Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-018-0157-5

ORIGINAL RESEARCH

Optimizing semantic LSTM for spam detection

Gauri Jain[1] · Manisha Sharma[2] · Basant Agarwal[3]

Received: 11 August 2017 / Accepted: 4 April 2018

Abstract Classifying spam is a topic of ongoing research in the area of natural language processing, especially with the increase in the usage of the Internet for social networking. This has given rise to the increase in spam activity by the spammers who try to take commercial or non-commercial advantage by sending the spam messages. In this paper, we have implemented an evolving area of technique known as deep learning technique. A special architecture known as Long Short Term Memory (LSTM), a variant of the Recursive Neural Network (RNN) is used for spam classification. It has an ability to learn abstract features unlike traditional classifiers, where the features are hand-crafted. Before using the LSTM for classification task, the text is converted into semantic word vectors with the help of word2vec, WordNet and ConceptNet. The classification results are compared with the benchmark classifiers like SVM, Naïve Bayes, ANN, k-NN and Random Forest. Two corpuses are used for comparison of results: SMS Spam Collection dataset and Twitter dataset. The results are evaluated using metrics like Accuracy and F measure. The evaluation of the results shows that LSTM is able to outperform traditional machine learning methods for detection of spam with a considerable margin.

Keywords Spam detection · Machine learning · Recursive neural network (RNN) · Long Short Term Memory (LSTM) · Deep learning · Twitter spam · SMS spam

✉ Gauri Jain
jain.gauri@gmail.com

Manisha Sharma
manishasharma8@gmail.com

Basant Agarwal
basant@skit.ac.in

1 Department of Computer Science, Banasthali Vidyapith, Banasthali, India

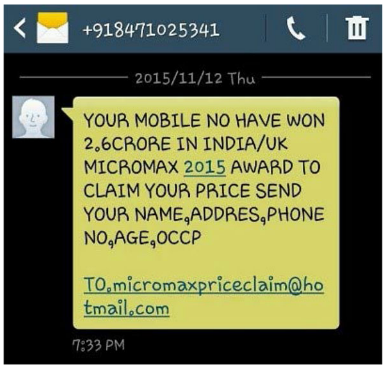
2 Department of Computer Science, Banasthali Vidyapith, Jaipur, India

3 Department of Computer Science and Engineering, SKIT, Rajasthan Technical University, Jaipur, India

1 Introduction

With the ease and increase in the use of smart phones the frequency of web surfing by an individual has increased. It has also changed the way people express themselves and interact on the web. The most popular medium used in exchanging the data/message are messages via mobile phones and social networking sites like Twitter, Facebook etc. It has given rise to the increase in the volume of data that is exchanged over the network which has resulted in people sending unwanted spams messages trying to take advantage by invading their privacy. Initially, spams started spreading with email spam. According to M3AAWG report, abusive email content amounts to 87.1–90.2% of the total email content [1]. This has resulted in the financial strain, increased requirement of storage, spreading of offensive material like pornographic content and above all it has violated the privacy of the people at the receiving end. The source of sending spams through mobile are bulk SMS messaging services, which allows the companies to send the messages at a very economic rates. An example of spam SMS is shown in Fig. 1.

Micro blogging sites like Twitter have much deeper reach due to the use of mobile applications on smart phones. The tweets are public by default, openly accessible by everyone and can be posted with the help of Twitter account. The spammers often create automated fake


+918471025341

2015/11/12 Thu

YOUR MOBILE NO HAVE WON 2.6CRORE IN INDIA/UK MICROMAX 2015 AWARD TO CLAIM YOUR PRICE SEND YOUR NAME,ADDRES,PHONE NO,AGE,OCCP

TO.micromaxpriceclaim@hotmail.com

7:33 PM

Fig. 1 Screenshot of a spam message

accounts called spam bots which keep on posting tweets automatically. Spammers on Twitter are detected by exploiting content level features like presence of url, presence of average no. of hashtags, tweets having spam words, age of the user account etc. [2, 3].

Due to steady increase in the number of spams on social network, companies and researchers have sought ways to stop them or avoid them. For e.g. a message marked as spam on our mobile phone will be added to the spam numbers list. On Twitter, one can mark a user as spammer as spam by tweeting "@spam @username". There are methods that take into account the weights of the tokens in the text for classification of text in different categories. One such approach is discussed by Mittal for classification of sentiments on Twitter [4]. These methods manual labelled the text after user had received it.

Spam Detection is a fairly mature research area. Early work in the area of development of spam detection model was content based spam identification that was dependent on the extracting and defining the rules that constitute the spam messages from the content [5]. Most of the later work was carried out using traditional machine learning classifiers like SVM, Naïve Bayes, Decision trees, Random Forest, etc. These classifiers were dependent on the handcrafted features extracted from the training data. Agarwal discussed various feature selection techniques for classification task [6]. As in an overview paper written by Khorsi summarized most of the statistical techniques used to filter spam emails [7]. He concluded that no technique may be alone sufficient for fighting with the spams since they have their own disadvantages. Other significant work on traditional spam classifier are discussed in [8–10].

Ntoulas has used various heuristics like no. of words in the page, no. of words in the page title, average length of the words, amount of anchor text, fraction of visible content, compressibility, fraction of page drawn from the globally popular words, independent n-gram likelihoods, conditional n-gram likelihoods on the contents of the web pages [11]. These heuristics were combined with various classifiers for identification of spams. In 2011, Mccord and Chuah used various content based methods along with various statistical classifiers to detect spams on microblogging site Twitter [12].

