



## Article

# BSTC: A Fake Review Detection Model Based on a Pre-Trained Language Model and Convolutional Neural Network

Junwen Lu <sup>1</sup>, Xintao Zhan <sup>1,\*</sup>, Guanfeng Liu <sup>2</sup>, Xinrong Zhan <sup>1</sup>  and Xiaolong Deng <sup>1</sup> 

<sup>1</sup> School of Computer and Information and Engineering, Xiamen University of Technology, Xiamen 361024, China; jwlu@xmut.edu.cn (J.L.); 15626257844@163.com (X.Z.); shannondeng@bupt.edu.cn (X.D.)

<sup>2</sup> School of Computing, Macquarie University, Sydney, NSW 2109, Australia; guanfeng.liu@mq.edu.au

\* Correspondence: 13510387483@163.com

**Abstract:** Detecting fake reviews can help customers make better purchasing decisions and maintain a positive online business environment. In recent years, pre-trained language models have significantly improved the performance of natural language processing tasks. These models are able to generate different representation vectors for each word in different contexts, thus solving the challenge of multiple meanings of a word, which traditional word vector methods such as Word2Vec cannot solve, and, therefore, better capturing the text's contextual information. In addition, we consider that reviews generally contain rich opinion and sentiment expressions, while most pre-trained language models, including BERT, lack the consideration of sentiment knowledge in the pre-training stage. Based on the above considerations, we propose a new fake review detection model based on a pre-trained language model and convolutional neural network, which is called BSTC. BSTC considers BERT, SKEP, and TextCNN, where SKEP is a pre-trained language model based on sentiment knowledge enhancement. We conducted a series of experiments on three gold-standard datasets, and the findings illustrate that BSTC outperforms state-of-the-art methods in detecting fake reviews. It achieved the highest accuracy on all three gold-standard datasets—Hotel, Restaurant, and Doctor—with 93.44%, 91.25%, and 92.86%, respectively.

**Keywords:** fake review detection; pre-trained language model; BERT; SKEP; TextCNN



**Citation:** Lu, J.; Zhan, X.; Liu, G.; Zhan, X.; Deng, X. BSTC: A Fake Review Detection Model Based on a Pre-Trained Language Model and Convolutional Neural Network. *Electronics* **2023**, *12*, 2165. <https://doi.org/10.3390/electronics12102165>

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 17 March 2023

Revised: 26 April 2023

Accepted: 6 May 2023

Published: 9 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Online product and business reviews have grown to be an increasingly valuable source of information for online customers, as these reviews have a considerable influence on their purchasing decisions, thanks to the rise of the Internet and e-commerce. However, online reviews are not always accurate and trustworthy. As early as 2007, Jindal et al. [1,2] discovered the issue of fake reviews of online products. Some businesses engage writers to post favorable reviews to promote their items or critical reviews to target competitors to increase profits [3]. On the other hand, it could be very difficult for consumers to distinguish fake reviews, and consumers frequently struggle to identify deceptive fake reviews [4]. The proliferation of fake reviews not only misleads potential customers, but also hinders the stable development of online platforms [5]. Therefore, an effective fake review detection method is a real need. Effectively detecting fake reviews on online platforms is critical for improving user experience and maintaining a safe and trustworthy online business environment.

The purpose of fake review detection is to distinguish if a review is real or fake, and its essence is text classification. Efficiently extracting features from reviews is the primary problem faced by fake review detection. The traditional classification model based on the bag-of-words [6] model and its variant methods ignore the relative position information of words and have limitations in representing the contextual information and semantic features of the text. Consequently, the traditional text classification model is

incapable of effectively capturing the multi-level features of a review's complex semantics. In comparison with the bag-of-words model, as a text feature representation method, deep learning can effectively capture a text's contextual information and semantic features by learning the vector representation of the text, resulting in good text classification results. However, deep-learning-based text classification methods typically use the traditional word vector model to vectorize the preprocessed text before extracting features. Because these word vectors ignore word polysemy, deep learning methods still have limitations in text representation. In addition to the aforementioned issues, product reviews also contain an array of topics, a large amount of information, and a wide range of subjective sentiments. Depending on the expression of sentiments, fake reviews are able to be positive, negative, or neutral [7]. At the same time, compared with real reviews, fake reviews are frequently more impactful and have a more complicated structure. It is difficult to extract fully effective features of review texts if these conditions are not considered when selecting the classification method [8].

As natural language processing (NLP) technology has advanced, some pre-trained models [9] have been applied to various NLP tasks, such as text classification and sentiment analysis, significantly improving their performance. In response to the aforementioned issues, we will adopt advanced pre-trained language models to enhance the model's ability to capture review features, thus improving the performance of the model in fake review detection.

This paper's contributions are summarized as follows:

- To capture review features, we first adopt the pre-trained language models BERT [10] and SKEP [11]. BERT captures generic semantic information from the reviews, while SKEP analyzes the sentiments and opinions expressed in the reviews.
- The advantages of pre-trained language models and convolutional neural networks are combined in our newly suggested model. The output of the pre-trained model is input into TextCNN [12], and TextCNN is utilized to further extract the local features and critical information of reviews to enhance the model's performance in fake review detection.
- Finally, we executed a variety of experiments to assess the performance of BSTC, and the findings illustrate that our model outperformed others.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 describes our proposed model, as well as the detection process for fake reviews. Section 4 introduces the relevant experiments and their outcomes. Finally, in Section 5, we summarize this study.

## 2. Related Work

Text classification [13] is a type of natural language processing task. Since the 1950s, researchers have been pursuing relevant studies. Fake review detection is the process of separating reviews into real and fake reviews, which is commonly regarded as a binary text classification task [14–16]. So far, many methods have been applied to detect fake reviews. Table 1 provides information comparing our model with those of prior studies.

