Journal on
Artificial Intelligence

Tech Science Press

DOI: 10.32604/jai.2024.049685

ARTICLE

Leveraging Pre-Trained Word Embedding Models for Fake Review Identification

Glody Muka[1,*] **and Patrick Mukala**[1,2,*]

[1]Department of Mathematics and Computer Science, National Pedagogical University, Kinshasa, P.O. Box 8815, Democratic Republic of Congo

[2]School of Computer Science, University of Wollongong in Dubai, Dubai, P.O. Box 20183, United Arab Emirates

*Corresponding Authors: Glody Muka. Email: glody.muka@aims-cameroon.org; Patrick Mukala. Email: patrickmukala@uowdubai.ac.ae

Received: 24 January 2024 Accepted: 08 July 2024 Published: 07 August 2024

ABSTRACT

Reviews have a significant impact on online businesses. Nowadays, online consumers rely heavily on other people's reviews before purchasing a product, instead of looking at the product description. With the emergence of technology, malicious online actors are using techniques such as Natural Language Processing (NLP) and others to generate a large number of fake reviews to destroy their competitors' markets. To remedy this situation, several researches have been conducted in the last few years. Most of them have applied NLP techniques to preprocess the text before building Machine Learning (ML) or Deep Learning (DL) models to detect and filter these fake reviews. However, with the same NLP techniques, machine-generated fake reviews are increasing exponentially. This work explores a powerful text representation technique called *Embedding models* to combat the proliferation of fake reviews in online marketplaces. Indeed, these embedding structures can capture much more information from the data compared to other standard text representations. To do this, we tested our hypothesis in two different Recurrent Neural Network (RNN) architectures, namely Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), using fake review data from Amazon and TripAdvisor. Our experimental results show that our best-proposed model can distinguish between real and fake reviews with 91.44% accuracy. Furthermore, our results corroborate with the state-of-the-art research in this area and demonstrate some improvements over other approaches. Therefore, proper text representation improves the accuracy of fake review detection.

KEYWORDS

Natural language processing; word embedding; deep learning; fake review detection

1 Introduction

For e-commerce sites, social media platforms, or any other online service, user reviews are very important [1]. Consumers rely heavily on the reviews of others to know whether the product or service is good or not before making their choice. These reviews also help online marketplaces to get feedback on their services from their consumers or users, and adjust their services. Years ago, there has been a proliferation of online reviews, and consumers' decisions are strongly influenced by them [1,2]. In the United States (US) for example, 80% of online consumers look at reviews before consuming a service or purchasing a product [3]. Additionally, 70% of online consumers consult reviews before making their final decision about a product, and 63% of them are more likely to finalize their decision only when a targeted product has many positive reviews and higher ratings [4].

However, according to the estimations of Zhang et al., about 20%–25% of online reviews on almost all online services are fake [5]. In 2020, Wu et al. estimated that the proportion of fake online reviews was between 16% to 20% and 25% to 33.3% [6]. We then witness the proliferation of "fake reviews". Indeed, a fake review is written or generated without having a real experience of the targeted product or service [7]. A fake review can be generated in two ways [8]. First, in a human-written way: by hiring professional spammers to write fake reviews (positive or negative) that look real and influence consumers of online products. Second, in a computer-generated way: by using machines and other techniques such as language models to generate rapidly a large number of reviews.

Since reviews seem to be so important to online marketplaces, fraudulent actors may try to attract consumers to their products by hiring spammers to post positive reviews on their platforms and negative reviews on their competitors' [8]. In 2013, for example, Samsung was forced by the Taiwan Federal Trade Commission to pay a total of 340,000 USD in damages for posting negative fake reviews to damage the image of its rival HTC [6]. On the other hand, with the emergence of Artificial Intelligence (AI) techniques such as NLP, using language modeling tools to generate fake reviews could prove fatal. For instance, in 2021, several online shopping giants claimed to have removed millions of allegedly fake reviews from their platforms. Amazon, which makes up 49% of online business in all US, removed more than 160 million and 200 million suspected fake reviews in 2019 and 2020, respectively [9]. Yelp indicated in its annual report that it had removed about 8% of suspected fake reviews, while TrustPilot removed 5.7% of reviews (approximately 2.2 million) [10].