We have proposed a deep learning approach which is an evolving technique in the area of classification. Deep learning architectures in general have multiple hidden implementation layers between input layer and output layer. One of the very effective deep learning architecture is Recurrent Neural Network (RNN). We have used a special type of RNN called long short term memory (LSTM) for building a model which proved to be very effective for classifying spams. LSTM is similar to normal neural network with added self-loops that helps to retain and pass the error information to the next layer and the number of hidden layers or the processing layers is very large. The use of LSTM helps to memorize the information up to many layers as it has the capability to choose which information to discard and which information to store and pass it on to the next layer, unlike normal neural network where processing is independent of the previous layers.

The experiments are performed on benchmark SMS spam dataset, available in UCI repository [13] and Tweets dataset, extracted from public live tweets from the microblogging site Twitter using Twitter API. The Twitter dataset is manually labelled. The research done in this paper studies the adaptation of spam classifiers where we have trained the classifier with already labelled spam and ham messages and then deployed the trained model to detect the spam messages on our testing corpus.

2 Background and related work

Until now, most of the research on spam detection was around traditional statistical machine learning techniques. Recently researchers realized that for each type of media (including text, image etc.) handcrafted features have to be extracted which is a complex and redundant task. In around 2006, the idea of Deep Learning started to take shape. The major advantage of Deep Learning technology is that they consists of set of algorithms that can learn high level features from the training data. This gave an edge over the traditional machine learning based spam detection methods. LeCun gave an insight about deep learning with special reference to convolution network in [14]. They shed light on the way the convolutional networks are helpful in processing large datasets especially images, audio and video data. Bengio in his monograph discussed in depth about how the data can be represented into abstract features, difficulty in successfully training deep architectures with various optimization methods. The monograph also focused on deep neural networks and deep graphical model architectures [15]. Deng [16], in his overview paper summarizes architectures, algorithms and application of Deep Learning. He discussed about deep autoencoders, deep stacking networks and deep neural networks in the area of signal and information processing of audio/speech, image/vision, language modeling, natural language processing and information retrieval. Hochreiter, discussed how neural networks can be trained to store information over long period of time using LSTM which enforce constant error flow [17]. This was not achieved by normal recurrent neural network. Deep learning had already given very good results in the area of sentiment analysis and domain adaptation of online reviews and recommendation by Glorot [18] and twitter sentiment classification [19], speech recognition [20]. Apart from statistical methods, deep learning approach was used by Xavier Glorot for domain adaptation for sentiment classification where Stacked Denoising Auto-Encoders with sparse rectifier units perform an unsupervised feature extraction [18]. Hong predicted the polarity of a movie review as positive or negative using deep RNN [21].

Work has already started in the area of spam detection using Deep Learning algorithms by various researchers. Tzortzis have performed spam detection on five benchmark spam email databases using Deep Belief Network (DBN) and found that the results are better when compared to traditional classification methods [22]. Mi and Gao applied Stacked Auto Encoder, one of the main type of Deep Learning method to various already existing spam corpus and found that they perform better than many of the existing machine learning methods like SVM, Naïve Bayes etc. and that too without any feature engineering [23]. Jain has applied another important architecture, CNN on the semantically pre-trained tweets to classify the short text as spam or ham [24].

Currently, there is no known attempt of classifying spams using LSTM architecture. But the use of LSTM is well known in the area of sentiment analysis, movie review classification etc. Hong and Fang predicted the polarity of a movie review as positive or negative using LSTM in [21]. Different LSTM models are also used for semantic relatedness and sentiment analysis and found that the results outperform the existing benchmark results [25]. Tang used gated RNN for sentiment classification on IMDB movie review dataset [26]. LSTM is also used for printed text recognition as in [27], the author used it for printing Urdu script. The bi-directional LSTM has also shown good results for emotion recognition through speech and facial expression [28]. The area of Deep Learning is comparatively new, and researchers are still working on it. Therefore, further work using LSTM in various fields is still expected.

3 Traditional machine learning techniques

Some of the most popular traditional machine learning methods that are used as benchmark techniques are discussed below:

3.1 Support vector machine (SVM)

Support vector machine (SVM) is based on non-probabilistic supervised learning classification. The label of the test data is based on the features extracted from the training data. The features of the training data is represented in the form of n-dimensional vector consisting of a set of real numbers. SVM finds the most suitable hyperplane that separates the classes of data objects in the vector space with maximum difference between them.

3.2 Naïve Bayes (NB)

Naïve Bayes (NB) classifier is a supervised statistical learning algorithm based on Bayes' Theorem. It is a probabilistic classifier which classifies the data instance according to the frequency/probability of the feature vector and assumes that the features are not dependent on each other. The various methods to extract the feature vectors are BoW, Information Gain (IG) etc. The Bayes' Theorem on which the NB classifier is based on is shown in the equation below:

$$P(C|D) = \frac{P(D/C) * P(C)}{P(D)} \quad (1)$$

where C and D are events and P(D) $\neq$ 0.

P(D) and P(C) are the prior probabilities of observing D and C without regard to each other.

P(D|C) is the probability of observing event D given that C is true.

3.3 Artificial neural network (ANN)

ANN classifier works on the principle of supervised learning taking inspiration from the human brain. The ANN system consists of three layers: an input layer, a hidden layer and an output layer. The input layer consists of neurons, which receives input from several other neurons and are connected with hidden layer via edges. These edges stores the values called weights and help in manipulating and sending the output to the output layer. The connection between the neurons are dependent on the weight value. The more the weight, the more information is passed. An activation function helps in adjusting these weights of the units so that network leaning can be optimised.

