Orwellian-times at SemEval-2019 Task 4: A Stylistic and Content-based Classifier

Jürgen Knauth
Institute of Computer Science
University of Goettingen
jknauth@uni-goettingen.de

Abstract

While fake news detection received quite a bit of attention in recent years, hyperpartisan news detection is still an underresearched topic. This paper presents our work towards building a classification system for hyperpartisan news detection in the context of the SemEval2019 shared task 4. We experiment with two different approaches - a more stylistic one, and a more content related one - achieving average results.

1 Introduction

Recent years have seen a noticeable change in the political discourse: Political polarization has increased and political opinions have become more hyperpartisan (Doherty, 2017). This affects the media, especially news media and is therefore a topic of considerable interest for science and society.

We present an approach for the detection of high polarization and hyperpartisan news articles. Our approach addresses stylistic and content related features. The latter are implemented by identifying n-grams that are typical for either a hyperpartisan or a more balanced perspective.

1.1 The Task

The goal of the SemEval2019 Hyperpartisan News Detection Task (Kiesel et al., 2019) is to build a system capable of classifying arbitrary articles either as non-hyperpartisan or hyperpartisan.

1.2 The Dataset

For building a classification system the organizers of the task provided several data sets extracted from different American news sites:

a) a set of 600.000 articles for training (classification: by publisher's general orientation)

b) a set of 150.000 articles for validation (classification: by publisher's general orientation)

c) 645 training articles (classified individually by humans using a crowd sourcing approach)

d) validation set (classified by publisher, unknown size as this data set has been hidden during the task)

e) validation set (classified individually, unknown size as this data set has been hidden during the task)

All data - the articles themselves as well as the ground truth data - is provided in an proprietary but simple and well parsable XML format defined by the task owners. The individual data records includes a globally unique ID, the title, the source URL and publication time. Additionally the ground truth data for a) and b) contains information about a left-right bias of the publisher in general.

2 Related Work

Not so much research has been done in regard to hyperpartisan news detection. Other work is primarily addressing related fields such as ideology detection, fake news detection. For example (Hutto et al., 2015; Hamborg et al., 2018) addresses identification and quantification of media bias. (Iyyer et al., 2014) addresses political ideology detection using recursive neural networks. (Rashkin et al., 2017) is analyzing language in fake news for automated political fact-checking. One interesting work directly targeting hyperpartisan news is the work of (Potthast et al., 2018) identifing hyperpartisan news articles via style.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 976–980
Minneapolis, Minnesota, USA, June 6–7, 2019.

3 Methodology

3.1 PoS Tagging

As we hypothesize that not all parts-of-speech are equally important for distinguishing hyperpartisan and non-hyperpartisan articles, we lemmatized and pos-tagged the dataset.

Initially, we experimented with the TreeTagger (Schmid, 1994), however since this turned out not to be sufficiently robust for the noisy input data, which included encoding errors as well as portions of JavaScript code, we later adopted the Stanford CoreNLP tagger (Manning et al., 2014).

3.2 Feature Extraction

We used a total of 108 features for our experiments. The next sections discusses our features in detail.

3.2.1 Linguistic Complexity and Style Features

Basic complexity: We hypothesize that hyperpartisan texts are stylistically less complex than non-hyperpartisan texts (Potthast et al., 2018), hence we implemented a number of features measuring linguistic complexity. We measure the distribution characteristics of paragraph lengths, sentence lengths and word lengths. Individual features were derived from that data like minimum, maximum, variance, mean, et cetera.

Number of words in main part-of-speech categories: We collect the number of verbs, nouns, adjectives and adverbs in articles and derive distribution features from it. While not every part-of-speech category will have the same importance we rationalize that at least the distribution characteristics of nouns, adjectives and adverbs could be a style hint for hyperpartisan or non-hyperpartisan.

Simple form of lexical density: As "lexical density" we here consider the ratio of words being not part of the NLTK stop-words in comparison to the total number of words. The idea behind this feature is to detect articles with lower or higher information character and take some stylistic aspect into account.

Huffman compression ratio: Our huffman compression feature is used with similar intention. First: The general idea behind huffman compression (Huffman, 1952) is to build a dictionary of words ordered by frequency in a binary tree. This is done in such a way that in the end high frequency words can be encoded with a shorter bit code than low frequency words. The rationale in our approach here is to perform a compression of individual articles: The better this compression works the more an author of an article reuses his own words. The more difficult this compression is, the higher is the variety of words used by an author. For speed reasons we intentionally do not perform a full compression here but build a huffman compression tree and then estimate the size the indices would take in a full compression. We then put this information into relation to the total number of tokens of an article and use this as a feature.

Readability scores: A set of readability scores is used. Readability scores express the simplicity of a text in various different ways - at least to some extent - as well as give a rough judgement for the reading competence level of an audience required to understand the text. We implemented features based on four readability scores: ARI (Smith and Senter, 1967), Coleman-Liau (Coleman and Liau, 1975), Flesch-Kincaid (Kincaid et al., 1975) and Gunning-Fog (Gunning, 1952).

