Deep Dive into Authorship Verification of Email Messages with Convolutional Neural Network

Marina Litvak(✉)

Shamoon College of Engineering, Beer Sheva, Israel
marinal@ac.sce.ac.il

Abstract. *Authorship verification* is the task of determining whether a specific individual did or did not write a text, which very naturally can be reduced to the binary-classification problem. This paper deals with the authorship verification of short email messages. Hereafter, we use "message" to identify the content of the information that is transmitted by email. The proposed method implements the binary classification with a sequence-to-sequence (seq2seq) model and trains a convolutional neural network (CNN) on positive (written by the "target" user) and negative (written by "someone else") examples. The proposed method differs from previously published works, which represent text by numerous stylometric features, by requiring neither advanced text preprocessing nor explicit feature extraction. All messages are submitted to the CNN "as is," after padding to the maximal length and replacing all words by their ID numbers. CNN learns the most appropriate features with backpropagation and then performs classification. The experiments performed on the Enron dataset using the TensorFlow framework show that the CNN classifier verifies message authorship very accurately.

Keywords: Authorship verification · Binary classification · Convolutional neural network

1 Introduction

Communication through electronic mail is a very basic everyday activity for almost every person these days. This communication can be personal or official and can have different purposes: work, study, commerce, or just chatting with friends. However, because we cannot trust every message that arrives to our account, we use spam filters on a daily basis. Email fraud is one of the most common types of illegal activity enabled by the Internet. Millions of fraudulent messages are sent every day. Statistics[1] say that in Q1 2017, the percentage of spam in email traffic amounted to 55.9%.

Email communication may be misused by various means. An intruder may disguise oneself as a legitimate user by forging messages after breaking into a

[1] https://securelist.com/spam-and-phishing-in-q1-2017/78221/.

J. A. Lossio-Ventura et al. (Eds.): SIMBig 2018, CCIS 898, pp. 129–136, 2019.
https://doi.org/10.1007/978-3-030-11680-4_14

mail server and fabricating SMTP messages [18], performing man-in-the-middle attacks [7], hacking an email account, or physically accessing the user's computer. The intruder's purpose can be spying, phishing, or other malicious goals. Therefore, performing authorship verification for suspect email messages may have a crucial role in cybersecurity and forensic analysis. In this paper we introduce an approach to the problem of authorship verification of short messages, which can be accurately applied on a "raw" text of emails and, in contrast to the state-of-the-art works, does not require either enhanced text preprocessing or feature extraction.

2 Related Work

The authorship verification problem has been studied for about decade. Most works used stylometry and relied on shallow classification models. Stylometry aims at reflecting personal writing styles, defined by numerous stylometric features [9]. In general, stylometric features can be categorized into four categories: lexical, syntactic, structural, and content-specific. The total amount of single features in stylometry-based work can reach hundreds, and, therefore, feature selection or dimensionality reduction must be performed prior to classification. Among the most frequently used classifiers in stylometry-based authorship verification models are: k-nearest neighbor (kNN), Naïve Bayes, decision tree, Markov chains, support vector machine (SVM), logistic regression (LR), and neural network. As can be seen from the literature, all authorship verification studies differ in terms of the stylometric features and the type of classifiers employed. An extended survey of stylometric features and authorship detection techniques is given in [8].

The first attempts of authorship verification focused on general text documents and were not realistic for application to online texts, which are usually much shorter, as well as being poorly structured and written. For example, the SVM-based model in [13] obtained 95.70% accuracy for documents containing at least 500 words. Many researchers subsequently investigated the effectiveness of stylometry techniques for authorship authentication on shorter text, including email messages. Their results were not as promising as the results for longer texts. Various classification and regression models with 292 stylometric features yielded an Equal Error Rate (EER) ranging from 17.1% to 22.4% on the Enron email dataset in [9]. Using 150 stylistic features in [5] resulted in an accuracy of 89% for 40 users from the Enron dataset. Authors of [14] combined stylometric representation with 233 features and various classification techniques, obtaining an accuracy of 79.6% on Facebook posts. SVM and SVM-LR classifiers were applied in [3] for authorship verification of short online messages, including email messages. About one thousand stylometric features, enriched by the N-gram model, have been extracted and then selected prior to classification. Experimental evaluation on the Enron email and Twitter datasets produced EER results varying from 9.98% to 21.45%. The SVM model with most frequent words as features [16] achieved 80% accuracy on 50 users from the Enron dataset. A

stylometry based authorship verification model based on the Gaussian-Bernoulli deep belief network [4] produced EER results ranging from 8.21% to 16.73% on Enron and Twitter datasets, respectively.

