Uncovering Text Features Embedded in Hyperpartisan News

Xuejun Ryan Ji
Department of ECPS, MERM
x.ryan.ji@ubc.ca

Chiyu Zhang
iSchool
zcy94@outlook.com

1 Introduction

Hyperpartisan news poses a major challenge to democracy in modern society, and always often bears biased and even polarized viewpoints against one party while excessively favouring another party. Therefore, we hypothesize that hyperpartisan news would bear its own text features. To be more specific, compared with mainstream news, it would have its own frequently used vocabularies, central topics, and styles. In short, this study aims to identify those typical text features of hyperpartisan news via a set of text analytic methods.

2 Related Works

There are three main types of widely used approaches in identifying the hyperpartisan news (Potthast et al., 2017): knowledge-based, context-based, and style -based. Knowledge-based approach mainly refers to using information retrieval to detect the degree of inconsistency between online claims and questionable relevant documents (Etzioni et al., 2008) and invoking information network to obtain knowledge graph (Shi and Weninger, 2016). The knowledge-based approach is, essentially, a fact-checking. However, the accuracy is highly susceptible to the credibility of data source and easily vulnerable to data manipulation (Potthast et al., 2017; Ginsca et al., 2015; Heindorf et al., 2016). Context-based approach mainly refers to using social network analysis to detect the false information via metadata, and to identify dissemination patterns through social media (Long et al., 2017). However, this approach requires large size of data, and as a posteriori method, it does not take the status quo and actual content of data into account. Style-based approach includes a set of methods such as rhetorical structure features extractions, inconsistency of stance detections, text categorization (Potthast et al., 2017). Likewise, this type of approach is also susceptible to data manipulation and generalizability issues. However, most of previous studies did not focus on the content features of hyperpartisan news. The present study further one more step with a set of text analytic methods to unmask the profiles and characteristics of hyperpartisan news.

3 Methods

3.1 Data Source and Description

Hyperpartisan news detection is the 4th task of SemEval 2019 [1], with the aim to identify whether given news articles display any hyperpartisan features. We used the hyperpartisan news dataset released by the organizers of this task. For heuristic purpose, we used only 10, 000 articles for our current study. Each article has been manually labelled as either hyperpartisan news (1) or mainstream news without party tilts (0; see Table 1).

Label	Hyperpartisan	Non-Hyperpartisan
Number	5,766	4,234

Table 1: Descriptive Statistics of Labeled Articles

3.2 Models

Before conducting text analytic models, we preprocessed the data with a set of feature engineering and selection techniques: 1) we employed domain-specific word embedding, to be more specific, word2vec to convert text to numeric vectors. The obtained word2vec dictionary was, then, used to identify the pools of words, which are semantically similar to the top-n importance features; 2) we also used bags of words, n-grams and TF-IDF

[1] https://pan.webis.de/semeval19/semeval19-web/

for word representations to form words frequency cloud for preliminary exploration; 3) based on TF-IDF scores, Random forest model was used to identify the overall important features of news text; 4) we, then, identified the unique features of hyperpartisan and mainstream news by matching the feature importance with the TF-IDF scores in each type of news; and 5)topic modeling was finally conducted to identify particular news topics related to political orientation.

3.2.1 Data Pre-processing

During data pre-processing for word2vec, we lowercased all text and tokenized articles into words. To obtain cleaner version, we removed special characters and white spaces as well as spelled out all the common word contractions. However, we did not remove stopwords for word2vec. The final dataset is a list of lists of tokenized words because all the articles form the list and each article contains a list of tokenized words. Likewise, in data pre-processing for Bag of Words, n-grams models, and TF-IDF, we joined the tokenized words in each article back to a string in the final stage. Therefore, the final obtained dataset is a list of strings. In other words, each original article was converted to a string of words.