**Table 1.** Comparison of relevant methods for fake review detection.

Method	Description
NB [15], K-NN [17], and SVM [18]	These traditional methods require manual feature extraction, which leads to a large amount of manual participation when processing large datasets, and they are prone to feature redundancy, which makes them difficult to expand and limited in accuracy.
SWNN [19]	SWNN is an improved document representation model based on a CNN. To better learn the semantics of documents, SWNN captures the importance of different sentences by synthesizing sentence representations into document representations.

Table 1. Cont.

Method	Description
CNN-GRNN [20]	This model uses a CNN to acquire sentence-level representations before integrating document-level representations of sentences through GRNN.
DFFNN [21]	Previous neural-network-based methods only considered word embedding and ignored the sentiment index of reviews. The improved DFFNN model proposed by Hajek et al. learns document-level representations of reviews by using n-grams, word embeddings, and three lexicon-based emotion indicators.
BERT [10]	BERT is a popular pre-trained language model. BERT can build dynamic word embedding representations to better represent contextual semantics and maintain the best results in most NLP fields. Although BERT has shown a great ability to learn the general semantics of texts, it does not explicitly study the emotional information of texts in the pre-training process, so it is difficult to expect it to provide the best results for sentiment analysis of review texts.
BSTC (this study)	BSTC is based on BERT, SKEP, and TextCNN. BSTC not only considers general semantic information, but also captures emotional features through SKEP. To obtain an emotional semantic representation, SKEP employs unsupervised approaches to autonomously mine sentiment knowledge. Compared with lexicon-based sentiment analysis methods, SKEP can capture more emotional information more comprehensively and accurately.

Traditional text classification methods based on machine learning, such as Naive Bayes (NB) [15], K-Nearest Neighbor (K-NN) [17], and the Support Vector Machine (SVM) [18], have been applied to detect fake reviews. These methods typically represent text by employing the bag-of-words [6] model, ignoring the natural sequence structure and contextual information of the text, which makes it impossible to adequately capture a review text's contextual semantic information. In addition, these traditional methods require manual extraction of text features and their input into the classifier. Although manually extracted features can improve a model's classification effect, they have low efficiency and low effectiveness in the model's generalization [22].

With the development of deep learning in NLP, various methods based on neural networks have been applied to fake review detection. The most often utilized models are convolutional neural networks (CNNs) [23] and recurrent neural networks (RNNs) [24]. Neural-network-based text classification models usually use word embedding as a feature representation method. Word embedding is a method for representing a word as a fixed-size vector with the use of contextual information, which preserves the word's contextual information [25]. These neural network models have been demonstrated to be excellent at extracting the semantic information concealed in fake reviews, which is difficult to convey when using traditional discrete manual characteristics [26]. Li et al. [19] detected fake reviews by using the text representation vector produced by a CNN. To capture semantic information more efficiently, they considered the influences of various sentences in the review text, and sentence weights were taken into consideration in the process of learning text representations. Ren et al. [20] studied a neural network model that combined a CNN and gated RNN in a framework to learn document-level representations for the detection of fake reviews. One disadvantage of these neural network models is that they solely consider word embeddings, omitting the sentiment features of the reviews. Hajek et al. [21] introduced an enhanced DFFNN model that considered the impacts of various word, sentence, and sentiment representations on the detection of fake reviews. Compared with the previous models, the model suggested by Hajek et al. could capture richer features from review texts. Methods based on neural networks have achieved excellent outcomes in

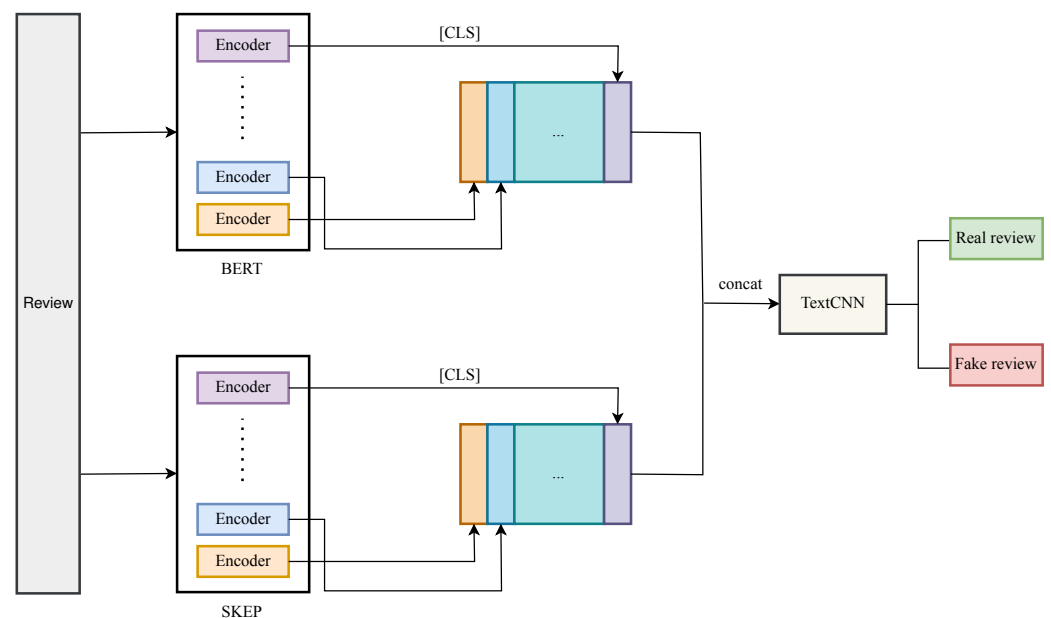
fake review detection, but there is an obvious disadvantage in that these methods generally use word embedding models, such as GloVe [27] and Word2Vec [28], to vectorize review texts for representation. GloVe and Word2Vec models can only generate fixed word vectors and cannot effectively reflect the semantics of words in varied contexts [29].

