sustainability

MDPI

Article

Content Noise Detection Model Using Deep Learning in Web Forums

Jiyoung Woo [1] and Jaeseok Yun [2,*]

[1] Department of Big Data Engineering, Soonchunhyang University, Asan-si 31538, Korea; jywoo@sch.ac.kr
[2] Department of Internet of Things, Soonchunhyang University, Asan-si 31538, Korea
* Correspondence: yun@sch.ac.kr; Tel.: +82-41-530-1447

Received: 8 May 2020; Accepted: 18 June 2020; Published: 22 June 2020

Abstract: Spam posts in web forum discussions cause user inconvenience and lower the value of the web forum as an open source of user opinion. In this regard, as the importance of a web post is evaluated in terms of the number of involved authors, noise distorts the analysis results by adding unnecessary data to the opinion analysis. Here, in this work, an automatic detection model for spam posts in web forums using both conventional machine learning and deep learning is proposed. To automatically differentiate between normal posts and spam, evaluators were asked to recognize spam posts in advance. To construct the machine learning-based model, text features from posted content using text mining techniques from the perspective of linguistics were extracted, and supervised learning was performed to distinguish content noise from normal posts. For the deep learning model, raw text including and excluding special characters was utilized. A comparison analysis on deep neural networks using the two different recurrent neural network (RNN) models of the simple RNN and long short-term memory (LSTM) network was also performed. Furthermore, the proposed model was applied to two web forums. The experimental results indicate that the deep learning model affords significant improvements over the accuracy of conventional machine learning associated with text features. The accuracy of the proposed model using LSTM reaches 98.56%, and the precision and recall of the noise class reach 99% and 99.53%, respectively.

Keywords: web forum; social media; content noise; posting quality; text mining; deep learning; machine learning

1. Introduction

Nowadays, social sustainability has been considered pivotal to sustainable development together with environmental and economic sustainability, although much attention has lately been focused on social sustainability [1]. It has been defined in a variety of ways, and the common aspect among them is to promote and sustain a high quality of life and wellbeing within communities by satisfying people's social needs [2]. Accordingly, it would be very helpful for our society to build and maintain a communication channel as a transparent platform on which all individual social needs can be merged into a social voice.

With the advent of information and communication technologies, the Internet has played an increasingly important role in everyday lives, in the way that people communicate via the Internet, and how they are socially connected with each other. In particular, the emergence of web-based applications, such as social media, blogs, wikis, and web forums, has radically altered how people communicate and interact with each other in the digital world.

Among the web-driven applications, social media is used by billions of people in the world as a communication channel. It lowers physical barriers and the cost of creating content, thereby accelerating information production and consumption. People not only consume provided content,

but also directly create and share content on the web, thus influencing the information acquisition and thinking of others. Accordingly, social media data are widely exploited in analyzing public opinion to evaluate social sustainability by involving scientific and computational techniques [3].

Meanwhile, web forums can provide an opinion platform where people discuss various topics of similar interests by sharing their ideas and experiences. They contain a wealth of information and intensive discussions on specific topics, making them the most important opinion-forming media [4]. However, improper use by certain users is degrading the value of web forums. Forums open to ordinary users typically allow anyone to write content, which often results in many promotional posts unrelated to the topics. Moreover, unproductive personal attacks among users do not lead to practical discussion. Such postings in web forums can be considered as "noise".

This work was started to improve the quality of the datasets of postings in web forums for opinion mining. We aimed to build business intelligence, focusing on Walmart, by listening to various stakeholders online, which is a new channel compared to traditional customer surveys. The main task was to investigate the main topics related to the company and to analyze the sentiment on the company. This information can be used to improve the service or product quality of the company and to develop or improve the brand image of the company. To capture what is going in online communities regarding the company, the basic process is to derive the main topics, which are frequent keywords. In this regard, it is noted that the importance of a post is evaluated in terms of the number of involved authors or comments, but content noise distorts the implications of the analysis by adding unnecessary text in the opinion analysis. To accurately identify user opinions in a web forum, this noise must first be removed through pre-processing. In terms of users, unnecessary posts have the effect of hindering web forum activity. Moreover, in terms of operating the web forum, unnecessary posts, conflicts among users, and spam can lower user experience and lead to users leaving the forum. Addressing these issues has become increasingly important today.

Walmart is the world's largest retail company according to the Fortune Global 500 list in 2019. It has 11,484 stores and clubs in 27 countries (as of 30 April 2020). Accordingly, Walmart is widely known, and various discussions take place in several forums. A case study on Walmart found that preprocessing is necessary for accurate opinion mining, so the experiment venue is set to Walmart.

An automatic noise detection model is proposed for web forums using representative machine learning approaches accompanied by feature generation and deep learning without feature generation. In the traditional machine learning approach, text features are extracted from the posts in the web forum, and a classification model is built by learning differences between two classes—noise and normal posts. Second, a deep-learning-based detection model is built using raw texts by adopting the following deep learning models: A deep neural network (DNN) without processing of the word sequence and a recurrent neural network (RNN) with processing of the word sequence.

The paper is organized as follows. In Section 2, related works in opinion mining are introduced to highlight the motivation of our work, the role of noise content detection in opinion mining is explained, and finally, previous works from the perspective of algorithms focusing on deep learning models recently widely adopted in text processing are reviewed. Section 3 presents the proposed models for noise content detection built with conventional machine learning and deep learning algorithms, such as simple RNN and long short-term memory (LSTM). Section 4 presents the experimental results of comparing the learning models according to the performance metrics, including accuracy, recall, precision, and F-value. Section 5 discusses the limitations and remaining challenges, and finally, Section 6 offers concluding remarks.

2. Research Background

2.1. Opinion Mining

First, the value of social media replacing the traditional marketing tool is highlighted, and then the related works dealing with the virtues of text mining for web posts are reviewed. Finally, studies

on text mining from the perspective of two kinds of algorithms are reviewed: Machine learning with feature engineering and deep learning that is applicable to raw text. Text mining is a technique that applies natural language processing and document processing technology to unstructured data (e.g., web forum posts) to extract and process useful information.

