Biased News Data Influence on Classifying Social Media Posts

Marija Stanojevic, Jumanah Alshehri, Eduard Dragut, Zoran Obradovic
Center for Data Analytics and Biomedical Informatics (DABI)
Temple University
Philadelphia, Pennsylvania, USA
{marija.stanojevic, jumanah.alshehri, edragut, zoran.obradovic}@temple.edu

Abstract

A common task among social scientists is to mine and interpret public opinion using social media data. Scientists tend to employ off-the-shelf state-of-the-art short-text classification models. Those algorithms, however, require a large amount of labeled data. Recent efforts aim to decrease the compulsory number of labeled data via self-supervised learning and fine-tuning. In this work, we explore the use of news data on a specific topic in fine-tuning opinion mining models learned from social media data, such as Twitter. Particularly, we investigate the influence of biased news data on models trained on Twitter data by considering both the balanced and unbalanced cases. Results demonstrate that tuning with biased news data of different properties changes the classification accuracy up to 9.5%. The experimental studies reveal that the characteristics of the text of the tuning dataset, such as bias, vocabulary diversity and writing style, are essential for the final classification results, while the size of the data is less consequential. Moreover, a state-of-the-art algorithm is not robust on unbalanced twitter dataset, and it exaggerates when predicting the most frequent label.

In: A. Aker, D. Albakour, A. Barrón-Cedeño, S. Dori-Hacohen, M. Martinez, J. Stray, S. Tippmann (eds.): Proceedings of the NewsIR'19 Workshop at SIGIR, Paris, France, 25-July-2019, published at http://ceur-ws.org

1 Introduction

In recent years, social media platforms have become leading channels for the exchange of knowledge, debates, and product or opinion advertising [PP10, WD07, Gly18, SKB12]. Social scientists routinely use data from social media platforms to survey public opinion on specific topics [Mos13, CSPR16, HBK+17, BM18] and computer scientists use the data to improve the performance of state-of-the-art natural language processing (NLP) algorithms [CXHW17, ACCF16, GPCR18, ZWWL18].

Social media data, while abundant, pose many challenges in usage: 1) user demographics are rarely available; 2) posts are short and sometimes hard to understand without context, and 3) it is challenging to label millions of posts manually in short time. One may overcome the first challenge by selecting only information from users where demographic information is available using multiple social platforms. However, this may bias the data. In order to solve the other two problems, we need systems that classify data into different opinion classes with limited human involvement.

Figure 1: ULMFiT model training-flow overview

Numerous algorithms have been proposed to cope with large amounts of short text [ZZL15, ZQZ+16, LQH16, XC16, CSBL16, YYD+16, MGB+18]. All these algorithms are supervised in nature, and therefore, require hundreds of thousands of labels in order to achieve adequate performance levels. In the last two years, algorithms such as CoVe [MBXS17], ELMo [PNI+18], ULMFiT [HR18] and OpenAI GPT

