Media Bias Detecting based on Word Embedding

Yihan Geng*

Department of Systems Engineering and Engineering Management, the Chinese University of Hong Kong, Hong Kong, China

*Corresponding author: 1155141668@link.cuhk.edu.hk

Abstract. Machine learning research to detect political bias in articles has boomed in recent years. However, there is still no widely accepted and effective word embedding technique for detecting bias. This paper explores the connection between political bias and word embedding models and deduces factors to consider when selecting and developing word embedding techniques. Three classic word embedding models are introduced into experiments to conduct comparisons to achieve this goal. Contextual meaning is observed to lose efficiency in the task. In contrast, frequency is the most relevant feature in predicting media bias. Simultaneously, this paper discovers a unique accuracy distribution generated by Random Forest through experiments. Experiments reveal that it has apparent advantages in accuracy when predicting left-biased articles, which may relate to features undiscovered.

Keywords: Natural language processing, media bias, machine learning, word embedding, news slant.

1. Introduction

As conflicts intensify in many aspects around the world, the political stances of news have gradually become an essential criterion for readers to evaluate the quality of the content. Typically, in the United States, there are two main political divides, liberal and conservative, with the most biased news falling into these two camps. In recent years, biased media content has been observed to fuel polarization and hostility between supporters of both parties [1]. The growing concern about such media bias has led to many studies evaluating and flagging the content through machine learning.

Various methods have been applied from sentence-level to article-level to detect media bias. M.Vu mainly applied Recurrent Neural Networks (RNN) and Multilayer Perceptron (MLP) in analyzing political bias at the sentence level [2]. P. Patil et al. also conducted experiments based on RNN at the article level [3]. W. Chen et al. described a detailed classification procedure from sentence-level to article-level with different classifiers, including Support Vector Machine (SVM), Logistic Regression, and BERT [4, 5]. Other word embedding techniques have been introduced into practice, apart from varying biases and models. Inverse document frequency (IDF) is employed [6, 7]. Meanwhile, S. Gerrish et al. implemented bag-of-words (BoW) to conduct experiments [8]. Additionally, word choice and labeling (WCL) is used by [9].

Moreover, there are concerns about whether datasets used for machine learning are inherently biased. T. Spinde et al. discussed systematic approaches to devise media bias datasets with high reliability [10]. Moreover, several datasets have been introduced into practice over the past few years. It is common practice to use college student judges for grading concerning reliability. In contrast, workers from crowdfunding sites accomplish the dataset presented in [11]. It was proved to achieve similar reliability with college student judges. Researchers proposed a dataset including the detailed characteristics of media bias and corresponding information of human judges in [12].

Although studies have been conducted from various aspects, there is still a lack of a widely accepted standard for word embedding techniques in this field. Apart from self-customized models and high-cost-high accuracy models like BERT, common approaches such as bag-of-words and paragraph vectors have not yet been compared and studied the differences. This paper tries to detect media bias at the article level using three word embedding techniques: TF-IDF, Word2Vec, and Doc2Vec. A comparison of these word embedding techniques is carried out in this paper.

The remainder of this paper is organized as follows: in the methods section, this paper discusses the word embedding techniques and machine learning models used in this paper. The experiment section presents the data preprocessing, Experiment results, and corresponding analysis. The Discussion section conveys the limitations of current work and discusses the potential future work. In the conclusion section, the research process and findings are summarized.

2. Methods

In this section, we introduce the methods applied to detect media bias. First, we describe different word embedding techniques. Then, we discuss the models implemented to conduct machine learning.

2.1. Word Embedding

We implemented multiple word embedding techniques to provide different views of documents, including TF-IDF, Word2Vec, and Doc2Vec. TF-IDF is known as the term frequency of inverse document frequency. TF-IDF is the multiplication result of the word frequency in a document and its information volume across all documents, expressed by a logarithmic function. It only shows the word frequency distribution regardless of the contextual meaning. The weight of the i-th word in the j-th document is defined by:

$$\omega_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \times \log\left(\frac{N}{df_i}\right) \tag{1}$$

where ni,j denotes the number of times the i-th word appears in the j-th document, k denotes the number of distinct words in all texts, N denotes the number of documents in the dataset, and dfi denotes the number of texts which contains the i-th word.