Regarding the analysis of web content, it is noted that beyond conventional questionnaires and telephone surveys, studies of web-based market research (including customer surveys and public opinion surveys) began with analysis of customer responses through email. Sampson predicted that the advent of the web would usher in online user communities as a source of market research [5]. Gillin pointed out the growing online influence of customers on each other and its impact on market competition [6]. Moreover, studies on online opinion analysis have primarily investigated opinions on certain products through product reviews on the company website or online stores, and then used the results to develop products or devise market strategies. For example, Morinaga et al. investigated product awareness through online surveys [7], and Liu et al. developed a system to visualize the influence of online customer evaluations of product characteristics on the purchasing behavior of other customers [8]. Glance et al. proposed a model for identifying and analyzing information on product reviews from blogs or web forums [9].

Many researchers have conducted studies to extract opinions from social media, particularly blogs, and determine their correlation with company performance indicators, such as movie box office rankings and product sales over the past 10 years [5,7,10]. Over the past decade, numerous such studies have also used data from direct user evaluations. These studies are indicative of the trend of research on directly analyzing web content as a means of replacing the traditional consumer surveys [4]. Gruhl et al. demonstrated that the number of mentions of a certain product in blogs and the link structures among blogs are related to sharp increases in product sales [10]. Liu et al. used the sentiment probabilistic latent semantic analysis (SPLSA) model, which applies probabilistic latent semantic analysis (PLSA) to blogs, to derive the potential sentiments of blog content, and used the autoregressive sentiment-aware (ARSA) model to analyze the relationship between product sales and product-related sentiment [8].

2.2. Spam Detection

Some of the literature also includes prior studies on noise in social media. Wanas et al. developed a model for automatically evaluating the quality of forum posts [11]. Features including topic relevance, originality, forum-related characteristics (number of citations and comments), surface-level features of the post (post length, speed of author responses to replies, quality of post formatting), and content-related features (numbers of links and questions) were derived, and the support vector machine (SVM) was used to build an automatic evaluation model. Regarding spam, noting that spam in forums degrades the search engine quality, Niu et al. compared spam between forums and blogs [12]. They additionally proposed a context-based (redirection and cloaking) model that can automatically detect forum spam. Noting the issue of spam in Web 2.0, Hayati and Potdar compared and analyzed current techniques for detecting or preventing spam [13]. Lin et al. proposed a spam-detection model for blogs that uses the characteristics of content and the regularity of posting times [14]. Mishne et al. presented a model for detecting spam in blog comments [15], and Han et al. presented a model for detecting promotional posts in blogs, including comments and trackbacks [16]. To detect opinion spam in online opinions, particularly product reviews, Jindal and Liu proposed a model encompassing review content, reviewer characteristics, and product characteristics [17]. Zinman and Donath considered the issue of spam in social networking sites and proposed a model to detect spam originating from fake and illegal user profiles [18]. Benevenuto et al. proposed a spam extraction model to address the issue of content spam on YouTube (titles and other promotional videos) [19].

2.3. Deep Learning for Text Mining

Regarding text mining, previous works have utilized machine learning algorithms after deriving text features. Meanwhile, as deep learning is known to work well in natural language processing, recent

works have adopted the deep learning model, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). In particular, long short-term memory (LSTM) that can learn long-term dependencies is used in text mining, such as in text classification, text summarization, and text generation.

Song et al. proposed an LSTM–CNN-based abstractive text summarization framework that can construct new sentences [20]. Rather than utilizing semantic phases, their framework explored fine-grained fragments from source sentences and then generated text summaries with deep learning, outperforming the state-of-the-art models in terms of both semantics and syntactic structure. Zhang et al. demonstrated a new topic-enhanced LSTM model to deal with the document representation problem [21]. They first built an attention-based LSTM model to generate a hidden representation of a word sequence in a given document, and then created a tree-structured LSTM to generate a semantic representation of the document. Their model was evaluated with typical text mining applications, including document classification, topic detection, information retrieval, and document clustering. Wei et al. proposed a method for malfunction inspection report processing for power grid inspection personnel to deal with unstructured text data by combining text mining-oriented RNN with LSTM [22]. The experimental results showed that the RNN–LSTM-based method could successfully diagnose labeled malfunction inspection reports when given unstructured text data.

From the application perspective, there are some works applying deep learning models in spam detection. Regarding the appropriateness of content and language, Yenala et al. pointed out that inappropriate content on the web includes hurling abuses and passing rude and discourteous comments to individuals [23]. Consequently, they proposed the convolutional bi-directional LSTM, which combines the strengths of both convolutional neural networks and bi-directional LSTMs. Their application includes query completion suggestions in search engines and user conversations in messengers. Jain et al. implemented an LSTM-based spam classification method and compared the performance on two datasets (i.e., SMS Spam Collection and Twitter) with various conventional machine learning algorithms, such as SVM, Naïve Bayes, ANN, and random forests [24]. The experimental results showed that the proposed LSTM-based method outperformed traditional machine learning methods. They also proposed a CNN- and LSTM-based spam detection method in social media by working with knowledge bases, such as WordNet and ConceptNet; their experimental results on two benchmark datasets (SMS and Twitter datasets) showed its effectiveness [25]. Ren and Ji also proposed a combined model of CNN and RNN for deceptive opinion spam detection [26]. Recently, Roy et al. proposed a method for classifying spam and non-spam SMS text messages with CNN and LSTM models [27]. The model showed 99.4% accuracy on a benchmark dataset consisting of 747 spam and 4827 non-spam text messages.

As mentioned above, deep learning has started being applied to spam detection mainly for Twitter and SMS, in which texts are relatively shorter than texts in web forums. In contrast to that in social media, spam in web forums containing longer texts has not yet been studied with state-of-the-art deep learning models, which echoes our motivation for this research. The promotional messages, opinion spam, and abuse contents have not yet been studied under the deep learning framework, while they were previously studied under a framework combining text feature extraction and machine learning.

Research on web forums where people who have similar interests can have in-depth conversation is lacking. Furthermore, the overall judgement of quality of posts on the web by detecting noise is a novel approach. Recent works on deep learning started to combine the CNN and RNN, but this will cause a high computational cost. This work will focus on building a lightweight detection model and on addressing the issue of content noise for opinion mining, which has not been addressed in previous works.

3. Spam Detection Model

The framework of the spam detection model for web forum content cleansing is shown in Figure 1.