We propose a different approach to the problem of authorship verification of short messages. This approach is based on the deep sequence-to-sequence CNN model, which *does not require either enhanced text preprocessing or feature extraction*. Originally invented for computer vision, CNN models have been lately proven to be effective for various natural language processing (NLP) tasks [6]. For example, a simple one-layer CNN was successfully applied for the sentence classification tasks in [10]. We adapt a similar approach and train the CNN classifier on a two-class training data, composed of positive (written by the "target" user) and negative (writen by "someone else") examples. No pre-trained word vectors are required. This approach, while saving much time and effort that could be invested in feature extraction and selection, produces a very high accuracy.

3 Authorship Verification with CNN

The traditional authorship verification approach, based on stylometry and classification, is usually composed of: (1) extracting a rich set of hand-designed features, (2) selecting the most significant ones, and then (3) feeding them to a standard classification algorithm (for example, SVM). The choice of features is a completely empirical process, mainly based on our linguistic intuition; and to a large extent, this determines the key to success.

Following the idea of application of CNN to NLP tasks [2,6,10,11,19], we propose a radically different approach: we apply a multilayer neural network (NN), trained in an end-to-end fashion, on a "raw" text, after a very basic preprocessing. The NN architecture takes the input text and *learns several layers of feature extraction* that process the input. The features computed by the deep layers of the network are *automatically trained by backpropagation to be relevant to the task* (of authorship verification in our case).

Typical CNN is composed of several convolutional modules that perform feature extraction. Each module is a sequence of a convolutional and pooling layers. The convolutional layer performs mathematical calculations (filter) to produce features in the feature map. The pooling layer reduces the dimensionality of the feature map. A commonly used pooling algorithm is max pooling, which extracts sub-regions of the feature map and keeps their maximum value, while discarding all other values. The last convolutional module is followed by one or more dense layers. Dense layers perform classification on the features extracted by the convolutional layers and reduced by the pooling layers. In a dense layer, every node is connected to every node in the preceding layer. The final CNN dense layer contains a single node for each target class in the model, with a softmax activation function that generates a probability value for each node. We can interpret the softmax values for a given input as its likelihood to belong to each target class. Figure 1 shows the model architecture adapted to the binary classification of text messages. We explain it in more detail below.

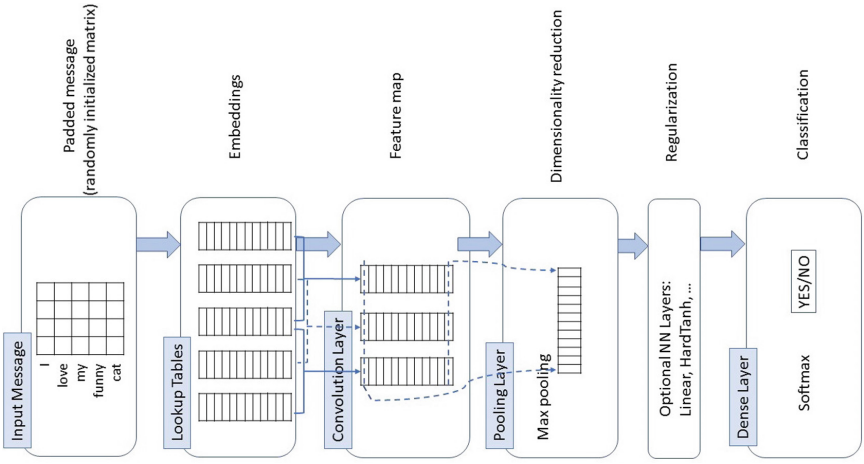

Fig. 1. CNN architecture for email classification task.

Let m_i be the k-dimensional word vector corresponding to the i^{th} word in the message. An email message of length n (zero-padding strategy as in [10] is applied) is represented as $m_{1:n} = m_1 \oplus m_2 \oplus \ldots \oplus m_n$, where $\oplus$ is the concatenation operator.

Let $m_{i:i+j}$ refer to the concatenation of word vectors $m_i, m_{i+1}, \ldots, m_{i+j}$. A convolution operation applies a filter $\varphi \in R^{hk}$ to a window of h words to produce a new feature. For example, a feature c_i is calculated from a window of word vectors $m_{i:i+h-1}$ as $c_i = f(\varphi \cdot m_{i:i+h-1} + b)$, where b is a bias term and f is a non-linear function such as the hyperbolic tangent. Filter φ is applied to each possible window of h word vectors for words in the input text $\{m_{1:h}, m_{2:h+1}, \ldots, m_{n-h+1:n}\}$ to produce a feature map $c = [c_1, c_2, \ldots, c_{n-h+1}]$.

A max-overtime pooling operation [6] is then applied over the feature map and the maximum value $\hat{c} = \max\{c\}$ is taken as the feature corresponding to this particular filter. This stage is aimed at capturing the most important feature—one with the highest value—for each feature map. One feature is extracted from one filter. The model uses multiple filters—with various window sizes—to obtain multiple features. These features are passed to a fully connected dense layer activating softmax function that produces the probability distribution of the input text over target classes.