In 2018, the pre-trained language model Bidirectional Encoder Representation from Transformers (BERT) [10] released by Google swept the optimal results of 11 tasks in the NLP field. BERT can build dynamic word embedding representations, which solves the problem of the polysemy of a word, which traditional word vector methods such as Word2Vec cannot solve, so as to better model contextual semantics [30]. Although BERT demonstrated its powerful function in learning general semantic representations, it did not explicitly study the emotional information of text in the pre-training process [11]. The Sentiment Knowledge Enhanced Pre-training (SKEP) model was suggested by Tian et al. [11]; it combines diverse kinds of sentiment knowledge and produces a unified and powerful sentiment representation for a variety of sentiment analysis tasks. We saw from the work of Hajek et al. [21] that integrating sentiment features can boost a model's performance in detecting fake reviews.

Based on the research presented above, we propose the BERT-SKEP-TextCNN (BSTC) fake review detection model, which takes into account the benefits of pre-trained language models and convolutional neural networks for text feature extraction. We used BERT and SKEP to capture the contextual semantic features and emotional features of comments, respectively, and then integrated these two features and fed them into TextCNN. TextCNN was used to further extract local features and crucial information, resulting in high-quality feature representation. To demonstrate the validity of BSTC, we compared it with multiple baseline models on three gold-standard fake review datasets.

### 3. Proposed Method

The architecture of BSTC is displayed in Figure 1, which shows the BERT-SKEP layer and TextCNN layer. In the following, we will introduce the structure of BSTC and the fake review detection process in detail.



**Figure 1.** The BSTC model's architecture.

#### 3.1. BERT-SKEP Layer

BERT [10] is a deep bidirectional transformer encoder-based language understanding model that is divided into two models with different parameter sizes: BERT<sub>BASE</sub> and BERT<sub>LARGE</sub>. Figure 2 shows the structure of BERT.  $E$  represents the input vector of the

model, and the corresponding word vector  $T$  is output after passing through the multilayer bidirectional transformer encoder.  $Trm$  is the abbreviation of transformer encoder.

The transformer encoder is made up of a multi-head attention mechanism that allows for parallel operation. The following is the formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

$$Multihead(Q, K, V) = concat(head_1, \dots, head_h)W^O \tag{3}$$

where  $Q, K,$  and  $V$  are vectors from the same input, and the dimensions are  $d_k$ . After a linear transformation, new matrices  $W^Q, W^K,$  and  $W^V$  are obtained.  $W^O$  represents the mapping vector of Multihead.

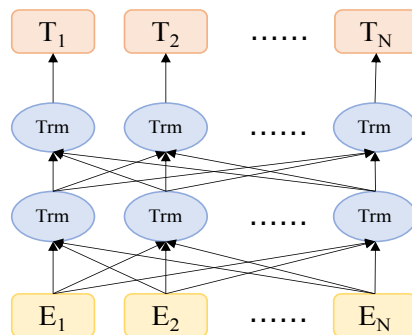


Figure 2. The BERT model’s architecture.

As shown in Figure 3, BERT’s input representation is made up of the total of token embeddings, segment embeddings, and position embeddings. The tokens [CLS] and [SEP] represent the beginning and end of a sentence, and they are automatically added by BERT. The token embeddings include the vector embedding of each word, as well as the character embeddings of words that are not in the vocabulary. The segment embeddings are employed to train the model on the next sentence prediction task. The position embeddings are due to BERT using a self-attention [31] mechanism. The relative distance between words in the input sentence is 1, which overcomes the issue of the text’s long-distance dependence but loses the original word order information. By adding the information on the position to each word in the form of position encoding, the model is provided with the relative locations between words, which enhances the integrity of the data in the text representation.

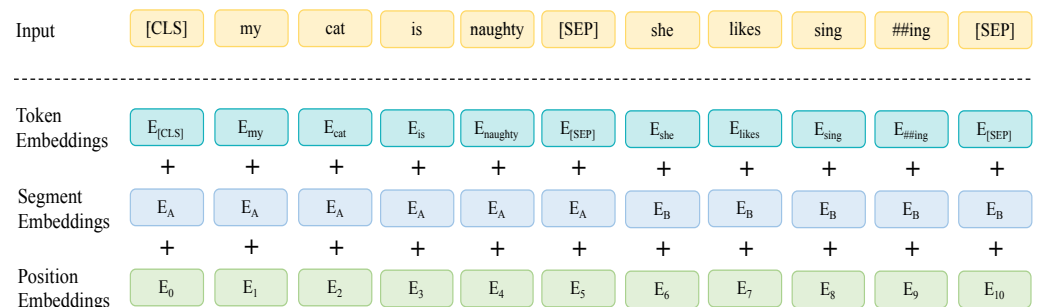


Figure 3. BERT’s input representation.

The position embedding encoding formula is as follows:

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}) \tag{4}$$

$$PE(pos, 2i + 1) = \cos(pos/10000^{2i/d_{model}}) \tag{5}$$

where  $i$  represents the index value of the position vector,  $pos$  represents the word’s position in the text vector, and  $d_{model}$  is the text vector’s dimension.

SKEP [11] was proposed by the Baidu research team. Like BERT, SKEP applies a transformer encoder layer.