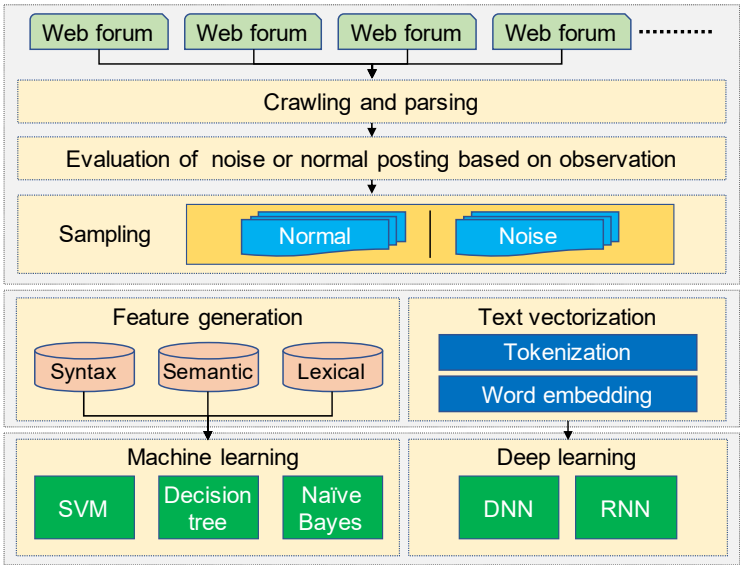

Figure 1. Content noise detection model.

First, a crawling operation is performed to collect web forum posts. The crawler utilizes the CSS (cascading style sheet) element applied to display formatted content in the web browser. With this model, HTML-format posts are parsed to extract the necessary contents, which are then stored in a database. Basically, parsing is performed by extracting the text part wrapped in a specific HTML tag.

After collecting content, training data tagged as noise and normal text are required. For this purpose, an evaluator judges whether or not each post can be considered content noise for randomly selected examples. Then, two different types of processes are performed; one is based on a traditional machine learning model using text features and the other is based on a deep learning model using raw text. For feature generation, the text features that can characterize the raw text are extracted from the raw text, tagged as normal or noise, and are ready to be fed to the machine learning models. Regarding the second approach, raw text is fed to the deep learning model. To explore the importance of punctuation in content noise detection, two scenarios that either include or exclude punctuation as features were developed.

To experimentally determine the best-performing model, the classification performance of the three models for noise detection in web forums was assessed. The cross-validation method was used to combine learning and testing with limited data, and these results were compared.

Here, it is noted that the efficacy of the trained automatic classification model is measured based on the precision and recall of the noise class. Our dataset is biased; the number of noise-class posts is much lower than the number of normal-class posts. The overall accuracy considering the two classes equally can be higher even if the model does not detect the minor class. Precision indicates how accurately the model can detect the noise class. Recall indicates how sensitively the model can detect the noise class. Precision is calculated using the proportion of posts determined to be actual noise among the noise posts classified by the learning model. Recall is calculated using the proportion of noise identified by the learning model among the real noise posts. To compare these two measurements between the models, their averaged F-values are compared as follows:

$$\begin{aligned}
&\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \\
&\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\
&\text{F-value} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\
&\text{TP}: \text{ True Positive (correctly classified spam)} \\
&\text{TN}: \text{ True Negative (correctly classified normal post)} \\
&\text{FP}: \text{ False Positive (incorrectly classified spam)} \\
&\text{FN}: \text{ False Negative (incorrectly classified normal post)}
\end{aligned} \tag{1}$$

3.1. Conventional Machine Learning Model Using Feature Engineering

Our proposed learning model automatically classifies noise in a web forum. To detect noise, post features extracted by means of text mining technology are used as variables to build a classification model that distinguishes noise from normal posts. Here, noise refers to spam, such as promotional posts, posts containing links to illegal sites or malicious code, posts unrelated to the topic, and prolonged discussions due to abuse or slander among users.

In general, features extracted from the textual data include content-free features (i.e., lexical features, syntactic features, and structural features) and content-specific features (e.g., word N-grams). Our machine learning model also uses the syntactical, semantic, and lexical features of the posts (see Table 1). These features typically frame the analytical perspective when analyzing text in linguistics. In the past, text analysis has mainly utilized semantic features, i.e., words or combinations of words [28,29]. Here, the syntactic and lexical features of the sentences are included to examine text features from various perspectives.

Table 1. Text features.

Classification	Text Features
Syntax	Part of speech
Semantics	Unigram, bigram
Lexical	Punctuation, line length, contains non-stop words, stemming, rare words

The proposed model uses the following variables for each type of feature: Unigrams and bigrams for semantic features; part of speech for syntactic features; and punctuation, line length, contains stopwords, stemming, and rare words for lexical features. Here, a unigram is one word, whereas a bigram is a combination of two consecutive words. For example, "forum" is a unigram and "web forum" is a bigram. The term "part of speech" (POS) refers to a combination of grammatically distinct words. This variable indicates the syntactic features of the post, which are normally divided into verbs, nouns, adjectives, and adverbs. Accordingly, POS is a feature that indicates verb + adverb, adjective + noun, noun + noun, etc. The variable "punctuation", which refers to the mark at the end of a sentence, characterizes the post according to whether it contains a period, comma, question mark, or quotation mark. "Line length", the length of the text, is an evaluation criterion that represents the depth and detail of the text. The variable "contains stopwords" indicates whether the text contains words excluded from the text analysis because their meanings are not distinct (e.g., prepositions and relative pronouns). This variable indicates how much meaningful information the post contains. "Stemming" measures the frequency of words with the same root but different forms in the sentences, while "rare words" measures the proportion of non-repeating words in the text.

Next, a classification model was constructed to distinguish between spam and normal text based on the extracted text features.

To construct the automatic learning model, the Naïve Bayes model, SVM, and decision tree, which are three different representative machine learning models, were adopted. The Naïve Bayesian probability model assumes independence between variables and enables learning with only a small

amount of data; as the variables are assumed to be independent, there is no need to calculate the covariance between them. SVM, a non-probabilistic classifier, constructs a multi-dimensional plane to distinguish classes in a multi-dimensional space. It is known for its excellent performance in classification. Meanwhile, a decision tree constructs a tree of variables in a direction that reduces the diversity of the classes included in the tree. Unlike the two aforementioned models, rather than using all variables, the tree variables are selected based on the degree of diversity reduction.

3.2. Deep Learning Model

Text is one of the most widespread forms of sequenced data. It can be understood as either a sequence of characters or a sequence of words. In general, deep learning models are applied to text for document classification, sentiment analysis, author identification, etc. A deep learning model automatically derives the statistical patterns in written language; thus, extensive feature generation is not necessary for deep learning.