In 2013, Word2Vec became a Natural language processing (NLP) model for word embedding generation [13]. It comprises a shallow two-layer neural network that digests a given corpus to produce vector space with different words in the data. Word2Vec contains two model architectures, Continuous Bag-of-Words (CBOW) model and the Continuous Skip-gram model. We chose the latter because it usually performs better than the CBOW model when the size of the dataset is large. Furthermore, Word2Vec does not explicitly generate vectors for documents. It outputs word vectors by a given the word, which predicts the occurrence probability of surrounding words. Hence, we applied the typical approach that calculates the mean of word vectors derived from the vocabulary of a document.

Doc2Vec is an extension of Word2Vec, introducing paragraph vectors into word embedding construction. Its new features enable it to generate document vectors explicitly and absorb contextual information into their creation. It includes two methods, known as the paragraph vector with distributed memory (PV-DM) and the paragraph vector with a distributed bag of words (PV-DBOW) [14]. Q. Le et al. concluded from experiments that PV-DM performs better than PV-DBOW consistently [14]. In contrast, the combination of two methods usually has better performance. Thus, for our training task, we generated the concatenation of two groups of vectors encoded with these two methods.

2.2. Training Models

Since we use a labeled dataset to perform machine learning, two supervised machine learning models are implemented in the experiment, including Support Vector Machine (SVM) and Random Forest. SVM is a supervised machine learning model which possesses high effectiveness in high dimensional spaces. This advantage gives rise to the general use of this model in NLP tasks. SVM aims to tackle the optimization problem:

$$\min_{\omega,b,\zeta} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\zeta_i$$

$$s.t.\, y_i(\omega^T \Phi(x_i) + b) \geq 1 - \zeta_i \quad (2)$$
$$\zeta_i \geq 0, \quad i = 1, \ldots, n$$

where $\omega^T\omega$ denotes the normal vector which creates a line passing through the origin of the coordinates, C denotes a regularization parameter which controls the level of penalty, ζi denotes the distance to the correct margin, y_i denotes the i-th target value, b denotes the bias parameter, and $\Phi(x_i)$ denotes the kernel function which performs the transformation from the original input space to a higher dimension.

SVM aims to find a hyperplane which separates two classes with best accuracy. Although it is initially designed for dual problems, it also works well in multiclass by either One-to-One or One-to-Rest approach. The logic is as follows:

Suppose there are m classes in total, denoted by R = {$class_1$,$class_2$, ... ,$class_m$}. The i-th one-to-Rest classifier will try to find the hyperplane which separates the i-th class from all other classes. It requires m SVM classifiers in total to combine their prediction results and accept the majority as the final output.

Meanwhile, the One-to-One classifier requires to construct an SVM classifier for every single pair of classes in R. Thus, it requires $\frac{m(m-1)}{2}$ SVM classifiers in total.

We chose the One-to-Rest classifier in our experiment concerning the time and space complexity. Previous investigations have proved the feasibility of SVM in media bias classification. We decided to use this classic base model in our experiment [4].

Random Forest is also a supervised machine learning model used in NLP tasks. It constructs a set of decision trees to provide the prediction results by selecting the output chosen by most decision trees. Random Forest strengthens its accuracy by diversifying the individual trees. A lower correlation between trees usually leads to higher accuracy. Ensuring such diversity is called bagging, which randomizes the selection of samples for each tree from the dataset with replacement.

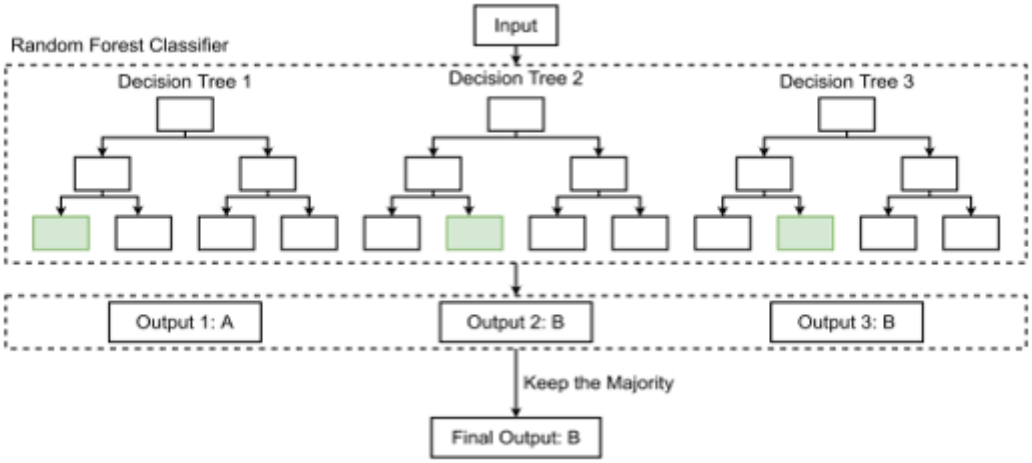


Figure 1. Workflow of Random Forest Classifier.

