IJACT 17-1-12

A Study of Efficiency Information Filtering System using One-Hot Long Short-Term Memory

Hee sook Kim[1], Min Hi Lee[2*]

[1]Department of Computer Information, Inchon Campus of Korea Polytechnic, Korea
primama@naver.com
*[*2]Department of Architecture, Howon University, Korea*
lmh@howon.ac.kr

Abstract

In this paper, we propose an extended method of one-hot Long Short-Term Memory (LSTM) and evaluate the performance on spam filtering task. Most of traditional methods proposed for spam filtering task use word occurrences to represent spam or non-spam messages and all syntactic and semantic information are ignored. Major issue appears when both spam and non-spam messages share many common words and noise words. Therefore, it becomes challenging to the system to filter correct labels between spam and non-spam. Unlike previous studies on information filtering task, instead of using only word occurrence and word context as in probabilistic models, we apply a neural network-based approach to train the system filter for a better performance. In addition to one-hot representation, using term weight with attention mechanism allows classifier to focus on potential words which most likely appear in spam and non-spam collection. As a result, we obtained some improvement over the performances of the previous methods. We find out using region embedding and pooling features on the top of LSTM along with attention mechanism allows system to explore a better document representation for filtering task in general.

Keywords: *information filtering system, spam filtering, region embedding, term weighting.*

1. Introduction

Information filtering plays an important role in many large-scale documents processing systems such as web search, document classification, recommender systems. In common, filtering component is used to filter out unwanted documents and keep only necessary or relevant documents. To be able to perform filtering task, document representation is required and it does affects filtering result.

Bag-of-Words (BoW) is one among other popular methods used in filtering systems in which each document is represented by a set of words and focus on frequency of each word that appear in the document texts and all word positions or other syntactic information are ignored. Under a general implementation of bag-of-words method, word collection is extracted from training datasets to construct user profile which can

Manuscript Received: Feb. 3, 2017 / Revised: Feb. 16, 2017 / Accepted: Mar. 4, 2017
Corresponding Author: lmh@howon.ac.kr
Tel:+82-063-450-7247, Fax: 82-063-450-7248
Howon University., Dep. of Architecture, Korea

be used as a model to perform filtering task on new incoming documents. By using constructed user profile, the system learns to filter documents by computing probability of each word occurs in new documents and examine their relevance. Furthermore, bag-of-words is fixed-length features commonly used in many machine learning techniques. Support vector machine (SVM) is a very popular for classification as well as filtering task [1], [2]. SVM uses the BoW representation with binary, term frequency (TF) and Inverse Document Frequency (IDF) features. Latent semantic indexing (LSI) proposed in [3] is an extension of the Vector Space Model that tries to uncover term dependencies by incorporating semantic information.

Neural network has become a new trend in natural language processing which many research try to propose neural network approach for extracting semantic information and to provide a richer information for filtering system. Long Short-Term Memory (LSTM) [4] is a deep learning technique among other neural network-based approach which is well known for its capability in solving the problem of long-term dependencies and it is a suitable method for sematic extracting related task.

In recent studies [5,6], LSTM is successfully used to incorporate with region embedding and pooling features of Convolutional Neural Network (CNN) for text categorization task and improvement of performance could be obtained. Inspired by previous studies, the purpose of the study in this paper is to extend incorporated models and apply them on spam filtering task. Unlike previous studies on information filtering task, instead of using word occurrence and word context as in probabilistic models, text messages are converted to low-dimensional word vectors. Using region embedding and pooling features of CNNs and LSTM allows filtering system to explore a better representation of spam and nom-spam messages. Also, the model can be generalized to perform other filtering tasks.

2. Methods of Research

2.1 Long Short-Term Memory

Long Short-Term Memory is a novel model of recurrent network proposed for a better learning method which make it possible to store information over a longer period [4]. In natural language processing tasks, LSTM has been used for text data related tasks such as sentiment analysis and semantic representation [7], [8]. There are several variations of LSTM exist, but in this paper, we use the following LSTM notation and formulations as used in [9]:

$$\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}),\\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}),\\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}),\\
u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh(c_t),
\end{aligned} \tag{1}$$

where in standard LSTM notations, it , ft , ot , ct , ht are input gate, forget gate, output gate, cell state and hidden state at each time step t respectively. And xt is the input vector at time step t, σ is denoted as logistic sigmoid function and ⦿ as element-wise multiplication. The forget gate controls the previous memory cell that is forgotten, the input gate and output gate control the input and output of memory cells. Lastly, W and b are model parameters need to be updated during training.