3.4 k-Nearest neighbor (KNN)

k-Nearest Neighbor (KNN) is non-parametric supervised learning classifier based on the class of its neighbors. The KNN classifier works on the basis of k number of neighbors in its vicinity. The test data is assigned the label of the class of the most frequently occurring data among the k training samples nearest to the test data. For calculating the distance of the nearest neighbor, measure like Euclidean distance, Manhattan distance is considered.

3.5 Random forest (RF)

Random forest is the collection or forest of many decision trees. It is an ensemble methodology to build a classifier model. It combines various decision trees to enhance the results as compared to a single tree. The dataset is divided into sub-sets repeatedly and the decision trees are formed on the basis of various combinations of variables. These randomly generated trees forms a forest.

4 Deep learning approach

The main challenge of Artificial Intelligence is the ability of a machine to represent and model the data and information in a way our human brain does. Unlike traditional classification methods, which requires handcrafted features to be extracted and pre-processed so that the classification techniques can be applied, deep learning techniques learn features while going through multiple hidden layers. The deep network learns high level abstract features progressively. This is possible due to the passing information learned in the previous layers to the future layers.

4.1 Recurrent Neural Network (RNN)

The human brain has an ability to continuously learn and recalibrate the response. It is able to process and understand information because it can co–relate with the past learnings i.e. in order to understand the problem, it looks for the related information in the past. But, the same is not the case with traditional neural networks that claims to represent human brain. A normal neural network cannot reason with the events that have occurred in the past, as each computation layer of this type of network is independent and has no effect on each other. Thus, they are "stateless" and cannot learn the information from the past sequences which is their major drawback. Recurrent Neural Network (RNN) is a promising solution to tackle the problem of sequential learning in the traditional neural network. They have a unique property of saving the information while reading the sequence of input with each time step thus they are "stateful". Parameter sharing is an important property of RNN that helps to generalize the model to the input sequence of different lengths.

The basic structure of RNN is same as feed forward neural network with the difference in the connections between the neurons. Instead of unidirectional connections, where information flow between the neurons from one layer to another, the neurons are allowed to have directed cycles in the network. They can have self-loop or connections. The Fig. 2 shows RNN with the input units labelled as x_1, x_2, x_3… and the output units labelled as p_1, p_2, p_3…. In between these two units are the hidden units. Within the hidden units there are directed cycles as well as self-loops. The information flows between the hidden units back and forth and sometimes between output unit and the hidden unit.

4.2 Long short term memory (LSTM)

LSTM is a variant of Recurrent Neural Network (RNN). RNN works well in theory, but it has difficulty in remembering long term memories, i.e. as the distance between the current layer and the previous layers increases, the memory becomes weak. This is due to the backpropagation of error which requires the network to calculate the gradient for the updating weights. For this reason, RNNs face the problem of vanishing/exploding gradient (gradient becoming too small or too large). To solve this problem, Long Short Term Memory (LSTM) [18] is used instead of the RNN. LSTM network is able to remember the information for a longer period of time or for longer cycles without vanishing/exploding gradient problem. The basic structure of the LSTM is similar to that of RNN and it is also unfolded at each time step as shown in the Fig. 3. LSTM consists of repeating LSTM units having special memory cells that has ability to store information for the extended period of time without being degenerated.

The memory cell that has three logistic gates to control the output that goes to the next memory cell—the forget gate, the input gate, and output gate as shown in Fig. 4.

These gates do not send their activities as input, instead they set the weights on the connections between the neural network and the memory cell. The memory cell has a self-connection which works as follows:

When the forget gate has an activity of 1 i.e. it is turned on and the self-connection also has a weight 1, then the memory cell writes its contents to itself. When the forget gate is set to zero, the memory cell discards its contents. The input gate is set to 1, it allows the rest of the neural net to write into the memory cell and when the output gate is set to 1, the network can read from the memory cell. With the help of these three cells the LSTM is able to protect the network from exploding and vanishing gradient problem.

