2016 15th IEEE International Conference on Machine Learning and Applications

Author Identification using Deep Learning

Ahmed M. Mohsen
Faculty of Engineering
Alexandria University
Email: ahmed.youssif5@alex-eng.edu.eg

Nagwa M. El-Makky
Faculty of Engineering
Alexandria University
Email: nagwamakky@alexu.edu.eg

Nagia Ghanem
Faculty of Engineering
Alexandria University
Email: nagia.ghanem@alexu.edu.eg

***Abstract*—Authorship identification is the task of identifying the author of a given text from a set of suspects. The main concern of this task is to define an appropriate characterization of texts that captures the writing style of authors. Although deep learning was recently used in different natural language processing tasks, it has not been used in author identification (to the best of our knowledge). In this paper, deep learning is used for feature extraction of documents represented using variable size character n-grams. We apply A Stacked Denoising Auto-Encoder (SDAE) for extracting document features with different settings, and then a support vector machine classifier is used for classification. The results show that the proposed system outperforms its counterparts**

Keywords*—denoising autoencoder, author identification, deep learning.

I. Introduction

Authorship Identification is the task of identifying who wrote a given piece of text from a given set of candidate authors (suspects). From machine learning perspective, it can be viewed as multiclass single-label text classification task where author represents a class (label) of a given text. Semantic features are used for topic-based classification, while stylometric features are used for author-based classification. Stylometric features focus on the patterns that appear in the text for the same author.

The study of stylometry and authorship goes back to the 19^{th} century, with Mendenhall [1] taking the lead by characterizing the style of different authors through the frequency distribution of words of various lengths. During the first half of the 20^{th} century, many statistical studies were followed introducing measures for writings styles (Stylometry) including *Zipfs distribution* and *Yules K measure*. Modern authorship identification started by Mosteller and Wallace [2] work on The Federalist Papers, where they applied Bayesian statistical analysis on the frequencies of a small set of function words (e.g. "and", "to", "the"), as stylistic features of the text.

In the literature many features have been proposed to capture stylistic features including vocabulary richness measures [3], syntactical features [4], function words frequencies and character n-gram frequencies [5].

Deep learning has been successfully applied to various natural language processing tasks producing performance results beating previously state of the art technique. For example, [6] applied deep learning on the Domain adaption of sentiment analysis by using the high-level feature representation extracted using a deep neural network and outperformed the state of the art methods on the classification task. Also, deep learning was applied in [7] to find a compact document representation to be used in the topic-based classification task. However, deep learning has not been used in author identification for free-form texts yet (to the best of our knowledge), although it has been applied in [8] for source code author identification, the domain studied in this paper is free-form text not source code.

Knowing that deep learning algorithms can learn better representation of data, we propose a deep learning approach for the problem of author identification. The intuition behind Deep Learning is that unsupervised learning could be used to set each level of a hierarchy of features, one level at a time, based on the features extracted at the previous level. These features have successfully been used to initialize deep neural networks. The contributions of this study are as follows:

- Proposing a Deep Learning feature extraction system for authorship identification.
- Investigating the usage of different feature selection and feature normalization on the input of the deep autoencoder.

The rest of the Paper is organized as follows. Section II describes the related work. Section III describes methodology of stacked denoising autoencoder and its training procedures. Section IV describes the proposed system architecture and implementation details. Section V reports the experiments results. Section VI states our conclusion. And finally future work is presented in Section VII.

II. Related Work

In this section, we review the previous work done in authorship analysis focusing on the stylometric features used for representing the text.

Several features have been proposed in the literature to describe the writing style of a given author. The most used features that have been used in previous studies to represent the text are listed below.

Lexical Features: Viewing a text as a sequence of tokens grouped into sentences. Those features could be divided into character-based and word-based lexical features. [5], [9], [10] Used character based features, while [11] used both character-based and word-based features.

Syntactic Features: The usage of syntactic information to fetch the unconscious syntactic patterns used by the authors at the sentence level, such as: part-of-speech, sentence structure, function words frequency and typos. [12] used a set of 150

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.45

function words, [13] used a set of most frequent words in the data set and [14] combined function words and punctuation features.

Content-specific Features: A set of features to be defined for a specific domain (topic) by domain experts. Those features are often avoided due to its inability to generalize in cross-topic settings. [12] Used features related to on-line messages domain, such as the usage of greetings in the messages, while [9] used measures related to HTML tag distribution in email messages domain.