From the above, we can understand that the war on detecting fake reviews is directed against malicious actors and machines that generate fake reviews. The emergence of NLP products such as ChatGPT-4 challenges the ability of humans to distinguish between human-generated text and artificial intelligence-generated text. One of the best remaining options would be to use machines to fight other machines. In recent studies, several NLP techniques such as BERT (Bidirectional Encoder Representations from Transformers) and artificial neural network models are used to achieve more accurate results in the text generation task [8,11]. NLP, an emerging field of AI, has several applications such as speech recognition, machine translation, etc. Each of these tasks involves specific data preprocessing steps. Text preprocessing is therefore an important part of any NLP project. Indeed, characters, words, phrases, or paragraphs noticed at this point are the fundamental units passed on to all subsequent preprocessing steps, from analysis and tagging components, such as morphological analyzers and part-of-speech recognition, to emerging applications, such as text generation, spoken dialogue systems, etc. [12]. Text data must be transformed into a numerical format to train an ML model. Several numerical representations of text have been proposed, but not all of them can capture the syntactic and semantic aspects of text. There are text representation structures called "embeddings" that can represent the numerical format of a text according to its context [12]. In this fight against the increase of fake reviews, the text representation with its full meaning based on the context can have a significant impact on the extraction of patterns that distinguish real from fake reviews. This paper leverages the performance of Word embedding models namely: Word2vec, GloVe, and FastText. To assess these performances, the second stage is to train neural network models such as LSTM and GRU, using fake review data from Amazon and TripAdvisor platforms.

The continued development of Generative AI tools challenges the effectiveness of existing techniques for fake review identification. According to the literature, there is no specific characteristic for distinguishing real from fake reviews generated by both humans and machines. Therefore, the objective of this work is to improve the model accuracy for fake review identification by utilizing powerful embedding models and comparing their performance on human and machine-generated review spam.

We address the proliferation of fake reviews through the following research questions:

a) How effective are embedding techniques for distinguishing real from fake reviews?
b) How well can a machine detect text generated by a human and another machine?

Here is the summary of the contribution of this work:

1. The proposed approach enhanced the effectiveness of three advanced pre-trained word embedding techniques on both human and machine fake review datasets.
2. The methodology followed by this work can serve as a baseline for solving further problems related to fake review identification and filtering in online marketplaces.

The rest of this paper is organized as follows: Section 2 summarizes some relevant studies related to word embedding and ML/DL models for fake reviews. Section 3 describes some concepts of the general methodology followed by this research. Our experimental results are presented in Section 4. Finally, Section 5 summarizes our findings and gives some perspectives.

2 Related Work

In this section, we explore the literature on the impact of fake reviews in online marketplaces, as well as techniques used to detect and combat this unfortunate phenomenon, including word embedding and ML/DL models.

A study by CHEQ and the University of Baltimore in 2021 indicates that the online economy of the G20 countries is estimated to be worth 4.2 trillion USD, which represents 5.3% of their total GDP. And 89% of that income is influenced by online reviews. The bad news in this report is the loss caused by fake online reviews, which is estimated at 4% [10]. Since this phenomenon seems to be so prolific for fraudulent online actors, we should expect it to grow as e-commerce evolves [13]. In recent years, papers have been published proposing tools and techniques to detect fake reviews. A survey by Mohawesh et al. [11] summarizes the studies conducted on existing approaches and available datasets between 2007 and 2021. Experiments performed on two benchmark datasets showed that RoBERTa (Robustly Optimized BERT Pretraining Approach, an improved version of BERT) outperformed state-of-the-art methods. Elmogy et al. [14] used several supervised ML techniques to identify fake reviews. Several classifiers were used and K-Nearest-Neighbor (KNN) performed better than the others in terms of F1-score. Alsubari et al. [15] used also several supervised ML models such as Adaptive Boosting (AB), Support Vector Machine (SVM), Naïve Bayes (NB), and Random Forest (RF), and achieved remarkable results. Salminen et al. [8] used two language models, namely Universal Language Model Fine-Tuning (ULMFiT) and Generative Pre-trained Transformer (GPT-2), to generate fake reviews from the Amazon review data. The fake review dataset created with GPT-2 was assessed using an NB version of the SVM model (NBSVM) and a fine-tuned version of RoBERTa (fakeRoBERTa). A binary classification model showed the outperformance of fakeRoBERTa over NBSVM.

Fake reviews are not only in English. Other rich languages such as Hindi can provide insights for text classification and fake reviews identification. In 2021, Sharma et al. [16] constructed an Indian reviews dataset for the Hindi language using Devanagari lipi. The authors used a scrapping tool called

Parsehub to collect Indian Hindi news and the performance of their proposed model was satisfactory. In the same logic, Pathak et al. [17] worked on an Aspect-Based Sentiment Analysis (ABSA) task, by leveraging ensemble models based on multilingual BERT for the Hindi language. For different domains such as Electronics, Mobile Apps, Travel, and Movies, this paper achieved encouraging results. Majumber et al. [18] used a pre-trained embedding model at the sentence level called Universal Sentence Encoder (USE) for classifying fake news broadcasters on Twitter in both English and Spanish languages.