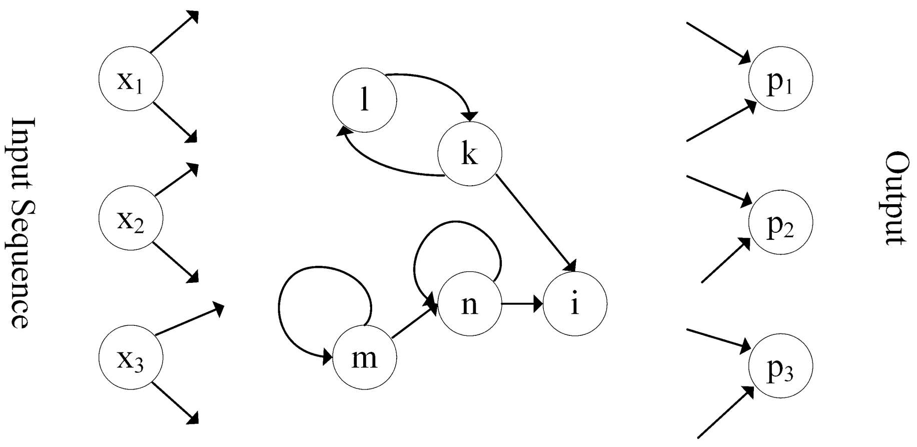


Fig. 2 Recurrent neural network

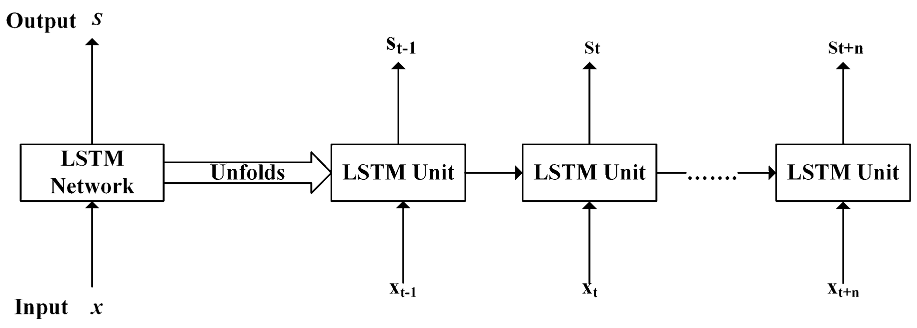


Fig. 3 Unfolding of LSTM

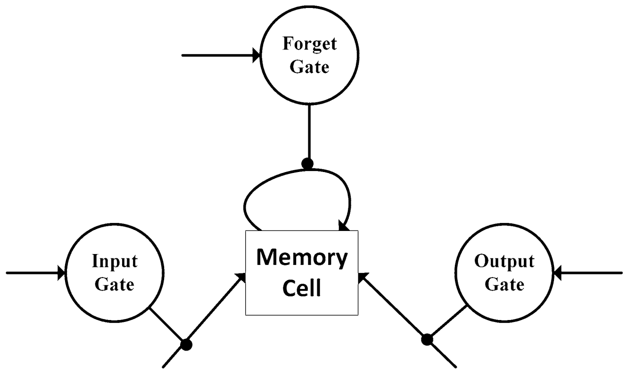


Fig. 4 The gates and memory cell in a single LSTM unit

The LSTM unit consists of a memory cell that has three gates described below:

- *Input gate*: The or input gate calculates the amount of input that is allowed to passed through it and is calculated using Eq. (2):

  $$i = \sigma(x_t U^i + s_{t-1} W^i). \tag{2}$$

  The sigmoid function maps the value of the input between [0, 1] and this value is multiplied by the weight vector (U_i). This helps the gate manage the amount of input which is passed through the input gate.
- *Forget gate*: The forget gate helps the network choose what and how much information from the previous level to pass to the next level. The sigmoid function maps the value of this function between 0 and 1. It is given by:

  $$f = \sigma(x_t U^f + s_{t-1} W^f). \tag{3}$$

  If no input needs to be passed to the next level, the previous memory is multiplied with the zero vector, which makes the input value zero. Similarly, if the memory at s_{t-1} needs to pass to next level it is multiplied by 1 vector. If only some portion of the input is to be passed, then the corresponding vector is multiplied with the input vector.
- *Output gate*: The output gate, defines the output passed at each step of the network. It is given by:

  $$o = \sigma(x_t U^o + s_{t-1} W^o). \tag{4}$$

In the case of spam classification, the final output is the classification label. The output at each time step is required in the problems like language modelling [29], language translation [30] etc. The equations of the three gates described above are the similar equations with different parameter matrices U and W. Each LSTM unit also perform various calculations given below.

- Candidate hidden state g of the network is calculated as:

$$g = \tanh(x_t U^g + s_{t-1} W^g) \tag{5}$$

c_t is the internal memory of the LSTM unit. It is calculated as:

$$c_t = c^{\circ}_{t-1} f + g^{\circ} i \tag{6}$$

It can be seen from the equation that the cell's memory is the combination of the portion of the previous cell state c_{t-1}. The information from the previous memory and the new calculated hidden state multiplied (element wise) by current input state gives the current cell memory.

- The output hidden state s_t is calculated as the product of the internal memory and the output gate:

$$s_t = \tanh(c_t)^{\circ} o. \tag{7}$$

The algorithm for LSTM is described in Algorithm 1.

```
INPUT Text Documents, Max Length = 160
OUTPUT class label Spam or Ham
  (1) For each (Document) input x at time t do
          Input gate $i_t = \sigma(W_i x_i + U_i s_{t-1} + b_i)$
          Candidate $\bar{C} = \tanh(W_c x_t + U_c s_{t-1} + b_c)$
          Forget gate $f_t = (W_f x_t + U_f s_f + b_f)$ d
          The Output gate $o_t = \sigma(W_o x_i + U_o s_{t-1} + b_o)$
          $s_t = o_t \cdot tanhC_t$
      End For
  (2) Apply Softmax function on $s_t$ to get the output label
```

