis of biases

Proceedings of the 2017 EMNLP Workshop on Natural Language Processing meets Journalism, pages 13–18
Copenhagen, Denmark, September 7, 2017. ©2017 Association for Computational Linguistics

in news sources in order to create a large silver standard of bias-marked text that will be used to train language-based bias detection models. This is done by collecting the articles reacted upon (liked/linked/posted) by the members of opposing political groups in social networks. We thus conceptualize the bias of a news article in terms of how likely it is to be referenced by one of the opposing groups, following the idea that any publicity is good publicity, and any reference to a source can in a some sense be considered a positive reference. The resulting "silver" corpus is slightly noisier than a manually annotated gold standard such as the one used in (Iyyer et al., 2014), but makes up for this deficiency by not being limited in size.

In this work, we use the Russia-Ukraine Maidan conflict of 2013–2014 as a case study for predicting bias in a polarized environment. We collect a large silver corpus of news articles using the posts in the user groups dedicated to the discussion of this conflict in a Russian social media network VKontakte, and evaluate several methods of using this data to predict which side is likely to like and share a given article. We use features derived both from a source's URL as well as the text of the article. We also analyze the news sharing patterns in order to characterize the specific conflict represented in our case study. Lastly, we annotate a small corpus of news articles for bias in relation to the Maidan crisis. We are then able to test the effectiveness of classifiers on gold-standard data when trained solely with silver-labeled data.

Our results show that predicting bias based on the frequency of sharing patterns of users representing opposing communities for our case study is quite effective. Specifically, a Naive Bayes classifier using only the domain name of a link as a feature (a one-hot input representation) achieves 90% accuracy on a bias prediction task. We compare an SVM-based classification method with a Feed Forward Neural Network (FFNN), and find that the best accuracy of 93.5% is achieved by the FFNN.

2 Dataset

In this study, we use data from Russian-speaking online media, posted during the Ukrainian events of 2013-2014. We use the largest Russian social network "VKontakte" (VK)[1]. According to liveinternet.ru, VKontakte has 320 million registered users and is the most popular social network in both Russia and Ukraine. During the conflict, both pro-Russian (also known as "Antimaidan") and pro-Ukrainian side (also known as "Pro-" or "Evromaidan") were represented online by large numbers of Russian-speaking users.

Domain Name	Google News	Antimaidan groups	Evromaidan groups
segodnya.ua	102	95	232
unian.net	78	160	2311
zn.ua	72	38	395
lenta.ru	70	869	146
news.liga.net	61	63	777
ru.tsn.ua	54	65	809
korrespondent.net	52	333	571
rbc.ua	34	91	115
ria.ru	21	8968	109
vestifinance.ru	19	104	6
glavred.info	19	12	117
forbes.ua	18	11	66
rian.com.ua	17	58	11
pravda.com.ua	17	197	6307
vz.ru	16	2092	8
vesti.ru	15	831	54
lb.ua	15	18	222
biz.liga.net	15	6	56
slon.ru	14	29	77
gordonua.com	14	34	762
gazeta.ru	12	454	94
interfax.com.ua	12	45	131
obozrevatel.com	11	57	670
podrobnosti.ua	10	60	275
top.rbc.ru	10	406	118
interfax.ru	9	1166	39
ntv.ru	8	408	36
mk.ru	8	150	44
pravda.ru	7	282	4
gigamir.net	7	5	16
focus.ua	6	8	101
forbes.ru	6	54	6
nbnews.com.ua	6	27	117
ng.ru	6	33	5
rosbalt.ru	6	90	61

Table 1: Statistics of the occurrences of domains extracted from Google News.

We have built a scalable open stack system for data collection from VKontakte using the VK API. The system is implemented in Python using a PostgreSQL database and Redis-based message queue. VK API has a less restrictive policy than Facebook's API, making it an especially suitable social network for research. Our system supports the API methods for retrieving the group members, retrieving all posts from a wall, retrieving comments and likes for a given post, and so on.

