Team Kermit-the-frog at SemEval-2019 Task 4: Bias Detection Through Sentiment Analysis and Simple Linguistic Features

Talita Anthonio
University of Groningen
University of the Basque Country
t.r.anthonio@student.rug.nl

Lennart Kloppenburg
lennartkloppenburg@live.nl

Abstract

In this paper we describe our participation in the SemEval 2019 shared task on hyperpartisan news detection. We present the system that we submitted for final evaluation and the three approaches that we used: sentiment, bias-laden words and filtered n-gram features. Our submitted model is a Linear SVM that solely relies on the negative sentiment of a document. We achieved an accuracy of 0.621 and a f1 score of 0.694 in the competition, revealing the predictive power of negative sentiment for this task. There was no major improvement by adding or substituting the features of the other two approaches that we tried.

1 Introduction

With the growing role of social media in politics it becomes ever more important to safeguard the integrity of information people consume. News articles about important events in the world can affect political choices. It is therefore crucial to pinpoint what information people know to be trustworthy, factual and unbiased. One way to do this is by using a computational system that detects an author's or publisher's bias in a news article. In Potthast et al. (2018) we have seen that it is possible to build such a system by relying on the writing characteristics of a text.

To shed more light on potential linguistic computational methods for hyperpartisan news detection, we present our participation in the SemEval 2019 shared task on hyperpartisan news detection, of which the purpose is to identify whether a news article contains *hyperpartisan* (Kiesel et al., 2019) content. For our contribution, we set out to experiment with various types and levels of features, such as *a*) **sentiment** that could indicate an author's bias (Recasens et al., 2013), *b*) **bias-laden words** such as assertives, factives and hedges, and *c*) part-of-speech (from now on POS) **filtered n–grams**. In the end, we decided to submit a model that only uses the negative sentiment of an article as a feature. We obtained an accuracy of 0.621 and a f1 score of 0.694 on the by-article test set, which resulted to the 30th place in the competition. On the by-publisher test set, the systems accuracy was 0.589 and its f1 score 0.623 (20th place).

2 Related Work

One of the first studies on detecting linguistic bias in online texts were mainly focused on detecting biased language in Wikipedia articles. Despite the domain difference, this task is related to ours because Wikipedia is also a source of information which should contain unbiased language. Systems that were employed for this task used a combination of linguistic features, such as POS n-grams and binary features representing the usage of bias words, assertive verbs, factive verbs, hedges and sentiment features (Recasens et al., 2013; Hube and Fetahu, 2018). Most of these features were derived from existing lexicons. For sentiment features, both studies used a sentiment polarity lexicon from Liu et al. (2005).

A similar set of features was used in Hutto et al. (2015) to detect sentence based bias in news articles. Yet, they obtained sentiment features using the VADER sentiment analysis tool (Hutto and Gilbert, 2014). Because of their focus on sentence-based bias detection and the high relatedness of their study, it was interesting to investigate whether we could use the same features on document-level classification.

The studies that we discussed so far proved that it is possible to detect bias by using simple computationally derived linguistic features. On the other hand, we work with a much larger amount of documents which also come from a different genre.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1016–1020
Minneapolis, Minnesota, USA, June 6–7, 2019.

Because of these aspects, it was fruitful to investigate how effective these features would be.

3 Data

We worked with the data provided by the organizers of the task. We show an overview of the data we used for the competition in Table 1. The sets named 'by-publisher' are automatically labeled using the publisher of the article, whereas the articles from the 'by-article' set were manually labeled through crowd-sourcing (Vincent and Mestre, 2019).

For the competition, we trained our models on the by-publisher training set. We used this data because of its size and the equal frequency distribution of the labels. We took the model with the highest accuracy on the by-publisher validation set for our final submission. After the evaluation period, we submitted several models that were trained on the by-article training data to find out whether we should have submitted a model that was trained on the by-article data.

name	function	size	distribution
by-publisher	training set	600,000	50-50
by-publisher	validation set	150,000	50-50
by-publisher	test set	4,000	50-50
by-article	training set	645	37% hyper. 63% mainstr.
by-article	test set for competition	638	50-50

Table 1: An overview of the data that we used for the shared task.

4 Final System

4.1 VADER Sentiment Analysis

The system we submitted for final evaluation is a simple SVM classifier with a linear kernel. It uses the LinearSVM[1] implementation from *scikit-learn* (Pedregosa et al., 2011) with default hyperparameter settings *C=1.0*, the *'squared hinge'* loss function and the *'l2'* penalization norm. The only features that the classifier uses to make a prediction is the intensity of the negative sentiment of each document which varies between 0 (neutral) and 1.0 (extremely negative). This score is computed by the freely available package **V**alence **A**ware **D**ictionary and s**E**ntiment **R**easoner[2] from NLTK(Loper and Bird, 2002). VADER was developed by Hutto and Gilbert (2014) and is a rule-based and lexical model for general sentiment analysis. Even though VADER is specifically developed to perform sentiment analysis in social media texts, the tool works reasonably well on determining the sentiment of news articles according to our observations.

