information

MDPI

Article

A Framework Model of Mining Potential Public Opinion Events Pertaining to Suspected Research Integrity Issues with the Text Convolutional Neural Network model and a Mixed Event Extractor

Zongfeng Zou , Xiaochen Ji * and Yingying Li

School of Management, Shanghai University, Shanghai 200444, China; zfzou@mail.shu.edu.cn (Z.Z.); loislyy@shu.edu.cn (Y.L.)
* Correspondence: jxc@shu.edu.cn

Abstract: With the development of the Internet, the oversight of research integrity issues has extended beyond the scientific community to encompass the whole of society. If these issues are not addressed promptly, they can significantly impact the research credibility of both institutions and scholars. This article proposes a text convolutional neural network based on SMOTE to identify short texts of potential public opinion events related to suspected scientific integrity issues from common short texts. The SMOTE comprehensive sampling technique is employed to handle imbalanced datasets. To mitigate the impact of short text length on text representation quality, the Doc2vec embedding model is utilized to represent short text, yielding a one-dimensional dense vector. Additionally, the dimensions of the input layer and convolution kernel of TextCNN are adjusted. Subsequently, a short text event extraction model based on TF-IDF and TextRank is proposed to extract crucial information, for instance, names and research-related institutions, from events and facilitate the identification of potential public opinion events related to suspected scientific integrity issues. Results of experiments have demonstrated that utilizing SMOTE to balance the dataset is able to improve the classification results of TextCNN classifiers. Compared to traditional classifiers, TextCNN exhibits greater robustness in addressing the problems of imbalanced datasets. However, challenges such as low information content, non-standard writing, and polysemy in short texts may impact the accuracy of event extraction. The framework can be further optimized to address these issues in the future.

Keywords: research integrity; TextCNN; SMOTE; event mining

check for updates

Citation: Zou, Z.; Ji, X.; Li, Y. A Framework Model of Mining Potential Public Opinion Events Pertaining to Suspected Research Integrity Issues with the Text Convolutional Neural Network model and a Mixed Event Extractor. *Information* **2024**, *15*, 303. https://doi.org/10.3390/info15060303

Academic Editor: Ralf Krestel

Received: 5 May 2024
Revised: 19 May 2024
Accepted: 21 May 2024
Published: 24 May 2024

1. Introduction

Public opinion was an early concern in political science. Lowell believed that public opinion refers to people's opinions on real events plus their subjective ideas. While the rapid development of the Internet has significantly changed people's lives, it has also brought new opportunities for the study of various public opinions [1]. Wang argues that online public opinion is the viewpoints or topics generated by public opinion events relying on self-media communication carriers, and that people generate many different viewpoints or topics around a specific public opinion event [2].

Prior research investigated the mining of public opinion events across various domains, focusing on both trending and technological perspectives.

Chen pointed out that online public opinion reflects people's social and political attitudes, and studying the trend prediction and evaluation of online public opinion is important for managers' decision-making [3]. Hassani et al. employed social trend mining techniques to investigate social dynamics and emerging patterns, extracting event trends through the analysis of time series data gathered from social media platforms and search engines [4].

Several studies analyzed events from the perspective of machine learning to enhance the capability of event mining. A study proposed an improved LDA module with sentiment discrimination learning capability and analyzed the sentiment intensity of the thematic arguments of different events in time series to effectively analyze online campus public opinion [5]. A study examining a case of rapid public health policy adaptation in China during the COVID-19 epidemic was carried out by employing K-means, TF–IDF, and HMM methods [6]. K-means clustering and the Baidu Application Programming Interface Gateway were used to explore why a routine government notice caused a series of unexpected public opinion crises, and results show that how the government releases information and issues clarifications significantly affects public risk perception and emotion [7]. Weng et al. used the event mining algorithm based on wavelet signal clustering (EDCoW) to process a large amount of event information from the Twitter social media platform, using the word frequency to construct word signals and filter trivial words by viewing the signals to improve the efficiency of event mining [8].

Several studies approach events from the semantic analysis standpoint, with the goal of exploring relationships between events or conducting sentiment analysis. A public opinion monitoring mechanism consisting of a semantic descriptor that relies on natural language processing algorithms was applied to the 2016/2020 US Presidential Elections tweet datasets to explore succinct public opinion descriptions [9]. Habibabadi et al. used natural language processing techniques to extract mentions of adverse events of vaccines from Twitter to gain early insights into vaccine safety issues [10]. Nallapati et al. captured the rich structure of events and their correlations in news topics through event modeling to address the problem of content loss associated with organizing news stories into a flat hierarchical structure by topic [11].

The above literature has realized the analysis of public opinion in various fields through various natural language processing techniques and event extraction methods, but few studies have conducted research on public opinion in the context of scientific research integrity.