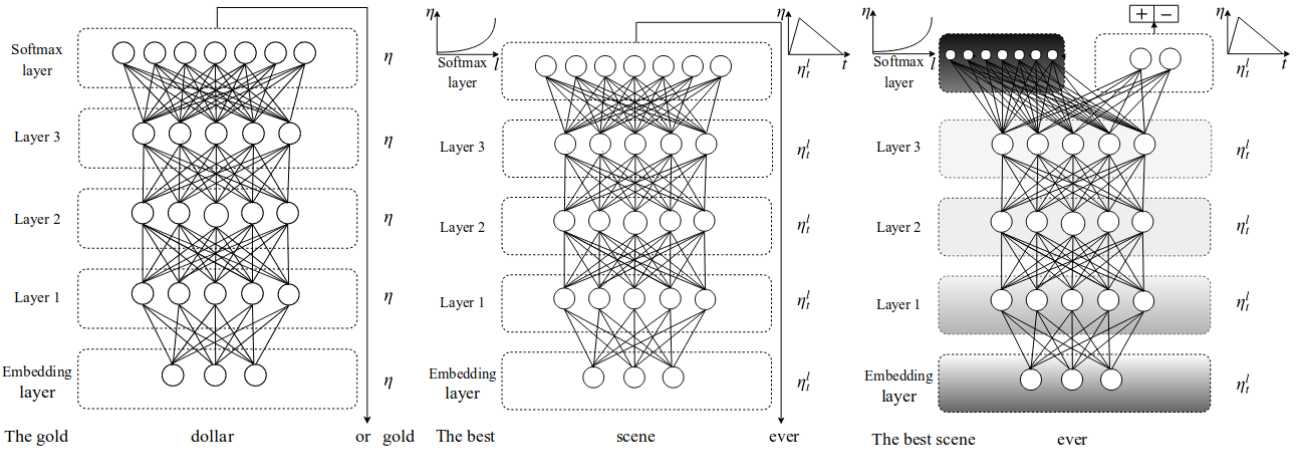

Figure 2: ULMFiT model training details

[RNSS18] have been proposed to minimize the need for labeled data and increase the performance by cleverly utilizing text characteristics. Those methods start from word-vectors pre-trained on general documents and fine-tune them on domain-specific documents by employing self-supervised learning. In the Universal Language Model Fine-tuning for Text Classification (ULMFiT) [HR18] self-supervised process predicts the next word based on the previous words in the context. After the fine-tuning step, we additionally train the model with a small number of manually labeled text instances (Figure 1).

Most of the datasets (e.g., AGNews, DBPedia, Yahoo Answers) [ZZL15] used for testing text classification algorithms are balanced and on average contain much longer texts than social media posts. On the other hand, social media data retrieved with a purpose to model opinion is usually unbalanced. The goal of this paper is to investigate the performance of ULMFiT model on classifying social media posts for different settings of fine-tuning and labeled datasets. We test balanced and imbalanced labeled social media datasets and fine-tuning news texts with different characteristics (e.g., size, bias, writing style).

Experiments utilize Twitter data related to USA midterm elections from 2018 and news data from the USA elections 2016. The news data is collected from six major outlets which are considered to have a bias towards the left or right political spectrum[1]. We test how fine-tuning with articles from different news outlets influences the accuracy of social media posts classification. The hypothesis is that fine-tuning with appropriate topic-related text from news can help improve classification, but bias in news articles can also hurt the performance. We test the hypothesis on ULMFiT algorithm described in the next section.[2]

2 Methods

The ULMFiT model [HR18] consists of three training components (Figure 2). Each component is based on the language model AWD-LSTM [MKS17] and consists of a word-embedding (input) layer, multiple LSTM-layers, and a softmax layer used to predict the output. Experimental results in literature prove that multiple LSTM-layers can learn more complex contexts [MBXS17, PNI+18, HR18] than single LSTM-layer models.

In the first part of ULMFiT, words and contexts embedding is learned from general texts (such as Wikipedia). In the second part, they are updated (fine-tuned) with topic-related data to learn domain-specific words and phrases. The third part is trained on labeled domain-specific examples so it can predict labels for the new examples. The output of each part is the input in the next step.

Even though the ULMFiT model is complex, it can be trained efficiently on GPUs when smartly implemented. Once trained, the first part of the model does not change, so we use WT103 pre-trained vectors to reduce training time. [3]

In order to speed up fine-tuning (Figure 2b), we use different learning rates for each LSTM layer. The top layer, which calculates softmax, has the largest learning rate, η^L. Learning rates for remaining layers are set to $n^{l-1} = n^l/2.6$ for $l \in (1, L)$ as suggested in a prior study[HR18]. Instead of having a constant

[1]Information about outlet bias is taken from: https://mediabiasfactcheck.com

[2]The latest text classification progress: http://nlpprogress.com/english/text_classification.html

[3]WT103 word-vectors can be found here: http://files.fast.ai/models/wt103

learning rate, slanted triangular learning rates are used for every layer to improve the accuracy of the model [HR18]. First, the learning rate sharply linearly increases so that the model can learn fast from the first examples. Once learning rate achieves the η^L, it slowly linearly declines as shown in the top-right corner of Figure 2.

In the third step (Figure 2c), layers are trained gradually. First, only the top layer is trained with labeled data for one epoch while other layers are frozen. In each new epoch, the next frozen layer from the top is added to the training.

3 Experiments

Experiments are conducted using Twitter data on USA midterm elections 2018 and news data from USA elections 2016.

Twitter data is collected by searching for posts published between November 4th and 7th 2018 which have one of the hashtags: "#vote", "#trump", "#election", "#midtermelection", "#democrats", "#republicans" and "#2018midterms". In total, we accrue 936,462 tweets. Most of the posts are retweets, which appear multiple times in the corpus. After retweets removal, 244,320 distinct posts remained, and we pre-process their text by removing all characters, except alphanumerics.