The logic of Random Forest appears to be analogous to categorizing media bias in news articles manually since the prediction results seem fairer and more accurate when enough judges with low correlation are involved. Thus, we integrated a Random Forest classifier into the experiment to demonstrate the feasibility of this model in detecting media bias.

3. Experiment results and analysis

In this section, we demonstrate the detailed experiment results step by step, from data preprocessing to the final result analysis.

C. Budak et al. presented a dataset that includes 10,502 articles manually classified by 749 human judges [11]. It conducted several tests to qualify the workers, including cross-validation, article slant

rating validation, etc. Researchers proved the interrater reliability with a lower bound of 81 percent approximately. The dataset contains the URLs of articles and the corresponding bias level on each article perceived by democrat and republican judges, respectively. The bias levels are denoted as “Negative”, ”SomewhatNegative”, ”Neutral”, ”SomewhatPositive”, and ”Positive”. We quantified the corresponding bias level ranging from -2 to 2, where negative values denote the positive impression claimed by democrats or negative impression according to republicans and vice versa. Thus, the numeric value of the overall bias can be derived from the combination result,s ranging from -4 to 4.

Table 1. Quantification of bias level.

Bias / Faction / Value	Negative	SomewhatNegative	Neutral	SomewhatPositive	Positive
Democrat	2	1	0	-1	-2
Republican	-2	-1	0	1	2

However, the original dataset only includes the URLs of articles. Five thousand five hundred twelve articles were missing or had restricted access during extraction, resulting in an imbalanced data distribution. Thus, as Figure 2 shows, we rebalanced the data with 2,251 articles on each topic. We focus on two features, text and bias, with the full derived dataset, to conduct the following experiments.

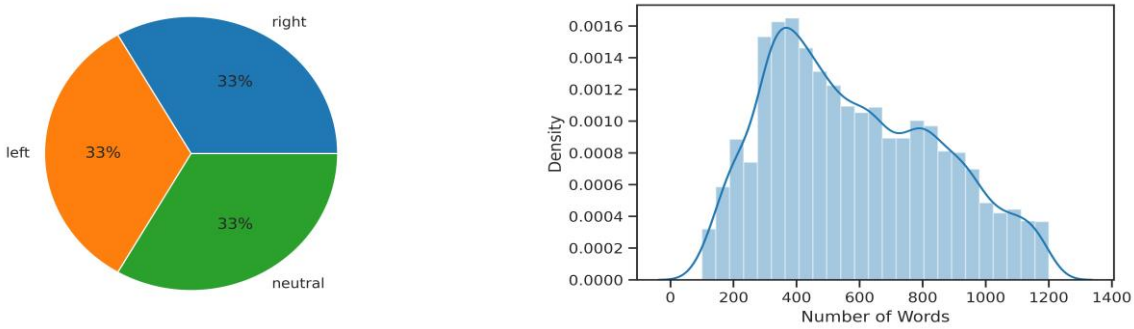


Figure 2. Distribution of media bias and text length in the dataset.

We use precision and recall to derive the F1 score of prediction results. F1 score is a balanced F-score, which is calculated by the harmonic mean of the precision and recall. Metrics including the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are implied inside the equations. The detailed equations are as follows:

$$precision = \frac{TP}{TP+FP} \tag{3}$$

$$recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1\ score = \frac{2\times precision\times recall}{precision+recall} \tag{5}$$

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

The prediction baseline is determined via an all-neutral strategy, which always produces neutral results regardless of inputs. According to the bias distribution shown in Figure 2, its accuracy is calculated as 33%. In the experiment, Random Forest classifiers have an optimal number of trees of 800. This number delivers the best results in terms of both stability and performance. Also, we determined the best C value discussed in equation (2) by grid search. The values of C are 0.004, 1.05, and 1.66 for Doc2Vec, Word2Vec, and TF-IDF, respectively. The detailed experiment result is shown in Table 2.

Table 2. Experiment results using different word embedding techniques.