2.2 One-hot vector and LSTM with pooling for text

In general, one-hot vector consists of many zeros and ones representing for presence and absence, positive and negative, or as binary representation. For text data, one-hot vector can be used to represent existence of word that appears in text and in word collection.

As proposed in [5], one-hot vector is used to represent word existences in text within a scope of vocabulary set. In this paper, to construct one-hot vector for text, we follow the same setting described in [5]. Given a document D= (w1,w2,w3….wn) and vocabulary set V, we embed all vocabularies in D and obtain n vectors with length |V|. By using embedded vector V, we check for existence of i-th word (wi) from document D in vector and produce a one-hot vector vi contains zeros for all indexes except an index that wi appears. For example, we have a document D= “I go to school” and a vocabulary set V= {“you”, “I”, “he”, “sleep”, “go”, “school”}. Then, we obtain one-hot vector for each word from document D such as V(I)= [010000], V(go)=[0000010], V(to)=[000000], and V(school)=[000001] where each vector has the same length as number of vocabularies in V, here |V|= 6. To represent document using one-hot vectors of each word, we concatenate each vector together and obtain document vector representation consists of n regions as below:

$$d= [010000 \mid 0000010 \mid 000000 \mid 000001]^{T},$$

where document vector length $|d|= n\ |V|$.

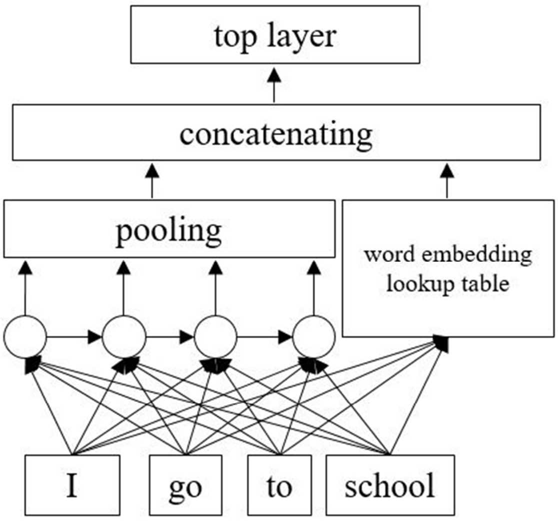

Figure 1. Concatenating output of one-hot LSTM with pretrained word embedding vectors (with attention mechanism)

Pooling is used in pooling layer for Convolution Neural Network (CNN) which most of application dataset are images. The main concept of pooling is to pool some areas of original image called “pixel region” and feed them into network instead of whole image. Pixel region consists of 2D dimension of pixels which can be flattened to 1D dimensional vector and it is generalized to be applicable for text data.

2.3 Combination of one-hot LSTM with self-information weight attention mechanism

One-hot vector is mainly used to preserved word position by mapping all words in document to vocabulary vector using binary values 0 and 1. However, this setting cannot be used to recover semantic information for words from text and semantic information should be used for a better classification performance. Therefore, we proposed an extended method of one-hot LSTM with pooling by adding some components and explore a better performance for filtering task.

We use word embedding vector to train classifier and allows network to learn more semantic features of related words in vector space. Furthermore, we also add words' self-information that will assign more importance to words during document vector compositional process [10].

First, we compute self-information of each word appear in observing documents. Self-information weight of each word can be obtained by extracting all words in spam corpus and calculate Inverse Document Frequency [11] as defined in (2):

$$\mathrm{IDF}(v) = \log\,(N - n(v) + 0.5)\,/\,n(v) + 0.5 \tag{2}$$

where N is total number of documents/messages in corpus, n(v) is the number of message that contain word v and a 0.5 is a fixed value for smoothing. Then we divide each word's weight by weight summation of all words in message. Then we multiply corresponding weight score to each word vector embedding of all words appear in document to construct document embedding vector. Suppose document D consists of N word, D={w1,w2…,wn} we can construct document embedding as follows:

$$\boldsymbol{d} = si_{w1}\cdot \boldsymbol{w_1} + si_{w2}\cdot \boldsymbol{w_2} + \ldots + si_{wn}\cdot \boldsymbol{w_n} \tag{3}$$

where siw refer to term weight or self-information weight of each word as mentioned in [10], wi is word embedding vector represent wi in document D.

