applied sciences

MDPI

*Article*

# A Multi-Stance Detection Method by Fusing Sentiment Features

**Weidong Huang and Jinyuan Yang ***

College of Management, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; huangwd@njupt.edu.cn

* Correspondence: 1022112313@njupt.edu.cn

**Abstract:** Stance information has a significant influence on market strategy, government policy, and public opinion. Users differ not only in their polarity but also in the degree to which they take a stand. The traditional classification of stances is quite simple and cannot fully depict the diversity of stances. At the same time, traditional approaches ignore user sentiment features when expressing their stances. As a result, this paper develops a multi-stance detection model by fusing sentiment features. First, a five-category stance indicator system is built based on the LDA model, then sentiment features are extracted from the reviews using the sentiment lexicon, and finally, stance detection is implemented using a hybrid neural network model. The experiment shows that the proposed method can classify stances into five categories and perform stance detection more accurately.

**Keywords:** stance detection; deep learning; LDA; sentiment lexicon



**Citation:** Huang, W.; Yang, J. A Multi-Stance Detection Method by Fusing Sentiment Features. *Appl. Sci.* **2024**, *14*, 3916. https://doi.org/10.3390/app14093916

Academic Editor: Douglas O'Shaughnessy

Received: 5 April 2024
Revised: 26 April 2024
Accepted: 1 May 2024
Published: 4 May 2024



## 1. Introduction

News hotspots can be easily discussed thanks to platforms such as Weibo and Twitter, which has resulted in a substantial increase in the amount of user-generated content. These messages represent users' opinions and stances and have significant influence on market strategy, government policy, and public opinion. The task of stance detection in Chinese microblogs was proposed in NLPCC-ICCPOL (2016), indicating that Chinese stance information is of tremendous research significance. Stance information is an important basis in fields such as market research, public opinion management, and policy formulation. In market research, stance detection helps companies understand public perceptions of their products. In the field of public opinion management, stance information is crucial for early warning and response strategies regarding public sentiment. Furthermore, when formulating policies, government agencies can utilize stance information to understand public opinions. Additionally, as one of the most widely used languages globally, Chinese stance detection presents unique challenges and opportunities for Natural Language Processing (NLP).

The current focus of stance detection is on improving stance detection accuracy by employing various models and methods. These approaches usually divide stances into three categories: support, opposition, and neutral. However, when expressing opinions, users not only differ in polarity but also in the degree of their opinions. Polarity typically refers to the positive or negative tendency of a stance, but the range of opinions extends beyond simple binary classification and includes a gradient of sentiments ranging from mild agreement to strong criticism. For example, when expressing support, some users may agree totally while others may agree with only one component. The same is true when users offer an opposing viewpoint. This reflects the complexity of human sentiments and the diversity of expressing viewpoints. In such circumstances, the traditional stance category is too simplistic to depict the diversity of users' stances. The degree of strength in expressing an opinion can be seen as the will or belief behind the opinion. An individual may express their viewpoint with a casual tone, or they may defend their stance with firm conviction. These varying degrees significantly influence people's perspectives. In the field of opinion mining, understanding the importance of this degree is just as crucial as understanding polarity. Grasping these subtle differences is essential for constructing more sophisticated sentiment analysis and stance detection models, enabling a more accurate capture of comprehensive public opinions.

Traditional approaches, on the other hand, usually consider only the text itself, ignoring the sentimental features buried in it. Sentiment features have a supportive effect on stance detection, and when stance detection is combined with sentiment analysis, it is possible to filter away the influence of moody expressions. Therefore, there are limitations to stance detection, and a more comprehensive picture of user attitudes can be acquired by combining sentiment analysis.

This paper aims to address the following issues: first, to classify stances into five categories based on the degree and polarity of users' attitudes to describe users' stances more thoroughly; second, to mine the reasons for users' different attitudes through the five categories of stances, which can be used to guide public opinion management; and third, to combine sentiment analysis with stance detection to gain a more comprehensive understanding of users' opinions. As a result, this paper introduces BACF (bi-LSTM-Attention-CNN-Fusion), a multi-stance detection model that combines deep learning with the sentiment lexicon to address the above issue. The experiment shows that BACF conducts stance detection more accurately than the traditional approach, which is significant for policy selection and public opinion analysis.

## 2. Related Research

Stance detection is a subtask of opinion mining that can automatically classify text stances based on a given target. Early research focused on political debates and online forums, whereas recent research has focused on online social platforms such as Weibo and Twitter.

### *2.1. Stance Category*

In the beginning, stance detection was a problem of binary classification, with stances categorized as "support" or "oppose". As stances were better understood, stance detection shifted from a binary to a three-class classification, and researchers began categorizing stances as "support, oppose, or neither" [1]. For instance, Jia et al. categorized stances as "support, oppose, and neutral" in their study on the stance recognition of users' viewpoints [2]. As can be seen, stance classification is a significant advancement in the field of stance detection. The simple three-class category can no longer effectively reflect user opinions, and some scholars have empirically proposed a four-class category. However, stance classification based on experience is limited in generalization and focuses on specific fields. Ma et al. categorized the stances of the two datasets used for rumor detection as "support, deny, question, and comment" and "agree, disagree, discuss, and unrelated" [3]. For the purpose of stance detection in Twitter rumors, Poddar et al. classified the stance into four categories: "remark, support, deny, and query" [4]. How to extend the stance category is a significant problem for stance detection.

Stance grading can assist in resolving the problem. Users' stances differ greatly in terms of polarity and degree, which allows for the possibility of stance grading. When users comment on various topics, this phenomenon becomes more obvious. For example, when a new policy is proposed, distinct discussions may focus on its current impact and potential future benefits. People will have different preferences, as with their levels of support and opposition. As a result, users with various stances may focus on different topics. It is feasible to acquire a more precise understanding of users' attitudes by grading their stances according to various topics. Extending segmented stances in terms of grading is a standard practice that can better represent user attitudes. For instance, five-level scales are frequently created in the marketing industry based on user satisfaction.

To achieve degree grading in the field of stance detection, we can first mine the event's topics and group them into various stances based on their degree and polarity to create a multi-stance indicator system. The LDA topic model (Latent Dirichlet Allocation) is an unsupervised learning technique for text analysis and topic mining that was first proposed by Blei et al. [5]. LDA acquires text topics through the statistical analysis and probabilistic modeling of words in the text. In order to analyze the experiences of Airbnb guests during the COVID-19 crisis, Keawtoomla examined the reviews posted on the Airbnb platform using the LDA [6]. To present a complete picture of the research field, Tomojiri employed the LDA to infer the research topics about anthropogenic marine debris [7]. As a result, the LDA model can effectively implement hidden topic mining. It can better reflect users' attitudes by mining text topics and building a multi-stance indicator system with LDA models.