Concerning leveraging word embedding models, Barushka et al. [19] sought to extract the linguistic context of words to detect fake reviews using Bag of Words (BoW) and Skip-Gram models in their neural network algorithms on two hotel review datasets. Hajek et al. [20] worked in the same direction to improve the model's performance for review spam detection using BoW, *n*-gram models, and sentiment mining techniques. Shi et al. [21] proposed a joint sentiment topic model combining word embeddings (JSTE), which is a combination of techniques for extracting sentiment and topics from the review text. Using LSTM as a classifier, the JSTE approach improved the accuracy of the fake reviews identification.

More recently, Cai et al. [22] used Bi-LSTM on Yelp labeled datasets to extract massive aspect words, which were clustered into different aspect categories by the K-means algorithm. The proposed approach achieved good results. In 2024, Dasgupta et al. [23] achieved remarkable results by leveraging MiniLM BERT and Word2vec on the Yelp fake review dataset.

Inspired by previous works, we intend to experiment with the effectiveness of some powerful techniques to combat the proliferation of fake reviews. However, as fake reviews can be generated by humans and machines, this work focused on using various embedding models on English datasets with both human and machine fake reviews. Therefore, compared to other approaches and to further enhance the performance of review spam identification, our work differs in two main aspects: (1) Three different pre-trained word embedding models are used and the experiments were conducted on two different datasets to investigate their performance. (2) We assessed the performance of these models with two RNN architectures namely LSTM and GRU, with corresponding best hyperparameters.

3 Methodology

This section describes some techniques of our general workflow for detecting fake reviews depicted in Fig. 1.

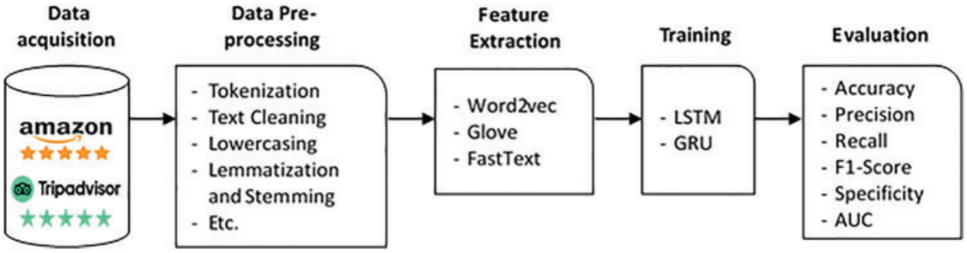

Figure 1: Workflow of the proposed approach

3.1 Data Preprocessing

Text preprocessing converts a raw textual data file, usually represented as a series of digital bits, into a sequence of linguistically meaningful components [12]. The process can sometimes be time-consuming, but it is critical to the performance of the model. Here are some key operations performed on our datasets: (1) Tokenization, which splits text into individual tokens (words or characters); (2) Expand contractions to reduce the number of the unique tokens; (3) Text Cleaning by removing special characters such as punctuations, emojis, numbers, and any other non-alphabetical token; (4) Lowercasing; (5) Removing stopwords, as they appear with high frequency in the text but do not carry much meaning (e.g., and, the, a, ...); (6) Lemmatization and Stemming for reducing words to their roots; and (7) Removing low- and high-frequency words.

3.2 Background of Word Embeddings

Words with similar contexts tend to have similar meanings [24]. Therefore, the numerical representation of a word in terms of its context is very important to capture the meaning of the text so that ML/DL algorithms can learn better. Word embedding models can represent individual words as real-valued dense vectors and capture inter-word semantics. Unlike BoW [19] or TD-IDF [17] methods which are sparse vectors with very high dimensions (the size of the vocabulary), word embedding models fit more data into a much lower dimensional space. There are two ways to obtain word embedding: (1) Learning word embeddings from scratch; and (2) Leveraging pre-trained word embeddings [25]. The latter are trained from real-world data and can better capture the syntactic and semantic similarities between words. Here are the pre-trained word embedding models used in this research:

3.2.1 Word2vec

Proposed by Mikolov et al. [26], Word2vec is a method that utilizes simple neural networks to produce word embeddings. Word2vec has two variants: (1) Continuous Bag-of-Words (CBOW), which predicts the center word based on its context. To predict a word from its surrounding context, the CBOW model maximizes the following probability:

$$P\left(w_t \middle| w_{t-c}, w_{t-c+1}, \ldots, w_{t-1}, w_{t+1,}, \ldots, w_{t+c-1}, w_{t+c}\right), \tag{1}$$

where w_t is the center word at position t and c is the window size (maximum context location at which word must be predicted, i.e., for $c = 2$, the word w_t is predicted at context locations $t - 2$, $t - 1$, $t + 1$, and $t + 2$). (2) Skim-Gram, a variant that predicts the context words given a center word. The model tries to maximize the following probability:

$$P\left(w_{t-c}, w_{t-c+1}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c-1}, w_{t+c} \middle| w_t\right). \tag{2}$$