To obtain final document representation, we combine pretrained word embedding vector with one-hot vector by concatenating manner.

3. Experiments and Evaluations

3.1 Testing corpora

In our experiment, we apply our proposed model on spam filtering task using two different types of spam corpus. The first corpora is Enron1 which a part of EnronSpam corpora (it can be found at http://www.iit.demokritos.gr/skel/i-config). Another one is SpamAssassin corpus (it is available at http://www.spamassassin.org).

The preprocessed version of EnronSpam dataset was introduced in [12] which contains only the subject and the body of the messages. In this paper, we did experiment only on Enron1 which has a total of 5172 messages in which 3672 are non-spam messages and 1500 are spam messages. So, we got a 29% spam ration. The SpamAssassin corpus contains 6047 messages in total and 4150 for non-spam messages and 1897 for spam messages with a 31.3% spam ratio.

3.2 Experimental settings

Next, we describe some settings used for training on our proposed model.

In our experiment, rather than using random initialized word embedding vector, we use pretrained word embedding vector with 300d (dimensions) and it can be publicly obtained from GloVe [13] (available at http://nlp.stanford.edu/projects/glove) which was trained with large scale external corpus. We have summarized and include statistic of all pretrained word embedding data obtained from GloVe project that were used in our experiment as in table 1.

Table 1. Pretrained Word Embedding Statistic

No	Glo Ve300d		
	Filename	Information	Size
1	Wikipedia2014 + Gigaword 5	6 B token, 400K vocab	822Mb
2	Common Crawl42B	42B tokens, 1.9M vocab	1.75Gb
3	Common crawl840B	480B tokens, 2.2M vocab	2.03Gb
	Total	4.1M vocab	

3.3 Evaluation measure

To evaluate the performance of filtering system, we calculate Accuracy which is a performance measure used to measure the percentage of correctly classified messages, define as:

$$Accuracy = (\mathrm{S} + \mathrm{NS}) / (\mathrm{TS} + \mathrm{TNS}) \qquad (4)$$

where S, NS are number of spam and non-spam messages that are classified to correct categories and TS, TNS are total number of spam and total number of non-spam messages in test dataset respectively.

3.4 Experimental results

In this sub section, we will summarize all experiment result obtained from experimental model described in previous section.

To evaluate and compare the performance of our proposed method, we also include the performance accuracy score of Support Vector Machine (SVM) and Deep Belief Network for spam filtering task on test dataset which were reported in [14]. Table 2 shows comparison between previous methods and best performance of our proposed method.

Table 2. Comparison of Performances Accuracy

Model	*Enron1*	*SpamAssassin*
SVM (Tzortzris & Likas, 2007)	0.9692	0.9732
DBN (Tzortzris & Likas, 2007)	0.9743	0.9750
LSTM_oh_pooling	0.9729	0.9851
LSTM_oh_pooling_addsi_attention	**0.9748**	**0.98675**

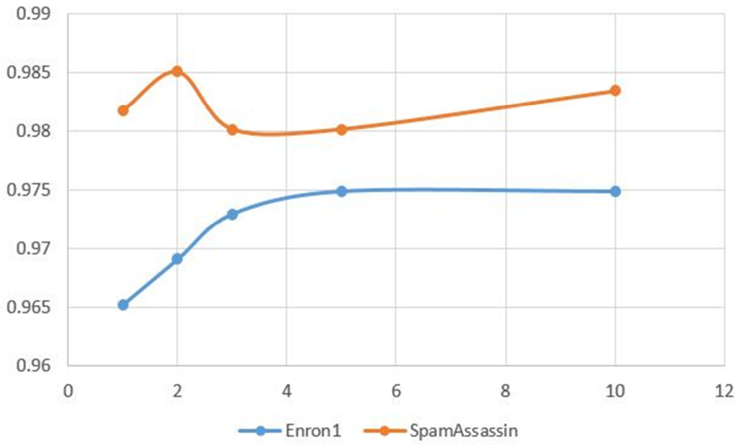

Figure 2. Performances of proposed method using different variants of max-pooling values (1, 2 , 3, 5, 10)