3.2.2 Bag of Words (n-grams) and TF-IDF

We used Bag of Words, n-grams and TF-IDF to vectorized words. In this way, we considered each unique single word as a feature. In addition, we also included n-grams to take into account phrases or words collection in a sequence. To avoid overshadowing from the highly frequent terms across documents, we used TF-IDF (Term Frequency-Inverse Document Frequency) model, a scaled version of raw Bag of Words model. As a result, each term has a TF-IDF score in each document. We ran all of these word vectorizing models via sklearn. When modeling Bag of Words, we set the number of features to be 1000, which resulted in 1000 terms in the output. More specifically, the obtained vocabulary only included the top 1000 terms ordered by the term frequency across the documents. The min_df was set at 50 to ignore the terms that appear in less than 50 documents.In addition, we also ran n-grams models with ngram_range (2, 4), indicating n-grams models are bi-gram, tri-gram, and quadri-gram. The n-gram frequency was visualized via word cloud. As for the TF-IDF parameter setting, we used a vocabulary of top frequent 1000 terms to represent each document. We also set min_df =2 , which indicated that the terms appeared in less than 2 percentage of documents were ignored. The obtained TF-IDF scores were, then, used for random forest to get the feature importance ranking.

3.2.3 Word Embeddings

We ran word2vec with gensim based on the unique words across 1,000,000 articles. During the hyperparameters setting, we set the size of features to be 300, which means we used 300 dimensions to represent the features. The window size of 10 indicated that 10 consecutive words before and after the feature word were considered during the training. Moreover, we also used batch training due to the large sample size of data. We set each batch to include 10,000 articles for training, and within each batch, the minimum word count is set at 50, which defines that the word will be is ignored during the training if its count frequency is less that 50. To enhance the global knowledge from training, we set epoch at 3, namely, we trained the word2vec model with 3 iterations. The final size of vectorized word obtained from word2vec model is 38,084.

3.2.4 Topic Modeling

Latent Dirichlet Allocation. We set the number of topics to be 10, pre-specifying 10 components. The maximum number of iterations is set to 5 (max_iter =5). Learning method is set as online online variational Bayes method along with learning offset parameter of 50. Random seed is set to 0 (random_state=0).

Non-Negative Matrix Factorization (NMF). Likewise, we set the number of topics for NMF model to 10. The nndsvd (nonnegative double singular value decomposition) was chosen for initializing the procedure. The l1_ratio was set to 0.5, which means the penalty is a combination of element wise L1 and L2. The constant (alpha) for multiplying the regularization terms is set to 0.5. Finally, the random seed is set at 1 (random_state =1).

3.2.5 Random Forest

Random Forest is an ensemble of decision trees that are individually created by means of drawing randomly the exact number of samples from the original training data (Breiman, 2001). We ran Random Forest via sklearn, based on the Bag of

Words model. The number of estimators for random forest classifiers was set to 100, i.e., 100 assembled decision trees models were required to run during modeling. The TF-IDF dataset obtained in the 3.2.2 section was used for Random Forest, with split ratio of 70% for training set and 30% for testing set. In the same way, the label dataset also followed the 70/30 split rule.

Random Forest with TF-IDF Ranking Matching. This time, we re-ran the Bag of Words by setting the parameters of CountVectorizer function as follows: binary = True, which means the value 1 refers to the occurrence of the event rather than the count of event. As a result, the X_vec was the sparse matrix with 10000(i.e, length of documents) by 95593 (i.e., length of unique words) dimensions to represent the text information. Then, the following Random Tree model was based on the word count matrix with binary transformer. The node split rule was based on entropy. The maximum depth of each decision tree was set as 55. The number of decision tree was set to 260 (n_estimator = 260). The number of candidate feature was fixed at 145. Ten-fold cross-validation was used for modeling training and validation.

4 Results

4.1 Bag of N-Grams

To capture a general picture of the news content, we created word clouds for bag of n-grams. The bigger size the words are, the higher frequent they would be. As shown in Figure 1, the top frequent word phrases are white house, continue reading, health care, north korea, prime minister, new york times, human rights, supreme court, hilary clinton, and president donald trump. Among these top frequent words, continue reading is nonsensical, since this phrase does not contain concrete meaning as other term features are. The high frequency of continue reading might be due to the original data file, which should be removed during the data preparation. However, as displayed in Figure 1, the collections of news for our present study are more likely to be political news, rather than entertainments, arts, sports or other genres.