3.2.2 Global Vectors for Word Representation (GloVe)

An improved version of Word2vec proposed by Pennington et al. [27], GloVe is a new global log-bilinear regression model for unsupervised learning of word representations. The model based on the Skip-Gram variant of Word2vec, not only considers the local context of the target word but also uses the statistical approach, that records the frequency at which two words w_i and w_j appear in the same context in the whole corpus, using the global word-word co-occurrence matrix [27]. The word-word co-occurrence matrix is a square matrix of size $|W| \times |W|$, where W is the vocabulary of the corpus. The co-occurrence matrix X being symmetric and sparse, each entry X_{ij} contains the frequency on which both word w_i and word w_j co-occur in a sentence, paragraph, document, or any other fixed-length window. The GloVe method exploits the statistical information by training only on the non-zero elements, rather than on the whole matrix. After obtaining the co-occurrence matrix, the co-occurrence probabilities are calculated as follows:

$$P_{ij} = P\left(j|i\right) = \frac{X_{ij}}{X_i}, \tag{3}$$

where P_{ij} is the probability that word j appears in the context of the word i and $X_i = \sum_k X_{ik}$ the number of times any word appears in the context of word i. Moreover, the GloVe approach learns word vectors based on the ratios of co-occurrence probabilities instead of using probabilities themselves. That ratio depends on three words w_i, w_j, and $w_{k,}$ and can be defined more generally as follows [27]:

$$F\left(v_i, v_j, \acute{\upsilon}_k\right) = \frac{P_{ik}}{P_{jk}}, \tag{4}$$

where v_i, and v_j are corresponding word vectors and $\acute{\upsilon}_k$ are separate context word vectors.

3.2.3 FastText

When considering a standard representation of a single word, the internal structure of that word can be ignored. This information can be meaningful for the representation of rare, misspelled, and out-of-vocabulary (OOV) words, especially in morphologically rich languages like Finnish or Turkish [28]. FastText is a powerful technique that allows the embedding representation of words at the character level. The model is Skip-Gram-based and uses subword information to overcome some limitations of previous models [29]. The subword information approach is a neural network that takes as input, a decomposition of each word w into its n-grams N, and each n-gram n is converted into a vector x_n. The sum of all x_n and the word vector itself represents the enriched word vector representation [28,29], i.e.,

$$v_w + \frac{1}{|N|} \sum\nolimits_{n\epsilon N} x_n. \tag{5}$$

To distinguish prefixes and suffixes from other characters' sequences, special symbols "<" and ">" are added at the start and the end of each word. The word w itself is added in the set of its n-grams to preserve its original representation [30]. For instance, the word "model" for $n = 4$ will be represented by the character n-grams: <mod, mode, odel, del>, <model>.

3.3 Training Algorithm Selection

Spoken and written language are temporal or sequential phenomena. RNN models are used to train such kinds of data. Indeed, RNN is an architecture that contains cycles in its network connections and these connections allow the model to depend on multiple words in the past, unlike classical ML algorithms [24]. However, the simple version of the model has some limitations, especially for large textual data. The model leads to a problem called "exploding and vanishing gradients", due to its difficulty in dealing with long-term dependencies in large sequences. To overcome this problem, two architectures have been proposed namely LSTM and GRU [25]. These architectures use special units that deploy gates to control the flow of the information into and out of components that make up the network layers, i.e., removing information that is no longer important from the context and retaining information that is likely needed for the next step. GRU can be faster and give good results even with small training data because of its reduced number of parameters. LSTM, on the other hand, has good default parameters that allow it to perform well even with large data. However, there is no evidence of the effectiveness of one over the other. The performance might differ depending on the data and the task at hand. Therefore, we will experiment with both for fake review detection in this work.

3.4 Performance Evaluation Measures

Analogous to [22], we produce a confusion matrix to extract other measures that quantify the model's performance. *Accuracy* (Acc) is usually the first and most used, although it is not always wise to rely on it alone, especially when dealing with a binary classification task and imbalanced data [24]. Therefore, we used *Precision* (Pr), representing the proportion of correctly predicted fake reviews out of all fake reviews predicted by the model, and *Recall* (Re), the proportion of correctly predicted fake reviews out of all actual fake reviews instances in the dataset. Another measure is the *F1-score* (F1), defined as the Precision and Recall's harmonic mean. Additionally, the proportion of real (truthful) reviews correctly identified by the model out of all actual real instances is called *Specificity* (Sp). Finally, the *Area Under the Curve* (AUC) measures the probability that the model will rank a randomly selected fake review instance higher than a randomly selected real review instance [20].

4 Experiment Setup and Results

According to [8], humans are limited in detecting fake comments as they seem realistic and are very abundant on online platforms. Machines are therefore a suitable solution for this purpose. This section presents experiments conducted and performance analysis of our suggested models.