In this study, the text classification method that uses machine learning after extracting various linguistic features was compared with deep learning that performs automatic feature generation. Deep learning models consider input text as numeric tensors, and this process is called vectorizing: The input text is tokenized into words, and each word is transformed into a vector.

In this study, the unigram was used for tokenization. Conventional machine learning uses unigrams and bigrams; however, in the deep learning model, consecutive words are considered as a sequence. Thus, only unigram tokenizing is required. N-gram features are extracted automatically in the deep learning model, even though humans cannot interpret which features are exactly extracted.

The token, which is a word that is associated with numeric vectors, is fed to the deep learning model. In the study, a 50-dimensional vector for word embedding to pack text information into lower dimensions was used.

Regarding the modeling procedure, first, an entire document is regarded as a sample. Next, a document is tokenized into words and an index is built for all tokens in the data. In addition, the punctuation and special characters from the sample are included in the first experiment using the deep learning case. In the second case, special characters are excluded. Next, a unique index is assigned to each word. All the words from all the samples are vectorized, and therefore, the vector dimension is set to the maximum value of the length of all sentences. If a sentence is shorter than the maximum length, zero padding is used to ensure that the sentence length is maximized. Two different word-embedding methods are set: One includes punctuation and special characters, and the other excludes them. In the study, the RNN was adopted, as it can learn features from sentences by examining continuous word sequences.

Figure 2 shows the conceptual description of DNN, simple RNN, and LSTM. The blue circle represents each token—in our case, a word. The orange square represents the processing layer used to incorporate input tokens. DNNs process all tokens together and transform them into output. In this case, the sequence of tokens is ignored. RNNs process sequences by iterating through the token from the text and maintaining a state containing past sequences. The box labeled as RNN is a loop that reuses the state computed during the previous iteration and computes it with a current input. LSTM has a carrying layer that saves previous sequences for later use. This prevents previous patterns from vanishing during the process. The comparison of the three models will tell us that the learning sequence of words is significantly important in detecting noise. Furthermore, the comparison between the simple RNN and LSTM will tell us how important remembering long-term sequences is in detecting noise.

The model architecture is built as follows. In the RNN, the first layer is composed of the recurrent layers or LSTM layers to learn features, and subsequent dense layers are used to perform classification. For comparison purposes, the DNN is also set up with two dense layers right after the word-embedding layer. The DNN derives abstract features based on the presence of word vectors, "forgetting" the sequences of word vectors. On the other hand, RNN models consider temporal sequences of word vectors. RNNs use their internal state to process variable-length sequences of word vectors.

The difference between simple RNN and LSTM is that the LSTM can "remember" a long-term sequence and process it with the current sequence.

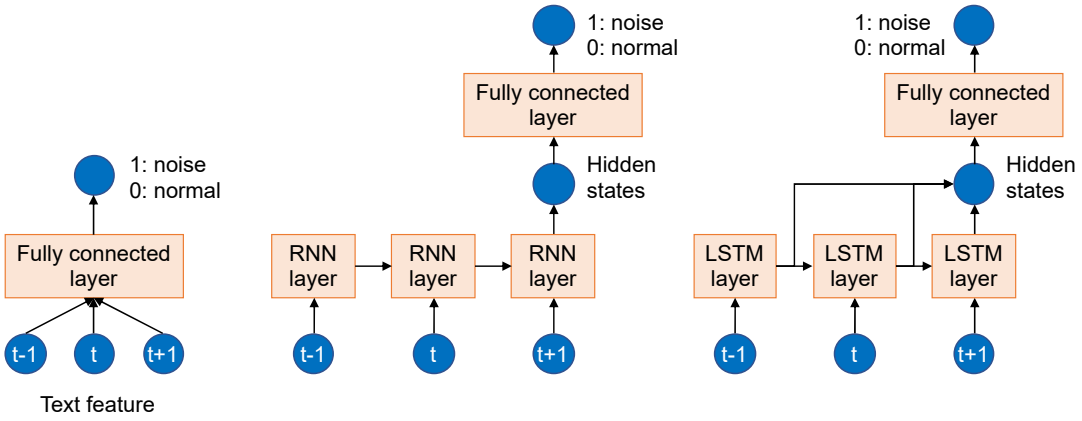

Figure 2. Conceptual description of the deep neural network (DNN), simple recurrent neural network (RNN), and long short-term memory (LSTM).

4. Experimental Results

4.1. Web Forum Data Collection

In this study, we used data from a Walmart web forum: The Walmart Message Board (within Yahoo! Finance), which is the Walmart-related web forum. We set the experiment venue to be the Walmart-related forum, since Walmart is the world's largest retail company, according to the Fortune Global 500 list in 2019. It has 11,484 stores and clubs across 27 countries (as of 30 April 2020) [30]. Walmart is widely known, and diverse discussions take place in various forums, so our experiment venue is set to Walmart; a case study of opinion mining on Walmart revealed that preprocessing is necessary for accurate opinion mining.

Though the forum is primarily for investor communication, employees and consumers also use it, which has resulted in the availability of a variety of topics. The data were collected from the "Wal-Mart sucks" board as customer opinions. The forum was crawled for data, after which the data were parsed and the necessary variables were stored in a database. Among the collected data, posts were randomly extracted, and four evaluators judged which posts were noise. The number of randomly selected posts of each forum is stated in Table 2.

Table 2. Experimental dataset.

Web Forums	Stakeholders	No. of Samples	No. of Content Noise	URL
Yahoo! Finance	Investor	868 (52.6%)	135 (59.5%)	http://messages.finance.yahoo.com/mb/WMT/
Wal-Mart sucks	Customer	783 (47.4%)	92 (40.5%)	http://www.walmartsucks.org
Total		1651	227	

The dataset is randomly split into four subsets, which means that each subset is used in turn as testing data, and the remaining subsets are used as training data. The split follows a stochastic process, so the split result differs with every turn. Thus, the repeated K-fold cross-validation was adopted to guarantee unbiased performance. K-fold cross validation method is a statistical skill to measure the performance of the model on new data after splitting the data into K-folds. Each fold is used as testing data in turns, and the remaining K-1 folds are used as training data. The K-fold split follows a stochastic process, so the split can differ per each implementation. As a result, the algorithm performance is affected by this randomness. The repeated K-fold validation complements this

weakness by repeating the step splitting samples into folds n times. In our case, we used 10 times four-fold cross-validation.