Algorithm 1 LSTM

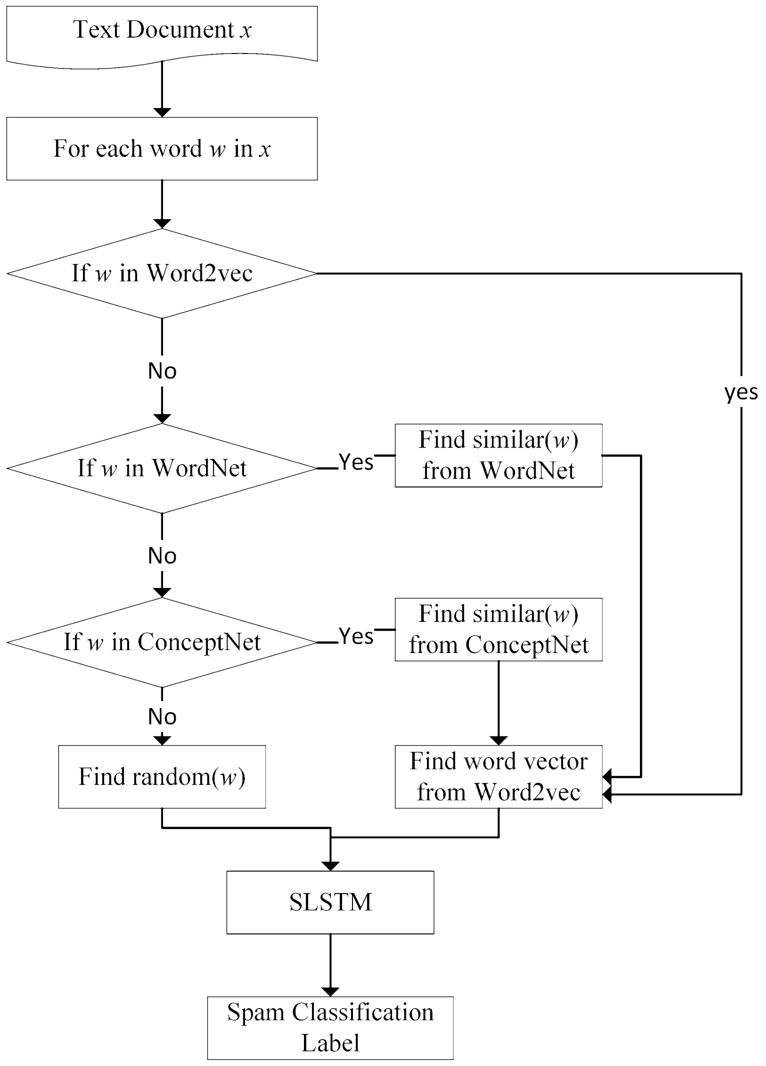
5 Semantic long short term memory (SLSTM)

Semantic LSTM (SLSTM) is a LSTM neural network with a semantic layer on the top of it i.e. before using the LSTM for spam classification the word vectors are enriched with semantically.

The text sentences are converted into word embedding using word2vec. With the help of these word vectors, the textual words from the sentences are mapped onto a multidimensional space and the vectors having similar meaning are mapped close to each other. Google's Word2Vec is trained on Google News Corpus, which has more than 3 Million running words. For the words not mapped in word2vec, a similar word is found using the semantic or conceptual dictionaries: WordNet [31] and ConceptNet [32]. For this task, a lookup is performed in WordNet, which gives a semantically similar word for a given word. Word2vec is again accessed to find the word embedding corresponding to this word.

If no similar word is found for given word in WordNet, a lookup for the similar word in ConceptNet. The word vector corresponding to the similar word is again looked from word2vec. This adds a semantic meaning to the vector representation rather than having random unrelated word vectors. This basic flow of SLSTM is shown in Fig. 5. The algorithm for SLSTM is given in Algorithm 2.

Fig. 5 Semantic LSTM



The details regarding the LSTM [18] are already discussed in the previous section. Figure 6 shows the SLSTM model which takes the sequence of text data as input and these input sequences are converted into word vector $x_1, x_2, \ldots, x_n$ using word2vec, WordNet and ConceptNet.

```
INPUT Text Documents x, Max length = 160

OUTPUT class label Spam or Ham

    (1) For each input x do
                If word present in word2vec.
                        word vector = word2vec(word)
                Else If word in WordNet
                        word1 = WordNet(word)
                        word vector = word2vec(word1)
                Else If word in ConceptNet
                        word1 = ConceptNet(word)
                        word vector = ConceptNet(word1)
        End For
    (2) Form the sentence matrix with the word vectors of the document.
    (3) For each input x at time t do
                Input gate $i_t = \sigma(W_i x_i + U_i s_{t-1} + b_i)$
                Candidate $\bar{C} = \tanh(W_c x_t + U_c s_{t-1} + b_c)$
                Forget gate $f_t = (W_f x_t + U_f s_f + b_f)$
                The output gate $o_t = \sigma(W_o x_i + U_o s_{t-1} + b_o)$
                $s_t = o_t \cdot tanhC_t$
        End For
    (4) Apply Softmax function on $s_t$ to get the output label
```

Algorithm 2 SLSTM

The sequences of input (sentences) are fed into the LSTM unit along with the output of the previous LSTM unit. This is repeated with each input sentence and in this way the LSTM units keep on saving the important features. The number of LSTM units save the most important features. Therefore, each LSTM layer will result in the sequence of output at each layer s_1, s_2, …s_n. Finally, softmax function is applied on s_t, the output of the last LSTM unit to get class label.

6 Experiments and results

6.1 Dataset and evaluation measures

The experiments are conducted on two datasets—SMS spam collection dataset which is available in UCI repository. The second dataset for conducting experiments is Twitter dataset. This dataset is created by scrapping the public live tweets from the micro blogging site Twitter using Twitter API. These tweets are manually classified as ham or spam. The class wise tweets distribution is shown in the Table 1.

For the evaluation of results metrics like precision, recall, accuracy and F1 score are calculated. The calculation of these matrices is based on the confusion matrix as shown in Table 2.