Character n-grams are a widely used approach to represent text for stylistic purposes since they are able to capture nuances in lexical, syntactical, and structural level. This feature set is considered the state of the art feature set for author identification task [15]. In [11], the robustness of character-level 3-grams was proved in a cross-domain setting. Also, in [16], character-level n-grams outperformed various feature sets in multiple cross-topic setting. [5] Showed the effectiveness of the variable length character n-gram approach (i.e., the combination of 2-grams, 3-grams, 4-grams, etc.). In this paper, authorship identification is studied using variable length character n-gram features that are fed to an SDAE deep learning network.

III. Methodology Of Deep Learning

A. Deep Learning

Deep learning is a kind of representation learning in which there are multiple levels of features. These features are automatically discovered and they are composed together in the various levels to produce the output. Each level represents abstract features that are discovered from the features represented in the previous level. So, it could discover intermediate abstractions that could distinguish between authors by finding their writeprints [17]. In the following subsections, the training of a stacked denoising autoencoder [18] is explained.

B. AutoEncoder (AE)

AE is a neural network that constrains the output values to be the same as the input values by having the same number of nodes in both the input layer and the output layer. The training of AE is unsupervised and the features produced are good representatives for the input.

C. Denoising AutoEncoder (DAE)

DAE [18] injects artificial noise in the input and its objective is to learn the clean input from the noisy one. The learnt features are more prone to actual noisy inputs so that it represents that input better. It consists of 2 components: encoder and decoder where the encoder resides between the input and hidden layers whereas the decoder resides between hidden and output layers. Listed below is the training procedure for DAE.

Training:

1) Noise is added to the input $x \in \mathbb{R}^d$ so that it got transformed into $\widetilde{x}$ by setting some elements to zero or addition of Gaussian noise.

$$f_n(x) = \begin{cases} Binomial & x \in [0,1] \\ Gaussian & x \in \mathbb{R} \end{cases}$$

2) $\widetilde{x}$ is encoded to the hidden representation y using a non-linear transformation function f_e.

$$y = f_e(W\widetilde{x} + b)$$

where $y \in \mathbb{R}^h$ is the output of the hidden representation, $W \in \mathbb{R}^{hxd}$ is the weight matrix of the encoding layer, b is the bias and f_e is the encoding function, sigmoid have been chosen.

$$f_e(t) = \frac{1}{1+e^{-t}}$$

3) The hidden representation y is decoded to the reconstructed input z.

$$z = f_d(W^{'}y + b^{'})$$

where $z \in \mathbb{R}^d$, $W^{'} \in \mathbb{R}^{dxh}$ is the weight matrix of the decoding layer, tied weights have been used $W^{'} = W^T$, f_d is the decoding function. For real-valued x, a linear decoding function is used, when x ranges from 0 to 1, sigmoid is used.

$$f_d(t) = \begin{cases} \frac{1}{1+e^{-t}} & x \in [0,1] \\ t & x \in \mathbb{R} \end{cases}$$

4) the objective of DAE is for the output z to reconstruct the input x, so that the reconstruction error is the cost function for the network where cross-entropy function is used when x ranges between 0 and 1 and squared error function is used with real-valued x.

$$cost = \begin{cases} cross-entropy & x \in [0,1] \\ squared-error & x \in \mathbb{R} \end{cases}$$

D. Stacked Denoising AutoEncoder (SDAE)

A deep neural network architecture is formed by stacking DAEs, where the output of the encoding layer of the current DAE is used as the input for the next one as shown in figure 1. SDAE initializes the weights of the network to prevent it from reaching suboptimal solution.

The training of SDAE consists of 2 procedures: unsupervised pre-training then supervised fine-tuning. Given SDAE, which consists of L DAE, the training procedures are as follow:

Unsupervised pre-training:

1) For the k^{th} DAE, the training procedure described in section III-C is applied and the encoder function f_θ^k is learnt.
2) f_θ^k is applied on the clean input x^k to produce the next input for the $(k+1)^{th}$ DAE.

$$x^{k+1} = W^k x^k + b^k$$

3) step 1 & 2 are repeated for each DAE, for $k \in [0, L]$ where, x^1 represents the original input and x^L is the code features.