The issue of research integrity has persisted since the inception of scientific research. Historically, due to the constraints of traditional mass media, this problem has predominantly been addressed within the scientific community itself for the purposes of self-evaluation and self-scrutiny. With the continued advancement of the Internet, the dissemination of information has reached unprecedented speeds, and the achievements of science and technology have incrementally captured people's widespread attention. The supervisors of research integrity are gradually expanding from within the scientific community to the whole of society. In addition to the traditional research integrity accusation, viewers on the Internet also may question the process and results of research by posting short texts on social media platforms. The events described in these words frequently elicit widespread discussion and possess the potential to shape public opinion. If not promptly addressed, they could potentially exert a significant influence on the oversight of scientific research integrity and undermine the credibility of research institutions and scholars. Therefore, this paper proposes a framework model based on TextCNN and a Mixed Event Extractor that is designed to mine potential public opinion event elements pertaining to suspected research integrity issues. The aim of this paper is to furnish research managers with public opinion mining tools, thereby expanding the scope of their inspection and management efforts and enhancing the development of the inspection system within the comprehensive accountability framework for research integrity.

The focus of this paper's research is on short texts related to potential public opinion events surrounding scientific integrity issues on online social platforms. Our model employs TextCNN [12] to distinguish potential public opinion events related to scientific integrity issues from common text and subsequently identify key elements through a mixed event extractor. TextCNN exhibits outstanding performance in extracting shallow features from text, rendering it an apt choice for application in short text classification tasks. Incidents of scientific misconduct in research activities are rare events, so predictably, the online short texts of potential public opinion events on scientific integrity issues represent only a small proportion compared to ordinary textual information. To avoid the impact of imbalanced datasets on TextCNN, SMOTE is utilized to process the training set. The Mixed Event Extractor, based on TF-IDF and TextRank, can more comprehensively mine important information related to potential public opinion events surrounding scientific integrity issues.

2. Methods

In this paper, we utilize the Text Convolutional Neural Network model (TextCNN) to identify texts pertaining to suspected research integrity issues, which are defined as suspicious short texts. This model is trained on a balanced dataset processed by SMOTE. Following this, we propose a Mixed Event Extractor based on TF-IDF and TextRank to extract and mine crucial event information from the short text related to research integrity issues. Figure 1 shows the flowchart of the research methodology of this paper.

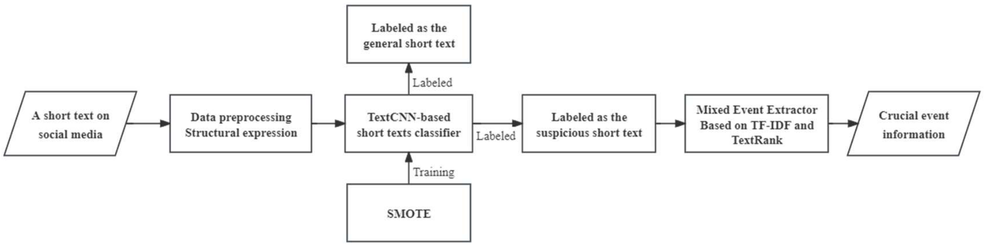
Labeled as the general short text
Labeled
A short text on social media
Data preprocessing Structural expression
TextCNN-based short texts classifier
Labeled
Labeled as the suspicious short text
Mixed Event Extractor Based on TF-IDF and TextRank
Crucial event information
Training
SMOTE

Figure 1. Flowchart of our approach.

2.1. TextCNN-Based Suspicious Short Text Classifier

2.1.1. TextCNN

Kim was the pioneer in proposing the fundamental architecture of TextCNN [12]. Figure 2 illustrates the general structure of TextCNN, which typically comprises an input layer, multiple convolutional layers, and pooling and fully connected layers, culminating in an output layer. This deep learning model leverages convolutional neural networks (CNNs) for effective feature extraction and text classification tasks. Unlike the traditional bag-of-words model, TextCNN utilizes word vectors as its input. It captures local features through multiple convolutional operations and subsequently maps these features to a low-dimensional space via pooling operations, resulting in a more efficient text representation. Finally, TextCNN classifies the text through a fully connected layer. TextCNN, as a deep learning algorithm, possesses the ability to automatically learn features within text. Simultaneously, it effectively handles high-dimensional and abstract data. Furthermore, TextCNN can process texts of varying lengths without the need to pad them to a fixed length. Its excellent performance in shallow text feature extraction renders it particularly suitable for short text classification tasks.

