applied sciences

MDPI

Article

A Discrete Hidden Markov Model for SMS Spam Detection

Tian Xia [1,*] and Xuemin Chen [2,*]

[1] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China
[2] Department of Engineering, Texas Southern University, Houston, TX 77004, USA
* Correspondence: xiatian@sspu.edu.cn (T.X.); xuemin.chen@tsu.edu (X.C.)

Received: 18 June 2020; Accepted: 18 July 2020; Published: 21 July 2020

Abstract: Many machine learning methods have been applied for short messaging service (SMS) spam detection, including traditional methods such as naïve Bayes (NB), vector space model (VSM), and support vector machine (SVM), and novel methods such as long short-term memory (LSTM) and the convolutional neural network (CNN). These methods are based on the well-known bag of words (BoW) model, which assumes documents are unordered collection of words. This assumption overlooks an important piece of information, i.e., word order. Moreover, the term frequency, which counts the number of occurrences of each word in SMS, is unable to distinguish the importance of words, due to the length limitation of SMS. This paper proposes a new method based on the discrete hidden Markov model (HMM) to use the word order information and to solve the low term frequency issue in SMS spam detection. The popularly adopted SMS spam dataset from the UCI machine learning repository is used for performance analysis of the proposed HMM method. The overall performance is compatible with deep learning by employing CNN and LSTM models. A Chinese SMS spam dataset with 2000 messages is used for further performance evaluation. Experiments show that the proposed HMM method is not language-sensitive and can identify spam with high accuracy on both datasets.

Keywords: short messaging service (SMS); spam detection; hidden Markov model (HMM); text classification; natural language processing (NLP)

1. Introduction

Nowadays, one of the most popular and common communication services is the short message service, known as SMS. SMS traffic volumes have risen from 1.46 billion in 2000 to 7.9 trillion in 2012 [1]. SMS-capable mobile phone users had reached 6.1 billion users by the year 2015 [2]. The growth of mobile users has generated a great deal of revenue [1]. Based on the latest statistics [3], global SMS revenue is predicted to hit 83.2 billion dollars in 2022 even though the revenue has continued to decrease after 2017. In addition, about half (43 billion dollars) of the global SMS revenue belongs to the global P2P (person-to-person) SMS messages market and the other half (40.2 billion dollars) belongs to A2P (application-to-person). A2P messages are sent by companies, such as bulksmsonline.com and bulksms.com, who provide bulk SMS sending services for commercial needs, e.g., verification codes, e-commercial notifications, express delivery notifications.

While enjoying the convenience of communication via electronic devices, unexpected advertising or even malicious information has flooded our email and phone message boxes. This spam information is usually unwanted or unsolicited electronic messages sent in bulk to a group of recipients [4]. It is being sent by spammers or even criminals who are driven by these most profitable spamming businesses. Spam first spread explosively but mainly in emails in the first decade of the 21th century, indicated by the statistical results provided in [5]. As SMS is low-cost, bulk-sending, and reliably reaches the

receiver, spam began moving to this most popular and globally used SMS service. The companies hosting the bulk SMS sending service have to continuously improve their spam filtering technology to combat against spam SMS. However, it is a challenging task as SMS contains limited words including abbreviations, etc. In addition, SMS spammers tend to use legitimate words to increase its rank in spam filters and use obfuscated words to confuse the spam filters.

Antispam techniques have been developed for decades. Many methods for antispam emails have been applied in the antispam SMS field. Traditional machine learning methods for spam detection including naïve Bayes (NB) [6,7], vector space model (VSM) [8], and support vector machine (SVM) [9], and novel methods such as long short-term memory (LSTM) [10] and the convolutional neural network (CNN) [11] all stand on the well-known bag of words (BoW) model, which assumes documents are an unordered collection of words. Based on this assumption, word or term occurrences are the only concern but the order of words in the document is ignored. Based on the BoW assumption, many feature extraction algorithms [12–14] were invented to make use of term frequency, such as TF.IDF (stands for term frequency and inversed document frequency) and word distribution. However, these algorithms do not work well in SMS spam detection, because of the strict length limitation of SMS. Most of the terms (words) occur only once in a single SMS. Human language is actually sequential data. Word order is critical information for SMS spam detection.

Motivated to address the aforementioned issues, we propose a new method based on the discrete hidden Markov model (HMM) for spam SMS detection in this paper. An HMM is a statistical model with two stochastic processes [15]. The underlying stochastic process that has the Markov property is a sequence of hidden states, and the observable stochastic process is a sequence of observation states that reflect the hidden states through some probabilistic distribution. HMMs are a formal foundation for building probabilistic models of linear sequence labeling problems [16]. In this paper, the SMS messages are preprocessed by removing the recommended stop words (e.g., *#&gt*, *&lt*, **, *&amp*, *that*, *shall*, *is*) and punctuation to form the observation sequence with original word order. The HMM is trained by the labeled observation sequence. Then, the SMS classification is a typical decoding problem in HMM by using the Viterbi algorithm. To the best of our knowledge, this is the first research on applying HMM for SMS spam detection based on word order.

The main contributions of this research are threefold:

1. We first propose to use a hidden Markov model for spam SMS detection based on word order. This method uses the word order information that consists of the key importance for human language, but it has been ignored by many traditional methods based on the BoW model.
2. This research solves the issue where the TF.IDF algorithm for word weighting does not work well in SMS spam detection, due to the extremely low term frequency.
3. The proposed method can be applied to alphabetic text (e.g., English) and hieroglyphic text (e.g., Chinese). It is not language-sensitive.

The rest of the paper is organized as follows. Related work is discussed in Section 2. The problem formulation and the proposed SMS spam detection method based on the discrete HMM are presented in Section 3. The experimental results and performance comparisons with well-known models are outlined in Section 4. The conclusions are drawn and future work is discussed in Section 5.

2. Related Work

2.1. Rule-Based Filtering Technologies

The rule-based filtering techniques are popular in commercial business. SpamAssassin [17] is a successful forerunner of typical rule-based systems (RBSs). It has been adopted by antispam industry companies, such as Symantec and McAfee [18]. The next-generation RBS of Wirebrush4SPAM [19] was then developed to increase its throughput. Both SpamAssassin and Wirebrush4SPAM host a set of scored rules and run a score-based mechanism. A spam email is detected when the sum of scores

from triggered rules is greater or equal to the value of a global threshold, which is called required score. However, throughput is a challenging issue of RBSs, and their time complexity of filtering algorithms could not be reduced to an acceptable level. To address the throughput issue, a constant time complexity spam detection algorithm was developed by Xia [20].

2.2. Content-Filtering Technologies

Content-filtering technologies have utilized machine learning technologies to combat spam. One of the most common technologies is the Bayesian classification filter [21]. Bayesian methods, such as naïve Bayes, worked efficiently and had become an important machine learning algorithm in information retrieval. It is based on Bayes theorem with a strong naïve independence assumption that treats each and every word as single, mutually exclusive, and independent. It is defined as a graphical probabilistic model for multivariate analysis. The nodes of the directed graph represent problem variables and the edges represent conditional dependencies between such variables. Jiang et al. [6] put forward a deep feature weighting (DFW) for naïve Bayes and applied it to text classification. Moreover, to enhance its accuracy, Bayesian methods are often hybrid with other algorithms. Sable et al. [7] introduced a hybrid system of SMS classification based on a naïve Bayes classifier and Apriori Algorithm. Ebadati and Ahmadzadeh [22] proposed a genetic algorithm (GA)-naïve Bayes for spam email detection with a genetic algorithm (GA) for feature extraction. Arifin at al. [23] focused on spam detection for SMS by a naïve Bayes classifier and frequent patterns (FP) tree mining, known as FP-Growth.