4.2 Random Forest and Features Importance Ranking

We conducted a Random Forest model with the word vectors obtained from TF-IDF. With 70/30

Collocation	Frequency
white house	1,833
continue reading	1,740
health care	1,112
north korea	949
prime minister	868
new york times	807
human rights	794
supreme court	772
hillary clinton	768
president donald trump	760

Table 2: Most Frequent N-gram collocation

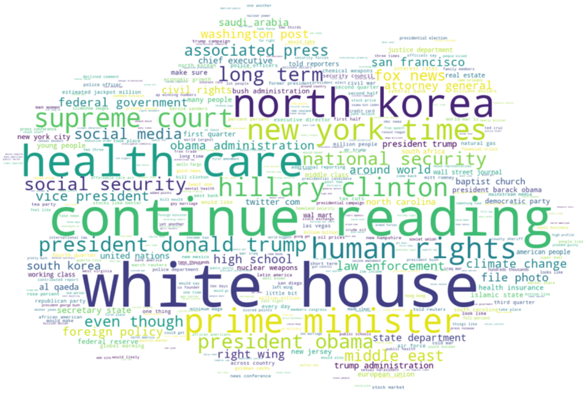


Figure 1: Word Cloud of N-gram collocation

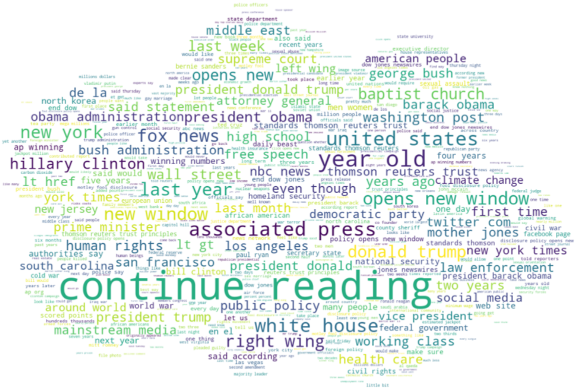
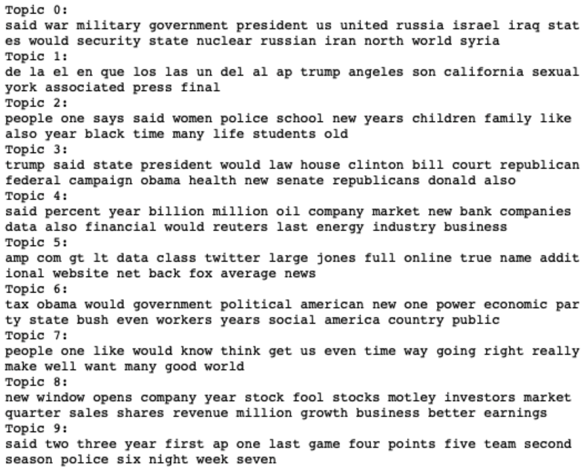


Figure 2: Word Cloud of Importance Ranking

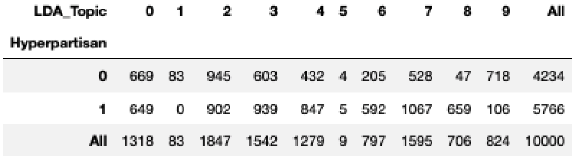
split rule for training and validation, the model accuracy rate is 0.804, meaning about 80 out of 100 events were correctly classified. The word reading and continue were ranked as a few most important features, which was consistent with Bag of n-grams, and should be removed during the data preparation. Other importance features ranked by random forest were ap, advertisement, obama, Baptist, and other function words like said, says and even. Compared with results of bag of n-grams frequency ranking, random forest did not give a clean and sensible picture of the collections of news included in our present studies.