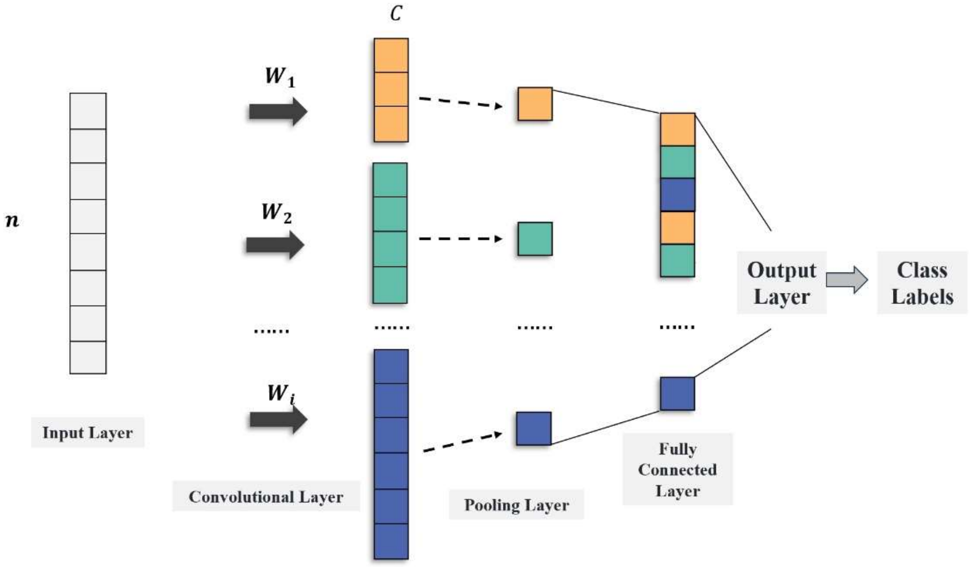

Figure 2. The general TextCNN model architecture diagram. Colors differ to indicate the outcomes of applying different convolutional kernels.

2.1.2. SMOTE for Imbalanced Data

Due to the imbalance in the dataset, the TextCNN classifier often correctly identifies the prevalent short texts belonging to the majority class during training but overlooks the infrequent short texts of the minority class. To overcome the limitations caused by the imbalanced dataset, we employ the SMOTE sampling technique, which solves this problem from a data perspective.

SMOTE is an oversampling technique that aims to achieve balance in the sample set by enhancing the quantity of minority class samples [13]. SMOTE utilizes the linear interpolation method to introduce new, additional, suspicious short texts. This method synthesizes new minority class samples at random positions based on the linear distance between two minority class samples. The primary steps to synthesize a new minority class sample using linear interpolation are as follows:

A minority class sample x_i is selected from a given minority set S_{min}, which is also a set of suspicious short texts.

The distance between x_i and all the other samples in S_{min} is calculated, then k nearest minority class samples are searched for to form a sample set M_i, $M_i \subseteq S_{min}$ and $x_i \notin M_i$. The typical value for k is often set to 5 or another odd number.

n denotes the required multiplication factor, which determine the number of synthetic minority class samples x_{new} to be generated.

x_{ij} is randomly selected from M_i, then x_{new} is obtained through (1). The parameter $\gamma \in [0,1]$ determines the position where x_{new} is generated.

$$x_{new} = x_i + \gamma\left(x_i - x_{ij}\right) \tag{1}$$

All newly synthesized x_{new} are added to the new minority class sample set S^*_{min} to obtain the new data sample set S^*.

2.2. Mixed Event Extractor Based on TF-IDF and TextRank

The aim of the event extractor is to identify important event information with semantic meaning in the text, which is further explained as identifying the keyword elements of the event "5W1H", i.e., when, where, who, what, why, and how, from a given text [14]. As suspicious short texts vary in length, it is necessary to remove redundant information and quickly capture critical information. This subsection describes how to perform further keyword extraction of events from the suspicious short text through the Mixed Event Extractor. The flow of mixed event extraction based on TF-IDF and TextRank is shown in Figure 3.

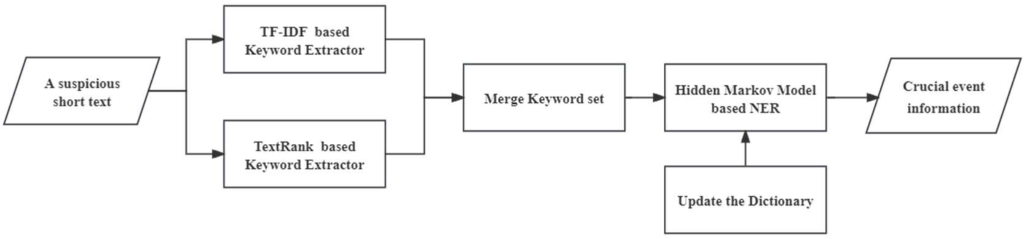
A suspicious short text
TF-IDF based Keyword Extractor
TextRank based Keyword Extractor
Merge Keyword set
Hidden Markov Model based NER
Update the Dictionary
Crucial event information

Figure 3. Flowchart of TF-IDF and TextRank-based Mixed Event Extractor.