Some samples are shown in Figure 3. Every post, including threads and replies, is tagged normal or noise.

	A	B
1526	normal	It'd be nice if my night crew would stop fucking plugging my department...I'm to the point where I'm going to go get a chainsaw and a hockey mask...
1527	normal	@@@START OF QUOTATION@@@masterofdisharmony wrote:It'd be nice if my night crew would stop fucking plugging my department...I'm to the point where I'm going to go get a chainsaw and a hockey mask...@@@END OF QUOTATION@@@ Hey Smart Ass, since you are s
1528	noise	I was a stockman when I was 16 it sucked i hated it biggggg time...
1529	noise	man joey, if you were from idaho i'd swear i know who you are.
1530	normal	I stocked overnights for about five months, at the anaheim store. worst decision of my life. I just stopped showing up. best decision of my life.
1531	noise	@@@START OF QUOTATION@@@mariposa93 wrote:man joey, if you were from idaho i'd swear i know who you are.@@@END OF QUOTATION@@@ Um. You're from idaho? What store?
1532	noise	i did work at caldwell, 2780 but i moved from there a couple months ago. you're in idaho?
1533	noise	Let's just say caldwell is in my market.
1534	normal	I really like Boise and Caldwell. Some of the nicest people I ever met were living in that area. I love northern ID the best! Trout Fishing!!
1535	normal	people think it's so hillarious when i tell them that we have a security guard at the caldwell store. they're like a security guard for a walmart? in I-DA-HO? well, yeah. half the shootings in the town happen in the walmart parking lot dontcha know. what cracked me up though was i overheard the guy talking one day about how he'd taken on drug lords in columbia and all this other stuff to a customer. then like two weeks later we had all these girls fighting outside the doors and he was too scared to intervene. i really love boise too. i miss it a lot. (i also miss the fact that cost of living was way cheaper there and i took a 60 cent an hour cut in pay to transfer here.) i lived in northern idaho too, in moscow. it is straight up wonderful there! although i never fished. . .
1536	normal	Wow, you had a security guard at Caldwell? WOW. We REALLLLLLLLLLY need one at my store.
1537	normal	I started at Wal-mart a year ago pushing carts. It was the worst job I ever worked. Our machine broke right before christmas and they didn't have it repaired until March(after I transfered to Garden Center, now I'm in furniture). With wally cutting hours, I always ended up working by myself on the weekends for the majority of my shift. The worst part is when you actually get the job done so the other stockmen dissapear for hours and let you do all the work.

Figure 3. A snapshot of data samples.

For the statistical test on the dataset, a Welch two-sample *t*-test was performed. The percentage of each text feature by class (noise or normal) was calculated, and the statistical significance was calculated for the two classes. As shown in Table 3, the low *p*-value implies that the true difference in distribution of text features exists between the two groups of normal and noise text.

Table 3. Statistical test on the experimental dataset.

Test	Group	Mean	*t*-Value	DF	*p*-Value
Welch two-sample *t*-test	noise	4.024784×10^{-5}	64.465	1,204,167	2.2×10^{-16}
	normal	1.533009×10^{-5}			

To perform the classification task, posts judged to be spam by four evaluators were classified as noise, and the remainder as normal posts. Accordingly, spam comprised 27.6% of the total posts. The kappa statistic, which measures inter-evaluator reliability, was calculated at 0.657, thus indicating reliable agreement among the evaluators. Next, the 513 text-based features were derived for each user post. A total of 1651 posts were used as training data, and 10 rounds of four-fold cross-validations were performed.

This work developed two scenarios, focusing on the fundamental differences in the learning algorithms of traditional machine learning and deep learning, the three representative algorithms of decision tree, SVM, and Naïve Bayes in machine learning, and the three different models of simple RNN, LSTM, and DNN in deep learning.

For each deep learning model, various architectures of heavy models or light models with different numbers of filters are explored. For the simple RNN and LSTM, the overall architecture, the number of recurrent layers and dense layers, the number of filters in recurrent layers, and the number of nodes in dense layers were varied and the performance was tracked. This provides information on the best-performing architecture. With the best model, two scenarios that either include or exclude special characters were explored. The 'tm' package was used to derive text features, and RWeka and Keras were used to build an automatic model.

4.2. Spam Detection with Machine Learning Models

Next, three different representative machine learning models, including a Naïve Bayes model, SVM, and decision tree, were constructed. In the SVM, the kernel function depends on what the data looks like. In our case, the number of features is relatively low and has little risk of overfitting. A polynomial kernel was adopted to use a non-linear hyperplane formed with features to divide noise and normal contents. For the decision tree, J48 was used for simple and interpretable results. Naïve Bayes had no specific parameters. The experiment was performed using R installed in a Windows server with a 2.3 GHz CPU and 640 GB of RAM. RWeka was used for the traditional machine learning algorithms. Table 4 summarizes the model specifications of the selected machine learning algorithms.

Table 4. Model specifications of the traditional machine learning algorithms.

Classification Model	Model Specifications
Decision tree	J48 (C4.5), Batch size: 100, confidence factor: 0.25, minimum number of data in a leaf: 2, number of folds: 3
Naïve Bayes	Batch size: 100, no kernel estimation
Support vector machine (SVM)	Batch size: 100, poly-kernel for non-linear hyperplane

Table 5 lists the training results of the three machine learning classification models. As the main objective of the proposed model is to detect noise, the model performance was assessed based on precision and recall for the noise class. The SVM exhibits the highest precision at 0.5417. Overall, the three machine learning algorithms generated poor performance in detecting content noise.

Table 5. Machine-learning-based detection results.

Classification Model	Precision for Content Noise	Recall for Content Noise	F-Value for Content Noise	Accuracy
Decision tree	0.5459	0.4956	0.5195	0.8734
Naïve Bayes	0.3527	0.7456	0.4789	0.7759
SVM	0.5417	0.5702	0.5556	0.8740

In the study, the decision tree model generated classification rules with −, ≤, POS, and nine unigrams. The model used a total of 31 variables, POS including preposition + space, basic_verb + preposition, and past_verb + preposition for POS, and unigrams including "blitz", "work", "thei", "thank", "that", and so on. The number of features, 31 in our case, was determined in terms of information gain. The variable with the highest normalized information gain was chosen to make the decision.