Vector space model is based on the BoW model. It represents documents as document vectors that are full of word weights and classify documents based on the cosine similarity value of the vectors. VSM is often used for text classification and information retrieval. Santos et al. [24] filtered spam by representing e-mails with the enhanced topic-based vector space model (eTVSM).

Support vector machine is a set of binary SVM classifiers. It trains a decision equation from an n-dimensional space representation of the data into two regions using a hyperplane, which leads to high accuracy. It is popular because it is robust for many circumstances with high classifying speed. In natural language processing (NLP) research, the SVM n-dimensional space is the same BoW vector space of VSM. Chan et al. [25] proposes a word attack strategy and a feature reweighting method toward the SMS scenario in SVM when the length of a message is limited. Tekerek [9] compared the result of NB, K-nearest neighborhood (KNN), SVM, random forest (RF), and random tree (RT), and found that SVM had the best result.

Entropy, as an information theory, is also used with co-training by Zhang et al. [26,27] to combat spam reviews, which promote sales or defame competitors by misleading consumers. Decision tree is also a method for combating spam. Gashti [28] proposed a hybrid of harmony search algorithm (HSA) and decision tree for selecting the best features and classification.

Deep learning has aroused extensive attention these years. Pumrapee et al. [10] proposed an SMS spam detection method based on long short-term memory (LSTM). Research by Roy et al. [11] used convolutional neural network (CNN) and LSTM models in spam SMS detection to achieve the highest accuracy so far.

In addition, researchers also investigated hybrid models for performance improvement. Uysal et al. [29] investigated the impact of feature extraction and selection of the BoW model and then used KNN and SVM for spam SMS filtering. Karthika et al. [30] applied a latent semantic indexing (LSI)-based SVM model for email spam classification. Arijit et al. [31] filtered SMS spam by a recurrent neural network and LSTM. Yang et al. [32] used a multi-modal fusion, which applied LSTM and CNN models to process the text. Zhao et al. [33] applied six classifiers in the basic module and a deep neural network in the combination module. There are also other models for SMS spam detection, such as the neural network [34], KNN [35], and negative selection algorithm (NSA) [36]. Recently, Shang [37] developed a score-based filtering mechanism in consensus of hybrid multi-agent systems with malicious nodes, which can also be applied for spam detection.

2.3. Hidden Markov Model for Spam Detecions

HMM and its variants have found a wide variety of applications. There was a hierarchical hidden Markov model (HHMM) for real-time finger motion synthesis [38], a hierarchical multivariate HMM with reactive interpolation functionality for full-body motion reconstruction [39], a combining speaker-specific Gaussian mixture model (GMM) with a syllable-based HMM for speaker recognition [40], and a Spherical-Self Organizing Map (S-SOM) with HMM for classifying sets of time series [41], to list just a few examples. HMMs have also laid a solid foundation for their applications in NLP, including part-of-speech tagging in many languages [42,43] and name entity recognition [44].

However, based on our literature review including the latest review papers about SMS spam detection techniques [4,45,46], there is no report on using HMM for SMS spam detection based on the word order. Rafique and Farooq [47] used HMM for SMS spam detection on byte-level, which is the low communication level of SMS delivery. Gordillo and Conde [48], as forerunners in this field, proposed a HMM for detecting spam mail in 2007. The paper focused on obfuscated words detection, such as the example in the paper, *m0ney* or *mo.ney* for the word *money*. Therefore, instead of words, they focused on spam detection at the language character level, such as letters in English. They treated characters in spam emails like a DNA chain and used a similar DNA chain classification method for spam emails. Ebrahimi et al. [49] built a HMM for detection and classification of duplicate bug reports (BRs) by focusing on the relation of current BRs and incoming BRs. Similarly, Washha et al. [50] put forward a topic-based HMM for spam tweet filtering and predicted the tweet sample classification based on an assumed high dependency among successive tweets. Vennila et al. [51] used HMM in spam detection over internet telephony in voice, which belongs to a far different research field from this study. The existing work focused on emails, bug reports, and tweets, etc. Unlike the existing work, we aim to use a hidden Markov model for spam SMS detection based on the word order, which is a new application of HMM.

3. The Discrete HMM for SMS Spam Detection

3.1. Problem Formulation and Notations

A typical SMS contains sequential words with punctuation. In English, as words are divided by blank spaces, English SMS is easy to be split, whereas in some other languages, e.g., Chinese, there are no blank spaces between words. These SMS messages have to first feed into a segmentation algorithm to extract words. In any case, each SMS text is first split or segmented into sequential words with punctuation at the very beginning.

Not all words are suitable for NLP. Punctuation and words only for positioning do not have much semantic information. Especially for SMS, many informal words, shortened and abbreviated words, social media acronyms, and some strange character sequences often appear in SMS. Part of them is also meaningless. They are called the stop words. Therefore, these stop words and punctuation are removed from the sequential words.

After these preprocesses, each SMS is refined to a word sequence, which is full of meaningful words. Let N denote the total number of all rest meaningful words in SMS including the duplicated ones. This set with N sequential words is the observation sequence denoted as $\underline{X} = \{o_1, o_2, o_3, \cdots, o_N\}$. The corresponding hidden state sequence is denoted as $\underline{Y} = \{q_1, q_2, q_3, \cdots, q_N\}$, satisfying the Markov property. The structure of these two sequences is represented by the directed graph in Figure 1.

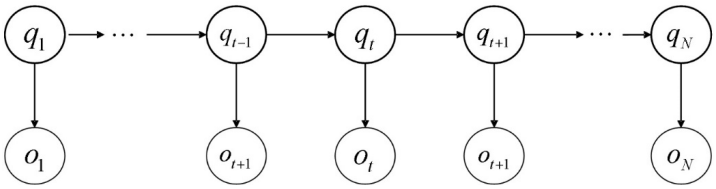

Figure 1. Directed graph of hidden Markov model.

The HMM, denoted by λ, can be defined by a three-tuple:

$$\lambda = (\pi, A, B) \tag{1}$$

where π is the initial probability distribution, A is the state transition probability matrix, and B is the observation probability distribution matrix.

For SMS spam detection, the set of hidden states is $S = \{s_1, s_2\} = \{ham, spam\}$. Thus, $A = (a_{ij})_{2\times 2}$, where $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$, *for* $i, j = 1, 2$ and $\sum_{j=1}^{2} a_{ij} = 1$, *for* $i = 1, 2$.