We conducted another Random Forest model, but utilized the word vectors obtained from n-gram models. With 70/30 partition rule for training and validation, the model accuracy rate is 0.761. As shown in Figure 2, the top-ranking results included some nonsensical phrases such as continue reading, open new window, new window. These words were imbedded in the original news publishers websites. Other concrete words, such as white house, united states, and fox news, indicate that again the collection of news was mainly related to politics. It is worthy to note that ap refers to associated press appeared in both feature importance ranking, indicating that the news source might be largely stem from Associated Press(a US-based long-standing news agency).



```
Topic 0:
said war military government president us united russia israel iraq stat
es would security state nuclear russian iran north world syria
Topic 1:
de la el en que los las un del al ap trump angeles son california sexual
york associated press final
Topic 2:
people one says said women police school new years children family like
also year black time many life students old
Topic 3:
trump said state president would law house clinton bill court republican
federal campaign obama health new senate republicans donald also
Topic 4:
said percent year billion million oil company market new bank companies
data also financial would reuters last energy industry business
Topic 5:
amp com gt lt data class twitter large jones full online true name addit
ional website net back fox average news
Topic 6:
tax obama would government political american new one power economic par
ty state bush even workers years social america country public
Topic 7:
people one like would know think get us even time way going right really
make well want many good world
Topic 8:
new window opens company year stock fool stocks motley investors market
quarter sales shares revenue million growth business better earnings
Topic 9:
said two three year first ap one last game four points five team second
season police six night week seven
```

Figure 3: Example of Topic Modeling



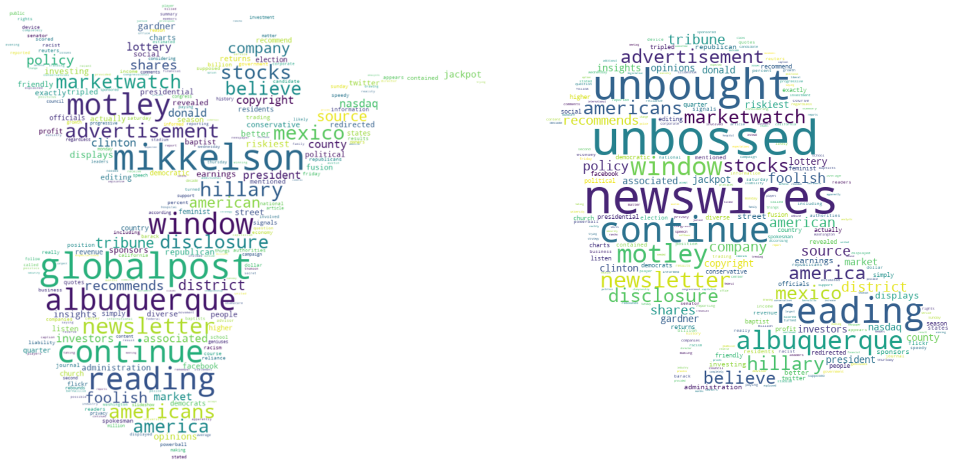
LDA_Topic	0	1	2	3	4	5	6	7	8	9	All
Hyperpartisan											
0	669	83	945	603	432	4	205	528	47	718	4234
1	649	0	902	939	847	5	592	1067	659	106	5766
All	1318	83	1847	1542	1279	9	797	1595	706	824	10000