Figure 4 shows the tree output from the decision tree model. The four most important features are sequentially listed as blitz (unigram), line-length, work, and "Preposition + Adjective". In the figure, POS is represented in the following abbreviations: CC: Coordinating conjunction, CD: Cardinal number, DT: Determiner, EX: Existential there, FW: Foreign word, IN: Preposition/subordinate, JJ: Adjective, JJR: Adjective, comparative, JJS: Adjective, superlative, LS: List item marker, MD: Modal, NN: Singular Noun, NNS: Plural Noun, plural, NNP: Singular proper noun, NNPS: Plural proper noun, PDT: Predeterminer, POS: Possessive ending, PRP: Personal pronoun, PP: Possessive pronoun,

RB: Adverb, RBR: Comparative adverb, RBS: Superlative adverb. Content that contained the "blitz" word, had a longer line length, and excluded the "DOLLAR" special character turned out normal content. Content noise excluded "blitz" and "work" words and also excluded the "Preposition + Adjective" POS composition. This indicates that "blitz" is relevant to investors and that "work" is relevant to workers, so the noise content does not require those words. In addition, the noise content is relatively short and tends to be grammatically broken, so the "Preposition + Adjective" composition does not show up in the noise.

```
blitz <= 0
|   line_length <= 0.014382
|   |   work <= 0
|   |   |   IN_JJ <= 0
|   |   |   |   VBP_DT <= 0
|   |   |   |   |   thei <= 0
|   |   |   |   |   |   IN_NNP <= 0
|   |   |   |   |   |   |   NN_VBZ <= 0
|   |   |   |   |   |   |   |   thank <= 0
|   |   |   |   |   |   |   |   |   that <= 0
|   |   |   |   |   |   |   |   |   |   my <= 0
|   |   |   |   |   |   |   |   |   |   |   and <= 0
|   |   |   |   |   |   |   |   |   |   |   |   BOL_WP <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   NN_NN <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   from <= 0: noise (92.0/25.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   from > 0: normal (4.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   NN_NN > 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   IN_DT <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   LESS_THAN <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   DASH <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   IN_NN <= 0: normal (11.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   IN_NN > 0: noise (3.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   DASH > 0: normal (3.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   LESS_THAN > 0: noise (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   IN_DT > 0: noise (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   BOL_WP > 0: normal (7.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   and > 0
|   |   |   |   |   |   |   |   |   |   |   |   |   BOL_CC <= 0: normal (12.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   BOL_CC > 0: noise (2.0)
|   |   |   |   |   |   |   |   |   |   |   my > 0
|   |   |   |   |   |   |   |   |   |   |   |   DT_NN <= 0: normal (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   DT_NN > 0: noise (3.0/1.0)
|   |   |   |   |   |   |   |   |   that > 0
|   |   |   |   |   |   |   |   |   |   do <= 0: normal (12.0/1.0)
|   |   |   |   |   |   |   |   |   |   do > 0: noise (3.0/1.0)
|   |   |   |   |   |   |   |   thank > 0: noise (9.0)
|   |   |   |   |   |   |   NN_VBZ > 0: normal (13.0/3.0)
|   |   |   |   |   |   IN_NNP > 0: normal (10.0/1.0)
|   |   |   |   |   thei > 0: normal (14.0/2.0)
|   |   |   |   VBP_DT > 0
|   |   |   |   |   have_a <= 0: normal (16.0)
|   |   |   |   |   have_a > 0: noise (3.0/1.0)
|   |   |   IN_JJ > 0: normal (12.0)
|   |   work > 0: normal (13.0/1.0)
|   line_length > 0.014382
|   |   DOLLAR <= 0
|   |   |   IN_DT <= 0
|   |   |   |   back <= 0: normal (358.0/43.0)
|   |   |   |   back > 0
|   |   |   |   |   TO_NN <= 0
|   |   |   |   |   |   FW_FW <= 0: normal (18.0/1.0)
|   |   |   |   |   |   FW_FW > 0: noise (2.0)
|   |   |   |   |   TO_NN > 0: noise (6.0)
|   |   |   IN_DT > 0
|   |   |   |   NN_ <= 0: normal (880.0/23.0)
|   |   |   |   NN_ > 0
|   |   |   |   |   forum <= 0: normal (30.0/1.0)
|   |   |   |   |   forum > 0: noise (4.0)
|   |   DOLLAR > 0
|   |   |   199 <= 0
|   |   |   |   DT_NN <= 0: noise (5.0)
|   |   |   |   DT_NN > 0
|   |   |   |   |   hd <= 0
|   |   |   |   |   |   CD_JJR <= 0: normal (55.0/2.0)
|   |   |   |   |   |   CD_JJR > 0: noise (3.0/1.0)
|   |   |   |   |   hd > 0: noise (4.0/1.0)
|   |   |   199 > 0: noise (5.0)
blitz > 0
|   VB_VBG <= 0: noise (27.0/1.0)
|   VB_VBG > 0: normal (3.0)
```

Figure 4. Tree results from the decision tree model.

When compared with the other two training models that use entire variables, the decision tree is a learning model that includes relatively few variables; hence, it is considered to have poor classification performance when using text features in posts.

For the Naïve Bayes model, unigrams and POS terms were used with high precision; among syntactic features, line length was ranked. Thus, in this model, lexical and syntactic features are important variables for detecting noise. Meanwhile, for the SVM model that uses a poly-kernel function, among the top 10 important variables for classifying normal and spam posts according to the coefficient values in the polynomial model, one POS (Verb + Preposition) was ranked, and nine unigrams (i.e., nice, receipt, fish, guess, manag, call, bentonvil, overtim, law, repli) were ranked. The SVM model was constructed mainly using semantic features. In the classification model generated from the three learning models, unigram was the most important feature, followed by POS and punctuation.

4.3. Spam Detection with Deep Learning Models

For the deep learning experiments, the same machine as that used in the experiments for the machine learning models was used. In addition, Keras was adopted as a deep learning library, and the parameters were set as listed in Table 6. In the study, the parameters were varied and the performance was checked in each case. The best performance was achieved with the optimal number of epochs = 10 and batch size = 10; Adam optimization was used to achieve this result. The number of unique words, including punctuation, was 17,475. The maximum length of samples was 5,649. Sentences shorter than this length were padded. To express the 17,475 words in lower dimensions, we set the Word2Vec dimension to 50.