Let $W = \{w_1, w_2, w_3, \cdots, w_n\}$ denote the set of observation states, which includes all different words in every spam and ham SMS. n is the total number of different words in both sets. Thus, $B = (b_{ij})_{2\times n}$, where $b_{ij} = P(o_t = w_j | q_t = s_i)$, *for* $i = 1, 2$, $j = 1, \cdots, n$.

π is a 2×1 initial probability distribution over the state, $\pi_1 = P(q_1 = s_1)$, $\pi_2 = P(q_1 = s_2)$, and $\pi_1 + \pi_2 = 1$.

The proposed hidden Markov model for spam detection is shown in Figure 2.

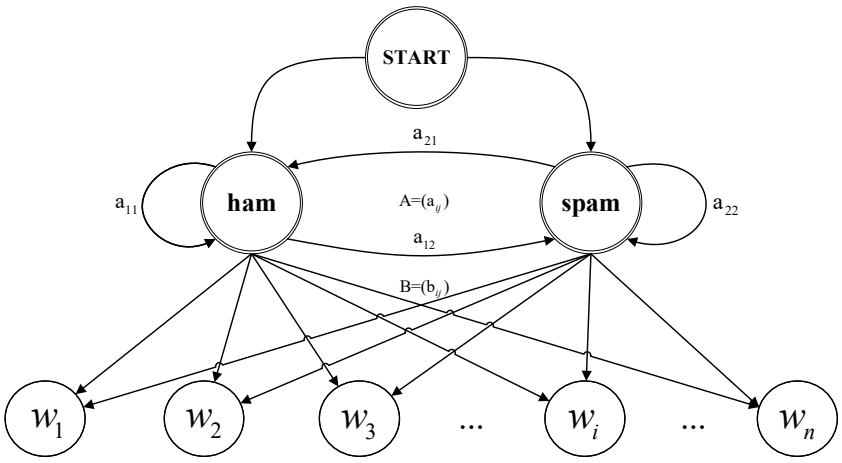

Figure 2. The discrete hidden Markov model for spam detection.

3.2. Observation States and Observation Sequence

The i^{th} SMS can be represented as the rest meaningful sequential words in the set of observation states W: $SMS_i = \left\{w_1^{SMS_i}, w_2^{SMS_i}, w_3^{SMS_i}, \cdots, w_{Lengthof SMS_i}^{SMS_i}\right\}$, $(0 < i \leq M)$. M is the total number of SMS. $w_i^{SMS_i}$ is one of the meaningful words in the i^{th} SMS.

For example, take the following two SMS messages from the UCI repository dataset, which can be downloaded from http://www.dt.fee.unicamp.br/~{}tiago/smsspamcollection. The same dataset has been used in [2,4,8–11,13,21,23,25,31,46,52–57] for performance evaluations.

ham: What you thinked about me. First time you saw me in class.

spam: Are you unique enough? Find out from 30th August. www.areyouunique.co.uk.

Please note that the example dataset only contains 2 SMS messages and words *you* and *me* are duplicated in the two SMS messages.

The two SMS messages can be segmented as word and punctuation sequences with the original order: [*What*, *you*, *thinked*, *about*, *me*, *.*, *First*, *time*, *you*, *saw*, *me*, *in*, *class*, *.*] and [*Are*, *you*, *unique*, *enough*, *?*, *Find*, *out*, *from*, *30th*, *August*, *.*, *www*, *.*, *areyouunique*, *.*, *co*, *.*, *uk*]. After removing stop words and punctuation, the word sequences also keep its order and become refined ones: [*What*, *you*, *thought*, *me*, *First*, *time*, *you*, *saw*, *me*, *class*] and [*Are*, *you*, *unique*, *enough*, *Find*, *30th*, *August*, *www*, *areyouunique*, *co*, *uk*].

Then, the observation states set W is generated, i.e., W = {*What*, *you*, *thought*, *me*, *First*, *time*, *saw*, *class*, *Are*, *unique*, *enough*, *Find*, *30th*, *August*, *www*, *areyouunique*, *co*, *uk*}. Each word in W appears in

ham and spam SMS sets with a certain frequency. These occurrence frequencies will be used to obtain the observation probability distribution.

In addition, the refined two-word sequences combine together to form the observation sequence $\underline{X} = \{o_1, o_2, o_3, \cdots, o_N\}$. In this example, $\underline{X}$ = {*What, you, thinked, me, First, time, you, saw, me, class, Are, you, unique, enough, Find, 30th, August, www, areyouunique, co, uk*}. It is obvious that the original word order is kept.

Therefore, the observation sequence $\underline{X} = \{o_1, o_2, o_3, \cdots, o_N\}$ can be represented as

$$\begin{array}{c} \underline{X} = \Big\{ w_1^{SMS_1}, w_2^{SMS_1}, w_3^{SMS_1}, \cdots, w_{Lengthof SMS_1}^{SMS_1}, \\ \cdots, \\ w_1^{SMS_i}, w_2^{SMS_i}, w_3^{SMS_i}, \cdots, w_{Lengthof SMS_i}^{SMS_i}, \\ \cdots, \\ w_1^{SMS_M}, w_2^{SMS_M}, w_3^{SMS_M}, \cdots, w_{Lengthof SMS_M}^{SMS_i} \Big\} \end{array} \tag{2}$$

Please note that different SMSs may have different lengths. All sequential words in each SMS combine together to form the final observation sequence $\underline{X}$.

3.3. Label Each Word in Observation Sequence for HMM Learning

Among these training SMS from the UCI repository dataset, some of them are labeled as spam and the others are labeled as ham. The labeled SMS dataset can be described as:

$$\begin{array}{cccc} SMS_1 & w_1^{SMS_1}, w_2^{SMS_1}, w_3^{SMS_1}, \cdots, w_{Lengthof SMS_1}^{SMS_1} & ham \\ SMS_2 & w_1^{SMS_2}, w_2^{SMS_2}, w_3^{SMS_2}, \cdots, w_{Lengthof SMS_2}^{SMS_2} & spam \\ \cdots & \cdots & \cdots \\ SMS_i & w_1^{SMS_i}, w_2^{SMS_i}, w_3^{SMS_i}, \cdots, w_{Lengthof SMS_i}^{SMS_i} & ham \\ \cdots & \cdots & \cdots \\ SMS_M & w_1^{SMS_M}, w_2^{SMS_M}, w_3^{SMS_M}, \cdots, w_{Lengthof SMS_M}^{SMS_i} & spam \end{array} \tag{3}$$

Take a look at the instances above again. The SMS, *What you thinked about me. First time you saw me in class.*, is labeled as *ham* and the other one, *Are you unique enough? Find out from 30th August. www.areyouunique.co.uk*, is labeled as spam. Thus, it is represented as [*What, you, thinked, me, First, time, you, saw, me, class*] with the label *ham* and [*Are, you, unique, enough, Find, 30th, August, www, areyouunique, co, uk*] with the label *spam*.