Figure 4: LDA Topic Modeling

4.3 Topic Modeling

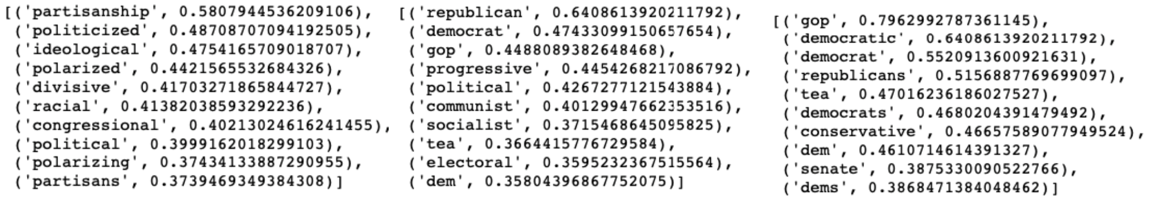
We ran Topic Modeling with Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) to cluster documents. LDA was conducted on word sparse matrix with ten assigned topics. Figure 3 presented the top ten representative features for each topic. Figure 4 displayed the topics by labels cross-table results based on LDA. As for the values of hyperpartisan label, 0 refers to non-hyperpartisan news, 1 to hyperpartisan news. As shown in Figure 4, the news belonging to Topic 1 was exclusively classified into non-hyperpartisan news. However, Topic 1 did not explicitly express concrete meaning based on the Figure 3, with exception for typical Latin language-based terms such as de, el, en, que, la and so forth. Therefore, we speculated that the Topic 1 news was mainly related to the events occurring in Latin and Francophone countries. By closely examining the topics by features result of LDA in Figure 4, Topic 0 could be interpreted as foreign affairs, middle-eastern, and wars. Topic 2 might be related to police power abuse or racial tension. Topic 3 had clear feature patterns of election, and internal affairs. Topic 4 was mainly related to financial and oil industry. Topics 6 and 8 were associated with economics and stock market. However, Topics 5, 7 and 9 did not show very clear features patterns. When checking topic labels as shown in Figure 4, the apparent hyperpartisan news can be found in Topics 2, 4, 6, 7, 8. In other words, we could summarize that news related to racial tension, financial and economic industry was more likely to be hyperpartisan or polarized.

To confirm the LDA findings, we also utilized Non-negative matrix factorization (NMF) for topic modeling. Topic 1 might be associated with stocks, investigation. Topic 2 was related to election campaign, and Russian interference. Topic 3 mainly referred to oil prices and Chinas market. Topic 4 indicated the news content was associated with abuse of police power and legal issues. Topic 5 involved middle eastern wars and foreign affairs. Topic 7 might be about the French or Latin American Arts or cultures. Topic 8 might be related to internal affairs like tax, insurance, or health care coverage. Topic 9 mainly concerned elec-



(a) Word Cloud for Non-hyperpartisan News (b) Word Cloud for Hyperpartisan News

Figure 5: Word Cloud of Top Features



```
[('partisanship', 0.5807944536209106),
 ('politicized', 0.48708707094192505),
 ('ideological', 0.4754165709018707),
 ('polarized', 0.4421565532684326),
 ('divisive', 0.41703271865844727),
 ('racial', 0.41382038593292236),
 ('congressional', 0.40213024616241455),
 ('political', 0.3999162018299103),
 ('polarizing', 0.37434133887290955),
 ('partisans', 0.3739469349384308)]
```

(a) Partisan.

```
[('republican', 0.6408613920211792),
 ('democrat', 0.47433099150657654),
 ('gop', 0.4488089382648468),
 ('progressive', 0.4454268217086792),
 ('political', 0.4267277121543884),
 ('communist', 0.40129947662353516),
 ('socialist', 0.3715468645095825),
 ('tea', 0.3664415776729584),
 ('electoral', 0.3595232367515564),
 ('dem', 0.35804396867752075)]
```

(b) Republican.

```
[('gop', 0.7962992787361145),
 ('democratic', 0.6408613920211792),
 ('democrat', 0.5520913600921631),
 ('republicans', 0.5156887769699097),
 ('tea', 0.47016236186027527),
 ('democrats', 0.4680204391479492),
 ('conservative', 0.46657589077949524),
 ('dem', 0.4610714614391327),
 ('senate', 0.38753300090522766),
 ('dems', 0.3868471384048462)]
```

(c) Democratic.

Figure 6: Example of Word Similarity

tion campaign without mentioning Trump. And finally, Topics 0 and 6 did not show any particular features patterns. The term features for each topic were similar to those in LDA. In short, similar to the features-document matrix based on LDA, when news was related to election campaign, economic and financial market, foreign affairs, internal affairs, it would tend to be hyperpartisan.