### 2.2. Stance Detection Models

Traditional stance detection frequently employs support vector machines [8], naive Bayes, logistic regression, random forests [9], K-means, and other machine learning models [10]. Machine learning models are well defined and simple in structure. For instance, a two-stage stance detection system based on SVM was proposed to characterize stances on Twitter [11]. Mourad et al. discovered that random forests, linear SVM, and Gaussian NB may be employed as majority vote stance identification classifiers [12]. However, machine learning models are linguistically demanding and prone to human error. As the volume of information increases, traditional stance detection models become more time-consuming and expensive, and mining stance information is becoming increasingly difficult in the information age.

Deep learning models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have become the standard methodologies for stance detection. RNNs are good at processing sequential data. In Twitter stance detection, Siddiqua proposed a variant that utilizes bi-LSTMs and nested LSTMs to capture long-term dependencies, where each module is enhanced with an attention mechanism [13]. In the two-target stance detection task, Liu et al. used multiple LSTM layers to encode target-related regions [14]. CNNs are well suited for extracting local features because of their multiple kernels. Zhang et al. proposed a CNN-based stance analysis approach to identify stance tendencies [15]. Tran et al. employed BERT and CNNs to create a stance detection model for Vietnamese [16]. With the advancement of deep learning, researchers have begun to investigate hybrid model structures. Li et al. produced an excellent result using a hybrid stance detection model built with GRU, position weight vector, and CNNs [17]. In the field of rumor stance detection, Li et al. proposed a framework based on multi-graph neural networks to capture the attribute and structural information of context [18]. In the task of stance detection on social platforms, Liu et al. employed a Gated Graph Neural Network (GGNN) approach to integrate structural information between reviews [19]. Furthermore, technologies such as sentiment lexicons and attention mechanisms have been used to improve the model's performance. The sentiment lexicon can be used to obtain sentiment information from a text. For example, Zheng et al. employed the sentiment lexicon to perform feature selection in microblog stance detection and found favorable results [20]. Although sentiment lexicons are commonly utilized as a supplement, the combination of sentiment analysis and stance detection is still in its early stages. Moreover, the attention process may give more weight to significant information, which can improve model accuracy. Dey et al. proposed a two-stage model based on LSTM in combination with an attention mechanism that can perform well in Twitter stance detection [21]. Additionally, Karande et al. implemented word embeddings for stance detection models using BERT [22]. In conclusion, deep learning has surpassed machine learning as the dominant model for stance detection, and the attention mechanism and sentiment lexicon have been employed to improve model efficacy. However, while binary classification has given way to three-class classification, the question of how to further extend stance classification remains unanswered. In addition, the sentiment information contained in stance expressions has not been fully utilized. To address these issues, this paper proposes a multi-stance detection model by fusing sentiment features.

## 3. Methodology

This paper aims to build a multi-stance detection method by fusing sentiment features. The method consists of three parts: the construction of a multi-stance indicator system, the acquisition of sentiment features, and the construction of a stance detection model. First, we build a 5-stance indicator system based on the degree and polarity of users' attitudes; then, the sentiment features of reviews are obtained via the sentiment lexicon; and finally, a hybrid neural network is utilized to achieve multi-stance detection. After the method was constructed, we implemented public opinion management based on a multi-stance indicator system. The research framework is shown in Figure 1.

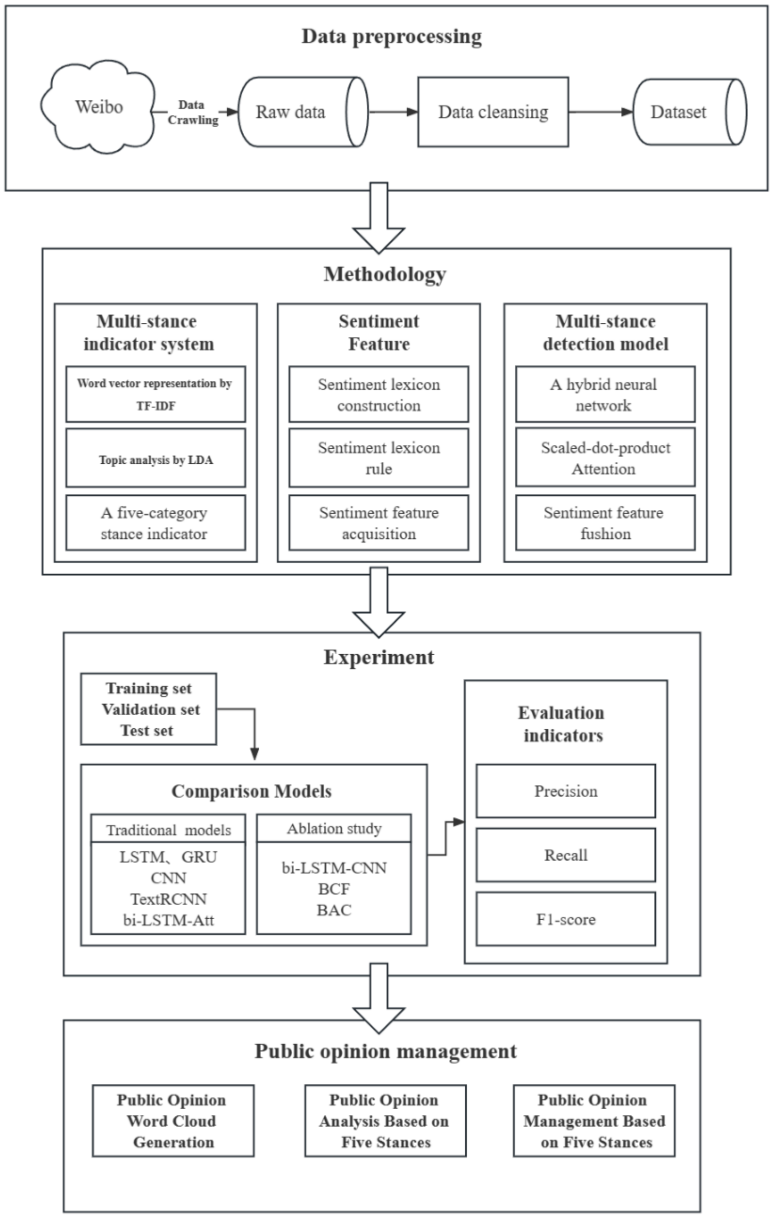


**Figure 1.** Research framework.

### 3.1. Multi-Stance Indicator System

User reviews differ not only in polarity but also in the degree to which they express their stance. The traditional approach usually divides stances into three categories, which fails to differentiate distinctions within the same polarity (support or against) and fails to explain why these discrepancies exist. As a result, this study separates stances into five categories based on polarity and degrees:

(1) Strong support (SS): a totally positive attitude towards the event and a favorable opinion of the event.
(2) Weak support (WS): a generally positive view of events, affirming most of them despite concessions in some parts.
(3) Neutral: neutrality towards events and the discussion of them.
(4) Weak opposition (WO): a generally negative view of events, opposing most of them despite concessions in some parts.
(5) Strong opposition (SO): a totally negative attitude towards the event and unfavorable to the event.

In this paper, a multi-stance indicator system based on the LDA model is developed to establish the association between segmented stances and topics. We use Gensim to implement LDA mining, which is a Python library for topic modeling. Parameter settings are shown in Table 1. The five stances categorized according to the topic results can provide a more detailed understanding of users' attitudes toward the event and the reasons for the differences, thus providing a guide to public opinion management.

**Table 1.** Parameter settings of LDA.

| Parameter | Value |
|---|---|
| corpus | TF-IDF |
| num_topics | 25 |
| passes | 10 |
| chunksize | 100 |

### 3.2. Sentiment Feature

Sentiment information plays an important role in stance expression. The core of the sentiment lexicon is rule-based calculation, from which sentiment features relating to reviews can be obtained.

#### 3.2.1. Sentiment Lexicon Construction

The model requires a comprehensive lexicon to work well. In this paper, an open-source sentiment lexicon known as Hownet is supplemented with the manual inclusion of new words to construct a sentiment lexicon. The lexicon consists of five sections, as shown in Table 2. A positive lexicon includes words that express joy, support, and other positive sentiments. A negative lexicon includes words that express disappointment and anger, representing negative sentiments. The adverb lexicon consists of adverbs that modify the degree of sentiments in other words. Negation words include words that express opposite meanings, and they hold a special position in sentiment analysis as they can reverse the polarity of a sentence's sentiment. Some conjunctions and prepositions have minimal impact on expressing sentiments and can be included in the stop-word list, as ignoring them can improve the efficiency of sentence analysis.

**Table 2.** Structure of the sentiment lexicon.

| Lexicon | Sentiment Words | Number |
|---|---|---|
| Positive lexicon | great, happiness | 6094 |
| Negative lexicon | annoyed, disheartened | 11,445 |
| Adverb lexicon | rather, extremely | 252 |
| Negation lexicon | not, no | 14 |
| Stop words | "、", a | 112 |

#### 3.2.2. Sentiment Feature Acquisition

Sentiment words are classified into positive or negative categories based on their polarity. Additionally, sentiment adverbs can be categorized according to their degree, such as "extremely", "very", "more," "slightly", "insufficient", and "excessive". These adverbs are assigned to their respective sub-lexicons as shown in Table 3.

**Table 3.** Structure of the Sub-Lexicon.

| Sub-Lexicon | Sentiment Words |
|---|---|
| Extremely | highly, extremely |
| Very | quite, fairly |
| More | even more, relatively |
| Slightly | slightly, a bit |
| Insufficient | not very, not so |
| Excessive | excessive, overly |

Based on the experimental outcomes, $v_i$ is quantified using weights with values of 2.5, 2, 1.5, 0.5, $-0.5$, and $-0.8$. Negation words are likely to completely reverse the original semantics; thus, the inverse number is utilized to reverse the semantics.

On this basis, the text's positive sentiment value $S_p$ and negative sentiment value $S_n$ are computed, and the sentiment feature E of the related review is obtained by subtracting the two, as shown in Equations (1)–(3):

$$n_S = (-1)^d \prod v_i \tag{1}$$

$$S_i = \sum n_S \tag{2}$$

$$E = S_p - S_n \tag{3}$$

where $n_S$ represents the sentiment value of a word, $S_i$ represents the sentence's specific sentiment value, $d$ is the number of negation words, and $\prod v_i$ is the product of all sentiment adverbs preceding the current sentiment word.

### 3.3. Multi-Stance Detection Model

This paper proposes a hybrid neural network stance detection model fused with sentiment features by combining deep learning and sentiment lexicon approaches, and the model structure is shown in Figure 2.