However, each word in the observation sequence should be marked as ham or spam for the HMM learning. As the UCI repository dataset only has a label for each SMS, we use a compromised method to label all words in the observation sequence, i.e., labeling the words in the SMS based on the label of the SMS. The labels for the observation sequence can be represented as:

$$\begin{array}{c} \{ham, ham, ham, \cdots, ham, \\ \cdots, \\ ham, ham, ham, \cdots, ham, \\ \cdots, \\ spam, spam, spam, \cdots, spam\} \end{array} \tag{4}$$

For the instances above, as the first SMS is a ham one, each word in the refined sequence is labeled as *ham*. As the second SMS is a spam one, each word in the sequence is labeled as *spam*. That is, the part of observation sequence [*What*, *you*, *thought*, *me*, *First*, *time*, *you*, *saw*, *me*, *class*] is labeled as [*ham*, *ham*, *ham*, *ham*, *ham*, *ham*, *ham*, *ham*, *ham*, *ham*] and [*Are*, *you*, *unique*, *enough*, *Find*, *30th*, *August*, *www*, *areyouunique*, *co*, *uk*] is labeled as [*spam*, *spam*, *spam*, *spam*, *spam*, *spam*, *spam*, *spam*, *spam*, *spam*, *spam*].

3.4. Observation Probability Distribution

We calculate the probability of each observation state, i.e., each word in W, appearing in ham and spam SMS sets. The probability distribution is depicted in Figure 3. The word order in W is fixed for easy comparison, as shown in Figure 3.

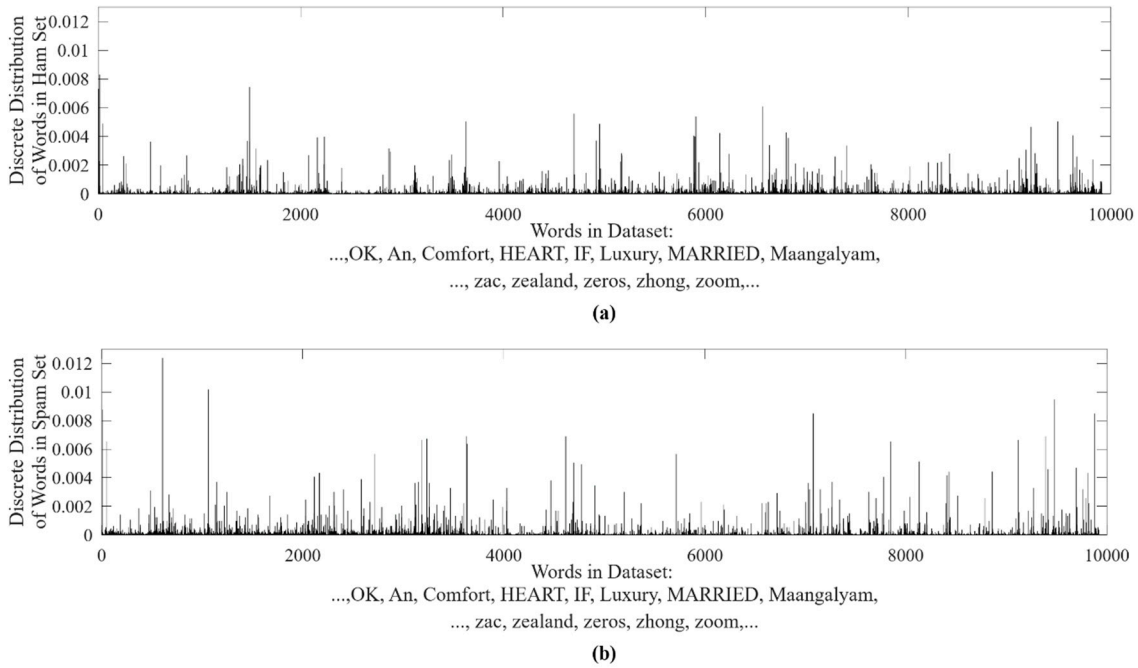

Figure 3. (**a**) Observation state distribution in ham messages set; (**b**) observation state distribution in spam messages set.

The higher the frequency of a certain word, the higher the probability in the distribution. As the word order in W is fixed in Figure 3, we can compare the probability of each word in ham and spam sets visibly. It is found that:

- As some words only have probability in a single dataset and their probability is equal to zero in another dataset, this indicates that these words only appear in the spam messages set or ham messages set;
- As the probabilities of many words are quite different in different datasets, it is referred that these words appear in both sets with much different word frequencies;
- Only a very small portion of them appear in both sets evenly.

It is true that the BoW also took advantage of this information to design term weights algorithms. However, the TF algorithm does not work well in the SMS scenario, because of the shortage of term occurrence.

In this paper, we first calculate two observation state distributions in spam and ham subsets. The two distributions are combined together to form the initial value of the HMM observation probability distribution matrix, $B = (b_{ij})_{2\times n}$.

3.5. HMM Learning

The Baum–Welch algorithm [15] is typically used for finding HMM parameters $\lambda = (\pi, A, B)$. That is, given HMM with initial parameters

$$\lambda_0 = (\pi_0,\ A_0,\ B_0) \tag{5}$$

and observation sequence

$$\underline{X} = \{o_t\}_{t=1}^{N} \tag{6}$$

to update $\lambda = (\pi, A, B)$ iteratively and find parameters that maximize the likelihood of observed data, i.e.,

$$\underset{\lambda}{\operatorname{argmax}} p\left(\{o_t\}_{t=1}^{N} \middle| \lambda\right) \tag{7}$$

The initial parameters $\lambda_0 = (\pi_0, A_0, B_0)$ are initialized as:

$\pi_0 = [0.5, 0.5]$ indicates that the state can start from ham or spam with the same probability.

$A_0 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$ means each state transmission has the same probability.

$B_0 = (b_{ij})_{2\times n}$ infers the initial observation probability distribution calculated in Section 3.4.

The Baum–Welch algorithm starts with initial parameters and then repeatedly takes two steps: Expectation step (E-step) and maximization step (M-step) until convergence (i.e., the difference of log-likelihood is less than small number d), as shown in Figure 4.

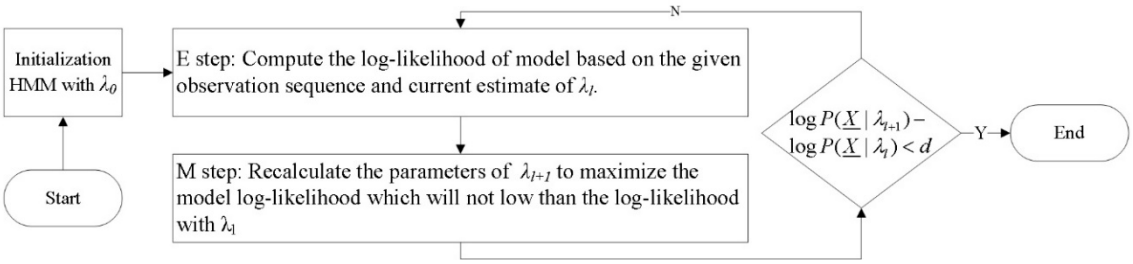

Figure 4. Baum–Welch algorithm.

