Comparative Evaluation of SMS Spam Filtering Techniques

Diego Antognini and **Panayiotis Danassis**
École Polytechnique Fédérale de Lausanne
Email: {diego.antognini, panayiotis.danassis}@epfl.ch

Abstract

SMS spam is an increasing threatening problem, especially in Asia and in developing countries, where the volume of SMS spam messages has dramatically increased over the past few years. The SMS spam filtering problem can be be approached by a plethora of different techniques, ranging from simple solutions such as white and black listing, to more elaborate content-based filtering, such as the methods employed to combat email spam messages. Nevertheless, compared to the email spam filtering problem, the sms one presents many unique challenges. This is partly due to lack of benchmark datasets, but mainly due to the limited size of a standard SMS along with the abundance of idioms and abbreviations. As such, in this work we focus on both message representation techniques along to classification algorithms, and present a brief comparative evaluation of different state-of-the-art techniques. **Experimental results suggest...**

1 Introduction

Spam refers to unsolicited electronic messages delivered to a large number of recipients customarily via email. In recent years returns from the email spaming are diminishing due to robust and effective filtering and user awareness (Delany, Buckley, and Greene 2012). Coupled with the emergence of low cost or free SMS messaging services, it has lead a growing number of marketers to use text messages (SMS) to target subscribers, especially in Asia and in developing countries (Gómez Hidalgo et al. 2006) (Yadav et al. 2011).

A spam classifier or filter, aims in recognizing and preventing the delivery of the aforementioned unsolicited messages. There is a vast literature in email spam filtering (e.g. see (Cormack and others 2008) (Blanzieri and Bryl 2008)), and state-of-the-art email spam filters are remarkably effective. Nevertheless, the sms spam filtering problem presents many unique challenges, and blindly adopting such techniques is not sufficient. This is partly due to lack of benchmark datasets, which further complicates the application of content-based filtering algorithms, but mainly due to the limited size of a standard SMS along with the abundance of idioms, informal speech (slang), and abbreviations [1]. As such, special care is required on both the message representation and the employed classification algorithm.

In this work we present a brief comparative overview on well established content-based filtering algorithms, ranging from simple models such as the naive Bayes classifier, to more complicated ones like a convolutional neural network. Moreover, we examine various message representation models, ranging from simple vector representations, to word and sentence embeddings. The latter constitute distributed vector representations able to capture a large number of syntactic and semantic word relationships, and have recently shown to be highly successful in a plethora of natural language processing applications (Mikolov et al. 2013) (Bojanowski et al. 2016) (Pagliardini, Gupta, and Jaggi 2017).

The rest of the paper is organized as follows. Section 2 provides a brief overview of SMS filtering techniques, Section 3 presents the evaluated models, both for message representation and classification, and Section 4 provides simulation results. Finally, Section 5 concludes the paper.

2 Related Work

In this section we provide a brief overview of related work in the area of SMS spam classification.

In (Gómez Hidalgo et al. 2006) the authors have tested a number of message representation techniques and machine learning algorithms, in terms of effectiveness. They have identified the tokenization step as the most important one, since a bad representation may lead to a poor classifier. Contrary to our work, were we have employed word and sentence embeddings which can capture syntactic and semantic word relationships, the authors feed their model with a big number of attributes and used the information gain metric (Yang 1999) as an attribute selection mechanism. They have identified as the most suitable learning algorithm the Support Vector Machines (SVM).

In (Xu et al. 2012) the authors follow an orthogonal approach and utilize non-content features from static, network and temporal views. Subsequently, they incorporated the aforementioned features to an SVM classifier, with a gain of $\approx 8\%$ in AUC (Area Under the ROC Curve).

Finally, a combination of content-based and non content-based features was studied in (Sulaiman and Jali 2016), in which the authors also focused on the effect of the employed model in the battery and processing performance.

[1] Here are two examples of such messages, drawn from the employed dataset: '*Ok lar... Joking wif u oni...*', and '*dun say so early hor... U c already then say...*'.

In this work, we limit ourselves to content-based filtering with emphasis on the message representation. Following the recent success in a variety of natural language processing applications (Mikolov et al. 2013) (Bojanowski et al. 2016) (Pagliardini, Gupta, and Jaggi 2017), we employ word and sentence embeddings to capture syntactic and semantic word relationships.

3 Overview of Employed Models

In this section we present a brief theoretical overview of the employed models for both message representation and the subsequent classification.

3.1 Message Representation Models

Naive Vector Representation

Word Embeddings Blablabla

As this representation leads to a $3-D$ matrix, we only use advanced neural networks as most of the methods cannot manage this kind of data easily.

Blabla

In order to fine-tune word embeddings to our specific domain, we update them during the backpropagation.

Sentence Embeddings

3.2 Classification Models

Naive Bayes The naive Bayes classifier determines the class $c \in \mathcal{C}$ of an input feature vector $\mathbf{h}$ by seeking the value that maximizes:

$$\max_{c \in \mathcal{C}} \pi_c \mathbb{P}(\mathbf{h} = h | \mathbf{c} = c) \tag{1}$$

Bernouilli - Multinomial - Gaussian are runned in the code

Logistic Regression This model implements regularized logistic regression using a limited-memory BFGS solver (Liu and Nocedal 1989). The L-BFGS solver is an optimization algorithm in the family of quasi-Newton methods which does not require to analytically compute the Hessian of a function. Instead it computes an estimation which uses to steer its search through variable space. The goal is to find the class that minimizes $f()$, where f is an L2-regularized cost function:

$$c^* = \arg\min_{c \in \mathcal{C}} f(c) \tag{2}$$

Decision Tree **TODO as we use it thereafter**

Random Forest We utilize a random forest meta-classifier which fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Adaptive Boosting The adaptive boosting (AdaBoost) algorithm (Freund and Schapire 1997) produces an accurate classification rule by combining rough and moderately inaccurate learning algorithms ('weak learners') into a weighted sum that represents the final output of the boosted classifier. We have deployed a variant called AdaBoost-SAMME (Hastie et al. 2009), using a decision tree classifier as the base estimator from which the boosted ensemble is built.

