# Clark Kent at SemEval-2019 Task 4: Stylometric Insights into Hyperpartisan News Detection

**Viresh Gupta, Baani Leen Kaur Jolly, Ramneek Kaur, Tanmoy Chakraborty**
Indraprastha Institute of Information Technology, Delhi (IIIT-Delhi), India
{viresh16118, baani16234, ramneekk, tanmoy}@iiitd.ac.in

## Abstract

In this paper, we present a news bias prediction system, which we developed as part of a SemEval 2019 task. We developed an XGBoost based system which uses character and word level n-gram features represented using TF-IDF, count vector based correlation matrix, and predicts if an input news article is a hyperpartisan news article.

Our model was able to achieve a precision of 68.3% on the test set provided by the contest organizers. We also run our model on the BuzzFeed corpus and find XGBoost with simple character level N-Gram embeddings to be performing well with an accuracy of around 96%.

## 1 Introduction

The problem of hyperpartisan news detection (Potthast et al., 2018) is based on predicting whether a news article is biased towards a specific political wing or not. The problem falls under the category of classification problems, and the task is to classify an article as being extremely one-sided or not. A closely related problem is that of fake new detection wherein the task is to analyze the veracity of an article, and classify it based on some predefined degrees of truthfulness.

Our problem has a high societal relevance, since one-sided news poses a great threat to democracy, particularly in the context of conducting fair elections. In this paper, we discuss our approach to solving this problem used during the contest *Hyper Partisan News Detection*, a competition task at SemEval 2019 (Kiesel et al., 2019).

More formally, our problem definition is:

**Definition 1 (Hyperpartisan News Detection)** *We are given a set of news articles $A$, where each article $a_i$ is marked with two labels: a Boolean label hyperpartisan $h_i$ which indicates if article $a_i$ is biased towards a political wing, and a bias label $b_i \in$ {left, right, left-center, right-center, least} which indicates which wing the article is biased towards. If $h_i$ = True, then $b_i \in$ {left, right}; if $h_i$ = False, then $b_i \in$ {least, left-center, right-center}. The objective is to learn a classifier **C** which predicts the hyperpartisan label $h_j$ for an unknown news article $a_j$.*

In this work, we identify the role of various traditional NLP features in determining the degree of partisanship. We utilise standard term-frequency and inverse document frequency vector features computed for uni, bi and tri-grams obtained from the corpus. We do this feature extraction at both character and word level and then train a gradient boosted decision tree as a classifier for identifying partisanship. We also compare other methods of classification such as SVM, KNN, Naive Bayes and Logistic Regression for the task using the same vector features. Furthermore, experiments exploiting the metadata information were also performed (explained in detail in the scalar features in section 3.2).

The experiments were performed on two corpora, the BuzzFeed corpus (created in (Potthast et al., 2018)) and the training corpus released by the task organisers (the SemEval corpus). Further we also discuss the results obtained on the final test corpus released for the final evaluation of the task in section 4.1. Due to computation infeasibility over the larger training corpus, we do not compute vector features for the SemEval corpus.

## 2 Related Work

The work done on hyperpartisan and fake news detection can be broadly classified into three categories – knowledge-based (Etzioni et al., 2008; Ginsca et al., 2015), context-based (Long et al.,



*Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 934–938
Minneapolis, Minnesota, USA, June 6–7, 2019. 