3.6. SMS Property Prediction

Given the observation sequence and the trained hidden Markov model to find the optimal hidden state sequence, this is a typical decoding problem in HMM. The Viterbi algorithm [15] is applied to find the most likely hidden state sequence based on the input of each word sequence of the testing SMS. In formalization, we are given the testing observation sequence $\{o_t\}_{t=1}^{N}$ and trained HMM with parameters $\lambda = (\pi, A, B)$ to find the most likely state sequence. That is,

$$\underset{\{q_t\}_{t=1}^{N}}{\operatorname{argmax}}\, p\left(\{q_t\}_{t=1}^{N} \middle| \{o_t\}_{t=1}^{N}\right) \tag{8}$$

Via the Viterbi decoding algorithm, for each word sequence of the testing SMS, the optimal hidden state sequence is produced. The state sequence is the combination of ham and spam. The prediction of the SMS property is based on the majority role, i.e., an SMS will be labeled as ham if the optimal hidden state sequence has more hams than spams. Otherwise, the SMS will be labeled as spam.

3.7. The Workflow of the Discrete HMM for SMS Spam Detection

3.7.1. Data Preparation and HMM Learning

Step 1: Training an SMS dataset by first splitting or segmenting it into word sequences to keep their original order. Then, stop words are removed from the sequence and rest meaningful words form the observation sequence.

Step 2: Observation state probability distributions in ham and spam datasets are statistically analyzed and obtained.

Step 3: The compromised word label sequences are generated based on the labeled training SMS messages.

Step 4: The discrete HMM model is initialized as $\lambda_0 = (\pi_0, A_0, B_0)$. The initial parameters are given in Section 3.5.

Step 5: The hidden state sequence and the observation sequence feed the discrete HMM. The discrete HMM is optimized by the Baum–Welch algorithm until convergence.

The training process workflow is shown in Figure 5.

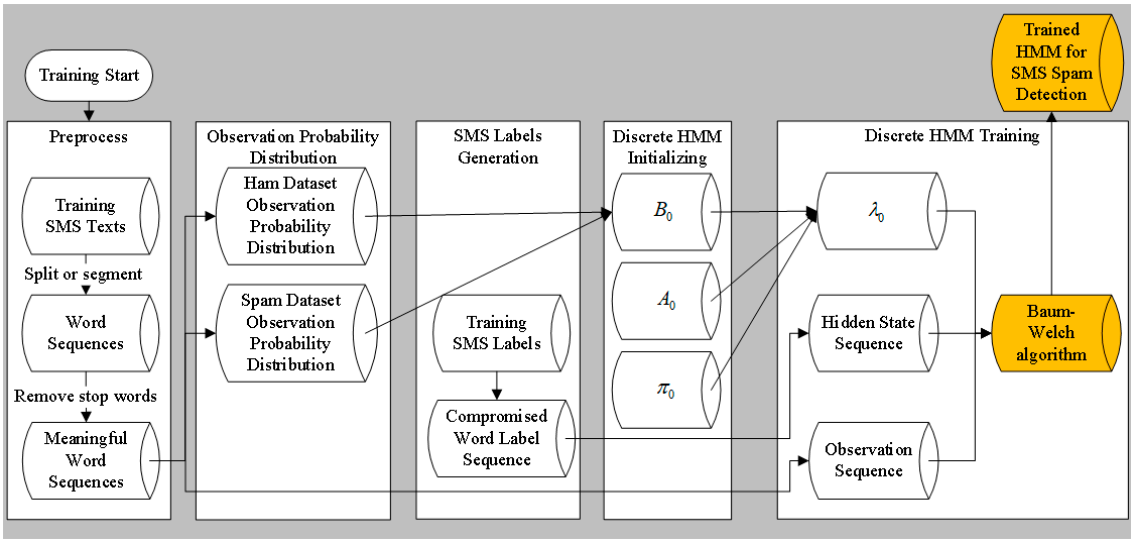

Figure 5. Training workflow.

3.7.2. SMS Classification

Given the trained HMM and observation sequence, the classification process involved finding out the optimal hidden state sequence and making a prediction of the SMS property based on the majority role. The SMS classification workflow is shown in Figure 6.

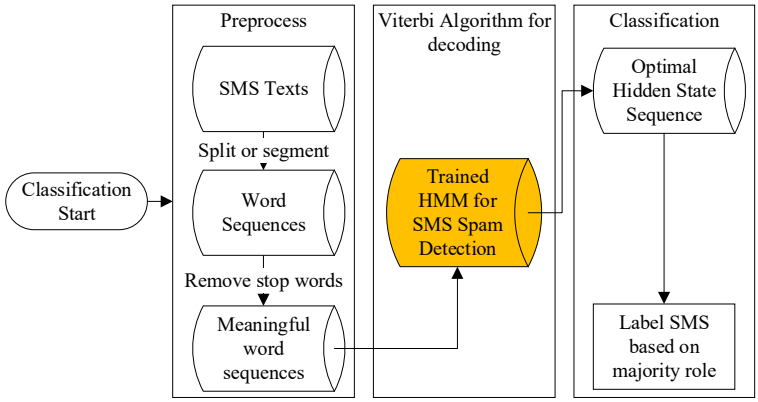

Figure 6. Short messaging service (SMS) classification workflow.

Step 1: Like the preprocess in the training workflow, the SMS dataset for classification is also first split or segmented into word sequences to keep their original word order. Then, stop words are removed from the sequence and the observation sequence is formed.

Step 2: Use the Viterbi decoding algorithm to find the optimal hidden state sequence for each SMS.

Step 3: Predict the SMS property based on the majority role.

4. Experiment Results and Discussion

4.1. Dataset and Analysis

This research uses the widely adopted UCI repository dataset for performance evaluations. This unbalanced dataset contains a total of 5574 English SMS messages, in which 747 SMS messages are spam and 4827 are ham, as shown in Table 1. These messages were collected from Grumbletext—a UK public forum (www.grumbletext.co.uk), the SMS corpus from the National University of Singapore, and Caroline Tagg's Ph.D. thesis [58].

Table 1. UCI dataset statistics.

	Number of SMS	Percentage of SMS
Spam	747	13.4%
Ham	4827	86.6%
Total	5574	100%

The experimental code is developed with Python 3.7 and Python packages including pyhanlp, pomegranate, and collections. The code runs on a MacBook with an Intel Core i7-7820 CPU and 16 GB of memory.

We first split the SMS, extract words, and remove the stop words. This resulted in 9955 meaningful words being extracted in total to form the set of observation states W. Then, we calculate the term frequency of each word in each SMS. The summary of the statistical results is shown in Table 2. We find that:

- 9272 words appear only once in a single SMS and accounts for 93.13%.
- The words that appear three times and above only account for 1.03% in total.

Table 2. Term frequency in spam and ham SMS set.

Term Frequency	Number of Words with the Same Term Frequency
1	9272
2	580
3	77
4	15
5	5
6	3
7	1
8–9	0
10	1
11–13	0
14	1

Apparently, most of the words only occur once in a single SMS. Therefore, the feature extraction algorithms, like TF.IDF, does not work well for SMS spam detection.

4.2. Evaluation Metrics

The well-known and persuasive evaluation metrics for classification are precision (P), recall (R), F-measure (F1), and accuracy (A) [59]. Their parameters for metrics calculation are shown in Table 3. Among these metrics, accuracy is the most important item [28].

Table 3. Metrics parameters.