We use single channel architecture, with one that is fine-tuned via backpropagation. This means that we do not need to provide pre-trained static word vectors (embeddings). All words in our input messages are randomly initialized and then modified during training.

4 Experiments

We evaluated our approach on the Enron email dataset [12]. The Enron dataset was used for different kinds of authorship analysis, including authorship attribution, authorship verification, authorship profiling (characterization), and authorship similarity detection. After discarding users with less than 1000 email messages, we trained our model to 52 remaining users. For each user, 1000 verified email messages were sampled. The parameters of our data are shown in Table 1.

Table 1. Enron data after filtering

# emails	52000
max email length	95 words
# users	52
# messages per user	1000

CNN model[2] was trained for each user. In order to get balanced data, we took the same amount (1000) of positive (all emails that were written by the "target" user) and negative (written by other users and randomly selected) examples. 90% of this data was used for training and 10% for testing. The TensorFlow framework [1] was used in our experiments. In preprocessing, every email was padded to the maximal length (95 words) and encoded by replacing its words by their ID numbers (integers from 1 to V, where V is a vocabulary size).

Figure 2 depicts the accuracy distribution for 52 users with a clustered bar chart. The average overall accuracy is 97%, which is significantly better than what most of the previously published works reported on the Enron dataset.[3] Unfortunately, we could not compare our performance with other works where only EER–that cannot be transformed to accuracy without additional information–was reported or other dataset–even if it is a subset of Enron dataset–was used. CNN performance depends on the amount of epochs (steps) performed during the training.[4] Figures. 3 and 4 show accuracy as a function of epoch number and loss as a function of epoch number, respectively.[5] Blue curves in these figures represent training accuracy and loss, while red curves represent test accuracy and loss.

[2] We kept the default settings of the CNN model in the TensorFlow framework, which are as follows: number of embedding dimensions is 128; filter sizes are 3, 4, and 5; number of filters is 128, dropout probability is 0.5, L2 regularization lambda is 0, batch size is 64.

[3] The best accuracy of 89% for 40 users from the Enron dataset was reported in [5].

[4] We ran our model with 500 epochs.

[5] Obtained from training on one of the users.

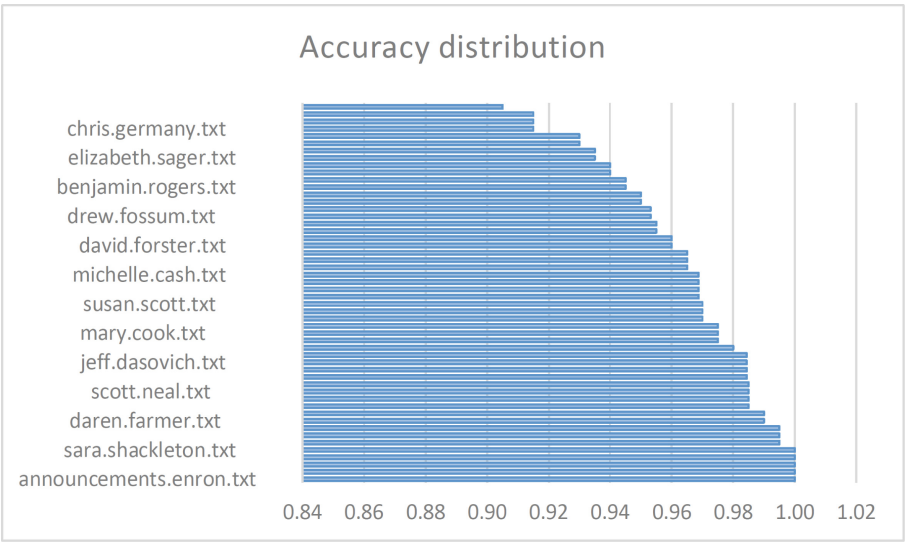

Fig. 2. Accuracy distribution.

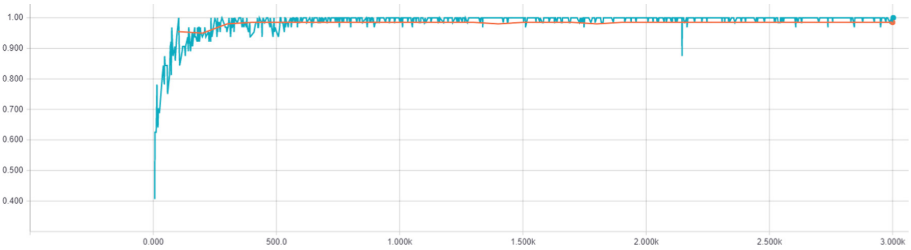

Fig. 3. Accuracy as a function of epochs number. (Color figure online)