4.4 Random Forest with TF-IDF Ranking Matching

We were also interested in identifying particular features for each labeled set, namely, the representative features for hyperpartisan news and non-hyperpartisan news. We re-ran the Random Forest again with ten-fold cross-validation. The average model accuracy rate was 0.773. Then we matched each TF-IDF scores for features of each labeled set with the feature importance ranking score obtained from the Random Forest.

As shown in Figure 5 , there were no clear differences on the feature patterns between hyperpartisan and Non-hyperpartisan news. In other words, there were no clear differences at the lexical level in terms of the news political orientation.

4.5 Word2Vec

We also attempted to identify the similarity among the top features using word2vec. However, because the top-ranking features in overall or labeled set did not indicate any apparent feature patterns. Then, we just used partisan, republican and democratic to test the quality of trained word2vec. As shown in Figure 6, the domain-specific word2vec was well trained, and successfully identified the associated words. For example, the words related to partisan were politicized, ideological, polarized, divisive and so forth.

5 Conclusion and Discussion

The present study aimed to identify the text feature patterns of a collection of news with a set of text analytic tools. In addition, we also attempted to identify the text features of each type of labeled news articles, especially hyperpartisan vs. non-hyperpartisan news articles. In doing so, we analyzed the features in both lexical and document level. More specifically, we utilized Bags of n-grams, TF-IDF, and Random Forest to identify the representative features of the collections of news article. We found that only n-grams model resulted in more sensible results than other methods, and the results indicated that the collection of 10,000 news articles was more likely to be related to politics based on the high ranking key features such as white house, health care, north korea, human rights, and hiliary clintion, and president Donald trump. In contrast, based on the results from Random Forest with TF-IDF Ranking Matching, we conclude that singe lexical units might not be a good information unit to represent the features of the news articles. For example, unbought and unbossed do not seem to be sensible as presented as two separate words. The Unbought and Unbossed is actually a book name authored by Shirley Chisholm. Another example are the motley and fool. The Motley Fool is actually a multimedia financial services company. It seems that those phrases would appear much less frequently in the articles. However, the TF-IDF overrated the importance of these keywords with less frequency. In stark contrast to the lexical text mining methods, the documents-level text analytics such as Topic modeling resulted in more sensible classification solution. We ran both LDA and NMF based on 10,000 news articles. The common findings from both models indicated that when the news article topics were specifically related to foreign affairs economics and financial investment, power abuse of police, and racial tension, news articles were more likely to be classified as hyperpartisan. Admittedly, we successfully identified the particular topics related to hyperpartisan news. However, the present study still bears a few limitations. First, the present study was conducted based on only 10,000 news articles. Second, we did not run optimizations for models used in the study. Third, the hygiene level of data provided by the competition organizers is not sufficiently high. Closer scrutiny of data is needed during the data pre-processing to remove those nonsensical words such as continue reading and open windows. Fourth, when conducting topic modeling, we only fixed 10 components. There would be other potential models needed to be tested.

6 Future Direction

The present study is focused on the application of computational linguistics to uncover the information hidden in the massive data. And it is part of our ongoing hyperpartisan news detection project. The results from this study will be also used for another classification task with deep learning algorithms we are currently working on. The text features identified in this study can provide a more concrete insights of hyperpartisan news. For future study, we would enumerate the number of components in the topic models. Model evaluation would be conducted to obtain the optimal number of components (Wallach et al., 2009). Moreover, because n-grams gave more sensible in feature extraction, we would attempt conduct topic modeling with n-gram to obtain more interpretable topics clusters. Finally, we might also conduct LDA2Vec (Moody, 2016) to taken the context into account.

References

Leo Breiman. 2001. Random forests. *Machine learning* 45(1):5–32.

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

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

Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism detection in wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pages 327–336.

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

Christopher E Moody. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019* .

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* .

Baoxu Shi and Tim Weninger. 2016. Fact checking in heterogeneous information networks. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 101–102.

Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*. ACM, pages 1105–1112.