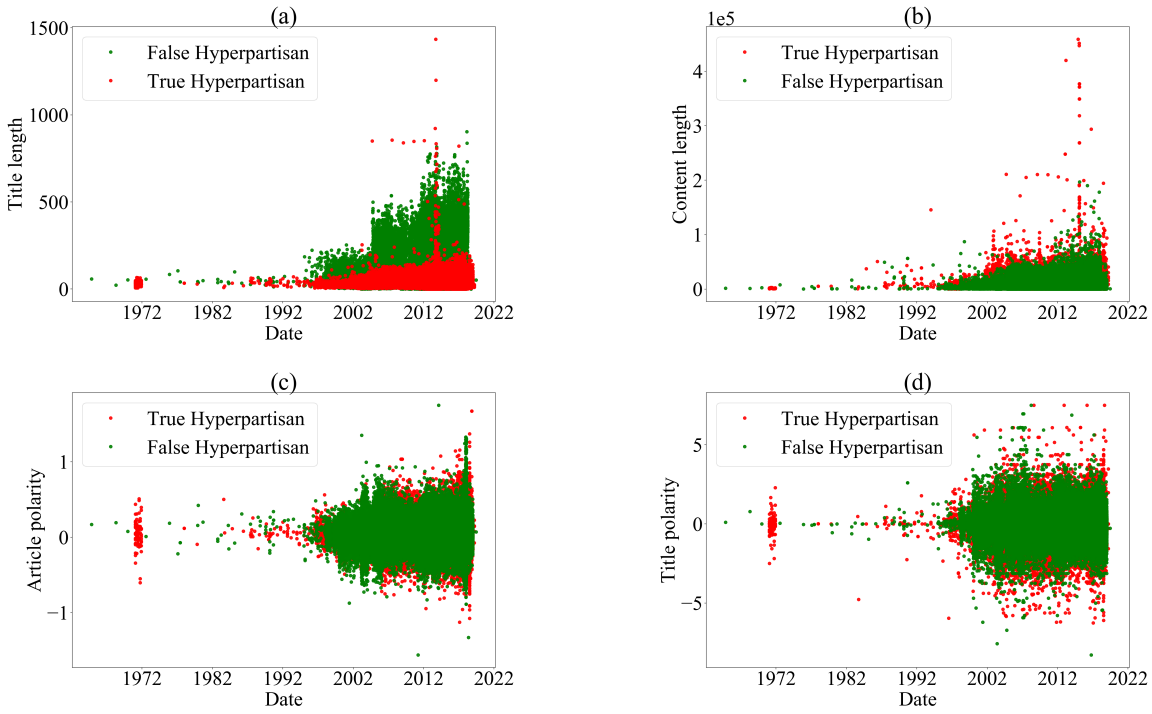


Figure 1: Variation in feature value w.r.t. time for true and false hyperpartisan articles in SemEval Corpus (a) Title length w.r.t. time (b) Content length w.r.t. time (c) Article polarity value w.r.t. time (d) Title polarity value w.r.t time . Red and green colors depict articles from hyperpartisan publishers and other publishers respectively.

2017a; Mocanu et al., 2015), and style-based (Bourgonje et al., 2017).

While the knowledge-based and context-based features may take some time to detect hyperpartisanship (after the news starts spreading on social media), the style-based features can be used to detect partisanship of a news article well in time before the damage happens (Potthast et al., 2018).

For exploiting style based features, (Long et al., 2017b) uses deep learning based methods, and (Shu et al., 2017) performs fake news detection on social media data using a data mining oriented approach.

### 2.1 Baseline

We take as our baseline the work done by (Potthast et al., 2018). Their work uses the author's writing style as features to detect hyperpartisanship. The stylometric features used in their work include POS-unigrams, POS-bigrams, POS-trigrams, char-unigrams, char-bigrams, char-trigrams, stopword-uniGrams, stopword-bigrams, stopword-trigrams, general inquirer categories, readability scores, quotation ratio, link amount and average paragraph length. A random forest classifier was used to make predictions.

We use their classifier as the baseline for the BuzzFeed corpus. For the SemEval corpus, we use the random baseline provided in the task as our baseline. The baseline results are mentioned in Tables 1 and 3 for both the datasets.

## 3 Methodology

In this section, we describe the dataset, the features that we selected and the models we trained using the selected features. A visual overview is shown in Figure 2.

### 3.1 Corpus

We used two corpora, which we name as BuzzFeed corpus and SemEval corpus.

**BuzzFeed corpus:** This corpus was produced by the baseline work. The dataset comprised 1,627 articles that were manually checked by four BuzzFeed journalists. Of these, 826 articles belong to the main-stream category of publishers, 256 belong to the left-wing category of publishers, and the remaining 545 to the right-wing category of publishers.

**SemEval corpus:** This corpus has been released for the SemEval 2019 Task 4 on Hyperpartisan News Detection. It comprises 800,000