SKEP includes sentiment masking and sentiment pre-training objectives, as shown in Figure 4. Sentiment masking detects and deletes the sentiment information from an input sequence by using autonomously mined sentiment knowledge, resulting in a corrupted version. To achieve three sentiment pre-training criteria, the transformer must restore the sentiment information for the corrupted version. SKEP employs a straightforward and efficient method based on point mutual information (PMI) [32] to extract sentiment knowledge from unlabeled data, including sentiment words, word polarity, and aspect–sentiment pairs. Sentiment masking seeks to create a corrupted version of each input sequence with masked sentiment information. Based on the automatic mining of sentiment knowledge, SKEP masks a few words in the original input sentence, replacing them with special characters [MASK]. Unlike the previous random word masking, the sentiment masking of SKEP is guided by sentiment knowledge. The goal of sentiment pre-training is to restore the sentiment information in the sequence after the original input sentence generates a defect sequence through sentiment masking.

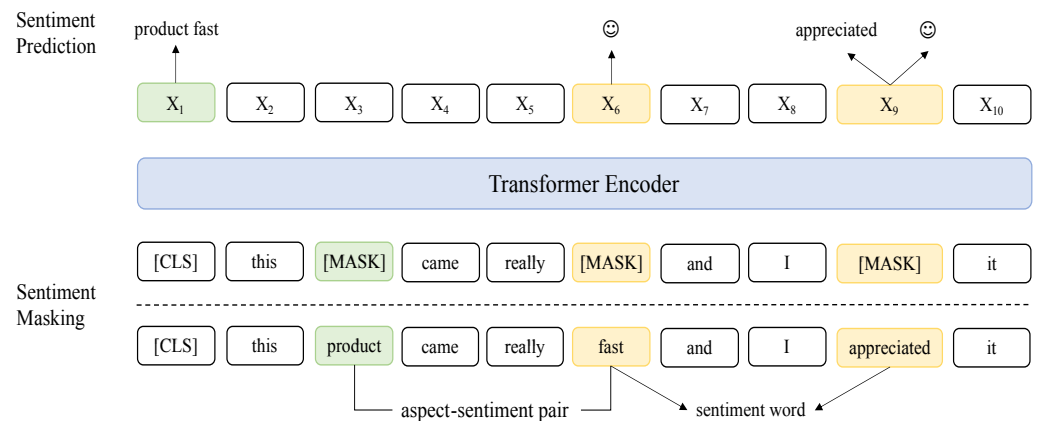


Figure 4. Sentiment knowledge enhanced pre-training (SKEP).

The BERT model adopted in this paper was BERT<sub>LARGE</sub>, with 24 encoder layers and a 1024 output vector dimension. The SKEP model also had 24 encoder layers, and the output vector dimension was also 1024. We used the BERT<sub>LARGE</sub> model mainly because of its larger number of parameters and deeper network structure. BERT<sub>LARGE</sub> is able to learn more detailed and complex semantic information than the BERT<sub>BASE</sub> model; thus, it can better understand and process text data. We began by entering the reviews into the BERT and SKEP models. In BERT<sub>LARGE</sub>, excluding the first embedding layer, there were 24 encoder layers, and the first token ([CLS] vector) of each encoder layer was able to be treated as a sentence vector. The deeper the encoder layer, the better the sentence vector can represent high-level semantic information. We spliced the [CLS] vectors of each encoder layer to obtain the vector  $C_b$ , and we obtained the output vector  $C_s$  of SKEP in the same way. The vectors  $C_b$  and  $C_s$  were directly spliced to obtain the vector  $C$ , which is represented below:

$$C = \text{concat}(C_b, C_s) \tag{6}$$

### 3.2. TextCNN Layer

TextCNN [12] is a CNN variant model that includes an embedding layer, a convolution layer, a pooling layer, and a fully connected layer. The architecture of TextCNN is shown in Figure 5.

The embedding layer is an  $n \times k$  matrix, where  $n$  is the number of words in a sentence, and  $k$  denotes the dimension of the word vector corresponding to each word. That is, each row of the embedding layer represents a  $k$ -dimensional word vector corresponding to a word.

Convolution is equivalent to filtering data; useless information is filtered to obtain useful features. Text convolution, unlike image convolution, only moves in a vertical direction. The convolution kernel has an equal width to that of the word vector encoding, and its height is a programmable hyperparameter. The convolution layer typically contains convolution kernels of varying sizes, allowing for the extraction of feature information of various dimensions in a text, which aids in the accuracy of the judgment results.

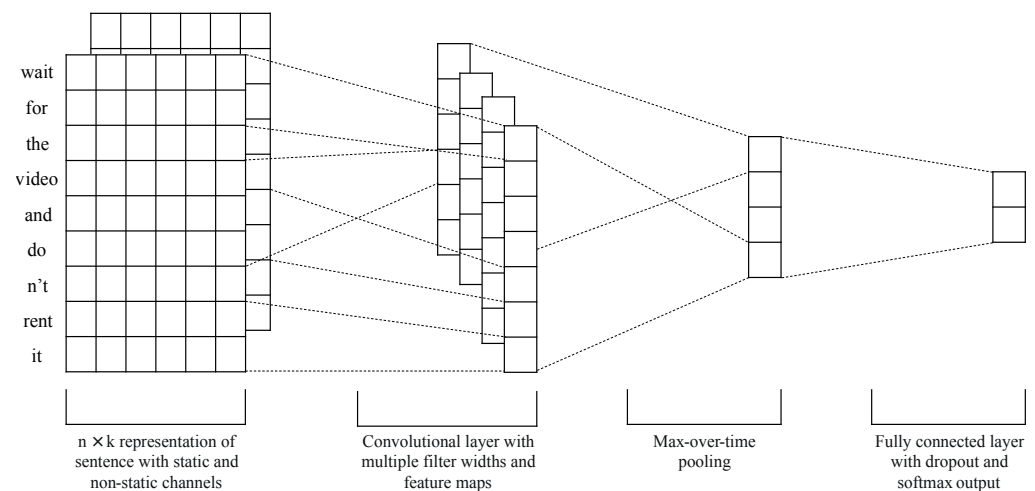
The primary goal of pooling is to lower the dimension of features to reduce the computational complexity, which is equal to secondary feature filtering. The pooling layer employs max-pooling [33] to select the maximum value for the output of the feature by each convolution kernel and then splices all of the maximum values into a one-dimensional vector, thus reducing the number of neural network parameters and feature vectors, achieving dimensionality reduction, and preventing overfitting to a certain degree.