2.2.1. TF-IDF-Based Keyword Extractor

The TF-IDF algorithm is a simple and efficient method to extract keywords from text. It first creates a dictionary based on the sample set then calculates the TF-IDF value for each word in the dictionary and sorts them accordingly, thus obtaining a keywords set $word_{TF-IDF}$. TF-IDF contains two important indexes, word frequency and inverse document frequency, and its calculation method is shown as follows:

Term frequency (TF refers to the frequency of words appearing in a short text, and words with high frequency can become the keywords of a short text. The formula of TF is shown in (2), where the numerator represents the number of times the word w appears in the text x_i, and the denominator represents the number of times all words appear in the text x_i.

$$TF(w, x_i) = \frac{count(w, x_i)}{count(*, x_i)} \tag{2}$$

Inverse document frequency (IDF) is utilized to measure the number of texts in which a word occurs. The formula for IDF is shown in (3), where the numerator in the logarithmic function represents the total number of short texts X, and the denominator represents the number of short texts in which the word w occurs in X.

$$IDF(w) = \log \frac{count(X)}{count(w, X)} \tag{3}$$

$TF - IDF$ is the product of TF and IDF, and its formula is shown in (4). When the value of $TF - IDF$ is larger, it indicates that the word w is more important, and vice versa.

$$TF - IDF(w) = TF(w, x_i) \times IDF(w) \tag{4}$$

The $TF - IDF(w)$ values of all words are sorted in the dictionary from largest to smallest. The keyword set $word_{TF-IDF}$ is obtained through taking the first t words in the dictionary.

2.2.2. TextRank-Based Keyword Extractor

Inspiring by Google's PageRank algorithm [15], Mihalcea and Tarau proposed TextRank in 2004 [16]. TextRank is a graph-based ranking algorithm mainly used for text summarization and keyword extraction from text. It computes the importance of words by constructing a co-occurrence relationship graph, where each word serves as a vertex, and edges represent the frequency or semantic connections between them. Based on the connection strength and link quality among the nodes in this graph, it determines the importance of words, enabling text summarization and keyword extraction.

The TextRank algorithm firstly segments the short text into sentences and then performs word splitting and lexical annotation for each sentence to obtain the words in the sentence. The purpose of lexical annotation is to determine the grammatical label of each word in the sentence and to grasp the function as well as the meaning of the word in the sentence. Through lexical annotation, words are generally tagged with lexical labels such as nouns, verbs, and adjectives.

The TextRank algorithm constructs a directed graph $G = (V, E)$ based on the co-occurrence of words in a sentence, where node V of the graph represents the set of words, and edge E represents the co-occurrence relationship between words. The strength of the co-occurrence relationship can be determined by word frequency as well as semantic and other information.

The TextRank algorithm uses iterative computation to calculate the importance score of each word based on the strength of the connection between nodes in the graph. The importance score $S(V_i)$ of vertex V_i is calculated by the formula shown in (5), where $In(V_i)$ denotes the set of all vertices with edges pointing to vertex V_i, and $Out\left(V_j\right)$ denotes the combination of all vertices with edges starting from vertex V_j. d is the damping factor, which determines the probability of jumping from one vertex to another, and is usually set to 0.85.

$$S(V_i) = (1 - d) + d \times \sum\nolimits_{j \in In(V_i)} \frac{1}{\left|Out\left(V_j\right)\right|} S\left(V_j\right) \tag{5}$$

After constructing the directed graph, the vertices in the graph are randomly selected, and iterative computation is carried out until the result is less than the threshold or reaches the upper limit of iteration number. The selection of initial vertices will not affect the result of the final iteration, but the number of iterations will determine the result of convergence. Finally, according to the result of iteration, we can obtain the importance score ranking of words, take the first t words as the keywords of the event cluster, and obtain $word_{TextRank}$.

2.2.3. Merge Keyword Set

The TF-IDF algorithm emphasizes the significance of words within a document, assigning weights to them based on their distribution across the entire corpus. Conversely, the TextRank model is a graph-based ranking algorithm that not only considers the frequency of words in the text but also accounts for the correlation between them, thereby effectively capturing semantic relationships between words. Given the distinct biases of the two keyword extraction methods, we integrate the TF-IDF algorithm with the TextRank model to extract events from suspicious short texts. The ensemble of these two methods forms the keyword set $word_{key}$, as defined by the formula in (6).

$$word_{key} = word_{TF-IDF} \cup word_{TextRank} \tag{6}$$

2.2.4. Hidden Markov-Model-Based NER

Given the unique nature of scientific research integrity, related events may trigger significant public outcry and cast doubts on the authority of scientific research within the community. However, relying only on the keyword set $word_{key}$ is insufficient to promptly extract crucial information that could spark public opinion, such as event characters, locations, and organizations. Named entity recognition (NER) is a technique that can identify words of particular significance in text, encompassing names of individuals, locations, organizations, and other proper nouns.