| Baseline Results | | | | |
|---|---|---|---|---|
| **Model** | **Precision** | **Recall** | **F1 score** | **Accuracy** |
| RF | 0.75 | 0.77 | 0.75 | 0.75 |
| Count Vector Results | | | | |
| **Model** | **Precision** | **Recall** | **F1 score** | **Accuracy** |
| XGB | 0.92 | 0.93 | 0.92 | 0.93 |
| LR | 0.92 | 0.92 | 0.92 | 0.93 |
| SVM | 0.89 | 0.90 | 0.89 | 0.91 |
| KNN | 0.75 | 0.78 | 0.76 | 0.76 |
| GNB | 0.75 | 0.77 | 0.73 | 0.73 |
| RF | 0.71 | 0.70 | 0.62 | 0.62 |
| Word N-gram Vector Results | | | | |
| **Model** | **Precision** | **Recall** | **F1 score** | **Accuracy** |
| XGB | **0.95** | 0.95 | **0.95** | **0.96** |
| SVM | 0.89 | 0.91 | 0.91 | 0.91 |
| LR | 0.87 | 0.90 | 0.88 | 0.89 |
| GNB | 0.82 | 0.85 | 0.82 | 0.82 |
| RF | 0.74 | 0.73 | 0.64 | 0.64 |
| KNN | 0.72 | 0.70 | 0.62 | 0.61 |
| Character N-gram Vector Results | | | | |
| **Model** | **Precision** | **Recall** | **F1 score** | **Accuracy** |
| XGB | **0.95** | **0.96** | **0.95** | **0.96** |
| SVM | 0.89 | 0.91 | 0.90 | 0.91 |
| GNB | 0.87 | 0.89 | 0.88 | 0.89 |
| RF | 0.87 | 0.89 | 0.87 | 0.89 |
| LR | 0.85 | 0.88 | 0.85 | 0.86 |
| KNN | 0.67 | 0.57 | 0.40 | 0.43 |

Table 1: Vector feature results (BuzzFeed Corpus only).

| | **Precision** | **Recall** | **F1 score** | **Accuracy** |
|---|---|---|---|---|
| RF | **0.81** | 0.64 | **0.72** | **0.74** |
| LR | 0.63 | 0.76 | 0.69 | 0.65 |
| KNN | 0.62 | 0.66 | 0.64 | 0.62 |
| GNB | 0.57 | **0.93** | 0.71 | 0.61 |
| SVM | 0.52 | 0.90 | 0.66 | 0.58 |

(a) BuzzFeed Corpus.

| | **Precision** | **Recall** | **F1 score** | **Accuracy** |
|---|---|---|---|---|
| KNN | **0.64** | 0.59 | 0.62 | **0.63** |
| RF | 0.55 | 0.76 | 0.64 | 0.58 |
| SVM | 0.62 | 0.08 | 0.15 | 0.52 |
| GNB | 0.51 | **0.94** | **0.66** | 0.51 |
| LR | 0.48 | 0.57 | 0.52 | 0.47 |

(b) SemEval Training Corpus.

Table 2: Scalar features results.

training articles and 200,000 test articles. These articles are annotated based on the publisher of the articles.

### 3.2 Feature Selection

Prior to the selection of features, we pre-processed our datasets to clean the text in articles to handle the encoding errors, perform text normalisation and stop word removal. The features we selected can be categorized into two categories, viz. scalar features and vector features. We train two sets of models, one for each category of features.

**Scalar features:** Here, we select four features, all used at the same time since they encode different information:

- Article length: This feature denotes the length of the articles in terms of the number of characters.
- Title length: The title length features denotes the length of the title of an article in terms of the number of characters.
- Article polarity: The article polarity denotes the sentiment score of the article text in the range $[-1, 1]$. A score value less than zero implies a negative sentiment, and a positive sentiment otherwise.
- Title polarity: Similar to the article polarity, the title polarity feature denotes the sentiment score of the article title in the range $[-1, 1]$.

**Vector features:** These include three kinds of features (considered separately since they encode the same information):

- Word count vectors: The count vector for a document denotes the vector of counts of words in the document from the set of possible words in a corpus/vocabulary.
- Word level n-gram vectors: The word level vector for a document denotes the vector of tf-idf values of words level n-grams in the document. We used unigrams, bigrams and trigrams for this feature.
- Character level n-gram vectors: The character vector for a document denotes the vector of counts of character level ngrams. For this feature too, we use unigrams, bigrams and trigrams.

**Visual inspection of the data:** In Figure 1 we provide a visual insight into the corpus based

| Dataset | Method | Precision | Recall | F1 score | Accuracy |
|---|---|---|---|---|---|
| By Article | Ours | 68.3 | 17.8 | 28.3 | **54.8** |
| | Baseline | 46.2 | 46.0 | 44.3 | 45.1 |
| By Publisher | Ours | 56.5 | 17.0 | 26.1 | **51.95** |
| | Baseline | 51.1 | 51.1 | 50.0 | 50.5 |

Table 3: Results for the submitted model.