Feature	Classifier	Category	Precision	Recall	F1 score	Accuracy
-	All-neutral Baseline	-	-	-	-	0.33
Doc2Vec	Random Forest	Left	0.63	0.69	0.66	0.62
		Neutral	0.61	0.65	0.63	
		Right	0.63	0.53	0.58	
	SVM	Left	0.55	0.55	0.55	0.59
		Neutral	0.62	0.67	0.65	
		Right	0.59	0.56	0.58	
Word2Vec	Random Forest	Left	0.68	0.72	0.70	0.64
		Neutral	0.64	0.64	0.64	
		Right	0.60	0.56	0.58	
	SVM	Left	0.55	0.57	0.56	0.60
		Neutral	0.65	0.69	0.67	
		Right	0.60	0.53	0.56	
TF-IDF	Random Forest	Left	0.61	0.59	0.60	0.61
		Neutral	0.59	0.63	0.61	
		Right	0.64	0.62	0.63	
	SVM	Left	0.62	0.65	0.64	0.65
		Neutral	0.63	0.68	0.66	
		Right	0.69	0.61	0.65	

The results demonstrate that TF-IDF achieved the best performance. Compared to the other two word embedding techniques, the Doc2Vec approach significantly falls in accuracy. The special relationship between news articles and political stances may account for the phenomenon. Unlike argumentative essays, the informational content of news is embedded in the framework of a storyline because facts have no intrinsic significance. It also implies that the sequential order of words is usually based on the storyline derived from the fact, but not the biased opinion. As a result, media bias is implicitly conveyed throughout the entire content. W. Chen et al. also observed that the sequential order of sentences has a weaker impact on the informational bias than the frequency and positions of biased statements [4]. We may further conclude that the prediction's paragraph vectors used in Doc2Vec produce noise. Meanwhile, the difference between Word2Vec and TF-IDF is slight, which shows the surrounding words used in Word2Vec may also produce noise but is in an acceptable range compared to paragraph vectors.

Furthermore, the Random Forest classifier outperforms SVM in Doc2Vec and Word2Vec but fails in TF-IDF. Generally, it proves the feasibility of the Random Forest classifier in media bias detection. However, Random Forest fails to outperform SVM in TF-IDF. First, it may show the instability of the Random Forest classifier. Second, the TF-IDF approach may fail to provide data with enough diversity compared to Word2Vec and Doc2Vec.

In the experiment, we also observed that Word2Vec and Doc2Vec perform better in left bias prediction. Intuitively, it may be caused by the imbalanced word distribution in documents. In this regard, imbalanced word distribution is likely caused by the imbalanced topic distribution. Left biased articles may focus more on specific issues, while the other two focus on different topics. Therefore, this leads to a more obvious distribution of related words in specific topics in this group. We visualized the topic distribution as Figure 3 shows.

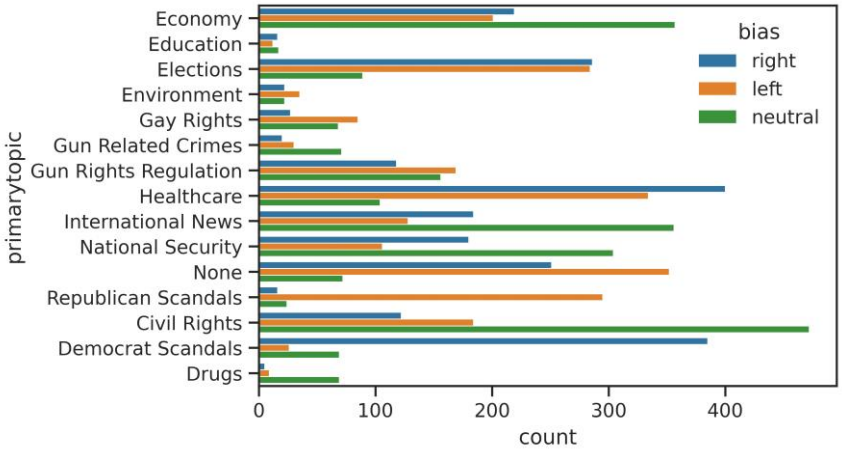


Figure 3. Topic distribution in the dataset.

The figure 3 shows that some highly imbalanced distribution exists in topics such as ”Drugs” and ”Scandals.” But we cannot confirm that it is the only reason because such a hypothesis should have shown its effect in TF-IDF. However, no specificity was found in this regard. The F1 score of left bias is slightly lower than the other two methods. This suggests that left bias articles might possess a trivial characteristic that uniquely identifies them.