4.2 Results

The system we submitted reached the 30th place with an accuracy of 0.621. The other scores are reported in Table 2. It also displays the performance of our model on the *by-publisher* development set and the *by-article* test set. We obtained a high recall on the official test set, which corresponds to the performance on the *by-article* test set. Nonetheless, since the evaluation script only tracked the *hyperpartisan=true* class and our model was apparently biased towards a hyperpartisan prediction, this metric is not informative because it implies that the mirrored *hyperpartisan=false* class has a much lower recall.

Furthermore, the scores in Table 2 show that our model neither performed well on the *by-article* (0.519 accuracy) nor the *by-publisher* (0.562 accuracy) development set. In particular, the accuracy on the *by-article* set is even lower than the accuracy of the baseline. Nonetheless, the model performed better on the official test set, since the accuracy and f1 score were substantially higher. We surmise that this is related to the inconsistent similarities of the data sets rather than the predictive power of the features.

	Accuracy	Precision	Recall	F1
official test set	0.621	0.582	0.860	0.694
by-publisher	0.562	0.559	0.585	0.572

Table 2: Evaluation metrics (of the true class) across different data sets of correctly detecting the hyperpartisan class.

5 Alternative Methods

Despite the low performance on the *by-publisher* development set, we submitted our final system because it had the highest accuracy on this development set (0.562 accuracy) and the competition evaluation was based on accuracy. In this section,

[1] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[2] https://www.nltk.org/_modules/nltk/sentiment/vader.html

Features	Set-up	Accuracy	Precision	Recall	F1 score
Tf-idf	Word uni-grams with default settings	0.5598	0.5412	0.7854	0.6408
	Word uni-grams with default settings + negative sentiment	0.5597	0.5411	0.7858	0.6409
Sentiment	positive score + negative score + compound	0.5247	0.5201	0.6373	0.5729
	compound score	0.4310	0.4389	0.4962	0.4658
	negative sentiment	0.5616	0.5589	0.5851	0.5717
	positive sentiment	0.4752	0.4779	0.5354	0.5050

Table 3: Performance of other models *trained* on the by-publisher training set on predicting hyperpartisan.

we outline the two other approaches with which we experimented: bias-laden words and filtered n-grams. We also present our attempts to improve the accuracy of our best model on the development set by combining features from the other two approaches. The performance of these models and other detrimental models on the development set are shown in Table 3.

5.1 Sentiment

VADER Sentiment In addition to the negative sentiment score, we also conducted experiments with systems that used several combinations of the negative, positive and compound score provided by VADER. However, none of the combinations outperformed the accuracy of the system that only took the negative sentiment into account. Moreover, as shown in Table 2, the compound score yielded the lowest performance. In our preliminary experiments, we also experimented with the neutral sentiment score but this led to low accuracy scores compared to the other scores. Besides, the evaluation procedure was based on predicting the hyperpartisan is true class, for which we can assume that its corresponding article is not neutral.

We tried to improve the score of the submitted model by using ordinal scales instead of interval variables, in which documents with a negative sentiment score exceeding 0.5 were labeled as having a "high" negative sentiment and "low" otherwise. This did not lead to improvement, which reveals that the raw sentiment score is a better predictor.

Other Sentiment Features We also developed systems that calculated the overall sentiment of a text by using the lexicons of positive and negative words from Liu et al. (2005). We experimented with two methods (1) by counting the amount of positive/negative words and (2) by using binary features where the value was True if it contained one of the words in the lexicons. This method was also used in Recasens et al. (2013). Nonetheless both methods did not even reach an accuracy of 30 percent.

5.2 Bias-Laden Words

Verbs We experimented with the same set of verbs as the mentioned previous studies: assertive verbs, factive verbs and hedges (Hooper, 1975). These words carry cues that may indicate bias. For instance, assertive verbs can be used to assert the truth of a proposition (i.e. *point out*, *claim*, *states*) and factive verbs can be used to presuppose the truth of their corresponding complement clause (i.e. *realize*, *revealed*, *indicated*). The usage of these verbs was encoded in the same way as we did for the lexical sentiment features. Yet, it was not possible to build accurate classifiers that used these features, since the accuracy fluctuated between 0.20 and 0.30. Also, we could not increase the performance of other systems by adding these features.

N-gram BOW We additionally tried to derive bias-laden words through BOW methods, as we surmised that the hyperpartisan texts contained more bias-laden words than legitimate news articles. Because of the size of the training set, we only experimented with uni-gram features (with tf-idf weighting). With this set-up, we obtained a similar accuracy score as when we used only the negative sentiment (see Table 2). Yet, we did not submit the uni-gram model for the competition because we surmised that the effectiveness of bag-of-word features would be more sensitive to the topic of the articles. As an effect, the generalizability on unseen data could be low.