Table 6. Specifications of the deep learning model.

Parameter	Specifications
Batch size	20
Max length of input	5649
Word2Vec dimension	50
Filters in RNN layer	50
Size of dense layer	20×1
Optimizer	Adam
Epoch	20

Next, the parameters of the deep learning architecture were also adjusted to construct a light model. The number of filters in the RNN and the dense layer size were reduced. The comparison between the DNN, RNN, and LSTM performances are shown in Table 7. The table reveals that the DNN that did not consider the sequence of words generated a better performance than the simple RNN, but the DNN performed slightly more poorly relative to the LSTM in the complex model with a large dense layer size. However, when the model was light with a small-sized dense layer, LSTM afforded the best performance. This result implies that the word sequence is important in detecting noise in web forums.

Table 7. Performance comparison for various specifications of the deep learning model.

Filters in RNN Layer	Dimension of Dense Layers		Algorithms
	50	20	
50	0.9843	0.9807	RNN
	0.9837	0.9845	LSTM
20	-	0.9789	RNN
	-	0.9856	LSTM
-	0.9847	0.9840	DNN

LSTM is slightly better than the simple RNN and DNN, even with the light architecture. This implies that memorizing the long-term sequence is efficient in content noise detection. The precision and recall of the noise class were checked because the sample class was unbalanced. The LSTM, as it afforded the best performance, gave a good performance for the noise class, which is different from the machine learning algorithms, as shown in Table 8.

Table 8. Performances for the noise class.

Classification Model	Precision	Recall	F-Value	Accuracy
LSTM with 20 LSTM filters, 20 × 1 dense layer	0.9900	0.9953	0.9926	0.9856

In addition, the role of punctuation for noise detection in text using the architecture affording the best performance was tested. The performance was reduced little when excluding the punctuation (see Table 9) in the case of the best performance. The results imply that special characters do not play an important role in noise detection.

Table 9. Performance comparison between cases of "including special characters" and "excluding special characters".

Classification Model	Including Special Characters	Excluding Special Characters
DNN with 20 × 1 layer	0.9847	0.9832
Simple RNN with 50 RNN filters, 50 × 1 dense layer	0.9843	0.9809
LSTM with 20 LSTM filters, 20 × 1 dense layer	0.9856	0.9855

5. Discussion and Remaining Challenges

Our work contributes to the related research fields from the following perspectives.

First, various deep-learning-based techniques have started to be applied to spam detection in social media, such as Twitter and SMS messages, of which texts are relatively shorter than those in web forums as main venues. The spam in web forums that usually contain long texts has not been studied with deep learning models yet, and thus, we have tried to explore the potential of deep learning techniques in spam detection for web forums.

Second, spam detection focuses on promotional messages; however, the content noise is a border concept including unnecessary posts, conflicts among users, and spam. The spam detection model, for example, cannot detect abusive posts or meaningless posts. This work aimed to develop a general model to be applied for multiple types of content noise.

Third, in this work, two different word embedding layers were developed; one includes special characters (e.g., ampersand '&', asterisk '*'), and one excludes them.

Finally, a comparative analysis of traditional machine learning and deep learning for content noise detection was performed. Two different types of processes were proposed; one is based on a traditional machine learning model using text features, and the other is based on a deep learning model using raw text. This work highlights the value of deep learning in text mining compared to previous works that extract extensive text features and develop models.

Although the experiments with deep learning models showed meaningful performance for the collected dataset, our work has a limitation in that two forums of a corporation were analyzed, resulting in a limited number of samples. A more comprehensive evaluation process would be necessary to validate our proposed model by applying it to other forums in diverse sectors where various topics of interest are discussed. Furthermore, all types of content noise, such as spam, abusive, and meaningless posts were taken into consideration as a 'single' class. For effective operation and usability of the web forum, a species class for each would be helpful.

Our future research will extend to detect spam through the syntax of posts in web forums. For example, the degree to which two users repeatedly converse in a text, as well as the post author

characteristics, can be used as variables. Because content noise can also be classified into more categories, whereas the present model classifies posts into noise and normal posts, future can be developed to further classify noises as promotional posts, personal attacks, violent and sexual posts, and off-topic posts. Such specified classifications can contribute to improving filtering systems and content quality in web forums.

6. Conclusions

Over the past decade, social media and web forum data have been widely utilized in collecting and analyzing public opinion to assess sustainability in various sectors, such as for governments, companies, communities, etc. In particular, web forums can powerfully influence public opinion and behavior, and thus mobilize and draw people to focus on ethics, social issues, and sustainability. Therefore, in order to keep such opinion platforms for the public transparent and sustainable, it is necessary to monitor and recognize "spam" opinions that are deliberately and sometimes illegally generated to mislead people or automated opinion mining systems.

The importance of a web post is normally evaluated in terms of the number of involved authors or comments. Thus, noise in the form of spam distorts the analysis results by adding large amounts of unnecessary data in the opinion analysis. To address this issue, an automatic noise detection model for web forums is proposed using representative machine learning approaches with feature generation and deep learning without feature generation. In the traditional machine learning approach, text features were extracted from the posts in the web forum through text mining, and a classification model was trained with two classes—spam and normal posts. For better performance, this work adopted the deep learning models of DNN without processing of the word sequence and RNN with processing of the word sequence.

To evaluate the proposed model's performance, the proposed model was applied to two Walmart-related forums: "Yahoo! Finance" for investor opinions and the "Wal-Mart sucks" board for customer opinions. It was found that the chosen conventional machine learning models afforded a poor performance of 87.40% for accuracy and 55.56% for the F1-score, at best, with SVM; however, the deep learning model afforded a better performance, with the highest accuracy of 98.56% and a 99.26% F1-score with LSTM. Among the chosen recurrent models, LSTM, which has memory to keep long sequences, showed a slightly better performance with 98.56% accuracy than the simple RNN with 98.43% accuracy and the DNN with 98.47% accuracy. In summary, we can conclude with confidence that our deep learning model that encompasses spam, abusive and unnecessary posts, and so on, especially by processing the sequences of words, could be a proper solution for content noise detection in web forums.

Author Contributions: Conceptualization, J.W.; methodology, J.W.; writing—original draft preparation, J.W.; writing—review and editing, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education under Grant NRF-2020R1I1A3A0403740911. This work was also supported by the Soonchunhyang University Research Fund.