To determine the category of reviews, we created a classifier by using the fully connected layer and the softmax function. The major function of the fully connected layer was to translate the feature space calculated by the front layer to the sample label space. The feature representation was incorporated into a value that reduced the influence of the feature position on the classification results while increasing the overall network's robustness. The softmax function could map the scores of each category label learned by the model to a value between 0 and 1, and the sum of all scores was 1. To prevent overfitting, the dropout technology was introduced to the fully connected layer so that the neurons could be turned to 0 with a certain probability, thus reducing the network's over-dependence on connections.

The softmax function is defined as follows:

$$\text{softmax}(x_i) = e^{x_i} / \sum_{i=1}^n e^{x_i} \quad (7)$$

where  $x_i$  represents the output value of node  $i$ , and  $n$  represents the quantity of output nodes.



**Figure 5.** The TextCNN model's architecture.

We used the front layer's output vector  $C$  as TextCNN's input, which was equivalent to using the BERT-SKEP layer as the embedding layer of TextCNN. The convolution layer of TextCNN was then employed to capture the local features of the review, and the important features in the review were obtained through the pooling layer. Finally, these feature vectors were entered into the fully connected layer to get the review label.

#### 4. Experiments and Discussion

We will first introduce the three benchmark datasets, experimental setup, and evaluation metrics. Then, by using some experiments, we evaluate BSTC's performance.

##### 4.1. Datasets and Experimental Setup

Table 2 shows the statistical information of the three benchmark datasets, which were the Hotel dataset, Restaurant dataset, and Doctor dataset. The details of these three datasets can be found at [34–36]. We assessed BSTC's performance on these three benchmark datasets, which are commonly used for fake review identification. These datasets were selected because they are regarded as gold-standard fake review datasets [35].

**Table 2.** Statistics of the datasets.

Dataset	# of Real/Fake Reviews	Polarity	Total
Hotel [34,35]	800/800	Positive and negative	1600
Restaurant [36]	200/200	Positive	400
Doctor [36]	200/356	Positive	556

The Hotel dataset was provided by Ott et al. [34,35], and it consisted mainly of real and fake hotel reviews for 20 of the most prevalent Chicago hotels, totaling 1600 reviews. The reviews included 800 real reviews and 800 fake reviews, and the real reviews were divided into 400 positive reviews and 400 negative reviews, as were the fake reviews. To obtain the real reviews, Ott et al. collected 6977 positive reviews of the 20 most prevalent Chicago hotels on TripAdvisor. These reviews were filtered through some constraints, such as by deleting some non-English reviews and some reviews with fewer characters than required. In the end, 400 positive reviews were chosen. The 400 real negative reviews came from six prevalent online review communities: Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp. A total of 800 real reviews were collected. The fake reviews were collected by Ott et al. by using the Amazon Mechanical Turk (AMT) crowdsourcing service. They created 400 human-intelligence tasks (HITs) and allocated them equally to 20 selected hotels. Turker received the hotel's name and website from an HIT. Turker was required by the HIT to pretend that they worked for the hotel's marketing department and that their supervisor would like them to write a fictional review and put it on a travel review website. Furthermore, the reviews needed to sound genuine and positively describe the hotel. Only one review per Turker was permitted to ensure that the review was written by a distinct writer. They also limited the task to Turkers based in the United States, and each Turker had an average approval rating of at least 90%. Turkers could work on an HIT for up to 30 min and receive one US dollar for each accepted review that they submitted. It took about 14 days to collect 400 positive fake reviews that were satisfactory. Using the same procedure, Ott et al. collected 400 fake negative reviews on AMT. A total of 800 fake reviews were gathered.

Li et al. [36] adopted the same collection techniques as those of Ott et al. to create two more datasets: the Restaurant dataset and the Doctor dataset. Twenty fake positive reviews were collected from each of the 10 most prevalent restaurants in Chicago for a total of 200 fake reviews. Likewise, 356 fake positive reviews were obtained from the field of doctors. The real review dataset came from customers, with 200 reviews collected in each of the restaurant and doctor fields.



In the experiment, the BERT model that we adopted was the “bert-large-uncased” model, and the SKEP model that we used was the “ernie\_2.0\_skep\_large\_en” model. Both models were from the Hugging Face community and contained 24 encoder layers, and the hidden dimension was 1024. We divided each dataset into a training set and test set according to the ratio of 8:2. Our model used AdamW as an optimizer with a learning rate of  $2 \times 10^{-5}$ , and a weight decay of  $1 \times 10^{-4}$ . Then, we trained the model for 10 epochs on each dataset, with a batch size of 2.