4.1 Data Description

In this paper, we used the following datasets, regardless of the sentiment polarity, ratings, and other attributes:

- **Amazon dataset:** This dataset was constructed by [8] in 2021, using Amazon review data as real reviews and GPT-2 to generate fake reviews. The dataset is composed of 20, 216 Original Reviews (OR) and 20, 216 Computer-Generated Reviews (CG).
- **TripAdvisor dataset:** also known as the gold-standard dataset and widely used in the literature [19–21], the dataset was constructed by [31] in 2013, based on reviews from 20 hotels in Chicago city. The data is also equally balanced with 800 truthful reviews from 6 online communities: TripAdvisor, Hotels.com, Expedia, Yelp, Orbitz, and Priceline. The other 800 deceptive reviews were gathered from Mechanical Turk, also called Workers.

After the processing operations performed in the datasets, we explore and understand the data. The distribution of the length of reviews is provided in Fig. 2. We notice that most reviews are shorter than the average review length. This is an important pattern when it comes to deciding the maximum review length to be processed by the model.

Through basic statistics, Table 1 gives some key characteristics of reviews in both datasets.

As shown in Table 1, the minimum review length for the Amazon dataset is zero. This is explained by the fact that some reviews have only one or two words and can therefore be removed during preprocessing (as stopwords, frequent words, etc.).

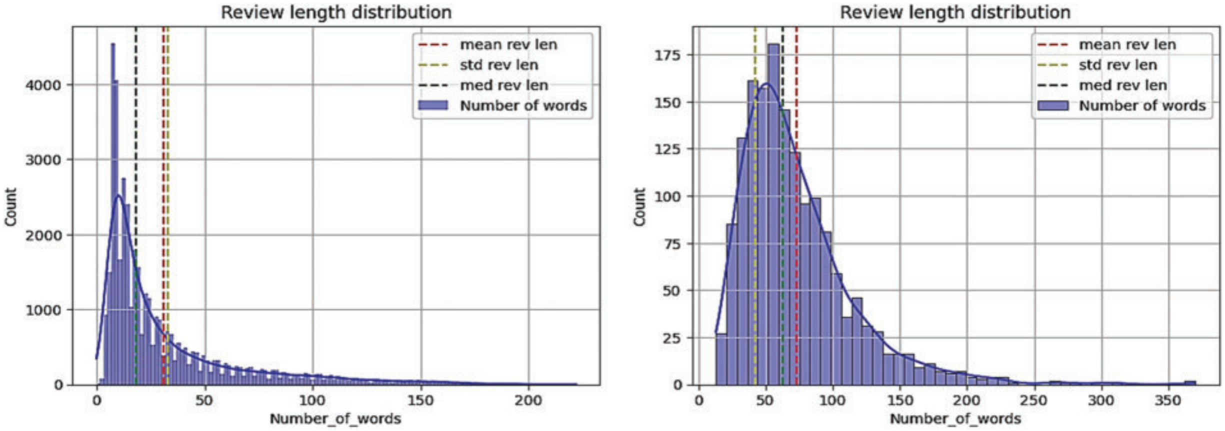

Figure 2: Amazon's (left) and TripAdvisor's (right) review length distribution

Table 1: Summary of basic statistics of the text

	Amazon	TripAdvisor
Number of reviews	40,432	1600
Vocabulary size	39,407	8312
Maximum review length	222	370
Minimum review length	0	13
Mean review length	31.47	72.62
Std review length	32.93	42.01
Median review length	18.00	63.00

4.2 Feature Extraction and Experimental Settings

The maximum review length is 222 and 370 for the Amazon and TripAdvisor datasets, respectively; which is quite long. Fig. 2 leads us to believe that the model can distinguish between real and fake reviews after just a few words, as the review length curves slope to the left. We therefore set the fixed length of all reviews to an optimal number denoted l through a process called *padding*. Indeed, the padding operation will add a special character to reviews whose length is less than l and truncate reviews whose length is greater than l. This is because the RNN model requires all input data to be of fixed size [25]. For all our experiments, we set $l = 100$. After that, we load pre-trained word embeddings (Word2vec, GloVe, and FastText), all of the 300-dimensional space, expecting better embeddings [32]. We then map them to our vocabulary to obtain the corresponding embedding vectors.

In addition to the embedding settings, we defined some hyperparameters for DL models according to each learning process we observed. For instance, we applied some Dropouts ranging from 20% to 50% in the LSTM/GRU layers. The Sigmoid, Adam, and Binary-Cross Entropy were used as the Activation, Optimizer, and Loss functions, respectively. We also defined an Early stopping parameter for the Amazon dataset, to prevent the Overfitting of almost all the models occurring after 20 Epochs. Finally, each dataset is randomly split into training, validation, and testing sets for 70%, 15%, and 15%, respectively.

4.3 *Results and Discussion*

In this paper, we trained six models for each dataset, combining three embeddings and two neural network-based models. We first presented in Figs. 3 and 4, the confusion matrices of the two best combinations of techniques for each dataset. Then, Tables 2 and 3 summarize the results of our experiments for the Amazon and TripAdvisor datasets, respectively.