We have compared two variants of one-hot LSTM with pooling and bidirectional one-hot LSTM with pooling and attention to previous methods, SVM and DBN. From table 2, we can see the basic one-hot LSTM with pooling beat two previous methods in experiment on SpamAssassin dataset. However, in case of using Enron1 dataset, it can only beat the SVM model while its performance is a bit lower that DBN model about 0.0014. But in case we apply bidirectional one-hot LSTM which is concatenation of input vector representation from two reversed directions and add term weight of attention mechanism, we can obtain the best performance among other methods in the table in both case of experiment on Enron1 and SpamAssassin dataset.

Furthermore, additional evaluation scores are also reported in Figure 2, where various setting on max-pooling values are used in both basic one-hot LSTM and extended one-hot LSTM with attention mechanism.

5. Discussion and Conclusions

We did experiment in spam filtering task using extended model of one-hot LSTM with pooling and extra components of self-information weights and attention mechanism. Unlike most of previous methods on spam filtering task, instead of using only term frequency, we apply inverse document frequency to increase efficiency of features learning of the system filter.

Using pretrained word embedding vector helps to recover semantic information of each word in message. The proposed method not only use one-hot (binary value) vector to represent existence of word in message, it also includes the semantic feature where it helps to learn similar words that appear closely to each other in vector space. Moreover, the extended model with self-information weight of each word in the message helps system filter to perform more accurately by giving penalty to most common words and assign more importance to words which become a help property to compose a better document vector representation.

In conclusion, the proposed method using extended components of term weights and combination with pretrained word embedding with attention mechanism can improve the efficiency of spam filtering system.

References

[1] H. Drucker, D. Wu, and V.N. Vapnik, “Support Vector Machines for Spam Classification,” *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, pp. 1048-1054, Sept 5, Sep. 1999.

[2] A. Kolcz and J. Alspector. “SVM-Based Filtering of E-mail Spam with Content-Specific Misclassification Costs”, in *Proc. of the Workshop on Text Mining* (TextDM’01), 2001.

[3] Deerwester, Dumais, Furnas, Lanouauer, and Harshman, “Indexing by Latent Semantic Analysis,” *Journal of the American Society for Information Science,* Vol. 41, No. 6, pp. 391-407, 1990.

[4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, Vol. 9, No. 8, pp.1735–1780, 1997.

[5] Rie Johnson and Tong Zhang, “Effective Use of Word Order for Text Categorization with Convolutional Neural Network,” In *NAACL HLT*, 2015.

[6] Rie Johnson and Tong Zhang, “Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings,” in *Proc. 33rd International Conference on Machine Learning*, Vol. 48, 2016.

[7] A.L. Maas, R.E Daly, P.T. Pham et al., “Learning Word Vectors for Sentiment Analysis,” in *Proc.49th Annual Meeting of the Association for Computational Linguistics*, pp. 142–150, June 19-24, 2011.

[8] K.S. Tai, R. Socher, and C.D. Manning, “Improved Semantic Representation from Tree-Structured Long Short-Term Memory Networks,” *in Proc.53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing,* pp. 1556-1566, July 26-31, 2015.

[9] Z. Wojciech and I. Sutskever, “Learning to Execute,” *under review as a Conference Paper at 5th International Conference on Learning Representations (ICLR)*, May 7 - 9, 2015.

[10] I. Vulic and M. Moens, “Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings,” in *Proc.38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 363-372, Aug. 9-13, 2015.

[11] K.S. Jones, “A Statistical Interpretation of Term Specificity and Its Application in Retrieval,” *Journal of Documentation*, Vol. 60, No. 5, pp. 493-502, 2004.

[12] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam Filtering with Naive Bayes – Which Naive Bayes?,” in *Proc. 3rd Conf. Email and Anti-Spam*, July 27-28, 2006.

[13] J. Pennington, R.Socher, and C.D. Manning, “GloveL Global Vectors for Word Representation,” in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543,October 25-29, 2014.

[14] G. Tzortzis and A. Likas, “Deep Belief Networks for Spam Filtering”, in *Proc. 19th IEEE International Conference on Tools with Artificial Intelligence,* pp. 306-309, Oct. 29 - 31, 2007.