In order to seed the data collection, we selected the most popular user groups from the two op-

[1] http://vk.com

posing camps, the Evromaidan group (154,589 members) and the Antimaidan group (580,672 members). We then manually annotated other groups to which the administrators of these two groups belonged, selecting groups with political content. This process produced 47 Evromaidan-related groups with 2,445,661 unique members and 51 Antimaidan-related groups with 1,942,918 unique members.

To create a dataset for our experiments, we randomly selected 10,000 links, 5,000 each from Antimaidan and Evromaidan-related group walls. Links are disregarded if they appear on walls from both sides, which is to ensure an unambiguous assignment of labels. We made a 90%/10% train/test split of the data. The labels for the links correspond to whether they came from an Antimaidan or Evromaidan related wall. We refer to these datasets as our silver-labeled training and test sets.

3 News Sharing Patterns in Polarized Communities

In this section we investigate whether the bias of a news article can be detected by examining the users who shared or liked this article. If the link to this article is predominantly shared by Evromaidan users, then it is more likely to cover the events in a way favorable to the Evromaidan side, and vice versa. Examining the links shared by “Antimaidan” and “Evromaidan” groups, we see that they have a very small number of shared links in common. The “Antimaidan” groups have posted 239,182 links and the “Evromaidan” groups have posted 222,229 links, but the number of links that have been posted by both sides is only 1,888, which are 0.79% and 0.85% of links posted to Antimaidan and Evromaidan groups, respectively, an alarmingly small number. This general mutual exclusion of link sharing makes our label assignment strategy realistic for our case study, since links are rarely shared by both communities.

In order to check how many links from a news aggregator are actually posted on the groups walls, we have collected links from the first 5 pages of Google News Russia by using “maidan” and “Ukraine” query words. This resulted in a total of 1,039 links. Out of these, 106 were posted on the “Antimaidan” group walls and 113 on the “Evromaidan” group walls.

In order to investigate the possibility of characterizing a news source, rather than a specific news article in terms of its bias, we also extracted domain names from the links collected from Google News, as well as the links from the group walls. This produced 126 unique domain names from Google News, out of which only 7 domains were not presented on the groups wall, for a total of 14 links, or 1.3%. Examining the number of occurrences of each domain name on each side’s group walls is quite instructive, since for most sources a clear preference from one of the sides can be observed.

4 Bias Annotation

In order to evaluate our methodology on gold-labeled data, as opposed to the silver-labeled dataset from Section 2, we have annotated the news articles from Section 3. Of the 1,039 links from the Google News query, only 678 were active at the time of the annotation. Two different annotators labeled the articles on a scale from -2 to 2, where -2 is strongly Antimaidan, -1 is weakly Antimaidan, 0 is neutral, 1 is weakly Promaidan, and 2 is strongly Promaidan. The annotators could also label NA if the article isn’t related to the Maidan crisis. We then merged the non-zero labels to be either Pro or Anti Maidan, like our silver data. In terms of labels where both annotators agreed, there are 40 Anti, 95 Pro, and 215 neutral articles. We test our methodology on the articles with a Pro or Anti bias (we were unable to scrape 3 of the Pro articles, so there are 92 Pro articles for testing).

5 Predicting Bias

In this section, we describe our experiments for predicting issue-based bias of links shared online, using the Maidan crisis as a case study.

5.1 Feature Representation

We define a feature representation for each article that will use the following types of features:

Domain Name This features is simply the domain name of the link. There are a total of 1,043 domain names in the training set. The use of this feature is inspired by the uneven distribution of domain name sharing present in Table 1. Most importantly, this feature provides a single non-zero value for its representation, which allows us to evaluate how effective domain names are for predicting bias.

Text-Based Features We initially scrape the full HTML page from links and strip the HTML content using BeautifulSoup[2], followed by tokenization of the text. We use a bag-of-words representation of the text with count-based features[3]. We filter the vocabulary to contain words that occur in at least 10 documents and at most in 90% of documents. This representation has 53,274 dimensions.

URL-Based Features Each article appears in our system as a link. We conjecture that we can better determine bias using features of this link. There are three features taken from the link: 1) domain name, 2) domain extension, and 3) path elements. For example, The URL `http://nlpj2017.fbk.eu/business-website-services` will have the following features: 'nlpj2017' and 'fbk' will be domain features, 'eu' will be an extension feature, and 'business-website-services' will be a path feature. We use the same vocabulary filtering strategy as with the text features – minimum frequency of ten documents and a maximum frequency of 90% of documents[4]. This representation has 277 dimensions.