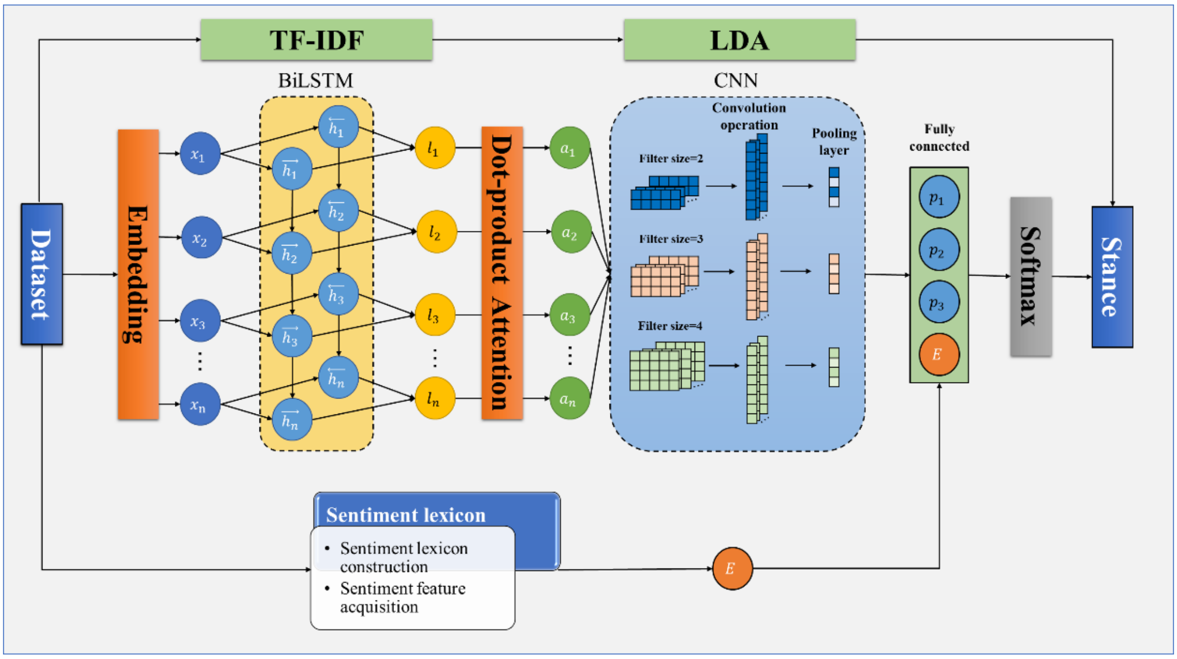


**Figure 2.** Structure of the multi-stance detection model.

#### 3.3.1. Word Embedding

The inputs to the neural network are vectors. The skip-gram model can train word vectors with high accuracy by predicting the context of the current word $w_t$. In review $T_i = [w_1, \ldots, w_i, \ldots, w_n]$, the word $w_i$ is converted to the vector $x_i = [v_1, \ldots, v_i, \ldots, v_d]$, and $T_i$ is converted to the matrix $S_{n \times d} = [x_1, \ldots, x_i, \ldots, x_n]^T$, where $n$ is the padding length and $d$ is the word vector dimension.

#### 3.3.2. bi-LSTM Layer

The bi-LSTM structure is adept at processing sequential data [23]. By introducing the forget gate, input gate, and output gate, bi-LSTM can selectively retain and forget information, as shown in Equations (4)–(9).

$$i_t = sigmoid(W_i * [h_{t-1}, x_t] + b_i) \tag{4}$$

$$f_t = sigmoid\left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{5}$$

$$\widetilde{g_t} = tanh(W_g * [h_{t-1}, x_t] + b_g) \tag{6}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{g_t} \tag{7}$$

$$o_t = sigmoid(W_o * [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \circ \tanh(c_t) \tag{9}$$

where $x$ is the input; $c$ is the cell; $h$ is the hidden state; $t$ is at time $t$; $\circ$ is the Hadamard product; $W$ is the weight of the gate; $b$ is the bias term; and $i, f, \widetilde{g}, o$ are the input, forget, cell, and output gates, respectively.

On this basis, the text is processed in both directions to capture bidirectional semantic information. The output vectors in both directions are collocated to produce the vector $l_t$, as shown in Equation (10).

$$l_t = \left[\overrightarrow{h_t} : \overleftarrow{h_t}\right] \tag{10}$$

#### 3.3.3. Scaled-Dot-Product Attention

Scaled-dot-product attention is one form of product attention that has greater computational efficiency. The correlation between the query vector and the key matrix is calculated, and the similarity is used as a weight. The obtained weights are weighted and summed with the value matrix and, via softmax normalization, the attention output matrix $A$ is obtained, as shown in Equations (11) and (12).

$$a(L_q, L_k) = \frac{{L_q}^T * L_k}{\sqrt{d_k}} \tag{11}$$

$$A = softmax(\frac{L_Q * {L_K}^\top}{\sqrt{d_k}})L_V \tag{12}$$

where $L_Q$, $L_V$, and $L_K$ are the query, key, and value matrices, respectively. $L_q$ is the query vector, $L_k$ is the key vector, and $d_k$ is the length of the $L_K$.

#### 3.3.4. CNN Layer

In this paper, a multi-kernel CNN is utilized to extract the features, which consists of three layers: a convolution layer, a pooling layer, and a fully connected layer [24].

(1) The convolutional layer is used for the initial extraction of features. The model uses a multi-size kernel $W_u$ to conduct a convolution operation with step length 1, and

the nonlinear transformation is realized using the ReLU, as shown in Equations (13) and (14).

$$c_i = f(W_u * A_i + b) \tag{13}$$

$$C = [c_1, \ldots, c_i, \ldots, c_{l-u+1}], u \in \{2, 3, 4\} \tag{14}$$

where $b$ is the bias terms, $l$ is the sentence length, and $u$ is the convolutional kernel size.

(2) The pooling layer abstracts and reduces the dimensionality of the convolution layer's result and employs the 1-max pooling operation to obtain the pooling vector $p_i$, as shown in Equation (15).

$$p_i = max(C) \tag{15}$$

(3) The fully connected layer expands the text feature into vectors. The local optimal features extracted from convolutional kernels of different sizes are collocated to obtain the final output $Q$ of the CNN layer, as shown in Equation (16).

$$Q = [p_2 : p_3 : p_4] \tag{16}$$

#### 3.3.5. Sentiment Feature Fusion

Feature fusion, which is a type of information fusion, can introduce useful information to improve model prediction. Traditional approaches to feature fusion include serial strategy and parallel strategy, and the serial strategy is used in this model because it is effective and simple to implement. The sentiment feature vector $E$ is collocated with $Q$, as shown in Equation (17).

$$M = [Q : E] \tag{17}$$

#### 3.3.6. Output Layer

The fused vector $M$ is outputted through a linear layer for five classifications, and the results are obtained via softmax normalization, with the highest value indicating the predicted stance, as shown in Equations (18) and (19).

$$R = Linear(M) = [r_1, r_2, r_3, r_4, r_5] \tag{18}$$

$$Stance = Max(softmax(r_i)) = Max(\frac{e^{r_i}}{\sum_1^k e^{r_i}}) \tag{19}$$

## 4. Experiment

### *4.1. Dataset*

In the experiment, we take "A doctor beat a 5-year-old child in Nanjing on 8 November 2022" as a case study. On 8 November 2022, an orthopedic doctor attacked the boy and shoved his grandfather at their home since his child had been pierced in the head by the boy. The event triggered a debate on parental protection and overreaction to bullying in schools. A dataset is built by crawling microblog reviews from 9 November 2022 to 15 November 2022. To improve the model's performance, data processing was performed. First, duplicate values were removed, and then the incorrect URL links were deleted. The text data were tokenized by performing jieba. After data processing, a total of 10,082 pieces of data were obtained. The dataset was divided into training, validation, and test sets in the ratio of 6:2:2.