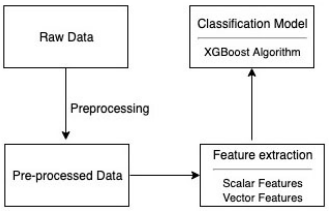


Figure 2: System Overview.

on the features selected. The figure depicts scatter plots showing variation in feature values w.r.t. time for both true and false hyperpartisan articles.

### 3.3 Models Used

We use the following learning models for our scalar features of the BuzzFeed corpus: K Nearest Neighbours (KNN), Gaussian Naive Bayes (GNB), Random Forest (RF), Logistic Regression (LR) and Support Vector Machine (SVM). For the vector features of the BuzzFeed corpus, we use: KNN, GNB, RF, LR, SVM and XGBoost (XGB).

## 4 Experiments

We divide this section into three parts – experimental setup, results on the BuzzFeed corpus, and results on the SemEval corpus.

### 4.1 Experimental Setup

The article polarity and title polarity features were computed using SentiWordNet[1] (Baccianella et al., 2010). All the vector features were computed using the scikit-learn package. To split the data into training and testing sets, we used 5-fold cross-validation.

### 4.2 Results on the BuzzFeed Corpus

The results for the scalar features for models trained on the BuzzFeed corpus are shown in Table 2a(a). The RF model performs the best with an accuracy of 74%. The scalar features, however, are insufficient in beating the baseline. We therefore train models on our vector features. The results for the vector features are shown in Table 1.

As evident from table 1 and 2a(a), vector features perform much better and are able to beat the baseline (Table 1) easily.

[1] https://github.com/anelachan/sentimentanalysis

Various sections in Table 1 represent the results when using different kinds of vector representations as features, with character level n-grams yielding the top results.

### 4.3 Results on the SemEval Corpus

Results on the SemEval corpus are shown in Table 2b(b). From all models, KNN performs the best, followed by RF, SVM, and GNB (in that order).

Since computing vector features and tf-idf features was computationally infeasible on this corpus, we did not train the vector features, however, based on our observations from buzzfeed dataset (i.e the character level vectors outperforming all others), we trained a supervised classifier using FastText (Joulin et al., 2016), (Bojanowski et al., 2016). The accuracy achieved for this model is 65%.

The results of our model using all the scalar features on the final evaluations (testing by article and testing by publisher corpus) of this competition are shown in Table 3. These results show that our model suffered from the inability to draw out more of the relevant results (low recall).

## 5 Conclusion

In this work, we have explored traditional sets of features and models for the Hyper-partisan News Detection problem. We worked on two corpora, of which one has been used in the state-of-the-art literature. For this corpus, we beat the baseline and achieve a remarkable accuracy of 96%. For the other corpus, we achieve an accuracy of 65% (with a fast text character level embedding based model).

From the results of the contest (Table 3), we were able to beat the baseline easily. Though our system did not achieve as high accuracy as other systems, we observe that this is due to a bad recall, i.e even though the features that we selected

are very useful for the model to produce relevant results, it cannot capture some of the correct results.

## 6 Code and Reproducibility

We provide all our code for both Buzzfeed and Semeval Corpus as a github repository located at https://github.com/virresh/hyperpartisan-semeval19-task4 . The same code was uploaded on TIRA (Potthast et al., 2019) and run for submission to the contest.

## Acknowledgement

Part of the research was supported by the Ramanujan Fellowship, Early Career Research Award (SERB, DST), and the Infosys Centre for AI at IIITD.

## References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. volume 10.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv e-prints*, page arXiv:1607.04606.

Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. 2017. From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 84–89. Association for Computational Linguistics.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Alexandru L. Ginsca, Adrian Popescu, and Mihai Lupu. 2015. Credibility in information retrieval. *Foundations and Trends in Information Retrieval*, 9(5):355–475.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv e-prints*, page arXiv:1607.01759.

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.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017a. Fake news detection through multi-perspective speaker profiles. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 252–256. Asian Federation of Natural Language Processing.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017b. Fake news detection through multi-perspective speaker profiles. In *IJCNLP*.

Delia Mocanu, Luca Rossi, Qian Zhang, Marton Karsai, and Walter Quattrociocchi. 2015. Collective attention in the age of (mis)information. *Computers in Human Behavior*, 51:1198 – 1204. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.

Martin Potthast, Tim Gollub, Matti Wiegmann, and Benno Stein. 2019. TIRA Integrated Research Architecture. In Nicola Ferro and Carol Peters, editors, *Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF*. Springer.

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.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *arXiv e-prints*, page arXiv:1708.01967.