5.2 Models

Our experiments are a binary classification task. We experimented with three types of classifiers. The first is a Naive Bayes classifier. The second classifier is an SVM. Both the Naive Bayes and SVM classifiers are implemented in scikit-learn (Pedregosa et al., 2011) using default settings. The second classifier is a FFNN, implemented in Keras (Chollet et al., 2015). The FFNN has two layers[5], each with size 64, and ReLu activation (Nair and Hinton, 2010) for the hidden layer.

6 Results and Discussion

The results of our experiments on the silver-labeled test set are shown in Table 2. Since the dataset is balanced, random guessing would produce 50% accuracy. We can see from the results that all systems perform very well when compared to random guessing, with the best accuracy posted by the FFNN at 93.5%. The main result that should be noted is the performance of the Naive Bayes classifier using only domain names, which is effectively determining bias purely based on which side has shared a given domain name the most. This method is highly competitive, outperforming all SVM models, and trailing the FFNN with URL features by only 1%. This result confirms the unbalanced sharing habits shown in Table 1. Furthermore, the high accuracy of the domain name/URL features could potentially be an indicator of just how polarizing the Maidan issue is, as the two sides are highly separable in terms of the sources and links they share in their respective communities.

Model	Features	Accuracy
Naive Bayes	Domain Name	90.3
SVM	URL	87.0
SVM	Text	90.2
SVM	URL+Text	90.2
FFNN	URL	91.3
FFNN	Text	**93.5**
FFNN	URL+Text	93.1

Table 2: Results of our supervised experiments for predicting bias on the silver-labeled test set.

Model	Features	Accuracy
Naive Bayes	Domain Name	82.6
SVM	URL	80.3
SVM	Text	73.5
SVM	URL+Text	72.7
FFNN	URL	78.0
FFNN	Text	71.2
FFNN	URL+Text	**85.6**

Table 3: Results of our supervised experiments for predicting bias on gold-labeled data.

One interesting result is that, regardless of the classifier, combining URL and text features does not increase the accuracy of text features alone, and even sees a drop in performance for the FFNN. This could potentially be explained by Karamshuk et al.'s (2016) assertion that the text on web pages contains markers of its URL features. However, when combining URL and text features, URL features are represented in different dimensions than the text features, so the classifier could potentially treat them differently than if they were just appearing in the text.

[2] `http://www.crummy.com/software/BeautifulSoup/`

[3] We also experimented with tfidf and binary features, but found count features to perform the best.

[4] Filtering of URL features greatly reduces the feature size, as it is 11,516 dimension in total. Also, the SVM classifier gains 11% accuracy with filtering.

[5] We also experimented with adding more layers, but did not find a gain in performance.

# Training Ex.	Accuracy
9,000	90.2
4,500	89.2
2,250	88.4
1,124	86.0
562	83.3
280	81.2
140	78.5
70	77.1
34	71.7
16	49.9

Table 4: Accuracy of the SVM model with text features based on differing amounts of training data. Evaluation is done on silver-labeled test set.

Table 3 shows the results of our models on the gold-labeled test set described in Section 4. First, we establish a trend of domain names being a highly informative feature. Secondly, we see a model that makes a dramatic improvement combining URL and text features; the FFNN. However, when using either URL or text features individually, the SVM performs better on this test set.

Effects of Training Set Size

Table 4 Shows the accuracy of the SVM model with text features based on differing amounts of training data evaluated on the silver-labeled test set. There are several interesting insights from these results. First, reducing the initial training set size by 75% reduces accuracy less than 2%. Second, even with just 280 training examples, the model still achieves above 80%; similarly, the model still achieves above 70% accuracy with only 34 training examples. Lastly, the model sees its accuracy drop to that of random guessing only once it is given 16 training examples.