Out of those posts, we label 1,526 examples with 0, 1 or 2. Label 0 is assigned to examples that support or promote the left political spectrum or denounce the right point of view. Label 1 is given to politically neutral posts (e.g., posts that encouraged voting). Label 2 is assigned to examples that support or advertise the right political spectrum or condemn the left point of view. We discard 500 examples ($\sim$ 25% of posts) because they are unrelated to elections.

News data is collected from six outlets that are perceived to have different political partisanship, ranging from the left-oriented to right-oriented outlets based on media bias fact check website (Table 1). Articles published between October 2015 and May 2017 that contain words "election", "ballot", "republican", "GOP", or "democrat" are selected. The news articles differ substantially in writing style, content diversity, bias, number of articles and number of words (Table 1). As with the tweets, news articles do not always discuss the U.S. elections. Sometimes, they debate Brexit or elections in France and other countries worldwide. In pre-processing, we remove all non-alphanumeric characters from news articles.

Experiments settings. We use the pre-trained WT103 token-vectors in the first ULMFiT step. WT103 has 103 million tokens from Wikipedia texts for training, 217K tokens for validation and 245K tokens for testing [MXBS16]. Our system is trained using the architecture in Figure 2a. The vocabulary has 267K unique tokens. In this paper word and token have interchangeable meanings.

Table 1: Outlets

Outlet	Bias	#Words
CNN News (CNN)	left	426,778
Washington Post (WP)	left-center	9,229,176
BBC News (BBC)	neutral-left	1,247,437
MarketWatch (MW)	neutral-right	1,505,107
Wall Street Journal (WSJ)	right-center	547,548
FoxNews (FN)	right	3,082,912

For the fine-tuning step, we explore ten different settings: 1) "all news" text with the data from all outlets + tweets text; 2) only the tweets; 3) text from "left-biased" outlets + tweets text; 4) text from "right-biased" outlets + tweets text. Remaining six experiments contain text from one outlet and tweets text. We randomly permute examples in a fine-tuning dataset before usage.

In the third step, experiments test two settings of labeled Twitter data. Mix 1 (balanced mix) contains 380 examples with label 0 (left), 323 examples with label 1 (neutral) and 323 examples with label 2 (right). Mix 2 (unbalanced mix) contains 380 examples with label 0 (left), 823 examples with label 1 (neutral) and 323 examples with label 2 (right). We randomly split labeled data into three disjoint parts: test (200 examples), validation (200 examples) and training (626 examples in Mix 1 and 1126 examples in Mix 2). Each experiment is repeated four times and accuracy mean, and the standard deviation is reported for each of the ten settings.

We do not clean Twitter, and news data of non-relevant examples in order to emulate the real-world situation. The data retrieval process is intentionally simple to mirror the information extraction process often used in research papers [Mos13, CSPR16, HBK$^+$17, BM18]. Those experiments test the robustness of the model to the bias and noise in data and robustness to the unbalanced classes.

4 Results and discussion

We repeat each experiment four times, and we report the accuracy mean and standard deviation in Table 2. High standard deviation ($1.2 - 5.3\%$) indicates the model's sensitivity to the order of examples in the fine-tuning data and a need for more labeled examples.

Results provide evidence that the model is not robust to unbalanced datasets. When Mix 1 and Mix 2 results are compared, the model always achieved better results for Mix 2 (Table 2) which has 54% of neutral labels as compared to 31.5% of neutral labels in

Table 2: Classification results

News sources included (Left : Neutral : Right)	**Mix 1 (380 : 323 : 323)**	**Mix 2 (380 : 823 : 323)**
All news	$53.2 \pm 3\%$	$59.4 \pm 3.7\%$
No news	$56 \pm 5.3\%$	$\mathbf{66.6 \pm 2.5\%}$
Left-biased (CNN+WP+BBC)	$49.2 \pm 2.9\%$	$61.1 \pm 3.3\%$
Right-biased (MW+WSJ+FN)	$51.7 \pm 3.8\%$	$63.0 \pm 3.2\%$
CNN	$\mathbf{58.7 \pm 1.2\%}$	$62.7 \pm 3.0\%$
Washington Post (WP)	$55.6 \pm 3.0\%$	$60.7 \pm 1.4\%$
BBC	$55.1 \pm 3.1\%$	$64.1 \pm 2.7\%$
MarketWatch (MW)	$56.5 \pm 2.6\%$	$64.2 \pm 1.8\%$
Wall Street Journal (WSJ)	$57.7 \pm 3.7\%$	$60.0 \pm 4.3\%$
FoxNews (FN)	$53.2 \pm 2.9\%$	$61.9 \pm 3.3\%$