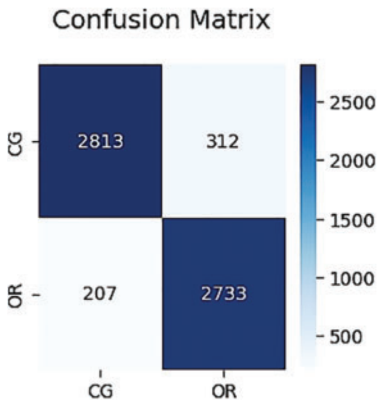

Figure 3: Confusion matrix of FastText-LSTM with Amazon dataset

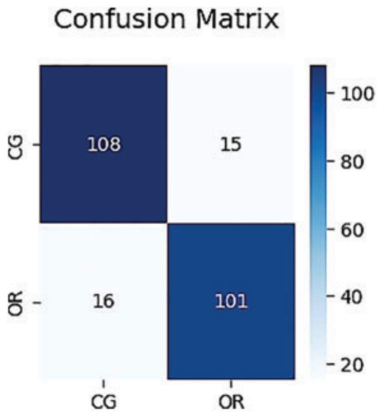

Figure 4: Confusion matrix of FastText-GRU with TripAdvisor dataset

Table 2: Performance of Amazon dataset (in %)

	Acc	Pr	Re	F1	Sp	AUC
Word2vec-LSTM	88.67	89.00	89.00	88.54	90.19	88.67
Word2vec-GRU	87.50	88.00	88.00	87.00	**94.56**	87.53
GloVe-LSTM	90.48	**91.00**	90.00	90.28	92.94	90.49
GloVe-GRU	90.46	90.00	90.00	90.48	90.66	90.47
FastText-LSTM	**91.44**	**91.00**	**91.00**	**91.32**	93.14	**91.44**
FastText-GRU	91.01	**91.00**	**91.00**	90.90	92.58	91.02

Table 3: Performance of TripAdvisor dataset (in %)

	Acc	Pr	Re	F1	Sp	AUC
Word2vec-LSTM	68.33	68.00	68.00	68.33	66.12	68.40
Word2vec-GRU	63.74	64.00	64.00	62.97	63.70	63.75
GloVe-LSTM	85.83	86.00	86.00	86.00	**91.93**	85.62
GloVe-GRU	77.49	78.00	78.00	77.31	75.80	77.55
FastText-LSTM	77.91	80.00	78.00	78.00	65.32	78.35
FastText-GRU	**87.08**	**87.00**	**87.00**	**86.69**	87.09	**87.08**

The results in Table 2 show the outperformance of FastText over Word2vec (Skip-Gram-based) and GloVe, with 91.44% accuracy for the Amazon dataset. Consistently, FastText scores better on all evaluation measures, regardless of algorithm, except for specificity, which is highest for Word2vec. Likewise, for the TripAdvisor dataset, FastText with GRU successfully distinguished real from fake reviews at 87.08% accuracy (Table 3). Additionally, for the Amazon dataset, this paper achieves remarkable results of 91.00%, 91.00%, 91.32%, 94.56%, and 91.44% in terms of precision, recall, F1-score, specificity, and AUC, respectively. It also achieved encouraging results of 87.00%, 87.00%, 86.69%, 91.93%, and 87.08% for precision, recall, F1-score, specificity, and AUC, respectively, on the TripAdvisor dataset. Note that, precision and recall performance are almost identical, as the datasets are well-balanced. However, the F1-score may differ slightly due to some rounding after averages.

Based on our experiments, LSTM achieved mostly better results with the Amazon dataset, whereas GRU gave higher results with the TripAdvisor dataset. This confirms that LSTM is suitable for larger datasets, while GRU can perform well with small datasets. On the other hand, the overall performance of the Amazon dataset is superior to that of the TripAdvisor dataset. This can be explained by the fact that the TripAdvisor dataset contains a small amount of data. In addition, the fake reviews detected in the TripAdvisor dataset are generated by humans. They are therefore more realistic and harder to identify by the model. In contrast, CG reviews from the Amazon dataset have some inconsistencies, which facilitate the model to learn relevant patterns and perform better.

4.4 Comparative Analysis

Before conducting experiments for fake review detection, it was important to know datasets and methodologies used in related previous works. Although this paper achieved remarkable results in terms of other evaluation metrics, Table 4 compares our results with those of some recent papers in terms of accuracy.