Vocabulary variety: The vocabulary variety classifier is calculating the ratio of uniquely used words in relation to the total number of words. This way this classifier assists in judging the complexity of the text in an article as well.

3.2.2 Arousal vs. Rationality

Distribution parameters of business words: We assume that news articles addressing business related topics are inherently not particularly hyperpartisan. Based on manual inspection of training data, we therefore created a list of 27 words, which we consider to be expressing business related topics, for example "sales", "growth", "CEO", "opportunity", "revenue", "Q1", "shareholder" and similar terms.

Distribution parameters of words of disgust: In a similar way to business words the corpus lemmas are judged whether they express some kind of disgust. For this purpose a hand picked vocabulary of 246 words had been created from publically available online dictionaries such as LEO (LEO), Wictionary (Wictionary) and similar that genuinely express some kind of disgust. Though this dictionary likely is not complete the assumption is, that it gives a general insight into whether a writer expresses disgust at least to some extend, e.g. "disaccord", "rupture", "distaste", "scandalous" or even words like "rotten". We intended here not to detect

only archetypical words such as "awful" but also more uncommon words that might typically not be seen in news so frequently. The rationale behind this is that we noticed the phenomenon of hyperpartisan authors to attempt to use a more vivid and strong language with sometimes less common words.

Cardinal number ratio: Detecting cardinal numbers is another feature addressing very specific aspects of articles: The idea behind this feature is that more fact-based communication might more likely make use of numbers in order to express and proof their positions. While we can not check the truth of claims involving cardinal numbers we at least try to detect the quantity of such claims.

Pronouns before "need" and "must": Two feature detectors address pronouns directly proceeding the words "need" or "must". We noticed hyperpartisan articles where the authors directly address the reader and give advice how society should proceed. This is done in an inclusive way, so sequences like "we must (do sth)" or "we need (to do sth)" could be observed.

3.2.3 Content Features

To address content specifically we implemented features derived from the provided test data itself, though this way these features can cover only limited and existing content.

Attributively used adjectives: According to the theories behind framing in psychology, political influence can be produced by repeating specific kind of wordings (Wehling, 2016). We noticed that this technique seems to be used sometimes quite extensively by authors of more extreme positions in recent years as they have a quite unchanging perspective about topics, persons and events. For example in the manually classified data the term "jewish" is used to characterize a following noun about five times more often in hyperpartisan than non-hyperpartisan articles, "holistic" about 30 times and "immediate" only about a third of the times compared to non-hyperpartisan news. Based on this phenomenon a dictionary of adjectives which discriminate a following noun have been extracted from non-hyperpartisan and hyperpartisan pos-tagged training data in a separate processing process, resulting in 2720 adjectives for our use. Our feature is then measuring whether more non-hyperpartisan or hyperpartisan use of such adjectives can be observed in an article.

Lemma-bigram similarity scores: While our attributively-used-adjectives-feature focuses on the adjectives themselves and is therefore a single word feature, we additionally used lemma based bigram features. We extracted all bigrams in sentences for a window of four tokens from the manually tagged training data (and for experiments from the larger data set) and associated them with either non-hyperpartisan or hyperpartisan labels. For example the lemmas "obama" directly followed by "administration" appear significantly more often in hyperpartisan than non-hyperpartisan articles. It's even more extreme with "obamacare" and "act", sequences of "bad" and "happen" or "disastrous" and "war": The latter having even no mentions at all in non-hyperpartisan articles. Interestingly some bigrams are less characteristic as one would expect: For example "illegal" and "immigration" is used quite frequently by both classes. Again other bigrams seem to be more typical for non-hyperpartisan news articles, e.g. "fake" and "story".

For our implementation we determined the relation of how often either hyperpartisan and non-hyperpartisan bigrams appeared per paragraph:

$$f = (nH - nNH)/(nH + nNH) \qquad (1)$$

where *nH* and *nNH* refer to the number of hyperpartisan/non-hyperpartisan bigrams. This value will be positive or negative depending on the surplus of non-hyperpartisan vs. hyperpartisan bigrams encountered in unseen text. We do this for directly adjacent lemmas, for two lemmas skipping one token, two tokens and three tokens and calculate the four medians so that we end up with a set of feature values, each one expressing content similarity to our reference data.

3.3 Machine Learning

We built two different models by training a support vector machine with an rbf-kernel (libsvm). The first one is based only on the stylistic features and has been submitted for the first evaluation run of the task, the second one is based only on the content features which has been submitted for the second evaluation run of the task.

