Steve Martin at SemEval-2019 Task 4: Ensemble Learning Model for Detecting Hyperpartisan News

Youngjun Joo
Department of Computer Engineering
Yonsei University, Seoul, Korea
yj.joo@yonsei.ac.kr

Inchon Hwang
Department of Computer Engineering
Yonsei University, Seoul, Korea
ich0103@yonsei.ac.kr

Abstract

This paper describes our submission to task 4 in SemEval 2019, i.e., *hyperpartisan news detection*. Our model aims at detecting hyperpartisan news by incorporating the style-based features and the content-based features. We extract a broad number of feature sets and use as our learning algorithms the GBDT and the n-gram CNN model. Finally, we apply the weighted average for effective learning between the two models. Our model achieves an accuracy of 0.745 on the test set in subtask A.

1 Introduction

The proliferation of misleading information in the media has made it challenging to identify trustworthy news sources, thus increasing the need for fake news detection tools able to provide insight into the reliability of news contents. Since the spread of fake news is causing irreversible results, near-real-time fake news detection is crucial. However, knowledge-based and context-based approaches to fake news detection can only be applied after publication; they may not be fast enough (Potthast et al., 2017).

As a practical alternative, style-based approaches try to detect fake news by capturing the manipulators in the writing style of news content. This approach captures style signals that can indicate a decreased objectivity of news content and thus the potential to mislead consumers, such as hyperpartisan style. Hyperpartisan style represents extreme behavior in favor of a particular political party, which often correlates with a strong motivation to create fake news. Linguistic-based features can be applied to detect hyperpartisan articles (Potthast et al., 2017). Deep network models, such as convolution neural networks (CNN), applied to classify fake news detection (Wang, 2017). In this paper, we employ the stylometry-based approach and N-gram CNN model for detecting hyperpartisan news.

2 System Overview

For this task, we extract a broad number of features from the training data and then apply the classifier model to make predictions. Our system employs a gradient boosting decision tree (GBDT) model and N-gram CNN model. In subsequent sections, we describe data preprocessing, feature engineering and learning algorithms.

2.1 Data Preprocessing

Before applying the models, we need to do some transforming tasks of the article texts (i.e., *xml parsing, text tokenizing, stemming, lemmatization, and removing stopwords*) and extracting tasks of the internal and external links for each article. Apart from these tasks, we construct the bias domain dictionary from the *mediabiasfactcheck* site [1] to check the bias on the external linked domain in the article. For this ends, we crawled the top-level domain information from the sites corresponding to the five categories associated with hyperpartisan (e.g., *Left, Center, Least Biased, Right-center Bias, and Right Bias*) respectively.

2.2 Feature Engineering

Since hyperpartisan news is intentionally created for political gain rather than to report objective claims, they often contain opinionated and inflammatory language. Thus, it is reasonable to exploit linguistic features that capture different writing styles to detect hyperpartisan news. Linguistic features are extracted from the text content in terms of document organizations at a different level, such as characters, words, and sentences. Typical common linguisitic features are: *lexical*

[1] http://mediabiasfactcheck.com/

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 990–994
Minneapolis, Minnesota, USA, June 6–7, 2019.

Type of Features	Feature Count
Count features	10
External link bias	3
Sentiment features	8
Readability features	14
Term features	44
Grammar transformation	45
Psycholinguistic features	54
POS tags	36
Word2vec features	301
TF-IDF	10,000

Table 1: Statistics of features.

features, including character-level and word-level features; *syntactic features*, including sentence-level features (i.e., *n-gram, POS tagging, etc.*).

We start by extracting several sets of linguistic features. These feature sets are designed to capture hyperpartisan article from the training datasets. Overall we selected 515 binary features and TF-IDF features. Table 1 provides extracted features on the training dataset.

Basic count features: Previous works on fake news detection (Rubin et al., 2016) as well as on opinion spam (Ott et al., 2011) suggest that the use of punctuation is useful to differentiate deceptive from truthful texts. We construct a basic count feature set including various punctuation characters and other features.

External link bias: We extract bias counts based on the bias domain dictionary for each external linked domain in the article (i.e.,*hyperpartisan links count, non-hyperpartisan links count, and unknown links count*). To determine biases of the external links, we exploit a biased domain dictionary crawling from the *mediabasisfactcheck* site, which consists of five categories for top-level domains(i.e., *left, right, left-center, center, right-center*). The external link bias is counted as the hyperpartisan when the externally linked site is belonging to *left* and *right* among these categories.

Sentiment features: Our system used the VADER sentiment analysis tools [2] to generate sentiment features on the title and body of articles. The VADER not only tells about the *Positivity* and *Negativity* score but also tells us about how positive or negative a sentiment is as shown in Figure 1.

[2] https://github.com/cjhutto/vaderSentiment

Title
Liberals wailing about gun control, but what about abortion?
{'h_neg': 0.175, 'h_neu': 0.825, 'h_pos': 0.0, 'h_compound': -0.1779}

Body
In response to Joyce Newman's recent letter about a conversation about guns: According to the National Right to Life Organization, approximately 600,000 babies are murdered ....
{'b_neg': 0.2, 'b_neu': 0.79, 'b_pos': 0.01, 'b_compound': -0.9831}

Figure 1: An example of sentiment analysis.