Table 4: Comparative analysis with existing works

Paper	Datasets	Feature extraction	Algorithms	Results (%)
Narayan et al. [33]	Hotel reviews from Amazon	TF-IDF	Logistic regression	86.00
Elmogy et al. [14]	Yelp	TF-IDF	Logistic regression SVM	86.89 86.90

(Continued)

Table 4 (continued)

Paper	Datasets	Feature extraction	Algorithms	Results (%)
Hajek et al. [20]	Amazon, mechanical turk	CBOW	Deep feed-forward NN (DFFNN)	89.56
			Convolutional NN	88.35
Mohawesh et al. [11]	Yelp and deceptive	GloVe	RoBERTa	91.02
			BERT	86.20
Shi et al. [21]	TripAdvisor, Amazon, and yelp	Word2vec	LSTM	91.05
			SVM	90.40
Dasgupta et al. [23]	Amazon	MiniLM BERT and Word2vec	LSTM	91.00
This study	Amazon and TripAdvisor	Word2vec, Glove, and FastText	GRU	**87.08**
			LSTM	**91.44**

5 Conclusion, Limitations, and Future Works

This paper presented the rapid increase of fake reviews, its impact on online marketplaces, and some techniques to combat this unfortunate phenomenon. Furthermore, we leveraged the performance of three pre-trained word embeddings, combined with two RNN-based models to classify original from fake reviews. To check the effectiveness of these models for fake review identification, we used Amazon and TripAdvisor datasets. The results showed that FastText is the suitable embedding model, as it managed to capture subword information and generalize better even with OOV words in the reviews. These results demonstrate its effectiveness for detecting fake reviews, especially when applied to larger datasets. Additionally, LSTM and GRU can be used interchangeably, depending on the specific task at hand.

Although this paper achieved some degree of success, there are still some limitations. Indeed, the generalization of our proposed approach can be challenged by multilingual datasets. Spammers can use machine translation techniques to thwart our efforts. As language evolves, so do writing and machine-generated text systems. It is therefore necessary to continuously update the proposed model with different datasets to maintain its effectiveness. In addition, other attributes of the datasets such as rating, sentiment polarity, etc., have not been taken into account, even though they can provide important insights when it comes to checking the consistency of the review.

For future works, our proposed methodology can be applied to fake reviews generated in other languages such as Hindi, French, etc., to enhance its effectiveness. Additionally, we can consider other attributes of datasets to extract further insights into reviews such as inconsistency, emotion, etc.

Acknowledgement: This work is supported by the African Institute for Mathematical Sciences (AIMS) Cameroon. Sincere gratitude to Martyn Parker from the University of Warwick for supervising the thesis version of this work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The study conceptualization: Glody Muka and Patrick Mukala; Methodology: Glody Muka; Data gathering: Patrick Mukala; Implementation and interpretation: Patrick Mukala and Glody Muka; Original draft writing: Glody Muka; Review and edition: Patrick Mukala. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The fake Review Dataset can be found at https://osf.io/tyue9/ and the TripAdvisor Dataset is available at http://myleott.com/op-spam.html (assessed on 22 November 2023).

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[2] S. Fernandes, R. Panda, V. G. Venkatesh, B. N. Swar, and Y. Shi, "Measuring the impact of online reviews on consumer purchase decisions—A scale development study," *J. Retail. Consum. Serv.*, vol. 68, no. 5, p. 103066, 2022. doi: 10.1016/j.jretconser.2022.103066.

[3] A. Smith and M. Anderson, "Online shopping and e-commerce," USA: Pew Research Center, 2016.

[4] A. K. Tang, "A systematic literature review and analysis on mobile apps in m-commerce: Implications for future research," *Electron. Commer. Res. Appl.*, vol. 37, pp. 100885, 2019. doi: 10.1016/j.elerap.2019.100885.

[5] D. Zhang, L. Zhou, J. L. Kehoe, and I. Y. Kilic, "What online reviewer behaviors really matter? effects of verbal and nonverbal behaviors on detection of fake online reviews," *J. Manag. Inf. Syst.*, vol. 33, no. 14, pp. 456–481, 2016. doi: 10.1080/07421222.2016.1205907.

[6] Y. Wu, E. Ngai, P. Wu, and C. Wu, "Fake online reviews: Literature review, synthesis, and directions for future research," *Decis. Support Syst.*, vol. 132, no. 12, pp. 113280, 2020. doi: 10.1016/j.dss.2020.113280.

[7] K. D. Lee, K. Han, and S. H. Myaeng, "Capturing word choice patterns with LDA for fake review detection in sentiment analysis," in *Proc. 6th Int. Conf. Web Intell. Min. Semant.*, no. 6, pp. 1–7, Jun. 2016. doi: 10.1145/2912845.

[8] J. Salminen, C. Kandpal, A. M. Kamel, S. Jung, and B. J. Jansen, "Creating and detecting fake reviews of online products," *J. Retail. Consum. Serv.*, vol. 64, no. 3, pp. 102771, 2022. doi: 10.1016/j.jretconser.2021.102771.

[9] E. Woollacott, "Amazon's fake review problem is getting worse," *Forbes*, vol. 3, 2021.