4. Discussion

There still exist some limitations in our work. First, the experiment lacks some new word embedding techniques specially designed for media bias detection, such as the state-of-the-art WCL approach proposed in [9]. A comparison with the latest models would further exhibit the current progress in the study of this field. However, since our chosen techniques are efficient and typical, they also possess the potential to guide us in evaluating the direction of media bias classification. Also, some standard classifiers have not yet been covered, such as the BERT model. This model has shown its accuracy in previous research [4, 7]. However, its cost is relatively too high compared to our chosen models. To better distinguish the trivial phenomena as previously discussed, such kind of models was not included in our experiment.

Our future work aims to discover the inherent value inside the trivial difference in left bias classification results to discover whether it is caused by dataset defects or the undiscovered features inside the frequencies of the word embedding vectors. New datasets could be introduced into the experiment, such as [12]. Thus, we may avoid the possible intrinsic bias carried by the current dataset. Moreover, we hope to utilize our future work to discover how classification using multiple features may affect media bias detection. The current research stage does not involve including other features to get rid of noise and therefore explores the relationship between different embedding methods and classification results. However, other features such as primary and secondary topics in the dataset could be used to generate better prediction results.

5. Conclusions

This paper explores different word embedding techniques and machine learning models to detect media bias in news articles. Besides observing the poor performance of paragraph vectors in detecting political leanings, we also discovered a noticeable increase in the accuracy of left bias prediction, consistent in both Doc2Vec and Word2Vec using the Random Forest Classifier. The reason is unknown and to be solved, but it is not limited to the imbalanced topic distribution. Moreover, we proved the feasibility of the Random Forest classifier in detecting political bias. It performs better than SVM in two of three experiments.

In general, our study further proves the theory that although the media bias of news articles is political, the measurement can be systematic. Additionally, the experiment results suggest some possible directions distinguish left-biased and right-biased articles in our future work. We believe that our current work is helpful for researchers to select and derive their word embedding techniques for this task. We also expect more customized word embedding techniques to be derived from future studies for media bias detection.

References

[1] A.E. Wilson, V.A. Parker, M. Feinberg, (2020). Polarization in the contemporary political and media landscape. Current Opinion in Behavioral Sciences, 34, 223-228.

[2] M. Vu, "Political news bias detection using machine learning", 2017. [Online]. Available: https://pdfs.semanticscholar.org/8445/2eb068bdfe7d5809734a5da8f5c7d10bebfa.pdf.

[3] P. Patil, S. Chaudhari, A. Dhote, M. Gorde and D. Palkar, "Study of Detecting the Political Bias in News Articles", Academia.edu, 2022. [Online]. Available: https://www.academia.edu/49651419/Study_of_Detecting_the_Political_Bias_in_News_Articles.

[4] W.F. Chen, K. Al Khatib, B. Stein and H. Wachsmuth. (2020, November). Detecting Media Bias in News Articles using Gaussian Bias Distributions. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 4290-4300).

[5] J. Devlin, M. Chang, K. Lee and K. Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT (pp. 4171-4186).

[6] S. Lim, A. Jatowt, M. Yoshikawa (2018). Towards bias inducing word detection by linguistic cue analysis in news. In DEIM Forum (pp. C1-3).

[7] L. Fan, M. White, E. Sharma, R. Su, P. K. Choubey, R. Huang and L. Wang. (2019). In plain sight: Media bias through the lens of factual reporting. arXiv preprint arXiv:1909.02670.

[8] S. M. Gerrish and D. M. Blei. (2011, October). Predicting legislative roll calls from text. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011.

[9] F. Hamborg, A. Zhukova, B. Gipp. (2019, June). Automated identification of media bias by word choice and labeling in news articles. In 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL) (pp. 196-205). IEEE.

[10] T. Spinde. (2021, December). An interdisciplinary approach for the automated detection and visualization of media bias in news articles. In 2021 International Conference on Data Mining Workshops (ICDMW) (pp. 1096-1103). IEEE.

[11] C. Budak, S. Goel, J. M. Rao. (2016). Fair and balanced? Quantifying media bias through crowdsourced content analysis. Public Opinion Quarterly, 80(S1), 250-271.

[12] T. Spinde, L. Rudnitckaia, K. Sinha, F. Hamborg, B. Gipp and K. Donnay. (2021). MBIC--A Media Bias Annotation Dataset Including Annotator Characteristics. arXiv preprint arXiv:2105.11910.

[13] T. Mikolov, K. Chen, G. Corrado and J. Dean. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[14] Q. Le and T. Mikolov. (2014, June). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196). PMLR.