Actual	**Predicted**	
	Negative	Positive
Negative	true negative (TN)	false negative (FN)
Positive	false positive (FP)	true positive (TP)

Precision (P) is the fraction of relevant instances among all retrieved instances.

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{9}$$

Recall (R), called Sensitivity, is the fraction of the total amount of relevant instances that are actually retrieved.

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{10}$$

F-measure (F1) is the harmonic mean of the precision and recall. It is a balance between precision and recall.

$$\text{F} - \text{Measure (F1)} = 2 \times \frac{P \times R}{P + R} \tag{11}$$

Accuracy (A) is the fraction of spam SMS messages that are correctly predicted among all SMS.

$$\text{Accuracy (A)} = \frac{TN + TP}{TN + TP + FN + FP} \tag{12}$$

Area under the curve (AUC) is also a well-known criterion for classification. It is the average of the true positive rate (TPR) and false positive rate (FPR). The greater the AUC value, the more accurate the model.

$$\text{AUC} = \left(\left(\frac{TP}{TP + FN}\right) + \left(\frac{TN}{TN + FP}\right)\right)/2 \tag{13}$$

4.3. Result of the Discrete HMM on the UCI Repository Dataset

To evaluate the performance of the proposed HMM method, we split the UCI repository dataset into two datasets. One is the training dataset containing 66% (about 2/3) spam and 66% ham SMS, and the other one is the testing dataset containing the rest 34% SMS, as shown in Table 4.

Table 4. Training and testing dataset statistics.

	Training Dataset	**Testing Dataset**	**Total Number of SMS**
Spam	493	254	747
Ham	3186	1641	4827
Percentage	66%	34%	100%

The reason we divide the dataset in this way is that we want to compare the performance of the proposed HMM method with the best performance achieved by the CNN method in [11]. Roy et al. [11] used the same UCI dataset and divided the database as 2/3 for training and 1/3 for testing in their experiment.

Following the procedures described in Section 3.7, the confusion matrix is shown in Table 5.

Table 5. Confusion matrix of the results.

Dataset	Actual	Predicted		Prediction %		
		Spam	Ham	Spam	Ham	AUC
The proposed HMM	Spam	222	50	0.892	0.031	0.900
	Ham	27	1559	0.108	0.969	

The evaluation results are shown in Table 6.

Table 6. Evaluation results.

Dataset	Class	Accuracy (A)	Precision (P)	Recall (R)	F-Measure (F1)
The proposed HMM	Spam	0.959	0.892	0.816	0.852
	Ham		0.969	0.983	0.976

To illustrate the performance of the proposed HMM method, we compare the results with those obtained by other machine learning models, including naïve Bayes (NB), support vector machine (SVM), non-negative matrix factorization (NMF), latent Dirichlet allocation (LDA), long short-term memory (LSTM), and convolutional neural network (CNN). The result comparison is shown in Table 7. As we pointed out before, all methods listed in Table 7 are evaluated by the same UCI dataset. Results of NB, SVM, NMF, and LDA methods were presented by Nagwani and Sharaff [56]. The results of LSTM and CNN were presented by Roy et al. [11]. The highest accuracy is achieved by the CNN method. The proposed HMM method has slightly better accuracy than LSTM. It is compatible with the CNN method.

Table 7. Summary of the experimental results with different models.

Model	Class	Accuracy (A)	Precision (P)	Recall (R)	F-Measure (F1)
NB [56]	overall	0.842	0.95	0.972	0.87
SVM [56]	overall	0.936	0.97	0.977	0.94
NMF [56]	overall	0.917	0.96	0.976	0.92
LDA [56]	overall	0.904	0.96	0.976	0.92
LSTM [11]	Spam	0.953	0.849	0.777	0.811
	Ham		0.972	0.976	0.973
CNN [11]	Spam	0.979	0.988	0.858	0.922
	Ham		0.982	0.996	0.988
The proposed HMM	**Spam**	**0.959**	**0.892**	**0.816**	**0.852**
	Ham		**0.969**	**0.983**	**0.976**

4.4. Result of the Discrete HMM on Other SMS Dataset in Chinese

The HMM presented in this paper can be applied to other languages. In order to evaluate its performance in different languages, we apply the proposed HMM method on a Chinese SMS dataset containing 2000 SMS messages. The dataset is derived from the production environment of our cooperated SMS service company that provides the SMS service in the East China area.

We choose 700 spam SMS and 700 ham SMS from the Chinese SMS dataset as the training set and the rest 300 spam SMS and 300 ham as the testing set. The statistics of the training and testing datasets are shown in Table 8.

Table 8. Training and testing dataset statistics.

	Training Dataset	Testing Dataset	Total Number of SMS
Spam	700	300	1000
Ham	700	300	1000
Percentage	70%	30%	100%

The confusion matrix turned out to be that shown in Table 9.

Table 9. Confusion matrix of the results.

Dataset	Actual	Predicted		Prediction %		
		Spam	Ham	Spam	Ham	AUC
The proposed HMM	Spam	293	7	0.977	0.023	0.985
	Ham	2	298	0.007	0.993	

The evaluation results of the proposed HMM are shown in Table 10.

Table 10. Evaluation results.

Dataset	Class	Accuracy (A)	Precision (P)	Recall (R)	F-Measure (F1)
The proposed HMM	Spam	0.985	0.977	0.993	0.985
	Ham		0.993	0.977	0.985

4.5. Result Discussions

4.5.1. UCI Repository Dataset Results

We use 2/3 dataset for training and 1/3 dataset for testing, which has the same division ratio of the paper [11]. The model classified spam and ham SMS with an excellent accuracy of 0.959, which is better than those of NB, SVM, NMF, LDA, and LSTM. In addition, the model performs even better in ham classification with precision 0.969, recall 0.983, and F1 0.976. Although the spam performance is a little under expectation with precision 0.892, recall 0.816, and 0.852, it still performs better than that of LSTM.

The performance could be better if the HMM model is trained enough. English words have many different forms. For verbs, they have past tense, present tense, future tense, and third-person singular forms. A similar situation happens in nouns and adjectives. Therefore, compared to the many English words, the total of 9955 words are not sufficient for observation states in the experiment, and words in the training dataset are less likely to reappear in the testing dataset. As a result, the spam classification performance is affected as the model does not know how to label the untrained words.

In addition, LSTM and CNN are very complex models that consume computer resources greatly. The HMM model proposed in this paper is relatively simple, which has fast training and predicting speed. Thus, our model is easy to implement in commercial applications to process other languages.

4.5.2. Chinese SMS Dataset Results and Its Non-Language-Sensitivity

Apparently, based on the experimental results, the proposed HMM works better in classifying Chinese SMS messages. Especially, it performed the best by securing a remarkable accuracy of 98.5% to classify spam and ham SMS. Compared to the experimental results on English SMS messages, the division ratio of training and testing SMS is similar, but the results are obviously much better.

The reason lies in English and Chinese languages themselves. In English, many words have different forms. As these synonyms have not merged in this research, the different forms of words are treated as different observation states. On the contrary, Chinese words never change. Compared to English, the words in the Chinese SMS training dataset are more likely to reappear in the testing dataset. Therefore, the HMM introduced in this paper works better in Chinese SMS spam detection.