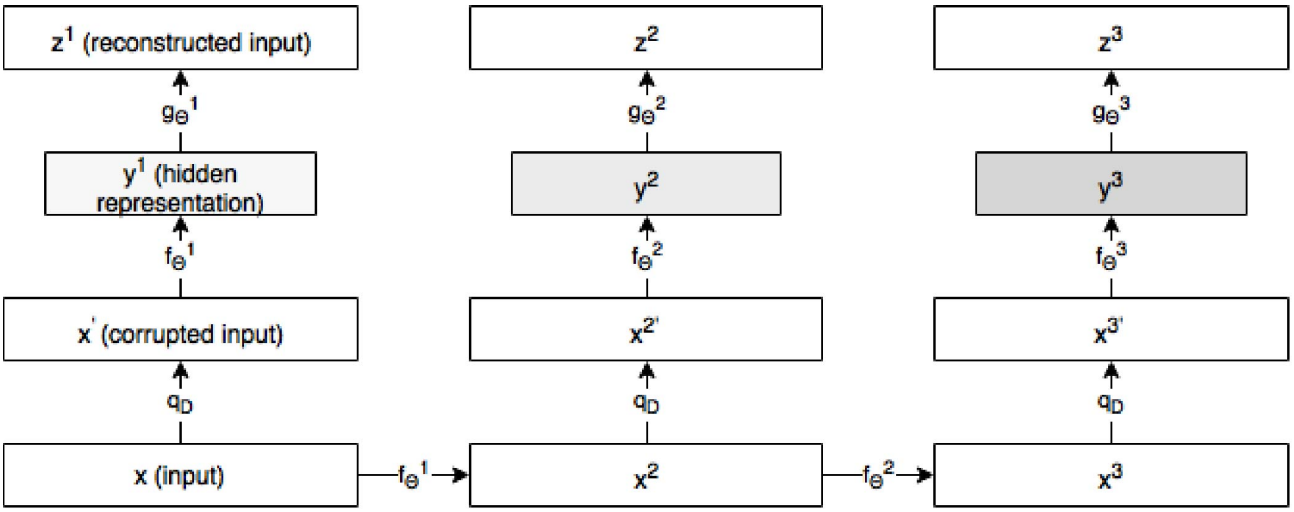

Fig. 1: Pre-training of a deep neural network, each autoencoder is trained separately in an unsupervised manner then the learnt f_{theta} is applied on the uncorrupted input to be fed for the next autoencoder as an input and the procedure is repeated layer by layer.

Pre-training is a layer-wise training as each layer is trained independently from the other layers. After pre-training is done the learnt weights are used to initialize the network instead of using random weights.

Supervised fine-tuning:

1) The encoding layers of each DAE is retained and got cascaded then a Logistic Regression (LR) layer is added on the top of the last encoder as shown in figure 2
2) The network weights are initialized using the learnt weights in the pre-training phase.
3) Using the labeled data, the whole parameters of the network is fine-tuned to minimize the cost function using backpropagation.

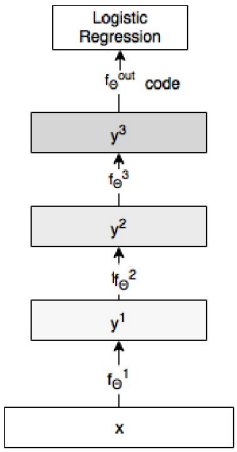

Fig. 2: Fine-tuning of a deep neural network, a logistic regression output layer is added at the top of the last encoding layer. The learnt weights from the pre-training phase are used as the initialization of the network weights and backpropagation is applied in a supervised manner.

After pre-training and fine-tuning, the feature extraction for a given input x is done by propagating the input from 1^{st} to L^{th} encoding layer and the extracted features are referred to as code (latent) features.

IV. The Proposed System Architecture and Implementation Details

In this section we describe the components of the proposed system for author identification task. As shown in figure 3, unstructured text input is converted to a vector of features using feature extraction, after that feature values are either being normalized or passed directly to feature selection where the most relevant features are being selected. The classifier takes 2 separate inputs, code features that have been compressed via SDAE and non-compressed features which, we refer to as "non-reduced features", The performance of the system under the two sets of features is reported in section V

A. Feature Extraction

Feature extraction is the conversion of unstructured text input into a vector of features to be used in machine learning algorithms.

Features: Many features have been proposed in previous studies. Character n-gram and frequent words are the most widely used and proven to be effective in the task of authorship identification.

Character n-gram: An n-gram is a sequence of n-contiguous characters. These features capture both the thematic as well as stylistic information of the texts, and hence have been proven to be very effective in previous identification studies [11], [5], [19].