7 Related Work

Most state-of-the-art work on bias detection deals with known pre-defined biases and relies either strictly on text or strictly on user reactions in order to determine the bias of a statement. For example, Recasens et al. (2013) developed a system for identifying the bias-carrying term in the sentence, using a dataset of Wikipedia edits that were meant to remove bias. The model uses a logistic regression classifier with several types of linguistic features including word token, word lemma, part-of-speech tags, and several lexicons. The classifier also looks at the edits that have previously been made on the article. Using the same dataset, Kuang and Davison (2016) build upon previous approaches by using distributed representations of words and documents (Pennington et al., 2014; Le and Mikolov, 2014) to create features for predicting biased language.

Iyyer et al. (2014) created a system that detects the political bias of a sentence using a recursive neural network to create multi-word embeddings. The model starts with the individual embeddings of the sentence's words and systematically combines them to create the sentence embeddings. These sentence embeddings are then used as input to a supervised classifier that predicts the author's political affiliation for the sentence. The model is trained on a set of sentences annotated down to phrase-level for political bias. The authors argue that, unlike bag-of-words models, the sentence embeddings capture the full semantic composition of the sentence.

The work most similar to ours is that of Karamshuk et al. (2016). While both their work and ours seek to predict the bias of a news source, the key difference is in how we construct our datasets. Karamshuk et al. manually annotate specific news sources to identify partisan slant, and label an article's bias based on its source. Our labeling is based on the sharing patterns of users in a polarized setting (see Section 2 for a further description of our dataset). Lastly, Karamshik et al. use a bag of (word vector) means to construct features for their classification experiments, which has been shown to be a poor representation for text classification (Zhang et al., 2015). The authors' best accuracy is 77% in their binary classification tasks.

A different approach to bias detection consists in analyzing not the texts themselves, but the way the texts circulate or are reacted upon within a social network. Examples of such an approach are found in the work of Gamon et al (2008) who analyze the links between conservative and liberal blogs and the news articles they cite, as well as the expressed sentiment toward each article. Zhou et al (2011) detected and classified the political bias of news stories using the users' votes at such collaborative news curation sites as diggs.com. Relatedly, Conover et al (2011) used Twitter political tags to show that retweet patterns induce homogeneous, clearly defined user communities with extremely sparse retweets between the communities.

8 Conclusion

In this paper we address the issue of predicting the partisan slant of information sources and articles. We use the the Russia-Ukraine Maidan crisis of 2013-2014 as a case study, wherein we attempt to predict which side of the issue is likely to share a given link, as well as its corresponding article. Our best classifier, a FFNN, achieves 93.5% accuracy on the binary classification task using a BOW representation of the link content, and 91.3% accuracy using only information from the URL itself. Moreover, a Naive Bayes classifier using only the domain name of a link can record 90.3% accuracy, outperforming an SVM with more complex features. This remarkably high accuracy dictates that this case study exhibits high polarization in terms of its news sources, as well as its semantic content. We also evaluate our methodology – training a classifier with silver-labeled data based on user actions – on a gold-labeled test annotated for bias in relation to the Maidan crisis. The classifier using only domain names continues its impressive performance, recording an 82.6% accuracy. Conversely, a FFNN records 85.6% accuracy. For our case study, we find that the situation when two opposing sides share the same links is extremely rare.

Acknowledgments

This work was supported in part by the U.S. Army Research Office under Grant No. W911NF-16-1-0174.

References

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political polarization on twitter. In *ICWSM*.

Michael Gamon, Sumit Basu, Dmitriy Belenko, Danyel Fisher, Matthew Hurst, and Arnd Christian König. 2008. Blews: Using blogs to provide context for news articles. In *ICWSM*.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*. pages 1113–1122.

Dmytro Karamshuk, Tetyana Lokot, Oleksandr Pryymak, and Nishanth Sastry. 2016. Identifying partisan slant in news articles and twitter during political crises. In *International Conference on Social Informatics*. Springer, pages 257–272.

Sicong Kuang and Brian D Davison. 2016. Semantic and context-aware linguistic model for bias detection.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. pages 807–814.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *ACL (1)*. pages 1650–1659.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. pages 649–657.

Daniel Xiaodan Zhou, Paul Resnick, and Qiaozhu Mei. 2011. Classifying the political leaning of news articles and users from user votes. In *ICWSM*.