#### 4.2. Evaluation Metrics

In this paper, the accuracy ( $Acc$ ), F1-score ( $F1$ ), precision ( $P$ ), and recall ( $R$ ) were used to assess the model’s effectiveness. The following are the evaluation metrics’ formulas:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$P = \frac{TP}{TP + FP} \quad (9)$$

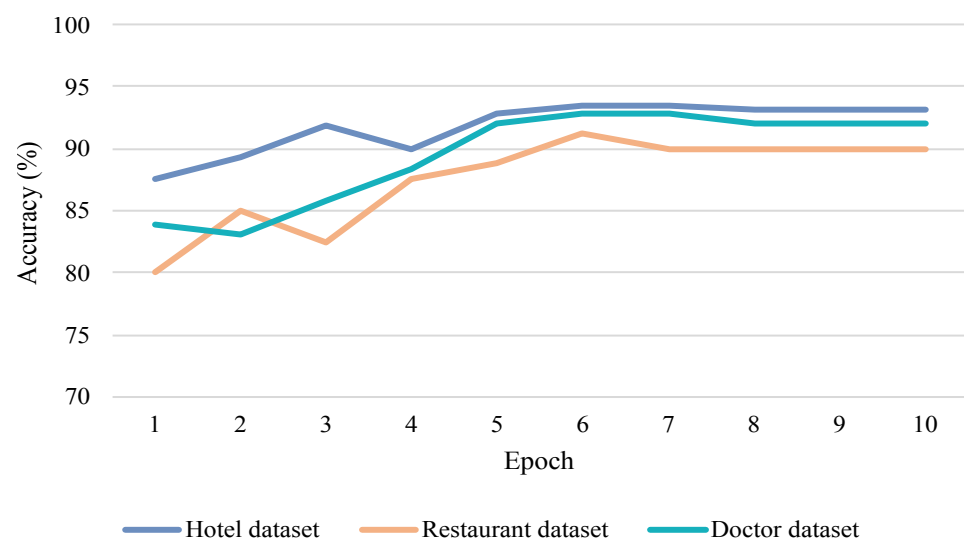
$$R = \frac{TP}{TP + FN} \quad (10)$$

$$F1 = \frac{2RP}{P + R} \quad (11)$$

where  $TP$  represents the number of correctly anticipated fake reviews,  $TN$  represents the number of correctly predicted real reviews,  $FP$  is the number of reviews that were wrongly forecasted as fake reviews, and  $FN$  is the number of reviews that were wrongly forecasted as real reviews.

#### 4.3. Effect of the Training Epochs

Figure 6 shows how the classification accuracy of BSTC changed with the number of training epochs. It could be found that when the training epochs numbered between 5 and 10, the training basically converged, and the performance of BSTC reaches its best.



**Figure 6.** Changes in accuracy with different numbers of epochs.

#### 4.4. Experimental Results

To reflect the advantages of BSTC, we compared it with the following baseline models:

- NB [15] and K-NN [17] represent the baseline classifiers used in earlier research.

- SAGE [36] is a generative Bayesian technique that combines the topic model and the generalized additive model.
- CNN [20] uses a pre-trained CBOW model, which includes 100 pre-trained word embeddings.
- The SCNN [19] model is made up of two convolution layers. The synthesis of each sentence through a fixed-length window is known as sentence convolution. The sentence vector is transformed into a document vector by using document convolution.
- SWNN [19] is an improved document representation model based on a CNN. SWNN learns the matching text representation and weight vector from the sentence and document levels, respectively, and combines them to generate a document representation vector, which is then employed to classify fake reviews.
- ST-MFLC [37] captures local, temporal, and weighted semantic information from reviews by using three different models. They are then combined to generate the final representation of the document.
- DFFNN [21] is a multilayer perceptron neural network that can deal with sophisticated sparse text representations. DFFNN learns document-level representations by utilizing n-grams, word embeddings, and three types of lexicon-based sentiment indicators.
- DSRHA [26] is a two-level hierarchical attention architecture used to detect fake reviews.
- EKI-SM [38] incorporates the TF-IDF algorithm. The n-gram model is used to capture high-dimensional sparse characteristics and sentiment features from reviews, and neural networks are used to classify the reviews.
- The BERT [10] model used in this paper is BERT<sub>LARGE</sub>, which contains 24 encoder layers, and the hidden dimension is 1024. By adding an additional output layer, BERT is employed for fake review detection.

Table 3 displays the experimental outcomes. We can see that using BERT alone allowed excellent results to be achieved, and it was very competitive compared with the other baseline models, demonstrating the benefits of the pre-trained language model in learning review semantics. Compared with BERT, BSTC significantly improved in all metrics on the three datasets, all of which increased by more than 1%. It should be noted that BERT is already a strong baseline. Simultaneously, BSTC achieved the best results on the three datasets, especially on the Hotel dataset, where the Acc, F1, P, and R reached 93.44%, 93.36%, 90.64%, and 96.88%, respectively, which were significantly higher than those of all of the baseline models. This unequivocally reflects that BSTC could detect fake reviews more effectively and accurately.