Furthermore, the LSTM and CNN models presented in the paper [11] were dependent on SMS written in English only. However, the HMM present in this paper may be implemented widely in the future because it is not language-sensitive.

5. Conclusions and Future Work

This paper proposed a discrete hidden Markov model for SMS spam detection, and it is the first research taking advantage of word order information to detect spam SMS. Compared to other traditional and even novel machine learning models, the proposed HMM method scored excellent results among them. In addition, HMM is a relatively simple machine learning model. It can be implemented into the spam filtering industry to meet the huge throughput requirement. In addition, this paper resolved the issue in which the traditional feature extraction algorithm, like TF.IDF, does not work well for SMS spam detection, due to the extremely low term frequency. The proposed HMM is not language-sensitive, which was also validated on Chinese SMS spam detection. The overall performance of the proposed HMM is better on the Chinese dataset than the English dataset.

The proposed HMM still has limitations. The accuracy depends highly on the size of the training set. The bigger the training set, the more likely SMS words will reoccur in the testing dataset and the better the achieved accuracy. In our future research, we will provide an improved HMM model to make it suitable to a small training set scenario. In addition, as there is no training dataset that has a label for each word of the SMS, each word was labeled based on the property of the SMS in this research. This compromised word labeling method also affected the spam classification accuracy. We will tackle this issue by applying artificial neural networks. Furthermore, the other HMM variants will be explored for SMS spam detection.

Author Contributions: T.X. and X.C. conceived the idea and developed the algorithm; T.X. developed the experimental software and performed the data analysis. T.X. prepared the manuscript; T.X. and X.C. wrote and made revisions to the paper. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Soft Engineering of Key Subjects Construction in Shanghai Polytechnic University, grant number xxkzd1604 and US National Science Foundation, grants number CNS-1801811.

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

References

1. *PortioResearch Worldwide A2P SMS Markets 2014–2017: Understanding and Analysis of Application to-Person Text Messaging Markets Worldwide*; Portio Research Limited: Chippenham, UK, 2014.
2. Ezpeleta, E. Short Messages Spam Filtering Combining Personality Recognition and Sentiment Analysis. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2017**, 175–189. [CrossRef]
3. Statista A2P and P2P SMS Market Revenue Worldwide from 2017 to 2022 (in Billion U.S. Dollars). Available online: https://www.statista.com/statistics/485153/a2p-sms-market-size-worldwide/ (accessed on 9 July 2020).
4. Abdulhamid, S.M.; Abd Latiff, M.S.; Chiroma, H.; Osho, O.; Abdul-Salaam, G.; Abubakar, A.I.; Herawan, T. A Review on Mobile SMS Spam Filtering Techniques. *IEEE Access* **2017**, *5*, 15650–15666. [CrossRef]
5. Arutyunov, V.V. Spam: Its past, present, and future. *Sci. Tech. Inf. Process.* **2013**, *40*, 205–211. [CrossRef]
6. Jiang, L.; Li, C.; Wang, S.; Zhang, L. Deep feature weighting for naive Bayes and its application to text classification. *Eng. Appl. Artif. Intell.* **2016**, *52*, 26–39. [CrossRef]
7. Sable, A.S.; Kalavadekar, P.N. SMS Classification Based on Naive Bayes Classifier and Semi-Supervised Learning. *Int. J. Mod. Trends Eng. Res.* **2016**, *3*, 16–25.

8. Waheeb, W.; Ghazali, R. Content-based SMS Classification: Statistical Analysis for the Relationship between Number of Features and Classification Performance. *Comput. Y Sist.* **2017**, *21*, 771–785. [CrossRef]
9. Tekerek, A. Support vector machine based spam SMS detection. *J. Polytech.* **2018**, *0900*, 779–784. [CrossRef]
10. Poomka, P.; Pongsena, W.; Kerdprasop, N.; Kerdprasop, K. SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit. *Int. J. Futur. Comput. Commun.* **2019**, *8*, 12–15. [CrossRef]
11. Roy, P.K.; Singh, J.P.; Banerjee, S. Deep learning to filter SMS Spam. *Future Gener. Comput. Syst.* **2020**, *102*, 524–533. [CrossRef]
12. Serkan, B.; Onur, K. Development of content based SMS classification application by using Word2Vec based feature extraction. *IET Softw.* **2018**, *13*, 295–304.
13. Barushka, A.; Hajek, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Appl. Intell.* **2018**, *48*, 3538–3556. [CrossRef]
14. Xia, T.; Chai, Y. An improvement to TF: Term distribution based term weight algorithm. *J. Softw.* **2011**, *6*, 413–420. [CrossRef]
15. Rabiner, L.R.; Juang, B.H. An Introduction to Hidden Markov Models. *IEEE Assp Mag.* **1986**, *3*, 4–16. [CrossRef]
16. Eddy, S.R. What is a hidden Markov model? *Nat. Biotechnol.* **2004**, *22*, 1315–1316. [CrossRef]
17. Group, T.A.S. The First Enterprise Open-Source Spam Filter. Available online: http://spamassassin.apache.org/ (accessed on 2 June 2020).
18. Ruano-Ordás, D.; Fdez-Glez, J.; Fdez-Riverola, F.; Méndez, J.R. Effective scheduling strategies for boosting performance on rule-based spam filtering frameworks. *J. Syst. Softw.* **2013**, *86*, 3151–3161. [CrossRef]
19. Wang, Y.H.; Wu, I.C. Wirebrush4SPAM: A novel framework for improving efficiency on spam filtering services. *Softw. Pract. Exp.* **2009**, *39*, 701–736. [CrossRef]
20. Xia, T. A Constant Time Complexity Spam Detection Algorithm for Boosting Throughput on Rule-Based Filtering Systems. *IEEE Access* **2020**, *8*, 82653–82661. [CrossRef]
21. Aragão, M.V.C.; Frigieri, E.P.; Ynoguti, C.A.; Paiva, A.P. Factorial design analysis applied to the performance of SMS anti-spam filtering systems. *Expert Syst. Appl.* **2016**, *64*, 589–604. [CrossRef]
22. Ebadati, O.M.E.; Ahmadzadeh, F. Classification Spam Email with Elimination of Unsuitable Features with Hybrid of GA-Naive Bayes. *J. Inf. Knowl. Manag.* **2019**, *18*, 1–19. [CrossRef]
23. Arifin, D.D.; Shaufiah; Bijaksana, M.A. Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier. In Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, Indonesia, 13–15 September 2016; pp. 80–84. [CrossRef]
24. Santos, I.; Laorden, C.; Sanz, B.; Bringas, P.G. Expert Systems with Applications Enhanced Topic-based Vector Space Model for semantics-aware spam filtering. *Expert Syst. Appl.* **2012**, *39*, 437–444. [CrossRef]
25. Chan, P.P.K.; Yang, C.; Yeung, D.S.; Ng, W.W.Y. Spam filtering for short messages in adversarial environment. *Neurocomputing* **2015**, *155*, 167–176. [CrossRef]
26. Zhang, W.; Bu, C.; Yoshida, T.; Zhang, S. CoSpa: A co-training approach for spam review identification with support vector machine. *Information* **2016**, *7*, 12. [CrossRef]
27. Zhang, W.; Bu, C.; Yoshida, T.; Zhang, S. CoFea: A novel approach to spam review identification based on entropy and co-training. *Entropy* **2016**, *18*, 429. [CrossRef]
28. Gashti, M.Z. Detection of Spam Email by Combining Harmony Search Algorithm and Decision Tree. *Eng. Technol. Appl. Sci. Res.* **2017**, *7*, 1713–1718.
29. Uysal, A.K.; Gunal, S.; Ergin, S.; Gunal, E.S. The Impact of Feature Extraction and Selection on SMS Spam Filtering. *Elektronika ir Elektrotechnika* **2013**, *19*, 67–72. [CrossRef]
30. Karthika, R.D.; Visalakshi, P. Latent Semantic Indexing Based SVM Model for Email Spam Classification. *J. Sci. Ind. Res.* **2014**, *73*, 437–442.
31. Chandra, A. Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory. In Proceedings of the 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 21–22 November 2019; pp. 118–122.
32. Yang, H.; Liu, Q.; Zhou, S.; Luo, Y. A spam filtering method based on multi-modal fusion. *Appl. Sci.* **2019**, *9*, 1152. [CrossRef]
33. Zhao, C.; Xin, Y.; Li, X.; Yang, Y.; Chen, Y. A Heterogeneous Ensemble Learning Frameworkfor Spam Detection in Social Networks with Imbalanced Data. *Appl. Sci.* **2020**, *10*, 936. [CrossRef]