### *4.2. Evaluation Metric and Experimental Environment*

The model was built and trained on the Pytorch deep learning framework. The Adam optimizer was selected for optimization and the AMD Ryzen 5 4600H was used in the experiment. The public parameter settings are shown in Table 4.

**Table 4.** Public parameter settings.

| Parameter | Value |
|---|---|
| padding_size | 64 |
| batch_size | 128 |
| learning_rate | $1 \times 10^{-2}$ |
| epochs | 15 |

The model performance is evaluated using precision, recall, and the F1-score. Precision is the proportion of true positive cases among the predicted positive cases; recall is the proportion of predicted positive cases out of the true cases; and the F1-score is used as a comprehensive evaluation metric, as shown in Equations (20)–(22).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{22}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

Since the distribution of each type of stance is not evenly distributed, precision, recall, and F1-score are weighted in proportion to the amount of each type of stance, as shown in Equations (23)–(25).

$$P_w = \sum_i^5 p_i P_i \tag{23}$$

$$R_w = \sum_i^5 p_i R_i \tag{24}$$

$$F_w = \sum_i^5 p_i F_i \tag{25}$$

where $P_i$, $R_i$, and $F_i$ are the precision, recall, and F1-score of the $i$-th stance classification, respectively, and $p_i$ is the proportion of the $i$-th stance classification sample to the whole sample.

### *4.3. Comparison Models*

Eight models were chosen for comparison investigations to confirm the efficacy of BACF in this paper.

(1) LSTM [25]: this model introduces forget gates, input gates, and output gates to regulate the retention and transmission of information.

(2) CNN [26]: this model extracts features from the input text by using convolutional kernels of various sizes.

(3) GRU: this model uses reset gates and update gates to retain and pass information from sequence data.

(4) bi-LSTM-Att [27]: a bi-LSTM model is utilized to process the input review text, and an attention mechanism is employed to improve the focus on important information.

(5) TextRCNN [28]: this model is based on an RNN model and extracts crucial features by employing a maximum pooling strategy.

(6) bi-LSTM-CNN: this model combines the bi-LSTM and CNN models. bi-LSTM is used to process the sequence information, and CNN is used to extract the local features.

(7) BCF: in this model, the sentiment features are obtained using a sentiment lexicon and combined with the bi-LSTM-CNN's pooling layer output.

(8) BAC: this model first processes the sequence information using a bi-LSTM model, then uses an attention mechanism to give weight to important information, and then uses CNN to extract local features.

### 4.4. Learning Rate Selection

The selection of hyperparameters is crucial in model training, and the learning rate is one of the most significant hyperparameters. If the learning rate is too high, the model will miss the global optimal point, and if the learning rate is too low, the difficulty of the model's convergence will increase. To select the appropriate value, this paper evaluates the model's $F_w$ under various learning rates, as shown in Table 5.

**Table 5.** Learning rate selection.

| Learning Rate | $F_w$ |
|---|---|
| 0.0001 | 0.5629 |
| 0.0003 | 0.7410 |
| 0.001 | 0.7586 |
| 0.003 | 0.7733 |
| 0.01 | 0.8403 |
| 0.03 | 0.3352 |
| 0.1 | 0.2194 |
| 0.3 | 0.2244 |

As can be seen from Table 5, $F_w$ reduces significantly to 33.52% when the learning rate exceeds 0.03, and the model may miss the global optimal point. When the learning rate is less than 0.0003, the model may hover at the local optimal point, resulting in an $F_w$ of less than 70%. As a result, the learning rate is selected as 0.01.

### 4.5. Results Analysis

(1) Comparison experiments with traditional deep learning models

The performance of BACF is contrasted with the traditional deep learning model to evaluate the model's efficiency. The experimental results are shown in Table 6.

**Table 6.** Experimental results of comparison with traditional deep learning models.

| Model | $P_w$ | $R_w$ | $F_w$ |
|---|---|---|---|
| LSTM | 73.82% | 75.01% | 73.86% |
| CNN | 78.21% | 78.33% | 77.33% |
| GRU | 78.71% | 78.53% | 78.57% |
| bi-LSTM-Att | 79.45% | 78.33% | 78.64% |
| TextRCNN | 79.82% | 80.12% | 79.20% |
| bi-LSTM-CNN | 80.32% | 80.37% | 79.33% |
| BACF | 84.42% | 83.99% | 84.03% |

Table 6 shows that the BACF proposed in this paper achieves the best prediction performance, with the highest $F_w$ of 84.03%. Among the traditional deep learning models, CNN outperforms LSTM, while GRU outperforms CNN. In terms of the dataset, reviews range in length from [1, 359], while Figure 3 depicts that almost 80% of texts have lengths of less than 61. LSTM processes sequence data through the forget gate, remember gate, and output gate, which is more suitable for long text. When the amount of data is not sufficient, it can easily lead to overfitting. CNN is excellent at processing local features, so the short texts are in its favor. However, when the LSTM extracts bidirectional information and combines an attention mechanism, the $F_w$ of bi-LSTM-Att improves significantly and is 1.31% greater than that of CNN. GRU has a better prediction effect than CNN in $P_w$, $R_w$, and $F_w$, respectively, because the structure of GRU is useful for processing sequence data. By combining the RNN structure with the maximum pooling layer, TextRCNN can appropriately capture textual information. Therefore, TextRCNN is the top performer among the traditional models. However, the $F_w$ of this model is 79.20%, which is still approximately 4% lower than BACF, and all other models are below 80% in all metrics. BACF is constructed by integrating deep learning, sentiment lexicon, and feature fusion. The experiment shows that, in comparison to the traditional deep learning model, the BACF proposed in this paper can more effectively achieve five-category stance detection.