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

References

1. Vallance, S.; Perkins, H.C.; Dixon, J.E. What is social sustainability? A clarification of concepts. *Geoforum* **2011**, *42*, 342–348. [CrossRef]
2. Eizenberg, E.; Jabareen, Y. Social sustainability: A new conceptual framework. *Sustainability* **2017**, *9*, 68. [CrossRef]
3. Ballestar, M.T.; Cuerdo-Mir, M.; Freire-Rubio, M.T. The concept of sustainability on social media: A social listening approach. *Sustainability* **2020**, *12*, 2122. [CrossRef]
4. Chen, H.; Zimbra, D. AI and opinion mining. *IEEE Intell. Syst.* **2010**, *25*, 74–80. [CrossRef]

5. Sampson, S.E. Gathering customer feedback via the Internet: Instruments and prospects. *Ind. Manag. Data Syst.* **1998**, *98*, 71–82. [CrossRef]
6. Gillin, P. *The New Influencers: A Marketer's Guide to the New Social Media*; Linden Publishing: Fresno, CA, USA, 2007.
7. Morinaga, S.; Yamanishi, K.; Tateishi, K.; Fukushima, T. Mining product reputations on the web. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AL, Canada, 23–26 July 2002; Available online: https://dl.acm.org/doi/10.1145/775047.775098 (accessed on 20 June 2020).
8. Liu, Y.; Huang, X.; An, A.; Yu, X. ARSA: A Sentiment-Aware Model for predicting sales performance using blogs. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 23–27 July 2007; Available online: https://dl.acm.org/doi/10.1145/1277741.1277845 (accessed on 20 June 2020).
9. Glance, N.; Hurst, M.; Nigam, K.; Siegler, M.; Stockton, R.; Tomokiyo, T. Deriving marketing intelligence from online discussion. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005; Available online: https://dl.acm.org/doi/10.1145/1081870.1081919 (accessed on 20 June 2020).
10. Gruhl, D.; Guha, R.; Kumar, R.; Novak, J.; Tomkins, A. The predictive power of online chatter. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21 August 2005; Available online: https://dl.acm.org/doi/10.1145/1081870.1081883 (accessed on 20 June 2020).
11. Wanas, N.; El-Saban, M.; Ashour, H.; Ammar, W. Automatic scoring of online discussion posts. In Proceedings of the 2nd ACM Workshop on Information Credibility on the Web, Napa Valley, CA, USA, 30 October 2008.
12. Niu, Y.; Chen, H.; Hsu, F.; Wang, Y.-M.; Ma, M. A quantitative study of forum spamming using context-based analysis. In Proceedings of the Network & Distributed System Security (NDSS) Symposium, San Diego, CA, USA, 28 February–2 March 2007.
13. Hayati, P.; Potdar, V. Toward spam 2.0: An evaluation of Web 2.0 anti-spam methods. In Proceedings of the 7th IEEE International Conference on Industrial Informatics, Cardiff, UK, 24–26 June 2009; pp. 875–880.
14. Lin, Y.-R.; Sundaram, H.; Chi, Y.; Tatemura, J.; Tseng, B.L. Splog detection using self-similarity analysis on blog temporal dynamics. In Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web, Banff, AL, Canada, 8 May 2007; Available online: https://experts.illinois.edu/en/publications/splog-detection-using-self-similarity-analysis-on-blog-temporal-d (accessed on 20 June 2020).
15. Mishne, G.; Carmel, D.; Lempel, R. Locking Blog Spam with Language Model Disagreement. *AIRWeb* **2005**, *5*, 1–6.
16. Han, S.; Ahn, Y.-Y.; Moon, S.B.; Jeong, H. Collaborative blog spam filtering using adaptive percolation search. In Proceedings of the 15th International Workshop on Peer-to-peer Systems, 3rd Workshop on Weblogging Ecosystem (Held in Conjunction with WWW 2006), Edinburgh, UK, 22–26 May 2006.
17. 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.
18. Zinman, A.; Donath, J.S. Is Britney Spears spam? In Proceedings of the 4th Conference on Email and Anti-Spam (CEAS 2007), Mountain View, CA, USA, 2–3 August 2007; pp. 1–10.
19. Benevenuto, F.; Rodrigues, T.; Almeida, V.; Almeida, J.; Zhang, C.; Ross, K. Identifying video spammers in online social networks. In Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb '08), Beijing, China, 22 April 2008; pp. 45–52.
20. Song, S.; Huang, H.; Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimed. Tools Appl.* **2019**, *78*, 857–875. [CrossRef]
21. Zhang, W.; Li, Y.; Wang, S. Learning document representation via topic-enhanced LSTM model. *Knowl. Based Syst.* **2019**, *174*, 194–204. [CrossRef]
22. Wei, D.; Wang, B.; Lin, G.; Liu, D.; Dong, Z.; Liu, H.; Liu, Y. Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report. *Energies* **2017**, *10*, 406. [CrossRef]
23. Yenala, H.; Jhanwar, A.; Chinnakotla, M.K.; Goyal, J. Deep learning for detecting inappropriate content in text. *Int. J. Data Sci. Anal.* **2018**, *6*, 273–286. [CrossRef]
24. Jain, G.; Sharma, M.; Agarwal, B. Optimizing semantic LSTM for spam detection. *Int. J. Inf. Technol.* **2019**, *11*, 239–250. [CrossRef]

25. Jain, G.; Sharma, M.; Agarwal, B. Spam detection in social media using convolutional and long short term memory neural network. *Ann. Math. Artif. Intell.* **2019**, *85*, 21–44. [CrossRef]
26. Ren, Y.; Ji, D. Neural networks for deceptive opinion spam detection: An empirical study. *Inf. Sci.* **2017**, *385–386*, 213–224. [CrossRef]
27. Roy, P.K.; Singh, J.P.; Banerjee, S. Deep learning to filter SMS Spam. *Fut. Gen. Comput. Syst.* **2020**, *102*, 524–533. [CrossRef]
28. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
29. Zhuang, L.; Jing, F.; Zhu, X.-Y. Movie review mining and summarization. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06), Arlington, VA, USA, 5–11 November 2006; pp. 43–50.
30. Walmart Financial Information. 2020. Available online: https://s2.q4cdn.com/056532643/files/doc_financials/2020/ar/Walmart_2020_Annual_Report.pdf (accessed on 20 June 2020).

© 2020 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 (http://creativecommons.org/licenses/by/4.0/).