5.3 POS-based N-gram Filtering

We experimented with POS-based features early on in an attempt to model *how* and *where* humans would perceive bias in a text on word-level. We found that adjectives, adverbs and (proper) nouns all somewhat contributed to the tone of a text. However, confidently identifying bias proved rather challenging in many cases. Nouns frequently provide thematic and topical information

Training data	System	Accuracy	Precision	Recall	F1 score
by-publisher	Submitted system: negative sentiment only	0.621	0.582	0.860	0.694
	Word uni-grams + negative sentiment	0.605	0.564	0.920	0.700
	POS filtering**	0.657	0.636	0.738	0.683
	Positive score + negative score + compound score	0.611	0.590	0.732	0.653
by-article	Character 3-to-5 grams	0.772	0.825	0.691	0.752
	Word uni-grams	0.755	0.803	0.675	0.734
	POS filtering	0.537	0.522	0.863	0.650

Table 4: Performance of models on the by-article test set submitted after the evaluation period. **only trained on 100k randomly obtained documents of the by-publisher set (with a balanced frequency distribution of labels).

about a text and adverbs and adjectives can indicate a level of subjectivity. Modals such as *would*, *could* and *must* could additionally carry assertiveness that could be related to bias (Recasens et al., 2013; Hube and Fetahu, 2018). We tried modelling this by extracting *n-grams* that followed certain patterns such as *a*) **nouns** in the middle of a trigram *b*) **particles** in the middle of a trigram *c*) **modal** verbs in the middle of a trigram *d*) **nouns** and their closest preceding adjectives/adverbs *e*) **adjectives/adverbs** and words after them. The n-grams essentially became a new, filtered representation of the *document* and would be weighted using tf-idf. We tested this intuition by splitting the training data. The results were quite promising as we (unofficially) achieved f1 scores of 75-85% using only 200,000 documents. However, results disappointingly floated between 53% and 58% when tested on the development data.

We concluded that the *n-grams* we extracted were not indicative enough to generalize well across different data sets, since they were essentially only a subset of the total body of possible *n-grams*.

6 Other Submissions

After the final submission deadline, we continued submitting models to investigate differences between the by-article and by-publisher training data sets. We also submitted models that solely relied on bag-of-words features, with which we experimented in early stages but discarded in our final submission because of the low performances on the by-publisher validation set.

The results of our submissions after the deadline (Table 4) reveal that bag-of-words and bag-of-characters are indeed useful when the model is trained on the by-article data. In particular, we could have obtained a high accuracy in the competition with a model trained on the by-article set, for instance by using character 3-to-5 grams (0.772 accuracy). Another observation is that the POS filtering model obtained a low accuracy on the test set, even when it was trained on the by-article data. Thus, this seems to indicate that bag-of-words are more effective than fine-grained POS filtering.

7 Conclusions

Detecting biased language is a difficult task because of the subjectivity of the task and the subtlety of linguistic context cues. *Bias* is a broad term which can be applied to many different areas and is not solely restricted to politics or economics. Per our own observations, it was difficult to exactly pinpoint the bias of a biased article.

We achieved promising results after our final submission with bag-of-words and bag-of-character n-grams. This indicates that a bag-of-words approach is able to identify token-based patterns in corpora that are related to bias. However, the reliability of a bag-of-words approach does depend on the lexical similarity between training and test data. We demonstrated this through our contradicting results on the provided validation and official test data.

Sentiment proved to be quite a strong feature that can already separate biased from unbiased articles, although more heuristics are needed. This could be combined with the title of the article which, much like sentiment, tells us something about the *entire* article. It could also be interesting to experiment with more general cues about *entire* texts rather than treating texts as only bags-of-words. This could help develop a system that scales better across different corpora and domains.

References

Joan B. Hooper. 1975. On assertive predicates. In *Syntax and Semantics Volume 4*, volume 4, pages 91 – 124. Academic Press, New York.

Christoph Hube and Besnik Fetahu. 2018. Detecting biased statements in wikipedia. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1779–1786, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

C.J. Hutto, Scott Appling, and Dennis Folds. 2015. Computationally detecting and quantifying the degree of bias in sentence-level text of news stories. *HUSO 2015: The first international conference on HUman and Social Analytics*, pages 30–34.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*. The AAAI Press.

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.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. page 342–351.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 231–240. Association for Computational Linguistics.

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

Emmanuel Vincent and Maria Mestre. 2019. Crowdsourced measure of news articles bias: Assessing contributors' reliability. In *CEUR Workshop Proceedings*, volume 2276.