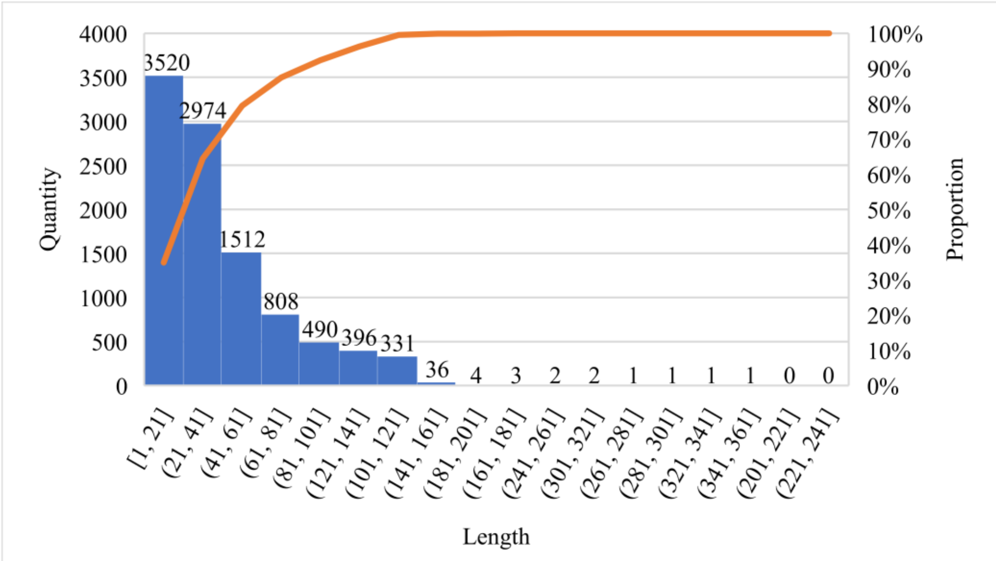


**Figure 3.** Length statistics of reviews.

Hybrid neural networks can extract features more comprehensively, which can improve the accuracy of stance prediction. A review may contain important information scattered throughout. When posting reviews, users sometimes express their opinions right away, while other times they reserve them for the end or bury them in the midst. RNNs can be used to process sequence data. However, words that appear later typically have an advantage in RNNs. In contrast, CNNs are an impartial model with the advantage of extracting features by using kernels of various sizes. Hybrid neural networks that combine CNNs with bi-LSTM can employ CNNs to extract local features while benefiting from the advantages of RNNs that are good at processing sequential data. Table 6 shows that bi-LSTM-CNN performs better on $P_w$, $R_w$, and $F_w$ when compared to the traditional single models, which shows that the hybrid neural network improves stance detection.

(2) Ablation Study

An ablation study was performed to demonstrate the importance of each module, and the results are shown in Table 7.

**Table 7.** Experimental results of ablation study.

| **Model** | $P_w$ | $R_w$ | $F_w$ |
|---|---|---|---|
| bi-LSTM-CNN | 80.32% | 80.37% | 79.33% |
| BCF | 82.17% | 81.95% | 81.91% |
| BAC | 82.18% | 82.20% | 82.07% |
| BACF | 84.42% | 83.99% | 84.03% |

Combining the sentiment lexicon with deep learning can help improve stance detection. Users usually express their attitudes while also expressing their sentiments. Taking sentimental factors into consideration might be helpful for identifying the user's stance. At the same time, combining stance detection and sentiment analysis can reduce the interference of moody expressions. In this paper, the sentiment lexicon is used to calculate the sentiment score of the review, and a serial strategy is used to fuse the sentiment features with the textual features. Table 7 shows that by fusing the sentiment features, BCF outperforms bi-LSTM-CNN on $F_w$ by 2.58%. Furthermore, compared to BAC, BACF improves

$P_w$, $R_w$, and $F_w$ by 3.24%, 1.79%, and 1.96%, respectively. The experiment demonstrates that this model can efficiently employ sentiment information to assist the model in achieving stance detection, which improves the model's prediction performance. Therefore, the fusion of sentiment features can help the model perform stance detection more effectively.

The attention mechanism can improve the model's predictions. By utilizing an attention mechanism, the model can concentrate on crucial information and improve prediction accuracy by giving different weights. BAC improves the model's performance by giving more weight to significant information and less weight to irrelevant information via the attention mechanism. Table 7 shows that BAC performs better than bi-LSTM-CNN on $P_w$, $R_w$, and $F_w$ through improvements of 1.86%, 1.83%, and 2.74%, respectively. The results show that the performance of the model can be significantly improved by introducing an attention mechanism. Therefore, by weighting the important information, the attention mechanism allows the model to predict the stance more accurately.

(3) Verification of generalization

To verify the generalization of BACF, this research selects open-source datasets for experiments. Since the current research does not have the same task, this paper chooses a dataset in the field of opinion mining. This dataset is about hotel reviews and includes 2444 negative reviews and 5322 positive reviews. One of the negative reviews is empty, resulting in a total of 7765 reviews. The dataset was divided into training, validation, and test sets in the ratio of 6:2:2. The experimental results are shown in Table 8.

**Table 8.** Experimental results of verification of generalization.

| Model | $P_w$ | $R_w$ | $F_w$ |
|---|---|---|---|
| LSTM | 73.75% | 74.52% | 71.21% |
| CNN | 83.04% | 83.40% | 83.02% |
| GRU | 84.12% | 84.43% | 84.11% |
| bi-LSTM-Att | 84.62% | 84.81% | 84.69% |
| TextRCNN | 84.08% | 84.23% | 83.59% |
| BACF | 85.80% | 85.07% | 85.29% |

Table 8 shows that the BACF proposed in this study has the best performance, with a $P_w$, $R_w$, and $F_w$ of 85.80%, 85.07%, and 85.29%, respectively. It is followed by bi-LSTM-Att, which incorporates the attention mechanism and has an $F_w$ of 84.69%. TextRCNN using a pooling layer also achieves good results with an $F_w$ of 83.59%. The lowest results come from LSTM, which has an $F_w$ of only 71.21%, which should be due to the fact that LSTM is not effective at processing short texts. As a result, the BACF proposed in this paper is more generalizable.

In conclusion, the BACF proposed in this paper employs a hybrid neural network structure, introduces sentiment information via feature fusion, uses the attention mechanism to weight essential information, and outperforms other approaches in the stance detection task. In comparison to traditional deep learning models, BACF can effectively carry out stance detection and has a better accuracy rate.

## 5. Public Opinion Management

In this work, a multi-stance indicator system is constructed using the LDA model to mine topics with varying degrees and polarities of stance. Topics are categorized according to the polarity and degree of the stance to which they are related. We employ the coherence $C_V$ to select the right number of topics [29]. Coherence measures the degree of semantic similarity of keywords in the topic [30]. The larger $C_V$ is, the better the model effect. To identify a reasonable number of topics, the experiment iterated the coherence under different topic counts and discovered that the model performed best with 15 topics, as shown in Figure 4.