[10] CHEQ and R. Cavaroz, "The economic cost of bad actors on the internet," University of Baltimore, Techical Report, Baltimore, MD, USA, 2019.

[11] R. Mohawesh *et al.*, "Fake reviews detection: A survey," *IEEE Access*, vol. 9, pp. 65771–65802, 2021. doi: 10.1109/ACCESS.2021.3075573.

[12] N. Indurkhya and F. J. Damerau, "Text preprocessing," in *Handb. Natural Lang. Process.*, 2nd ed, vol. 3, no. 3, pp. 7–10, 2010.

[13] N. Hu, L. Liu, and V. Sambamurthy, "Fraud detection in online consumer reviews," *Decis. Support Syst.*, vol. 50, no. 3, pp. 614–626, 2011. doi: 10.1016/j.dss.2010.08.012.

[14] A. M. Elmogy, U. Tariq, M. Ammar, and A. Ibrahim, "Fake reviews detection using supervised machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, 2021. doi: 10.14569/issn.2156-5570.

[15] S. N. Alsurabi *et al.*, "Data analytics for the identification of fake reviews using supervised learning," *Comput. Mater. Contin.*, vol. 70, no. 2, pp. 3189–3204, 2022. doi: 10.32604/cmc.2022.019625.

[16] D. K. Sharma and S. Garg, "Machine learning methods to identify Hindi fake news within social media," in *12th Int. Conf. Comput. Commun. Network. Technol. (ICCCNT)*, Kharagpur, India, Jul. 2021, pp. 1–6.

[17] A. Pathak, S. Kumar, P. P. Roy, and G. B. Kim, "Aspect-based sentiment analysis in Hindi language by ensembling pre-trained mBERT models," *Electronics*, vol. 10, no. 21, pp. 2641, 2021. doi: 10.3390/electronics10212641.

[18] S. B. Majumber and D. Das, “Detecting fake news spreaders on twitter using universal sentence encoder,” in *11th Conf. Labs Eval. Forum*, Thessaloniki, Greece, Sep. 22–25, 2020.

[19] A. Barushka and P. Hajek, “Review spam detection using word embeddings and deep neural networks,” in *Artif. Intell. Appl. Innova.: 15th IFIP WG 12.5 Int. Conf.*, Hersonissos, Crete, Greece, Springer, May 24–26, 2019, pp. 340–350.

[20] P. Hajek, A. Barushka, and M. Munk, “Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining,” *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17259–17274, 2020. doi: 10.1007/s00521-020-04757-2.

[21] L. Shi, S. Xie, L. Wei, Y. Tao, A. W. Junaid and Y. Gao, “Joint sentiment topic model with word embeddings for fake review detection,” *SSRN Electron. J.*, vol. 8, pp. 1369, 2022. doi: 10.2139/ssrn.4096565.

[22] M. Cai, Y. Du, Y. Tan, and X. Lu, “Aspect-based classification method for review spam detection,” *Multim. Tools Appl.*, vol. 83, no. 7, pp. 20931–20952, 2023. doi: 10.1007/s11042-023-16293-x.

[23] S. Dasgupta and J. Buckley, “A multi-embedding convergence network on Siamese architecture for fake reviews,” arXiv preprint arXiv:2401.05995, 2024.

[24] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. NJ, USA: Prentice Hall PTRUpper Saddle River, 2000.

[25] F. Chollet, “Deep learning for text and sequences,” in *Deep Learning with Python*, USA: Simon & Schuster, 2021, pp. 188–196.

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

[27] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543.

[28] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, “Advances in pre-training distributed word representations,” arXiv preprint arXiv:1712.09405, 2017.

[29] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Trans. Assoc. Computat. Linguist.*, vol. 5, no. 1, pp. 135–146, 2017. doi: 10.1162/tacl_a_00051.

[30] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu and B. Qin, “Learning sentiment-specific word embedding for twitter sentiment classification,” in *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist.*, Baltimore, MD, USA, 2014, pp. 1555–1565.

[31] M. Ott, C. Cardie, and J. T. Hancock, “Negative deceptive opinion spam,” in *Proc. 2013 Con. North Am. chap. Assoc. Comput. Linguist.: Human Lang. Technol.*, Atlanta, Georgia, USA, Jun. 2013, pp. 497–501.

[32] B. O. Deho, A. W. Agangiba, L. F. Aryeh, and A. J. Ansah, “Sentiment analysis with word embedding,” in *2018 IEEE 7th Int. Conf. Adapt. Sci. Technol. (ICAST)*, Accra, Ghana, IEEE, Aug. 2018, pp. 1–4.

[33] R. Narayan, J. K. Rout, and S. K. Jena, “Review spam detection using opinion mining,” in *Progress Intelligent Computing Techniques: Theory, Practice, and Applications*, Singapore: Springer, 2018, pp. 273–279.