**Table 3.** Results of the comparative experiments with the baseline models.

Model	Hotel Dataset				Restaurant Dataset				Doctor Dataset			
	Acc	F1	P	R	Acc	F1	P	R	Acc	F1	P	R
K-NN [17]	71.38	67.80	–	–	72.14	69.20	–	–	71.13	78.60	–	–
NB [15]	81.25	81.70	–	–	80.58	81.30	–	–	81.02	85.30	–	–
SAGE [36]	81.80	82.60	81.20	84.00	81.70	82.80	84.20	81.60	74.50	73.50	77.20	70.10
SWNN [19]	–	83.70	84.10	83.30	–	87.60	87.00	88.20	–	82.90	85.00	81.00
CNN [20]	84.88	85.00	–	–	79.61	80.30	–	–	77.96	83.90	–	–
SCNN [19]	86.44	86.30	–	–	89.30	89.80	–	–	87.81	90.60	–	–
ST-MFLC [37]	88.00	88.00	88.10	88.00	85.00	85.00	85.30	85.00	90.30	90.20	90.30	90.30
DFFNN [21]	89.56	89.60	–	–	88.31	88.40	–	–	86.21	89.30	–	–
DSRHA [26]	–	–	–	–	77.50	80.90	<b>90.50</b>	73.10	91.00	92.80	97.00	88.90
EKI-SM [38]	90.75	90.72	–	–	–	–	–	–	–	–	–	–
BERT	90.94	91.29	87.86	95.00	88.75	89.16	86.05	92.50	88.39	90.91	91.55	90.28
BSTC	<b>93.44</b>	<b>93.36</b>	<b>90.64</b>	<b>96.88</b>	<b>91.25</b>	<b>91.57</b>	88.37	<b>95.00</b>	<b>92.86</b>	<b>94.29</b>	<b>97.06</b>	<b>91.67</b>

The best results are shown in bold.

#### 4.5. Ablation Study

We performed ablation studies on BSTC to further assess the degree of contribution of each portion of the model to the overall model's performance and the importance of each part. Table 4 displays the outcomes.

**Table 4.** The results of the ablation study.

Model	Hotel Dataset				Restaurant Dataset				Doctor Dataset			
	Acc	F1	P	R	Acc	F1	P	R	Acc	F1	P	R
BSTC	<b>93.44</b>	<b>93.36</b>	<b>90.64</b>	<b>96.88</b>	<b>91.25</b>	<b>91.57</b>	<b>88.37</b>	<b>95.00</b>	<b>92.86</b>	<b>94.29</b>	<b>97.06</b>	91.67
w/o BERT	92.50	92.77	89.53	96.25	<b>91.25</b>	<b>91.57</b>	<b>88.37</b>	<b>95.00</b>	91.07	93.15	91.89	<b>94.44</b>
w/o SKEP	91.87	92.17	88.95	95.63	90.00	90.24	88.10	92.50	89.29	91.43	94.12	88.89
w/o BERT+TextCNN	91.87	92.07	89.88	94.37	90.00	90.24	88.10	92.50	90.18	92.41	91.78	93.06
w/o SKEP+TextCNN	90.94	91.29	87.86	95.00	88.75	89.16	86.05	92.50	88.39	90.91	91.55	90.28

The best results are shown in bold.

First, we removed the BERT model (w/o BERT), and we could see that the model had a significant decline in the accuracy and F1-score on the Hotel dataset and the Doctor dataset, reflecting that BERT could improve the model's ability to capture review features. Then, we removed the SKEP model (w/o SKEP). At this time, the model had a more significant decline than BSTC in all metrics on three datasets. It was proved that the combination of the advanced pre-trained language sentiment model could better integrate the contextual sentiment information of the reviews, thus improving the model's performance. Finally, we used the BERT model (w/o SKEP + TextCNN) and the SKEP model (w/o BERT + TextCNN) to conduct separate experiments. We observed that the performance of these two models declined with respect to the previous basis, which fully proved that the incorporation of TextCNN could help the model further extract the local features and key information of reviews, thereby improving the model's classification ability.

## 5. Conclusions

In this study, we proposed a new fake review detection model called BSTC, which was based on a pre-trained language model and a convolutional neural network and used BERT, SKEP, and TextCNN. To validate BSTC's effectiveness, we compare it with different baseline models on three benchmark datasets. In terms of overall performance, our model surpassed all other models and had the highest accuracy across all three datasets. It was demonstrated that our model could more comprehensively extract the contextual, semantic, and sentiment information of the reviews, resulting in excellent detection results.

We considered the effect of sentiment knowledge on the detection of fake reviews and, thus, introduced SKEP to boost the model's performance. The experimental outcomes demonstrated that BSTC achieved excellent results in fake review detection, but the factors taken into account were still not comprehensive enough. If this is combined with other pre-trained language models that take more types of knowledge into account in the pre-training stage, it may be possible to strengthen the model's performance in detecting fake reviews. In future work, we will continue our research based on this idea.

**Author Contributions:** Conceptualization, X.Z. (Xintao Zhan) and J.L.; Methodology, X.Z. (Xintao Zhan); Writing—original draft, X.Z. (Xintao Zhan); Writing—review and editing, J.L. and X.Z. (Xintao Zhan); Supervision, J.L., X.Z. (Xintao Zhan), G.L., X.Z. (Xinrong Zhan) and X.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the 2022 Central Government Guided Local Development Science and Technology Special Project (2022L3029) and the Fujian Provincial Department of Industry and Information Technology (202319).