In this paper, we use named entity recognition (NER) using the Hidden Markov Model (HMM) as our foundation [17]. Initially, we obtain the HMM model's parameters, including the state transition probability matrix and the observation probability matrix, through rigorous training on public corpora. In the context of part-of-speech tagging, states represent lexical properties, while observations correspond to individual words. Next, we employ the keywords within the event clusters of our established HMM model to carry out part-of-speech tagging. However, we recognize that the HMM's performance in part-of-speech tagging may be hampered by the limited textual information available in suspected research integrity public opinion events, particularly when it comes to learning specialized research-related institution names. To address this, we augment our lexicon with a compilation of research integrity-specific terms, ensuring the comprehensive extraction of crucial elements such as names of individuals and institutions from these events.

3. Results and Discussion

3.1. Dataset and Data Pre-Processing

3.1.1. Dataset

The data utilized in experiments were collected from Sina Weibo https://weibo.com/ (accessed on 20 October 2023 and 3 November 2023) using a crawler tool specifically designed to extract relevant short texts. Sina Weibo is a platform that can publish different forms of content besides text, including images, videos, and other information. However, the object of interest in this paper is textual content, so we only focused on textual information. Table 1 provides a comprehensive overview of the dataset, including the type of data and the distribution of labels. The majority class is common short texts, each labeled with "0", whereas the minority class consists of suspicious short texts, labeled with "1".

Table 1. Data types and labels.

No.	Data Type	Data Label	Sample Size
1	The common short texts	0	9130
2	The suspicious short texts	1	194

3.1.2. Data Pre-Processing

Doc2vec [18] was utilized to convert unstructured text into vectors and treat the entire short text as a vector embedding. Doc2vec is capable of learning variable-length texts, including sentences, paragraphs, and articles, and representing them as fixed-length, low-dimensional, and high-density vectors. This approach allows us to simultaneously capture both the semantic and sequential information embedded within the text. Therefore, we chose to employ Doc2vec for the vector representation of the text in this paper. Our Doc2vec model is trained on a comprehensive corpus, including 300,000 articles from the Chinese Wikipedia and over 100 texts related to policies, regulations, and case reports from the China Research Integrity Network https://www.orichina.cn/ (accessed on 22 October 2023).

3.2. Evaluation Metrics of TextCNN

Accuracy is the most commonly used evaluation metric. The formula is shown in (7), which is the ratio of the number of samples correctly identifying true and false to the number of all samples. $Accuracy \in [0,1]$, and a higher value indicates better performance of the model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

The formula of $Precision$ is shown in (8). It refers to the number of correctly predicted true samples in the proportion of all predicted true samples. $Precision \in [0,1]$, and a higher value indicates better performance of the model.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

The formula of $Recall$ is shown in (9). It refers to the number of correctly predicted true samples in the proportion of all true samples. $Recall \in [0,1]$, and generally a higher value indicates better performance of the model.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

During the experiments, we distinguished between ordinary short texts, labeled as 0 (false), and suspicious short texts, labeled as 1 (true). Our primary concern was accurately identifying the suspicious short texts. Therefore, when assessing the TextCNN model, our primary focus was on Recall, as it measures the model's ability to correctly detect suspicious texts.

3.3. Experimental Setup of TextCNN

Before training TextCNN, the detailed parameters of the model need to be set, The parameter settings are shown in Table 2. For the short text dataset used in this experiment, 80% was used as the training dataset, and 20% was used as the validation dataset.

Table 2. TextCNN model parameters setting.

NO.	Parameter	Value
1	Embedding size	(50, 1)
2	Number of hidden layers	3
3	Number of convolution kernels	64
4	Convolution kernel size	[3,4,5]
5	Padding	valid
6	Activation	Relu
7	Dropout	0.5
8	Activation function of output layer	Sigmoid
9	Loss	binary_crossentropy
10	Optimizer	adam
12	Epochs	10
13	Batch_size	64

3.4. Experimental Results

3.4.1. Experimental Results of TextCNN

(1) Experimental results of SMOTE

The dataset utilized in this experiment is detailed in Table 1 and primarily comprises common short texts and a minority of suspicious short texts. The proportion of the majority class to the minority class was approximately 25.8:1. Table 3 provides a breakdown of the number of samples for both the majority and minority classes across different sampling ratios applied using SMOTE.

Table 3. Proportion of majority and minority classes.