34. Sheikhi, S.; Kheirabadi, M.T.; Bazzazi, A. An Effective Model for SMS Spam Detection Using Content-based Features and Averaged Neural Network. *Int. J. Eng.* **2020**, *33*, 221–228. [CrossRef]
35. Liu, J.; Yuan, X. Spam Short Messages Detection via Mining Social Networks. *J. Comput. Sci. Technol.* **2012**, *27*, 506–514. [CrossRef]
36. Saleh, A.J.; Karim, A.; Shanmugam, B.; Azam, S.; Kannoorpatti, K.; Jonkman, M.; De Boer, F. An intelligent spam detection model based on artificial immune system. *Information* **2019**, *10*, 209. [CrossRef]
37. Shang, Y. Consensus of Hybrid Multi-Agent Systems with Malicious Nodes. *IEEE Trans. Circuits Syst. Ii Express Briefs* **2020**, *67*, 685–689. [CrossRef]
38. Mousas, C.; Anagnostopoulos, C.N. Real-time performance-driven finger motion synthesis. *Comput. Graph.* **2017**, *65*, 1–11. [CrossRef]
39. Mousas, C. Full-body locomotion reconstruction of virtual characters using a single inertial measurement unit. *Sensors* **2017**, *17*, 2589. [CrossRef] [PubMed]
40. Nakagawa, S.; Zhang, W. Text-independent speaker recognition by speaker-specific GMM and speaker adapted syllable-based HMM. In Proceedings of the EUROSPEECH 8th European Conference on Speech Communication and Technology, Geneva, Switzerland, 1–4 September 2003; pp. 3017–3020.
41. Niina, G.; Dozono, H. The Spherical Hidden Markov Self Organizing Map for Learning Time Series Data. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2012, Lausanne, Switzerland, 11–14 September 2012; pp. 563–570.
42. Okhovvat, M.; Minaei, B. Procedia Computer A Hidden Markov Model for Persian Part-of-Speech Tagging. *Procedia Comput. Sci.* **2011**, *3*, 977–981. [CrossRef]
43. Ptaszynski, M.; Momouchi, Y. Expert Systems with Applications Part-of-speech tagger for Ainu language based on higher order Hidden Markov Model. *Expert Syst. Appl.* **2012**, *39*, 11576–11582. [CrossRef]
44. Zhang, J.; Shen, D.; Zhou, G.; Su, J.; Tan, C. Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *J. Biomed. Inform.* **2004**, *37*, 411–422. [CrossRef]
45. Hussain, N.; Mirza, H.T.; Rasool, G.; Hussain, I.; Kaleem, M. Spam review detection techniques: A systematic literature review. *Appl. Sci.* **2019**, *9*, 987. [CrossRef]
46. Abayomi-Alli, O.; Misra, S.; Abayomi-Alli, A.; Odusami, M. A review of soft techniques for SMS spam classification: Methods, approaches and applications. *Eng. Appl. Artif. Intell.* **2019**, *86*, 197–212. [CrossRef]
47. Rafique, M.; Farooq, M. SMS Spam Detection by Operating on Byte-Level Distributions Using Hidden Markov Models (HMMs). In Proceedings of the 20th Virus Bulletin International Conference, Vancouver, BC, Canada, 29 September–1 October 2010.
48. Gordillo, J.; Conde, E. An HMM for detecting spam mail. *Expert Syst. Appl.* **2007**, *33*, 667–682. [CrossRef]
49. Ebrahimi, N.; Trabelsi, A.; Islam, M.S.; Hamou-Lhadj, A.; Khanmohammadi, K. An HMM-based approach for automatic detection and classification of duplicate bug reports. *Inf. Softw. Technol.* **2019**, *113*, 98–109. [CrossRef]
50. Washha, M.; Qaroush, A.; Mezghani, M.; Sedes, F. A Topic-Based Hidden Markov Model for Real-Time Spam Tweets Filtering. *Procedia Comput. Sci.* **2017**, *112*, 833–843. [CrossRef]
51. Ganesan, V.; Manikandan, M.s.K.; Suresh, M.N. Detection and prevention of spam over Internet telephony in Voice over Internet Protocol networks using Markov chain with incremental SVM. *Int. J. Commun. Syst.* **2016**, *30*, e3255. [CrossRef]
52. Almeida, T.A.; Gomez Hidalgo, J.M.; Silva, T.P. Towards SMS Spam Filtering: Results under a New Dataset. *Int. J. Inf. Secur. Sci.* **2012**, *2*, 1–18.
53. Adewole, K.S.; Anuar, N.B.; Kamsin, A.; Sangaiah, A.K. SMSAD: A framework for spam message and spam account detection. *Multimed. Tools Appl.* **2019**, *78*, 3925–3960. [CrossRef]
54. Rahmani, H.; Sahli, N.; Kamoun, F. Simple SMS spam filtering on independent mobile phone. *Int. J. Secur. Commun. Netw.* **2012**, *5*, 1209–1220. [CrossRef]
55. 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]
56. Nagwani, N.K.; Sharaff, A. SMS spam filtering and thread identification using bi-level text classification and clustering techniques. *J. Inf. Sci.* **2017**, *43*, 75–87. [CrossRef]
57. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the study of SMS spam filtering: New Collection and Results. In Proceedings of the 11th ACM Symposium on Document Engineering, Mountain View, CA, USA, 19–22 September 2011; pp. 259–262. [CrossRef]

58. Tagg, C. A Corpus Linguistic Study of SMS Texting. Ph.D. Thesis, University of Birmingham, Birmingham, UK, 2009.
59. Forman, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification George. *J. Mach. Learn. Res.* **2003**, *1*, 1289–1305. [CrossRef]