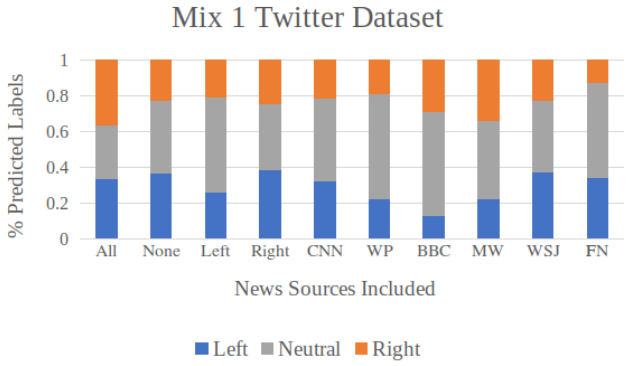

Figure 3: Balanced Twitter Dataset: Percent of predicted labels from each class when fine-tuned with ten different combinations of news outlets texts

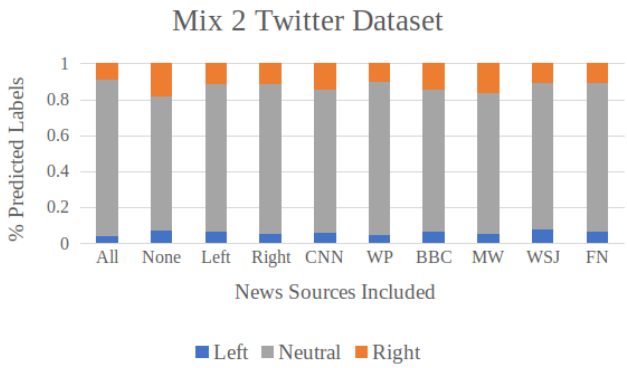

Figure 4: Unbalanced Twitter Dataset: Percent of predicted labels from each class when fine-tuned with ten different combinations of news outlets texts

Mix 1. As evident from Figure 4, $80 - 90\%$ of predicted labels for Mix 2 are neutral. Therefore, better results for Mix 2 are achieved because the algorithm exaggerates the most frequent (neutral) label in the imbalanced dataset (which contains 54% of examples of that class).

The classification accuracy difference between Mix 1 and 2 is the largest (11.9%) when "left-biased news" is used for fine-tuning. In this case, the accuracy on both Mix 1 and Mix 2 decreases compared to when "No news" is present. However, outlet bias has more influence on the accuracy of Mix 1.

Figure 3 reveals that using "all news" data for fine-tuning achieves the best balance among predicted labels for Mix 1. However, almost half of predicted labels are wrong, so accuracy is low.

Labeled Twitter data demonstrate diversity among posts with label "left". They often talk only about one particular issue and have fewer hashtags that support the left political spectrum. Additionally, the diversity of people and entities mentioned is more prominent in the posts labeled as "left" than those labeled "right" (which mainly mention president Trump). Hence, the best performance for Mix 1 is achieved when fine-tuning with "CNN" data because the model is trained to focus more on left-relevant contexts.

The next best results for Mix 1 are achieved when fine-tuning with news articles from The Wall Street Journal because its articles often discuss both sides in detail (sometimes even in the same sentence). Hence, when the model is trained with data from this outlet, it understands relevant phrases and predicts "left" and "right" labels with higher accuracy. On the other hand, "The Wall Street Journal" fine-tuned experiment predicts much more often "right" label for "left-labeled" example than the other experiments.

The confusion matrices created for each experiment and Figure 4 reveal that the algorithm recognizes the right label easier than the left label in Mix 2. A better understanding of the right label can be explained with the different writing style of left-labeled tweets, which reflects a more diverse set of topics and entities as discussed above. The best accuracy score for Mix 2 is achieved when "no news" data is used for the fine-tuning process. Most of the labels are neutral, and news data is mainly left or right oriented/biased, so it influences the accuracy negatively.