Value of 0:1	Number of 0 Samples	Number of 1 Samples	Size of Dataset	Size of Training Set	Size of Test Set
1:1	9130	9130	18,260	14,608	3652
2:1	9130	4565	13,695	10,956	2739
3:1	9130	3043	12,173	9738	2435
4:1	9130	2282	11,412	9129	2283
5:1	9130	1826	10,956	8764	2192

The training outcomes of the model on the test set when the dataset remained unprocessed are presented in the first row of Table 4. Notably, while the Accuracy appears to exceed 97%, the Precision and Recall hover around 0. Although a significant number of common short texts were accurately identified, these results are ultimately meaningless, as our primary focus lies in the accurate identification of a few suspicious short texts. The results indicate that the TextCNN model, when trained on imbalanced datasets, loses its capacity to effectively recognize minority class samples.

Table 4. Model performance for different sampling ratios of majority and minority classes.

Value of 0:1	Accuracy	Recall	Precision	Val_Accuracy	Val_Recall	Val_Precision
25.8:1	0.9795	0	0	0.9780	0	0
1:1	0.722	0.776	0.700	0.750	0.846	0.713
2:1	0.783	0.544	0.727	0.784	0.538	0.776
3:1	0.819	0.375	0.799	0.841	0.437	0.815
4:1	0.823	0.147	0.809	0.836	0.212	0.943
5:1	0.840	0.069	0.754	0.852	0.094	0.892

Other rows in Table 4 present the evaluation metrics of the TextCNN model on the test set, considering these varying sampling ratios. Overall, the experimental results are quite satisfactory after applying SMOTE sampling. The Accuracy, Recall, and Precision are all maintained above 0.7. However, as the sampling ratio of the majority class to the minority class increases from 1:1 to 5:1, while the Accuracy and Precision remain above 0.7, the Recall rapidly declines to 0.094. This observation indicates that the imbalanced dataset has a more significant impact on Recall. When the data become increasingly imbalanced, the model demonstrates good performance in classifying suspicious short texts but lacks the ability to detect all of them.

Maintaining a 1:1 ratio between common short texts and suspicious short texts can optimize the performance of the TextCNN model in terms of Accuracy, Recall, and Precision. If the primary focus is on enhancing the accuracy of suspicious short texts identified by the model, without the need to detect them from a larger pool of texts, a sampling ratio of 4:1 may be a suitable choice. As evident in Table 3, reducing the sampling ratio from 1:1 to 4:1 significantly decreases the number of samples by approximately 7000, potentially leading to a notable reduction in time complexity. Therefore, when contemplating the application of subsequent transfer learning based on this model to a larger training set, it is crucial to strike a balance between performance and time complexity, selecting a practical solution that best suits the requirements.

(2) Baseline models

Three models were selected as our baseline models.

The naïve Bayes classifier [19] is a classifier based on Bayes' theory. It assumes that the sample features are independent and calculates the prior probability and conditional probability between the categories and the features to classify the new sample points.

A support vector machine (SVM) [20] is a classification model based on statistical learning theory. Although primarily a linear model, it can handle nonlinear problems and is widely employed in binary classification, outlier detection, and other scenarios.

Logistic regression [21] is a linear model that evolved from linear regression and is utilized for binary classification tasks.

Tables 5–7 present the evaluation metrics for the Gaussian Bayesian classifier, SVM, and logistic regression, respectively, across various sampling rates using SMOTE. Based on past experience, it would be challenging to distinguish the experimental subjects in this paper using a linear classifier. However, surprisingly, both the SVM and logistic regression demonstrate great performance when the sampling ratio reaches 1:1. The evaluation metrics range from a low of 0.869 to a high of 0.906, indicating that these classifiers are capable of fitting appropriately and achieving a good classification effect. One possible explanation for this observation is that SMOTE's linear interpolation technique effectively oversamples the minority class, generating new minority class instances in the linear direction of the original minority class. Consequently, the 8936 newly generated minority class instances exhibit a clear linear relationship with the original minority class, enabling the linear classifier to achieve such a satisfactory training effect.

Table 5. The results of the Gaussian plain Bayesian model.

Value of 0:1	Val_Accuracy	Val_Recall	Val_Precision
1:1	0.764	0.829	0.734
2:1	0.764	0.816	0.619
3:1	0.747	0.827	0.504
4:1	0.755	0.792	0.448
5:1	0.767	0.807	0.407

Table 6. The results of the SVM model.

Value of 0:1	Val_Accuracy	Val_Recall	Val_Precision
1:1	0.897	0.889	0.904
2:1	0.905	0.843	0.876
3:1	0.925	0.817	0.882
4:1	0.928	0.773	0.865
5:1	0.935	0.721	0.876

Table 7. The results of the logistic regression model.