Support Vector Machine We have utilized and compared two support vector machine classifiers, one with a linear and one with radial basis function (rbf) kernel.

Feedforward Neural Network This model implements a multi-layer perceptron classifier with a rectified linear unit activation function. The weight optimization is being performed using Adam (Kingma and Ba 2014), an algorithm based on adaptive estimates of lower-order moments. Adam has the advantages of being computationally efficient, and has low memory requirements, which is ideal for deployment in real scenarios with large datasets used by telecommunication providers, and/or in mobile devices. In order to generalize better, for all the neural models we use in addition L2 regularization, dropout (Srivastava et al. 2014) where the idea is to cancel the activation of a neuron with a probability p and also clip the gradient norm to 1.0 to avoid gradient exploding.

Recurrent Neural Network As sentence are a sequence of words, this emphasize the try of using a recurrent neural network which can leverages this kind of sequenced data. In order to handle gradient vanishing/exploding problem, we use the propose Long-Short Term Memory units (LSTM) of (Hochreiter and Schmidhuber 1997). **To continue by describing the model**.

Convolutional Neural Network Convolutional neural networks are considered state-of-the-arts for text classification ((Cao et al. 2017) (Kim 2014) (Zhang and LeCun 2015) (Xiao and Cho 2016)) and therefore, we naturally implement a similar model based on (Kim 2014). **To continue by describing the model**.

4 Experimental Evaluation

We have conducted a series of simulations, and employed a variety of message representation and classification models, with goal to present a comparative overview of well-established classification techniques applied to the SMS spam filtering problem.

4.1 Dataset

To train and evaluate our models, we have utilized the *SMS Spam Collection v. 1* [2]. The SMS Spam Collection is a free corpus, collected for research purposes. It consists of 5.574

[2] https://www.kaggle.com/uciml/sms-spam-collection-dataset, http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

messages in English, tagged according to being ham (legitimate) or spam ($|\mathcal{C}| = 2$). As spam classification is usually similar to an anomaly detection problem, the dataset is highly imbalanced as only 13.5% is considered as spam.

4.2 Evaluation Setup

todo...

4.3 Simulation Results

todo...

5 Conclusion

As the cost of sending messages over the telecommunication network decreases, SMS messaging is becoming a perfect domain for abuse. SMS spaming is thriving, especially in Asia and in developing counties. The unique text style in SMS messaging, requires additional effort in terms of message representation models, which in turn have a significant effect in the performance of the employed spam classifiers. In this work, we have presented a brief comparative overview of various message representation models, ranging from simple vector representations to state-of-the-art word and sentence embeddings, and content-based filtering models. Experimental results show **add results.**

6 NOTES:

Challenges:

- People use abbreviations and when communicating via SMS. E.g. 'Ok lar... Joking wif u oni...', or 'dun say so early hor... U c already then say...'. Future work: use domain specific corpora.

References

Blanzieri, E., and Bryl, A. 2008. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review* 29(1):63–92.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2017. Improving multi-document summarization via text classification. In *AAAI*, 3053–3059.

Cormack, G. V., et al. 2008. Email spam filtering: A systematic review. *Foundations and Trends® in Information Retrieval* 1(4):335–455.

Delany, S. J.; Buckley, M.; and Greene, D. 2012. Sms spam filtering: methods and data. *Expert Systems with Applications* 39(10):9899–9908.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55(1):119–139.

Gómez Hidalgo, J. M.; Bringas, G. C.; Sánz, E. P.; and García, F. C. 2006. Content based sms spam filtering. In *Proceedings of the 2006 ACM symposium on Document engineering*, 107–114. ACM.

Hastie, T.; Rosset, S.; Zhu, J.; and Zou, H. 2009. Multi-class adaboost. *Statistics and its Interface* 2(3):349–360.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1746–1751.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming* 45(1-3):503–528.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Pagliardini, M.; Gupta, P.; and Jaggi, M. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958.

Sulaiman, N. F., and Jali, M. Z. 2016. A new sms spam detection method using both content-based and non content-based features. In *Advanced Computer and Communication Engineering Technology*. Springer. 505–514.

Xiao, Y., and Cho, K. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *CoRR* abs/1602.00367.

Xu, Q.; Xiang, E. W.; Yang, Q.; Du, J.; and Zhong, J. 2012. Sms spam detection using noncontent features. *IEEE Intelligent Systems* 27(6):44–51.

Yadav, K.; Kumaraguru, P.; Goyal, A.; Gupta, A.; and Naik, V. 2011. Smsassassin: Crowdsourcing driven mobile-based system for sms spam filtering. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, 1–6. ACM.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Information retrieval* 1(1-2):69–90.

Zhang, X., and LeCun, Y. 2015. Text understanding from scratch. cite arxiv:1502.01710Comment: This technical report is superseded by a paper entitled "Character-level Convolutional Networks for Text Classification", arXiv:1509.01626. It has considerably more experimental results and a rewritten introduction.