The various evaluation metrics are calculated as below:

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

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

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

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (11)$$

6.2 Hyperparameters

The parameters that helps in optimizing the LSTM network are defined below:

- Word Vector Initialization

The text needs to be converted into numeric form, so that they can be used as an input in the learning model.

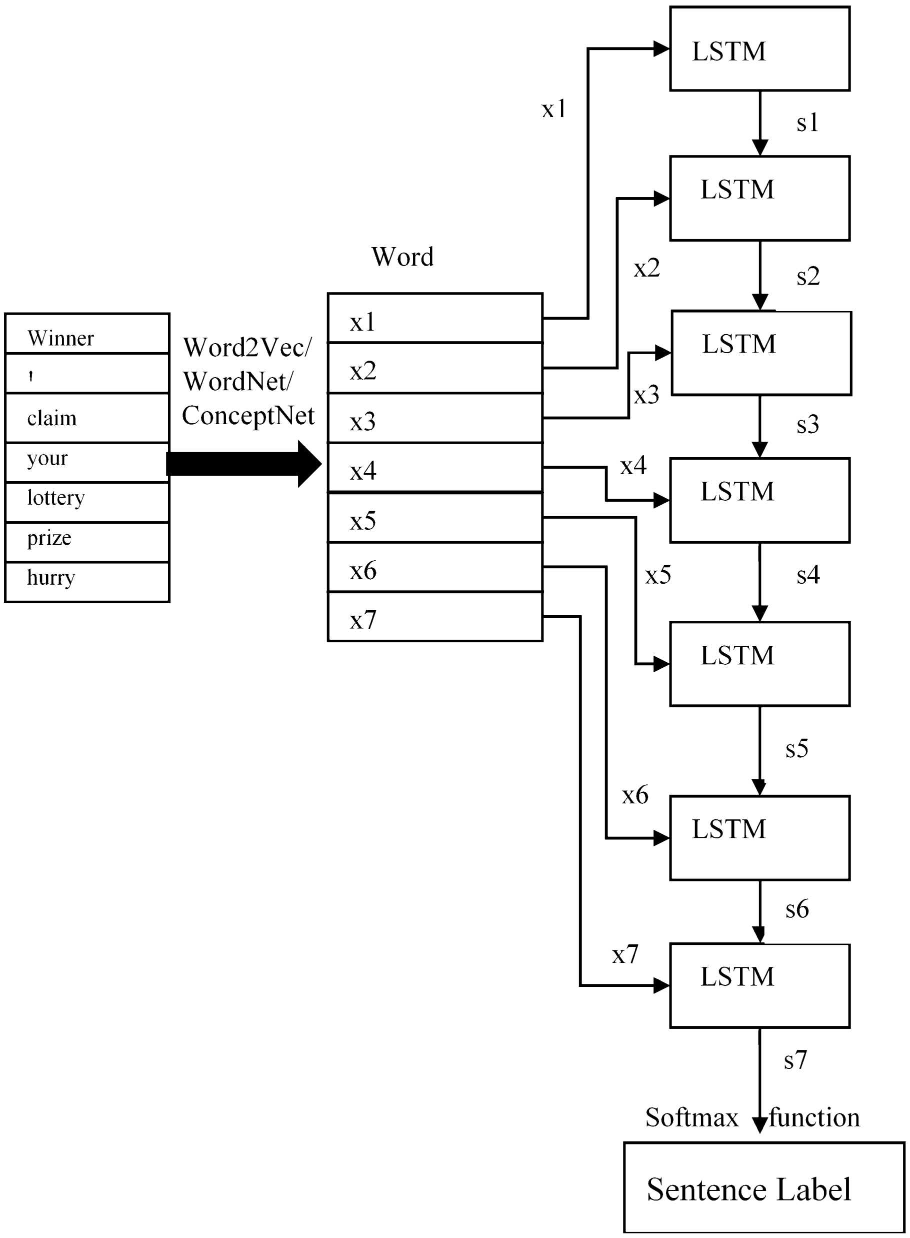


Fig. 6 Semantic LSTM classification model

Table 1 Dataset instance description

Dataset	No. of instances	No. of ham	No. of spam
SMS spam	5574	4827	747
Twitter	5096	4231	865

Table 2 Confusion matrix

	Ham	Spam
Ham	True positive (TP)	False positive (FP)
Spam	False negative (FN)	True negative (TN)

Therefore, they are converted into word vector that is initialized with a random value from the uniform distribution [− 0.25, 0.25]. The dimension of the vector can be chosen according to the data at hand. In the SCNN model, the word vector is initialized by finding the corresponding word embedding from Google's Word2vec [33] having a dimension of 300. In case of static CNN, word vectors are fixed while in non-static CNN, the word vectors are learned with the training process and updated continuously.

- Activation Function

The application of activation function on the convolved features gives the best feature from the word vectors. It is applied after the features of the same window with various filters are concatenated. The activation function is a non-linear function like *tanh*, *ReLU* or sigmoid function that squashes the value of the vectors between the specified ranges. For example: sigmoid function maps the value of the vector between the range [− 1, 1].

- Optimization

To optimize the training network, gradient descent method is used which is the most popular and common methods of optimization. Gradient descent aims at minimizing the error function that is calculated when the errors are backpropagated to the previous layer in a network. In the present work, Adagrad [34] algorithm is used for error optimization that adapts learning rate according to the parameters. The learning rate is adapted according to the parameters with larger update for frequently occurring parameters and smaller update for infrequent parameters.