Value of 0:1	Val_Accuracy	Val_Recall	Val_Precision
1:1	0.890	0.869	0.906
2:1	0.902	0.835	0.873
3:1	0.915	0.785	0.869
4:1	0.926	0.758	0.867
5:1	0.933	0.735	0.854

3.4.2. Experimental Results of the Mixed Event Extractor

Table 8 presents the keywords extracted using these two methods. The results reveal that each method exhibits its own biases in keyword extraction. TF-IDF is effective in capturing institutions and individuals. Conversely, TextRank tends to focus more on keywords pertaining to the event description.

Table 8. Two methods to extract keywords.

TF-IDF	TextRank
rubbish	National
Medical	Medical
* University	Research
Bo Song	Misbehavior
* Li	Encounter
* Du	Patient
Misbehavior	Rubbish
* Zhang	Industry
Naming	Take a look
Unlucky	Naming
	Funding
	Academic
	Should
	Start

Note: words marked with * have been redacted.

After obtaining the $word_{key}$, it becomes crucial to perform named entity recognition (NER) on the keywords. This step is aimed at extracting key nouns, such as the names of individuals and organizations. Given the availability of sophisticated and encapsulated NER models, in this experiment, we employed jieba to conduct further NER on the keyword set. Jieba's NER is based on the Hidden Markov Model (HMM). To enhance the quality of the keywords, we updated the relevant dictionaries by incorporating 3072 national institutions of higher education (excluding schools in Hong Kong, Macao, and Taiwan) that were enumerated by the Ministry of Education of the People's Republic of China in 2023.

Table 9 displays the speech tags obtained after NER. Specifically, "v" represents a verb, "n" indicates a noun, "nr" stands for a person, "m" signifies a numeral, "a" designates an adjective, and "an" represents a name. Additionally, "m" can also refer to a number word, "a" remains as an adjective, "an" can mean an adjectival noun, and "x" signifies that the term is not relevant in the original lexical cross-reference table.

Table 9. The results of named entity recognition.

NO.	Word	Part of Speech
1	National	n
2	Naming	v
3	* University	x
4	Rubbish	n
5	Patient	n
6	Rubbish	n
7	Unlucky	a
8	* Song	nr
9	* Li	nr
10	* Zhang	nr
11	* Du	nr
12	Academic	n
13	Misbehavior	n
14	Medical	n
15	Research	n
16	Founding	n
17	Rectification	m
18	Medical	n
19	Industry	n
20	Corruption	an

Note: words marked with * have been redacted.

Table 10 presents the outcomes of the Mixed Event Extractor's filtering and extraction process for a topical short text event element. Notably, crucial elements, including individuals, locations, and institutions, were effectively extracted, providing a solid foundation and valuable reference for subsequent event investigations.

Table 10. Results of event extraction.

Results of event extraction	{'* Li', '* Du', '* University', '* Song', '* Zhang'}

Note: words marked with * have been redacted.

Table 11 exhibits several short texts for which the model failed to extract crucial event-related information. Firstly, the original short texts lack standardization in their writing, presenting a significant challenge for NER and part-of-speech tagging. For instance, the redacted words in the first short text are essential event elements we aim to extract; however, the described institutions and individuals' names employ colloquial abbreviations rather than standardized, conventional terminology. Secondly, the presence of words with multiple meanings can introduce biases in the part-of-speech tagging of keywords. In the part-of-speech tagging of the keywords in the first original short text in Table 11, "bully" and "week" are erroneously classified as nouns, whereas in the actual context, "bully" is a verb and "week" represents a time period. Lastly, when the original short text lacks key event elements, such as the names of individuals and organizations, it poses a significant obstacle for NER and part-of-speech tagging. For instance, the second text in Table 11 fails to provide explicit event details, which is a common occurrence in real-world scenarios.

Table 11. Some examples of unsuccessful extraction of key information from events.

NO.	Original Text	Results of Event Extraction
1	#* give the public an answer as soon as possible # Rumour mongering and defamation, bullying roommates, academic fraud! Any one of them is enough to be expelled, almost a week has gone by without any movement, this student's energy is through the sky ah!	{'bully', 'week'}
2	I can't accept this kind of academic misconduct around me, and I can't change it, I can only put myself out of the picture and watch silently.	{'morals'}

Note: words marked with * have been redacted.

4. Conclusions

To identify potential public opinion events suspected of integrity issues amidst large volumes of short texts, our study introduces a modular framework model. This model employs TextCNN to distinguish suspicious short texts related to research integrity concerns from ordinary ones. Subsequently, it utilizes a Mixed Event Extractor based on TF-IDF and TextRank to extract crucial event elements from these suspicious texts. These extracted elements serve as valuable support and foundation for the subsequent management tasks undertaken by research managers. The experimental results showed that the TextCNN classifier after SMOTE sampling performed well in recognizing special short texts. Furthermore, the Mixed Event Extractor demonstrates its proficiency in extracting vital event components, including personnel and organization names.