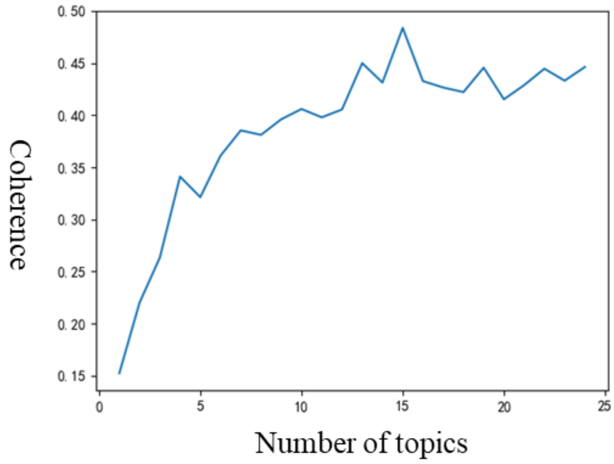


**Figure 4.** Identification of the number of topics.

According to the differences that exist in the degree and polarity of the topics in expressing their stances, this paper develops a five-category stance indicator system and classifies the 15 topics into five stances, as shown in Table 9. Word clouds are generated for each stance, and there is a substantial difference in polarity and the degree of stance, as shown in Figure 5.


**Figure 5.** Word clouds for each stance: (**1**) Strong support, (**2**) Weak support, (**3**) Neutral, (**4**) Weak opposition, (**5**) Strong opposition.

**Table 9.** Five-category stance indicator system and their corresponding keywords.

| Stance | Topic | Keywords |
|---|---|---|
| SS | # The beaten child had displayed unacceptable behavior. Understand and support the doctor. | Deserved, your child, his family, calm, powerful |
| WS | # Opposition to the beaten child's guardian's use of internet violence and the concealment of facts. | Internet violence, awesome, bad, dodge the issue, conflict |
| | # The guardian of the beaten child is primarily responsible for the event. | Understand, this family, moral values, neighborhood, post online |
| | # The violence of the beaten child is outrageous. | Poke, violence, reverse, good, anger |
| Neutral | # Discussion of the details related to the event. | Fights, sanity, comment, events, civil servants |
| | # Discussion of the conduct and responsibility of the person involved. | Bad child, public opinion, kindergarten, grandfather, Dr. Lu |
| | # Discussion of the causes of the event. | Judgment, face, puncture, truth, jab |
| | # Discussion of the impact of the event. | Devil, event, trauma, psychological, guardian |
| | # Making snide remarks about the event. | Slap, hope, correction, police department, nice |
| | # Both sides in the event are to blame. | Beat, bad woman, call the police, family, an eye for an eye |
| WO | # The doctor's assault was just impulsive. | Kid, doctor, adult, hit, impulsive |
| | # The main mistake of the doctor lies in the approach to treatment. | Hit back, my house, tell, claim, ability |
| | # The doctor's assault rendered his reasonable action unreasonable. | Dr. Lu, would have, reasonable, consequences, responsible |
| SO | # The assault will be severely punished under the law. | CD (abbreviation of criminal detention), cost, criminal detention, death, deal |
| | # Adults shouldn't hit children in any case. | Fight, kid, adult, certainly, inappropriate |

By analyzing keywords across 15 categories, we can categorize reviews into 15 topics. If a review strongly opposes the behavior of a child and supports the doctor, it is classified as "SS". If a review opposes the guardian of the child or online violence or simply considers the child's behavior as too extreme, it is classified as "WS". If a review suggests that the doctor is at fault for impulsively hitting the child or believes that the doctor is not at fault for hitting the child, it is labeled as "WO". If a review strongly opposes the doctor and suggests legal punishment, it is designated as "SO". All other topics fall under the neutral category.

Taking "SS" as an example, considering keywords such as "deserved", "your child", "his family", "calm", and "powerful", we can infer the following conclusions from the context of the event and the meaning of these terms. In this case, the phrase "deserved" implies that the child's behavior is believed to warrant corresponding consequences. The term "calm" may indicate that most people find it difficult to remain rational in this situation. This viewpoint may lead users to be unable to accept the child's behavior and hold a supportive attitude towards the doctor.

As can be seen from Table 9, SS believes that the fault lies with the beaten child and backs the doctor's conduct in defending his rights. WS shows solidarity with the doctor by opposing the guardian of the beaten child, etc. SO believes that using violence to uphold rights is utterly unacceptable, while WO believes that the doctor just acted impulsively. In addition, Neutral discusses the details of the event, including the cause and effect of the incident, etc.

Then, these issues can be dealt with specifically. For SS, we should pay attention to calming their anger and informing them about the relevant laws; for WS, we need to respond positively to the queries raised by them; for SO, we should promptly inform them of the processing results and meet their demands; and for WO, we must pay attention to their opinions and suggestions. Additionally, for Neutral, we must release information and respond to queries in a timely manner.

Therefore, the five-category stance suggested in this study not only reflects the user stance in a more detailed way but also shows differences in the user stance that cannot be realized by the three-category stance. The five-category stance helps to better realize opinion mining and has great application value.

## 6. Conclusions and Prospect

This paper proposes a multi-stance detection method by fusing sentiment features. A five-category stance indicator system is initially created based on the LDA model to distinguish stances of different polarities and degrees. Then, the sentiment lexicon and deep learning are combined to construct a stance detection model.

The experimental results indicate that the model achieved the highest score of 85.29% on the $F_w$ metric, demonstrating its ability to effectively identify stances of different polarities and degrees. Furthermore, thanks to the topic mining feature of LDA, the model accurately describes not only the users' stances and sentiments but also reveals the differences between stances. With a more comprehensive classification of stances, BACF can accurately capture users' stances and attitudes and recognize the differences and reasons between different stances. This understanding holds significant implications for public opinion management and has enormous potential for applications in areas such as policy-making and market planning.

Although the method proposed in this paper achieves improved results in multi-stance detection, it primarily relies on manual judgment when categorizing topics into different polarities and degrees of stance. A future study will focus on how to construct a quantitative relationship between topics and stances. Meanwhile, as online social networks evolve, more diverse and complicated user features may emerge. A future study will focus on how to constantly update the model to adapt it to new application conditions.

**Author Contributions:** Conceptualization, W.H. and J.Y.; methodology, W.H; software, J.Y.; validation, J.Y.; formal analysis, W.H. and J.Y. data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, W.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China—Research on the guiding and blocking mechanism of public opinion spread from the perspective of emotion model (72271128); Research on security and stability of international cooperation supply chain in cyberspace, a key work support project of the Research Base of International Governance in Cyberspace for 2023.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original data presented in the study are openly available in GitHubat (https://github.com/hello-world2024/A_Multi_Stance_Detection_Method_by_Fusing_Sentiment_Features.git, accessed on 25 April 2024).