Vocabulary richness and readability features: We also extract features indicating article understandability. These features include several vocabulary richness and readability scores, including the Brunet's Measure W, Hapax DisLegemena, Hapax Legomenon, Honores R Measure, Sichels Measure, Yules Characteristic K, Dale Chall Readability Formula, Flesch Reading Ease, Gunning Fog Index, Shannon Entropy, Simpson's Index etc[3]. Among this index, Simpson's index stems from the concept of biodiversity. We apply this index to measure the diversity of a text.

$$\text{Simpson's Index } (D) = \sum (n/N)^2$$

N = total number of words in a text

n = total number of unique tokens

Term features: Hyperpartisan news uses their language strategically despite the attempt to control what they are saying. This language occurs with certain verbal aspects and patterns of pronoun, conjunction, and negative emotional word usage. Based on this assumption, we extract term count features which count synonyms of several terms (e.g., to obtain the ORDER term Feature, we calculated the frequency of words such as *command, demand, instruction, prescription, order* in each article).

Grammar transformation: Analysis of the content-based approach is often not enough in predicting hyperpartisan news. Thus, we adopt language structure (syntax) to predict this task. We use spaCy tool [4] to transform news articles into a set of parse tree describing syntax structure.

Psycholinguistic features: For psycholinguistic features, we use the 2015 Linguistic Inquiry and Word Count (LIWC[5]) lexicon to extract the proportions of words that belong to the psycholinguistic categories. LIWC has two types of categories; the first kind captures the writing style

[3] https://en.wikipedia.org/wiki/Readability
[4] https://spacy.io
[5] http://liwc.wpengine.com/

of the author by considering features like the POS frequency or the length of the used words. The second category captures content information by counting the frequency of words related to some thematic categories such as affective processes(e.g., *positive emotion, negative emotion, anxiety, anger, sadness*), social processes (e.g., *family, friends, female references, male references*), etc. Regarding the use of this tool, we focus on the content information, and consequently, we decide to ignore the style categories.

Part-of-Speech (POS) tags: Syntactic features consist of function words and part-of-speech tags. Syntactic pattern varies significantly from one author to another. These features were extracted using more accurate and robust text analysis tools (i.e., part-of-speech taggers, and lemmatizers). In our system, we expand the possibilities of word-level analysis by extracting the utilities of features like POS frequency. For the extraction of syntactic features, we used NLTK POS tagger[1].

Word2Vec features : Recently, word representation model (e.g., *word2vec, GloVe*) based on neural networks which represents a word into a form of a real-valued vector have increased popularity (Mikolov et al., 2013). These approaches proved to be advantageous in many NLP tasks, such as Machine Translation, Question Answering, Document Classification, to name a few. We adopted a pre-trained 300-dimensional word vector [6] to create a vector representation of the article, with an average word2vec. Besides, we use the word2vec feature to extract the cosine similarity value between the news title and the text.

TF-IDF features: Finally, We extract unigrams, bigrams, and trigrams derived from the bag of words representation of each news article. To account for occasional differences in content length between train dataset and test dataset, these features are encoded as tf-idf values. We limit the number of features that the vectorizer will learn to 10,000 features.

2.3 Learning Algorithms

Based on the above multiple features, we explore several learning algorithms to build classification models. We adopt the average weighted value for effective learning between GBDT for the style-based and content-based features and the N-gram CNN model. (see Figure 2).

[1]https://www.nltk.org/

[6] https://code.google.com/archieve/p/word2vec/

Layer	# of layers		hyperparameters
Embedding	1	l	5000
		d	300
Convoulution	3	m	[500,500,500]
		w	[3,4,5]
		w	max
Dense Layer	2	t	128
		o	2

l: max sequence length d: embedding dimension
m: filter w: kernel size
w: max-pooling t: dense unit size
o: softmax

Table 2: N-gram CNN model hyperparameters.

For deep learning model, we adopt N-gram CNN model proposed in (Shrestha et al., 2017). As shown in Figure 2 (right), the model receives a sequence of character n-gram as input. These N-gram are then processed by four layers: (1) an embedding layer, (2) a convolution layer, (3) a max-pooling layer, and (4) a softmax layer. We briefly sketch the processing procedure.

The network takes a sequence of character bigrams $x =< x_1, ..., x_l >$ as input, and outputs a multinomial over class labels as a prediction. The model first look up the embedding matrix to generate the embeddings sequence for x (i.e., the matrix C), and then pushes the embedding sequence through convolutional filters of three bigram-window sizes $w = 3, 4, 5$, each yielding m feature maps. We then apply the max-pooling to the feature maps of each filter, and concatenate the result vectors to obtain a single vector y, which then generate a prediction through the softmax layer.