- Dropout Rate

Most of the neural networks suffer from the problem of overfitting especially when the amount of data is less. One way to reduce the extent of overfitting is to decrease the size of the network rather than increase the amount of data. This can be done through one of the regularization technique known as dropout. Dropout modifies the network for solving the problem of overfitting. In the dropout layer specified part of the network hidden nodes is deleted or dropped. Due to this, the co-adaptation of the nodes is prevented and thus reduces the overfitting of the network.

- No. of Epochs

The epoch refers to the number of times the training process is repeated using the same training data. The number of epochs depends on the training data. The complex and noisy data requires more training iterations. A careful choice of the number of epoch is needed since a very number might lead to overfitting of data and that would result in increased training accuracy but lower test accuracy.

- Number of Features

Features refer to the most important words in the corpus. The number of features limits the application of the model to the most frequently occurring words rather than taking all the features. This helps to reduce the effect of overfitting due to infrequent words that are zeroed out and does not take part in the spam classification.

- LSTM Units

The number of LSTM units is the number of memory cells in the LSTM network. It refers to the capability to memorize the information and correlate it with the past information. These information in these memory units are sent forward in the next time step for the further training.

6.3 Experiments

Extensive experiments are performed using LSTM for the spam detection on two datasets: SMS Spam Collection and Twitter Datasets already defined above.

The words are converted into word vectors using Google's Word2Vec which are pre-trained semantic vectors having dimension of 300. If the word is not present in word2vec similar words are found using WordNet and ConceptNet dictionaries which can then be converted into word vectors.

For gradient descend, the Adagrad update rule is used with a learning rate of 1.0. The implementation of the LSTM model was carried out on the Keras 2.0 API with Tensorflow backend using Python 2.7. It was implemented on Ubuntu 16.04.02 machine installed virtually on a Windows environment with the help of Virtual Box.

The first set of experiments find the optimal number feature for both the data sets with the initial parameters as in Table 3. Better results were achieved when these parameters were optimized for the SMS spam Collection and Twitter datasets.

For the first dataset, the accuracy of the model increased when the no. of features were increased from 5000 to 6000. However, it started decreasing when features were further increased. But, in the Twitter dataset, it kept on increasing with the number of features. The results are shown in Tables 4 and 5.

Another important hyperparameter for LSTM tuning is the number of LSTM units in the network used for training. Table 6 shows the accuracy of LSTM network with different LSTM units. The LSTM units help to reduce the error rate by backpropagating.

Table 3 Initial parameters

Parameter	Value
LSTM units	100
Dropout	0.1
Activation function	Sigmoid
Epochs	10

Table 4 Effect of feature size

No. of features	Accuracy (SMS spam)
5000	97.93
6000	98.92
7000	98.74
8000	97.57

Table 5 Effect of feature size

No. of features	Accuracy (Twitter)
5000	93.81
8000	93.91
10,000	94.21
14,000	95.09

Table 6 Effect of no. of LSTM units

No. of LSTM units	Accuracy (SMS spam)	Accuracy (Twitter)
50	98.92	94.11
100	98.83	95.09
150	99.01	94.21
200	99.01	94.40

However, in case of Twitter data, the best accuracy is achieved when the number of LSTM was 100. In case of SMS spam data, 150 and 200 LSTM units gave the best accuracy. Further increase in the number of LSTM units resulted in a decrease in the accuracy.

Regularization is performed using the most effective technique: dropout which skips the number of neurons while the training process. LSTM architecture is tested for the dropout value of 0.1–0.5 and the results are shown in Table 7 for both the dataset. The optimized value of 0.1 is observed for both the datasets.

Finally, Tables 8 and 9 shows the comparison of LSTM for both the datasets with the optimized parameters shown in Table 10.

Table 7 Effect of dropout

Dropout	Accuracy (SMS spam)	Accuracy (Twitter)
0.1	98.92	95.09
0.2	98.74	93.71
0.3	98.83	93.62
0.4	98.83	92.54
0.5	98.56	93.32

Table 8 Performance comparison of LSTM with classifiers: SMS spam dataset

Classifier	Precision	Recall	Accuracy	F1
KNN	91.37	90.40	90.40	88.13
NB	97.64	97.67	97.67	97.65
RF	97.88	97.85	97.85	97.77
ANN	97.41	97.40	97.40	97.40
SVM	97.45	97.49	97.49	97.44
SLSTM	98.74	99.35	99.01	99.24

Table 9 Performance comparison of LSTM with classifiers: Twitter dataset

Classifier	Precision	Recall	Accuracy	F1
KNN	91.61	91.96	91.96	91.38
NB	91.69	92.06	92.06	91.74
RF	93.25	93.43	93.43	93.04
ANN	91.80	91.18	91.18	91.41
SVM	92.91	93.14	93.14	92.97
SLSTM	95.54	98.37	95.09	96.84

Table 10 Optimized parameters for LSTM

Parameter	Value (SMS spam)	Value (Twitter)
LSTM units	100	150
Dropout	0.1	0.1
No. of features	6000	14,000
Activation function	Sigmoid	Sigmoid
Epochs	10	10

While comparing the SLSTM with the other classifiers in the Tables 7 and 10, the accuracy of SLSTM is the highest. This is due to the semantic representation of the text sentence with the help of word2vec and sequential processing of text with the help of LSTM. It also considers the long term dependency among the word vectors due to the number of LSTM units.