**Data Availability Statement:** The source code and the datasets used in the experiments are available at <https://github.com/byDream99/BSTC-Fake-Review-Detection> (accessed on 16 March 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jindal, N.; Liu, B. Analyzing and Detecting Review Spam. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 547–552. [\[CrossRef\]](#)
2. Jindal, N.; Liu, B. Review Spam Detection. In Proceedings of the 16th International Conference on World Wide Web, WWW'07, Banff, AB, Canada, 8–12 May 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 1189–1190. [\[CrossRef\]](#)
3. Jindal, N.; Liu, B. Opinion Spam and Analysis. In Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM'08, Palo Alto, CA, USA, 11–12 February 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 219–230. [\[CrossRef\]](#)
4. Ott, M.; Cardie, C.; Hancock, J. Estimating the Prevalence of Deception in Online Review Communities. In Proceedings of the 21st International Conference on World Wide Web, WWW'12, Lyon, France, 16–20 April 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 201–210. [\[CrossRef\]](#)
5. Ullrich, S.; Brunner, C.B. Negative online consumer reviews: Effects of different responses. *J. Prod. Brand Manag.* **2015**, *24*, 66–77. [\[CrossRef\]](#)
6. Zhang, Y.; Jin, R.; Zhou, Z.H. Understanding bag-of-words model: A statistical framework. *Int. J. Mach. Learn. Cybern.* **2010**, *1*, 43–52. [\[CrossRef\]](#)
7. Sudhakaran, P.; Hariharan, S.; Lu, J. A framework investigating the online user reviews to measure the biasness for sentiment analysis. *Asian J. Inf. Technol.* **2016**, *15*, 1890–1898.
8. Wu, Y.; Ngai, E.W.; Wu, P.; Wu, C. Fake online reviews: Literature review, synthesis, and directions for future research. *Decis. Support Syst.* **2020**, *132*, 113280. [\[CrossRef\]](#)
9. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [\[CrossRef\]](#)
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
11. Tian, H.; Gao, C.; Xiao, X.; Liu, H.; He, B.; Wu, H.; Wang, H.; Wu, F. SKEP: Sentiment Knowledge Enhanced Pre-training for Sentiment Analysis. *arXiv* **2020**, arXiv:2005.05635.
12. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014. [\[CrossRef\]](#)
13. Rubin, T.N.; Chambers, A.; Smyth, P.; Steyvers, M. Statistical topic models for multi-label document classification. *Mach. Learn.* **2012**, *88*, 157–208. [\[CrossRef\]](#)
14. Wu, F.; Huberman, B.A. Opinion Formation under Costly Expression. *ACM Trans. Intell. Syst. Technol.* **2010**, *1*, 5. [\[CrossRef\]](#)
15. Li, F.; Huang, M.; Yang, Y.; Zhu, X. Learning to Identify Review Spam. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence—Volume Volume Three, IJCAI'11, Barcelona, Spain, 16–22 July 2011; AAAI Press: Menlo Park, CA, USA, 2011; pp. 2488–2493.
16. Feng, S.; Banerjee, R.; Choi, Y. Syntactic Stylometry for Deception Detection. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, ACL'12, Jeju Island, Republic of Korea, 8–14 July 2012; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; Volume 2, pp. 171–175.
17. Elmurngi, E.; Gherbi, A. An empirical study on detecting fake reviews using machine learning techniques. In Proceedings of the 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Luton, UK, 16–18 August 2017; pp. 107–114. [\[CrossRef\]](#)
18. Harris, C.G. Detecting Deceptive Opinion Spam Using Human Computation. In Proceedings of the AAAI Workshop on Human Computation, Virtual, 6–10 November 2012.
19. Li, L.; Qin, B.; Ren, W.; Liu, T. Document representation and feature combination for deceptive spam review detection. *Neurocomputing* **2017**, *254*, 33–41. [\[CrossRef\]](#)
20. Ren, Y.; Ji, D. Neural networks for deceptive opinion spam detection: An empirical study. *Inf. Sci.* **2017**, *385–386*, 213–224. [\[CrossRef\]](#)
21. Hajek, P.; Barushka, A.; Munk, M. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Comput. Appl.* **2020**, *32*, 17259–17274. [\[CrossRef\]](#)
22. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29. [\[CrossRef\]](#)
23. Severyn, A.; Moschitti, A. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015. [\[CrossRef\]](#)
24. Nguyen, T.; Shirai, K. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2509–2514. [\[CrossRef\]](#)
25. Shahi, T.; Sitaula, C.; Paudel, N. A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 5681574. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Liu, Y.; Wang, L.; Shi, T.; Li, J. Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and Bi-LSTM. *Inf. Syst.* **2022**, *103*, 101865. [\[CrossRef\]](#)

27. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014. [[CrossRef](#)]
28. Mikolov, T.; Corrado, G.; Chen, K.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
29. Zhu, Y.; Zheng, W.; Tang, H. Interactive Dual Attention Network for Text Sentiment Classification. *Comput. Intell. Neurosci.* **2020**, *2020*, 8858717. [[CrossRef](#)] [[PubMed](#)]
30. Li, X.; Bing, L.; Zhang, W.; Lam, W. Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. *arXiv* **2019**, arXiv:1910.00883.
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
32. Turney, P.D. Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL'02, Philadelphia, PA, USA, 6–12 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 417–424. [[CrossRef](#)]
33. Christlein, V.; Spranger, L.; Seuret, M.; Nicolaou, A.; Král, P.; Maier, A. Deep Generalized Max Pooling. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1090–1096. [[CrossRef](#)]
34. Ott, M.; Choi, Y.; Cardie, C.; Hancock, J.T. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT'11, Portland, OR, USA, 19–24 June 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; Volume 1, pp. 309–319.
35. Ott, M.; Cardie, C.; Hancock, J.T. Negative deceptive opinion spam. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 497–501.
36. Li, J.; Ott, M.; Cardie, C.; Hovy, E. Towards a General Rule for Identifying Deceptive Opinion Spam. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–24 June 2014; Volume 1, pp. 1566–1576. [[CrossRef](#)]
37. Cao, N.; Ji, S.; Chiu, D.K.; Gong, M. A deceptive reviews detection model: Separated training of multi-feature learning and classification. *Expert Syst. Appl.* **2022**, *187*, 115977. [[CrossRef](#)]
38. Han, S.; Wang, H.; Li, W.; Zhang, H.; Zhuang, L. Explainable knowledge integrated sequence model for detecting fake online reviews. *Appl. Intell.* **2023**, *53*, 6953–6965. [[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.