For training of the first model 100.000 articles have been selected by random with stratified sampling, arriving at 25.000 articles classified as hyperpartisan left, 25.000 hyperpartisan right and 50.000 non-hyperpartisan. Selecting a sub-

	Dataset	Acc	Prec	Rec	F1
M1	by-pub.	0.505	0.503	0.949	0.657
M2	by-pub.	0.537	0.530	0.658	0.587
M2	by-art.	0.671	0.654	0.729	0.689

Table 1: Results, Model 1 and Model 2 with validation dataset used, accuracy, precision, recall and F1 score.

set of the available articles was necessary as part-of-speech tagging with first the TreeTagger and then the CoreNLP tagger took quite some time to complete. As mentioned before we ran into some tagging problems because of errors in the corpus data and limited capabilities of the existing Python adapters for CoreNLP. Additionally we encountered some problems with larger amounts of data which surprisingly caused crashes in the C implementation of the SVM (NuSVC of sklearn) for unknown reasons. So in the end we limited ourselves to these 100.000 random articles to cope with these difficulties.

As we recognized during our work that the training data classified by-pubisher was - by nature - not so accurately labeled, we train our second model on the gold standard data with 645 manually labeled articles to avoid any noise for our features as much as possible. For this model we used only the content features.

4 Results and Conclusions

To train our models we used the provided training data "by-publisher" and "by-article" as described in the last section. Evaluation runs have then been performed on the validation data "by-publisher" and "by-article" (which were hidden during the duration of the shared task). The results can be seen in table 1.

Model 1 (which was trained on the 100.000 randomly picked articles focusing on style features) was tested against the validation data labeled by-publisher. Model 2 (which was trained on the 645 articles focusing on content features) was tested against the validation data labeled by-publisher and the validation data by-article in two separate runs. Our model 1 achieved better results than model 2 during our evaluation runs on the by-publisher data. It has been selected by the organizers for ranking in the leader board.

Validation showed that our first model exhibits a trend to judge articles too easily as being hyperpartisan. Though our second model exhibits a trend to more easily classify articles as hyperpartisan as well, this effect is not that strong.

Our second, content feature model did not perform that well on the test data labeled by-publisher than the first, the style-based model. Interestingly it performed better on evaluation data labeled by-article. As our training data of 645 articles for that model is small the second model likely suffers from overfitting.

5 Further Work

In this paper we have presented a binary classification system that assign labels "non-hyperpartisan" and "hyperpartisan" for articles. While we could achieve some results in that field we still think that more work is needed here.

Results of competing teams in the SemEval2019 shared task indicate that our current approach has not yet been explored to full extent in that regard: Better classification could be possible. We assume that additional effort should be taken in selecting more and better style and content features. Though stylistic approaches seem to be promising – comp. (Potthast et al., 2018) – we assume that future work should focus more on content and empathic perception of the content by the reader. For example sentiment could be taken into consideration as news articles tend to have different point of views on different topics. As there exists a variety of different sentiment tools of varying quality experiments need to be performed to explore possibilities of improving our models. Attempts in this regard have been undertaken by ourselves already but could not be completed for this shared task. Additionally it would be interesting to combine both approaches, something we were not able to explore sufficiently during this shared task.

Acknowledgments

This work was funded by the ministry of science and culture of Lower Saxony ("Holen und Halten").

References

M. Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. In *Journal of Applied Psychology*, volume 60(2), pages 283–28.

Carroll Doherty. 2017. Key takeaways on americans growing partisan divide over political

values. http://www.pewresearch.org/fact-tank/2017/10/05/takeaways-on-americans-growing-partisan-divide-over-political-values/. Accessed: 2019-02-21.

Robert Gunning. 1952. *The technique of clear writing*. New York: McGraw-Hill.

Felix Hamborg, Karsten Donnay, and Bela Gipp. 2018. Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*.

D. A. Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IEEE, formerly Proceedings of the IRE*, 40(9):1098–1101.

C.J. Hutto, Dennis Folds, and Scott Appling. 2015. Computationally detecting and quantifying the degree of bias in sentence-level text of news stories. In *The First International Conference on Human and Social Analytics*, pages 30–34.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Ps Resnik. 2014. Political ideology detection using recursive neural networks. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 1, pages 1113–1122.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval 2019)*. Association for Computational Linguistics.

J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel. Technical report.

LEO. Leo online dictionary. https://dict.leo.org. Accessed: 2019-02-21.

libsvm. SVM implementation library libsvm. https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html. Part of Scikit-Learn; Authors: Chih-Chung Chang, Chih-Jen Lin; Accessed: 2019-02-21.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 231–240. Association for Computational Linguistics.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937. Association for Computational Linguistics.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/. Accessed: 2019-02-21.

E. A. Smith and R. J. Senter. 1967. Automated readability index. Technical report.

E. Wehling. 2016. *Politisches Framing: Wie eine Nation sich ihr Denken einredet - und daraus Politik macht*. Ullstein.

Wictionary. Wictionary, the free dictionary. http://en.wictionary.org. Accessed: 2019-02-21.