SLSTM architecture performed fairly well on both the datasets. The performance is better than the benchmark methods with the advantage of not having to extract and design handcrafted features. SLSTM has an ability to learn the features from the text automatically. When comparing with SCNN, SLSTM is able to capture long term dependency in the features of the sequential text while SCNN is able to detect short term correlations and temporal features in the texts. While dealing with the data related to social media, the user tends to include lots of slangs, abbreviation etc., precise grammar is not followed, therefore the language structure is not maintained. LSTM helps to sequentially process the input words while correlating with the past words. It better helps in understanding the slangs and new words.

7 Conclusion

LSTM architecture performed fairly well on both the datasets. The performance is comparable to some of the best benchmark methods with the advantage of not having to extract and design handcrafted features. Also, when we are dealing with the data related to social media, the data tend to include lot of slangs, abbreviation etc., also precise grammar is not followed, therefore the level of accuracy depends upon the abstraction of the features extracted. We can further improve the resulted by using pre-trained vectors for the words in our dictionary like `word2vec` etc. and increasing the volume of data for training.

References

1. MAAWG. Messaging anti-abuse working group. Email metrics report. Q1 2012 to Q2 2014. https://www.m3aawg.org/sites/default/files/document/M3AAWG_2012-2014Q2_Spam_Metrics_Report16.pdf. Accessed 30 Mar 2017
2. Mowbray M (2010) The twittering machine. In: WEBIST (2), pp 299–304
3. Benevenuto F, Magno G, Rodrigues T, Almeida V (2010) Detecting spammers on twitter. In: Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), vol 6, no. 2010, p 12
4. Mittal N, Agarwal B, Agarwal S, Agarwal S, Gupta P (2013) A hybrid approach for twitter sentiment analysis. In: 10th international conference on natural language processing (ICON-2013), pp 116–120
5. Ahmed S, Mithun F (2004) Word stemming to enhance spam filtering. In: The conference on email and anti-spam (CEAS'04) 2004
6. Agarwal B, Mittal N (2016) Prominent feature extraction for sentiment analysis. Springer International Publishing, Berlin, pp 21–45
7. Khorsi A (2007) An overview of content-based spam filtering techniques. Informatica 31(3):269–277
8. Kolari P, Java A, Finin T, Oates T, Joshi A (2006) Detecting spam blogs: a machine learning approach. In: Proceedings of the 21st national conference on artificial intelligence (AAAI), July 2006
9. Wang AH (2010) Don't follow me: spam detection in twitter. In: Proceedings of the 2010 international conference on security and cryptography (SECRYPT). IEEE, New York, pp 1–10
10. Tretyakov K (2004) Machine learning techniques in spam filtering. In: Data mining problem-oriented seminar. MTAT, vol 3, no 177, pp 60–79
11. Ntoulas A, Najork M, Manasse M, Fetterly D (2006) Detecting spam web pages through content analysis. In: Proceedings of the 15th international conference on World Wide Web. ACM, New York, pp 83–92
12. Mccord M, Chuah M (2011) Spam detection on twitter using traditional classifiers. In: International conference on autonomic and trusted computing. Springer, Berlin, pp 175–186
13. SMS Spam Collection v.1. http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. Accessed 27 Dec 2016
14. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
15. Bengio Y (2009) Learning deep architectures for AI. In: Foundations and trends® in machine learning, vol 2, no 1, pp 1–127
16. Deng L (2014) A tutorial survey of architectures, algorithms, and applications for deep learning. In: APSIPA transactions on signal and information processing, vol 3
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 513–520
19. Tang D, Wei F, Qin B, Liu T, Zhou M (2014) Coooolll: a deep learning system for twitter sentiment classification. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp 208–212
20. Deng L, Hinton G, Kingsbury B (2013) New types of deep neural network learning for speech recognition and related applications: an overview. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, New York, pp 8599–8603
21. Hong J, Fang M (2015) Sentiment analysis with deeply learned distributed representations of variable length texts. Technical report, Stanford University, pp 655–665
22. Tzortzis G, Likas A (2007) Deep belief networks for spam filtering. In: 19th IEEE international conference on tools with artificial intelligence, 2007. ICTAI 2007, vol 2. IEEE, New York, pp 306–309
23. Mi G, Gao Y, Tan Y (2015) Apply stacked auto-encoder to spam detection. In: International conference in swarm intelligence. Springer, Cham, pp 3–15
24. Jain G, Sharma M, Agarwal B (2018) Spam detection on social media using semantic convolutional neural network. Int J Knowl Discov Bioinform (IJKDB) 8(1):12–26
25. Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53st annual meeting on association for computational linguistics, ACL'15, Stroudsburg, PA, USA. Association for Computational Linguistics
26. Tang D, Qin B, Liu T (2015) Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1422–1432
27. Ul-Hasan A, Ahmed SB, Rashid F, Shafait F, Breuel TM (2013) Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks. In: 12th international conference on document analysis and recognition (ICDAR), 2013. IEEE, pp 1061–1065
28. Wöllmer M, Metallinou A, Eyben F, Schuller B, Narayanan S (2010) Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In: Proceedings on INTERSPEECH 2010, Makuhari, Japan, pp 2362–2365
29. Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association, pp 194–197
30. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112
31. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
32. Liu H, Singh P (2004) ConceptNet—a practical commonsense reasoning tool-kit. BT Technol J 22(4):211–226
33. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of international conference on learning representations (ICLR)
34. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(Jul):2121–2159