**Conflicts of Interest:** No potential conflicts of interest were reported by the authors.

## References

1. Dilek, K.; Fazli, C. Stance Detection: A Survey. *ACM Comput. Surv.* **2020**, *53*, 1–37. [CrossRef]
2. Jia, P.; Du, Y.; Hu, J.; Li, H.; Li, X.; Chen, X. An Improved BiLSTM Approach for User Stance Detection Based on External Commonsense Knowledge and Environment Information. *Appl. Sci.* **2022**, *12*, 10968. [CrossRef]
3. Jing, M.; Wei, G.; Kam-Fai, W. Detect Rumor and Stance Jointly by Neural Multi-task Learning. In Proceedings of the WWW'18: Companion, The Web Conference, Lyon, France, 23–27 April 2018; pp. 585–593. [CrossRef]

4. Poddar, L.; Hsu, W.; Lee, M.L.; Subramaniyam, S. Predicting Stances in Twitter Conversations for Detecting Veracity of Rumors: A Neural Approach. In Proceedings of the IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece, 5–7 November 2018; pp. 65–72. [CrossRef]
5. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
6. Keawtoomla, N.; Pongwat, A.; Bootkrajang, J. Using Latent Dirichlet Allocation to investigate guest experience in Airbnb accommodation during COVID-19 pandemic in the United Kingdom. In Proceedings of the 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), Bangkok, Thailand, 22–25 June 2022; pp. 1–6. [CrossRef]
7. Tomojiri, D.; Takaya, K.; Ise, T. Temporal trends and spatial distribution of research topics in anthropogenic marine debris study: Topic modelling using latent Dirichlet allocation. *Mar. Pollut. Bull.* **2022**, *182*, 113917. [CrossRef] [PubMed]
8. Xu, X.; Hu, F.; Du, P.; Wang, J.; Li, L. Efficient Stance Detection with Latent Feature. Web and Big Data. In Proceedings of the APWeb-WAIM 2017 International Workshops: MWDA, HotSpatial, GDMA, DDC, SDMA, MASS, Beijing, China, 7–9 July 2017; Volume 10612, pp. 21–30. [CrossRef]
9. Sherif, S.S.; Shawky, D.M.; Fayed, H.A. An Ensemble Model for Stance Detection in Social Media Texts. *Int. J. Inf. Technol. Decis. Mak.* **2023**, *22*, 737–775. [CrossRef]
10. Liu, W.; Peng, X.; Li, C.; Wang, P.; Wang, L.H. A Survey on Stance Detection. *J. Chin. Inf. Process.* **2020**, *34*, 1–8.
11. Dey, K.; Shrivastava, R.; Kaushik, S. Twitter Stance Detection—A Subjectivity and Sentiment Polarity Inspired Two-Phase Approach. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 365–372. [CrossRef]
12. Mourad, S.S.; Shawky, D.M.; Fayed, H.A.; Badawi, A.H. Stance Detection in Tweets Using a Majority Vote Classifier. In Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), Cairo, Egypt, 22–24 February 2018; Volume 723, pp. 375–384. [CrossRef]
13. Siddiqua, U.A.; Chy, A.N.; Aono, M. Tweet Stance Detection Using an Attention based Neural Ensemble Model. In Proceedings of the 2019 North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 1868–1873. [CrossRef]
14. Liu, H.; Li, S.; Zhou, G. Two-Target Stance Detection with Target-Related Zone Modeling. In Proceedings of the 24th China Conference, CCIR 2018, Information Retrieval, Guilin, China, 27–29 September 2018; pp. 170–182. [CrossRef]
15. Zhang, X.; Liu, C.; Gao, Z.Q.; Jiang, Y. Text Stance Detection Based on Deep Learning. In Proceedings of the 2021 IEEE International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, 17–19 December 2021; pp. 193–199. [CrossRef]
16. Tran, O.T.; Phung, A.C.; Bach, N.X. Using Convolution Neural Network with BERT for Stance Detection in Vietnamese. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 7220–7225.
17. Li, W.; Xu, Y.; Wang, G. Multi-target stance detection based on gru-pwv-cnn network model. *J. Internet Technol.* **2021**, *22*, 593–603.
18. Li, C.; Peng, H.; Li, J.X.; Sun, L.; Lyu, L.; Wang, L.; Philip, S.Y.; He, L. Joint Stance and Rumor Detection in Hierarchical Heterogeneous Graph. *J. IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2530–2542. [CrossRef] [PubMed]
19. Liu, C.; Zhou, K.X.; Zhou, L.X. Infusing external knowledge into user stance detection in social platforms. *J. Intell. Fuzzy Syst.* **2024**, *46*, 2161–2177. [CrossRef]
20. Zheng, H.Y.; Gao, J.B.; Qiu, J. Stance Detection in Chinese Microblog Topic Based on Word Embedding Technology and Thematic Words Feature. *Comput. Syst. Appl.* **2018**, *27*, 118–123. [CrossRef]
21. Dey, K.; Shrivastava, R.; Kaushik, S. Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention. In Proceedings of the 40th European Conference on IR Research, ECIR 2018, Advances in Information Retrieval, Grenoble, France, 26–29 March 2018; pp. 529–536. [CrossRef]
22. Karande, H.; Walambe, R.; Benjamin, V.; Kotecha, K.; Raghu, T.S. Stance detection with BERT embeddings for credibility analysis of information on social media. *J. PeerJ Comput. Sci.* **2021**, *7*, e467. [CrossRef] [PubMed]
23. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *J. Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
24. Le, C.Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *J. Neural Comput.* **1989**, *1*, 541–551.
25. Liu, P.F.; Qiu, X.P.; Huang, X.J. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
26. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
27. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.Y.; Li, B.C.; Hao, H.W.; Xu, B. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 207–212.
28. Lai, S.W.; Xu, L.H.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2267–2273.

29. Röder, M.; Both, A.; Hinneburg, A. Exploring the Space of Topic Coherence Measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 399–408. [CrossRef]
30. Keith, S.; Philip, K.; David, A.; David, B. Exploring topic coherence over many models and many topics. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Republic of Korea, 12–14 July 2012; pp. 952–961.