As hypothesized, results demonstrate that fine-

tuning with biased news datasets can influence accuracy in contrasting ways. Different influence of biased news is particularly visible in the results of Mix 1 where the difference between the best and the worst accuracy for different fine-tuning settings is 9.5%. In Mix 2 this difference is also notable, 7.2%. Influence of the bias is not uniform. While fine-tuning with "left-biased news" gives the worst result for Mix 1, its performance for Mix 2 is average when compared to other experiments. On the other hand, fine-tuning with "all news" gives the worst results for Mix 2 and average results for Mix 1.

The size of the fine-tuning data does not seem to influence the results. "Washington Post" has the largest amount of words, but it achieves average results in both mixes. "CNN" is the smallest dataset, but it achieves the best result for Mix 1. It is interesting to notice that "all news" achieves worse results than "no news" fine-tuning for both Mix 1 and Mix 2, even though in literature, training with more data often contributes to better results. This result suggests that the content (bias) of the fine-tuning dataset is more important than its size.

Accuracy behavior in many experiments requires further analysis in order to better understand the influence of fine-tuning text characteristics on the performance. Additionally, the effect of non-relevant text on the accuracy should be further tested since its frequency is high in both news and Twitter data. Since results clearly show that this model is not robust on bias and noise, other novel methods should be tested similarly. It is essential to create unbalanced and biased datasets for fine-tuning and testing of the future models to create robust methods that would be beneficial to the real-world applications.

5 Conclusion

In this work we have shown that bias, noise and text properties need to be accounted for when constructing data for fine-tuning language models. Text size does not seem to be an important dimension. We performed experiments with data collected from Twitter and six news outlets using ULMFiT language model. Results show that the algorithm is not robust to noise in data, to bias in the fine-tuning dataset, or to the dataset imbalance.

While conducted experiments show weaknesses of the existing system, further work is needed to understand better the relationship between properties of fine-tuning data and specific tasks. Additionally, better models are required that are more robust to bias and noise in order to be able to solve challenging real-world problems.

6 Acknowledgements

This research was supported in part by the NSF grants IIS-1842183.

References

[ACCF16] Orestes Appel, Francisco Chiclana, Jenny Carter, and Hamido Fujita. A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems*, 108:110–124, 2016.

[BM18] Marco Bastos and Dan Mercea. Parametrizing brexit: mapping twitter political space to parliamentary constituencies. *Information, Communication & Society*, 21(7):921–939, 2018.

[CSBL16] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[CSPR16] Fabio Celli, Evgeny Stepanov, Massimo Poesio, and Giuseppe Riccardi. Predicting brexit: Classifying agreement is better than sentiment and pollsters. In *Proceedings of the Workshop on Computational Modeling of Peoples Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 110–118, 2016.

[CXHW17] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.

[Gly18] Carroll J Glynn. *Public opinion*. Routledge, 2018.

[GPCR18] Aitor García-Pablos, Montse Cuadros, and German Rigau. W2vlda: almost unsupervised system for aspect based sentiment analysis. *Expert Systems with Applications*, 91:127–137, 2018.

[HBK+17] Philip N Howard, Gillian Bolsover, Bence Kollanyi, Samantha Bradshaw, and Lisa-Maria Neudert. Junk news and bots during the us election: What were michigan voters sharing over twitter. *Computational Propaganda Research Project, Oxford Internet Institute, Data Memo*, 1, 2017.

[HR18] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[LQH16] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

[MBXS17] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

[MGB+18] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[MKS17] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

[Mos13] Mohamed M Mostafa. More than words: Social networks text mining for consumer brand sentiments. *Expert Systems with Applications*, 40(10):4241–4251, 2013.

[MXBS16] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[PNI+18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[PP10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[SKB12] Pawel Sobkowicz, Michael Kaschesky, and Guillaume Bouchard. Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web. *Government Information Quarterly*, 29(4):470–479, 2012.

[WD07] Duncan J Watts and Peter Sheridan Dodds. Influentials, networks, and public opinion formation. *Journal of consumer research*, 34(4):441–458, 2007.

[XC16] Yijun Xiao and Kyunghyun Cho. Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*, 2016.

[YYD+16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[ZQZ+16] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.

[ZWWL18] Shunxiang Zhang, Zhongliang Wei, Yin Wang, and Tao Liao. Sentiment analysis of chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems*, 81:395–403, 2018.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.