Based on this model, we modified the network by adding a dense layer which helps detect hyperpartisan news features. After the experiment, the result shows that the character bigram CNN model outperforms the unigram CNN model. Table 2 summarizes the sizes of various parameters included in the N-gram CNN model. The official evaluation measure for subtasks A is *accuracy*.

Table 3

3 Experiments and Results

3.1 Datasets

The statistics of the datasets provided by SemEval 2019 task 4 (Kiesel et al., 2019) are shown Table

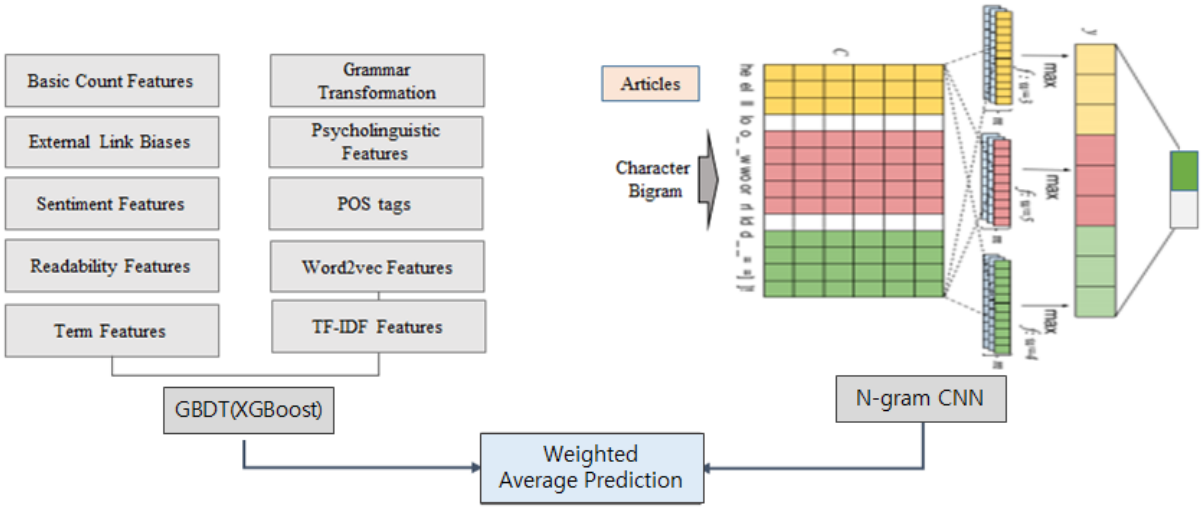

Figure 2: Hyperpartisan news detection model.

Subtask A	Hp(%)	NHp(%)
train (645)	238(36.9)	407(63.1)
Subtask B	Hp(%)	NHp(%)
train (600k)	300k(50%)	300k(50%)
valid (150k)	75k(50%)	75k(50%)

Table 3: Statistics of data sets in SemEval 2019 Task 4 Hp: hyperpartisan news; NHp: non-hyperpartisan news.

3.

3.2 Experiments on the Train Dataset

We conduct several experiments on each feature set to explore predictive separately. In these experiments, we use the GDBC (i.e., XGBoost) for the above feature set. For comparison with the N-gram model, we used the Char-level CNN model (Kim et al., 2016). The objective function was minimized through stochastic gradient descent over shuffled mini-batches with Adam(Kingma and Ba, 2014).

The performance is evaluated using 5-fold cross validation with accuracy and F-score. Table 4 lists the experimental results for each feature set on the training dataset. The prediction model through the incorporation of the entire feature showed higher accuracy than the prediction model for the individual feature.

3.3 Experiments on the Test Dataset

Our submission results to the subtask A on TIRA (Potthast et al., 2019)–the web service platform to facilitate software submissions into virtual machine– achieve an accuracy of **0.745** (precision: 0.853, recall: 0.592, F1: 0.6999). We ranked the 14th for subtask A in terms of accuracy. The prediction results of the test data are lower than the results of the training set, especially gains huge gap between precision and recall score.

Features (# of features)	**Acc**	**F1**
Count features (10)	0.6977	0.60
External link bias (3)	0.6512	0.60
Sentiment features (8)	0.6124	0.61
Readability features (14)	0.7442	0.74
Term features (44)	0.6512	0.65
Grammar transformation (45)	0.7829	0.78
Psycholinguistic features (54)	0.7984	0.79
POS tags (36)	0.7132	0.72
Word2vec features (301)	0.7752	0.77
TF-IDF (10,000)	0.7364	0.73
Char CNN (unigram)	0.7442	0.73
N-gram CNN (bigram)	0.7752	0.78
All Features (train dataset)	**0.8450**	**0.84**
All Features (test dataset)	**0.7450**	**0.70**

Table 4: Experimental results on the subtask A dataset.

4 Conclusion

Using a combination of the style-based approaches, the content-based approaches, and the N-gram CNN model, we construct the model for detecting hyperpartisan news. For this ends, we extract a broad number of linguistic features and employ GBDT model to make predictions. Finally, we adopted the weighted average value for effective learning between the two models.

References

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 *In Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval 2019)*. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

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. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 669–674.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.