This study aims to transform the traditional role of citizens as mere spectators in scientific research process into active supervisors. For policymakers, it offers an opportunity to integrate citizens' queries and concerns regarding scientific research processes and outcomes into the reporting and evidentiary framework for scientific integrity. Leveraging this approach, policymakers can enhance the relevant systems to bolster the compliance and reliability of scientific research, both in its processes and outcomes. Additionally, by opening the API for the model's output port, the extracted event information can be seamlessly integrated into higher-level information management systems, such as public opinion event warning systems and decision-making support systems, thereby broadening the model's utility and capabilities.

However, this study possesses two limitations. Firstly, the experimental data were flawed. Due to numerous constraints of public Internet platforms, acquiring the suspicious short texts brought significant challenges; thereby, the processes of data mining, cleaning, and tagging depended on artificiality. Secondly, the scarcity of information in short texts, coupled with non-standard word and sentence expressions, as well as polysemy, can adversely impact the accuracy of event extraction outcomes. Looking ahead, suspicious short texts could be mined from a broader range of channels and dimensions. Furthermore, this model holds potential for further enhancements of the corpus to tackle challenges like the ambiguity of short text words, irregular text formatting, and the absence of crucial information.

Author Contributions: Z.Z. contributed to the design and implementation of the research. X.J. contributed to the design and implementation of the research, to the analysis of the results, and the writing of the manuscript. Y.L. contributed to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lowell, L.A. *Public Opinion and Popular Government*; Longmans, Green: New York, NY, USA, 1913.
2. Wang, G.; Chi, Y.; Liu, Y.; Wang, Y. Studies on a multidimensional public opinion network model and its topic detection algorithm. *Inf. Process. Manag.* **2019**, *56*, 584–608. [CrossRef]
3. Chen, X.; Duan, S.; Wang, L. Research on trend prediction and evaluation of network public opinion. *Concurr. Comput.-Pract. Exp.* **2017**, *29*, e4212. [CrossRef]
4. Hassani, H.; Komendantova, N.; Rovenskaya, E.; Yeganegi, M.R. Social Intelligence Mining: Unlocking Insights from X. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1921–1936. [CrossRef]
5. Weng, Z. Application Analysis of Emotional Learning Model Based on Improved Text in Campus Review and Student Public Opinion Management. *Math. Probl. Eng.* **2022**, *2022*, 5135200. [CrossRef]
6. Wang, C.; Wang, X.; Wang, P.; Deng, Q.; Liu, Y.; Zhang, H. Evaluating public opinions: Informing public health policy adaptations in China amid the COVID-19 pandemic. *Sci. Rep.* **2024**, *14*, 5123. [CrossRef] [PubMed]
7. Sun, Q.; Chen, J.; Gao, S. From panic to banter: How do routine government releases and clarifications cause unexpected public opinion crisis—An analysis of public opinion toward a release by Chinese Ministry of Commerce encouraging the storage of necessities. *J. Cont. Crisis Manag.* **2024**, *32*, e12530. [CrossRef]
8. Weng, J.; Lee, B.S. Event detection in twitter. In Proceedings of the International AAAI Conference on Web and Social Media, Barcelona, Spain, 17–21 July 2011; pp. 401–408.
9. Karamouzas, D.; Mademlis, I.; Pitas, I. Public opinion monitoring through collective semantic analysis of tweets. *Soc. Netw. Anal. Min.* **2022**, *12*, 91. [CrossRef] [PubMed]
10. Khademi Habibabadi, S.; Delir Haghighi, P.; Burstein, F.; Buttery, J. Vaccine Adverse Event Mining of Twitter Conversations: 2-Phase Classification Study. *JMIR Med. Inf.* **2022**, *10*, e34305. [CrossRef]
11. Nallapati, R.; Feng, A.; Peng, F.; Allan, J. Event threading within news topics. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004; pp. 446–453.
12. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arxiv* **2014**, arXiv:1408.5882.
13. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
14. Turney, P.D.; Littman, M.L. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.* **2003**, *21*, 315–346. [CrossRef]
15. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
16. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
17. Kupiec, J. Robust part-of-speech tagging using a hidden Markov model. *Comput. Speech Lang.* **1992**, *6*, 225–242. [CrossRef]
18. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
19. Xu, S. Bayesian Naïve Bayes classifiers to text classification. *J. Inf. Sci.* **2018**, *44*, 48–59. [CrossRef]

20. Kecman, V. *Support Vector Machines–An Introduction*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 177.
21. Yen, S.J.; Lee, Y.S.; Ying, J.C.; Wu, Y.C. A logistic regression-based smoothing method for Chinese text categorization. *Expert Syst. Appl.* **2011**, *38*, 11581–11590. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.