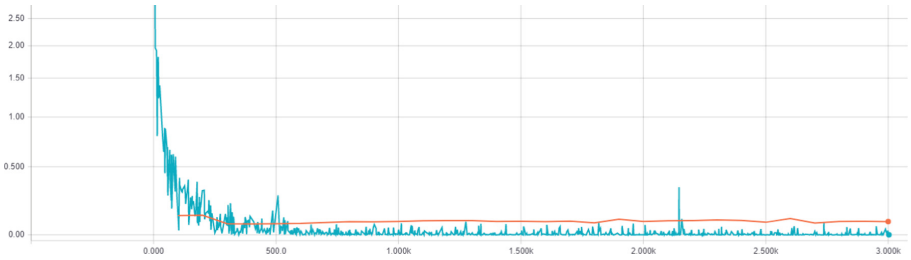

Fig. 4. Loss as a function of epochs number. (Color figure online)

5 Conclusion and Future Work

This paper describes an application of a deep sequence-to-sequence CNN model to the authorship verification task for email messages. In contrast to current state-of-the-art works, our model does not require explicit feature extraction. CNN gets "raw" text as an input, learns features, and performs the classification task on them. The results show that this model verifies email message authorship very accurately. The results can be fine-tuned by performing more epochs, that is generally improves the accuracy of the NN models. In conclusion, the main value of the proposed method is its accurate performance while being applied on a "raw" text. As such, it allows to save time required for the features design, implementation, and extraction and to avoid adding noise to the representation model.

In the future, we intend to experiment with different variations of the CNN architecture, such as the multichannel architecture with several 'channels' of word vectors. These channels will encompass static throughout pre-training with a neural language model [15,17] and non-static that is fine-tuned via backpropagation, as in [10]. Static vectors can be trained by word2vec [15] or Glove [17]. In such architecture, each filter must be applied to all channels, and the results must be summed to calculate a feature c_i in the feature map. Our experiments can be extended with different baseline methods, additional evaluation metrics (i.e. EER), and real-world (unbalanced) domains. Also, additional task-related features (email message structure and meta-data) can be incorporated into the neural network. In addition, we would like to apply our approach to a different task of authorship analysis—authorship attribution—that can be modeled as a classification task with multiple classes.

Acknowledgments. The author is grateful to Vlad Vavilin and Mark Mishaev for the implementation and running the experiments using the TensorFlow framework.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation OSDI 2016, pp. 265–283. USENIX Association, Berkeley (2016). http://dl.acm.org/citation.cfm?id=3026877.3026899
2. Britz, D.: Understanding convolutional neural networks for NLP (2015)
3. Brocardo, M.L., Traore, I., Woungang, I.: Authorship verification of e-mail and tweet messages applied for continuous authentication. J. Comput. Syst. Sci. **81**(8), 1429–1440 (2015)
4. Brocardo, M.L., Traore, I., Woungang, I., Obaidat, M.S.: Authorship verification using deep belief network systems. Int. J. Commun. Syst. **30**(12), e3259 (2017)
5. Chen, X., Hao, P., Chandramouli, R., Subbalakshmi, K.P.: Authorship similarity detection from email messages. In: Perner, P. (ed.) MLDM 2011. LNCS (LNAI), vol. 6871, pp. 375–386. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23199-5_28

6. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(Aug), 2493–2537 (2011)
7. Desmedt, Y.: Man-in-the-middle attack. In: van Tilborg, H.C.A. (ed.) Encyclopedia of Cryptography and Security. Springer, Boston (2005). https://doi.org/10.1007/0-387-23483-7
8. El Bouanani, S.E.M., Kassou, I.: Authorship analysis studies: a survey. Int. J. Comput. Appl. (0975 – 8887) **86**(12), 22–29 (2014)
9. Iqbal, F., Khan, L.A., Fung, B., Debbabi, M.: E-mail authorship verification for forensic investigation. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1591–1598. ACM (2010)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 2017 Proceedings of ICLR (2017)
12. Klimt, B., Yang, Y.: The enron corpus: a new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_22
13. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proceedings of the Twenty-First International Conference on Machine learning, p. 62. ACM (2004)
14. Li, J.S., Chen, L.C., Monaco, J.V., Singh, P., Tappert, C.C.: A comparison of classifiers and features for authorship authentication of social networking messages. Concurr. Comput.: Pract. Exp. **29**(14), e3918 (2017)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
16. Nirkhi, S.M., Dharaskar, R., Thakare, V.: Authorship identification using generalized features and analysis of computational method. Trans. Mach. Learn. Artif. Intell. **3**(2), 41 (2015)
17. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
18. Polychronakis, M., Provos, N.: Ghost turns zombie: exploring the life cycle of web-based malware. LEET **8**, 1–8 (2008)
19. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)