Frequent Words: The origins of these features returns to topic-based classification where text is represented as bag of words of the most frequent words. In topic based classification stopping words are removed as they have no semantic value

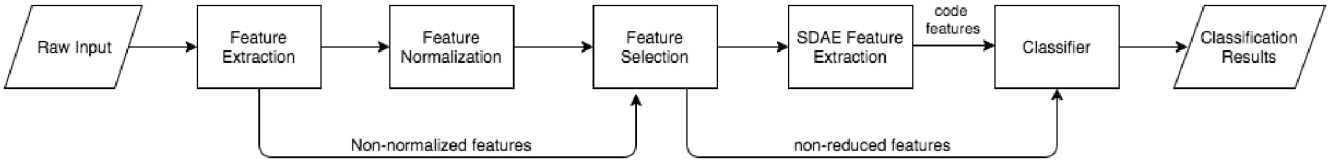

Fig. 3: The framework of the authorship identification

and words are stemmed to decrease the feature space, on the other hand stopping words (function words) carry a lot of stylistic information so, they are kept for author-based classification. [20], [11], [13] have applied those features in the author identification task.

B. Feature Normalization

Feature Normalization (scaling) is standardizing the range of the features. It has been shown that it helps gradient descent to converge much faster. The effeteness of normalization on the classification accuracy has been addressed.

Min Max Normalization: In this approach the data is scaled to a fixed range

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X is the original input vector, X_{max} is the upper bound of the range and X_{min} is the lower bound of the range. $[0, 1]$ range has been used (i.e. $X_{min} = 0$, $X_{max} = 1$).

C. Feature Selection

Feature selection techniques are used to remove redundant features and retain those features that have high information for a given category. The benefits of selecting useful features are: overfitting reduction, accuracy improvement, training time reduction.

We have compared 2 of the most popular technique with text classification and the results have been reported.

Frequency Based: Selects the top N most frequent features in the corpus.

Chi Square: χ^2 test measures the independence between the occurrence of a given feature within a given class. The top N features with highest scores are selected.

$$\chi^2 = \sum_{K=1}^{n} \frac{(O_K - E_K)^2}{E_K}$$

where K is the number of classes (authors), O_K is the observed value for the appearance in class k and E_K is the expected value for the appearance in class k.

D. SDAE Feature Extraction

A higher-level feature extraction is learnt using SDAE as described in section III, and the selected features are fed to the SDAE to extract code features. Comparisons have been made between code features and non-reduced features in the classification accuracies.

Theano [21] has been used for the SDAE implementation.

E. Classification

Support Vector Machine (SVM) is known to perform well on author identification task [15] due to its ability to deal with high dimensional data. In our research linear SVM have been used using 10-fold cross validation. *Scikit-learn* [22] implementation has been used with the default parameters.

V. Experiments

A. Corpus

The corpus used in this paper is a subset of the Reuters Corpus Volume 1 (RCV1). It has been labeled according to the writing authors. The top 50 authors of texts labeled with at least one subtopic of the class CCAT (corporate/industrial) were selected. The corpus is class-balanced where each author has written 100 documents. This corpus has been used in many previous studies including [5], [23]. The feature sets proposed by [13], [10] were applied to this corpus for comparing with our proposed system.

B. Setup

The dataset has been randomly divided into the following 3 sets.

- *Training set*: 60% of the dataset, used for both pre-training and fine-tuning of the models.
- *Validation set*: 10% of the dataset, used for choosing the best configuration of hyperparameters.
- *Test Set*: 30% of the dataset, used for the classification performance evaluation.

The whole dataset is then mapped to code features using the trained SDAE, and the classification accuracies between non-reduced features (without SDAE extraction) and code features have been reported.

C. SDAE Hyperparameters

Table I shows the parameters of the deep NN, a grid search have been used to choose the best parameter configurations regarding the validation set performance. Hidden units, noise level, and learning rate for all hidden layers have been chosen to be the same.

2 configurations for SDAE have been considered:

1) Linear decoder with linear squared error loss and Gaussian noise (for non- normalized inputs).
2) Sigmoid decoder with cross-entropy loss and Binomial noise (for normalized inputs).

Hyperparameter	Considered Values
number of hidden layers	3
number of hidden units	1000
pretraining epochs	50
pre-training learning rate	{0.00001, 0.0001, 0.001}
fine-tuning learning rate	{0.001, 0.01, 0.0001}
corruption noise level	{0.05, 0.10, 0.15, 0.30, 0.50}

TABLE I: List of hyperparameters for the deep network

D. Our proposed system vs. The previously used feature sets

In this section, we compare our proposed deep learning system with the most widely used features in literature.

The approaches of fixed-length character n-grams [10], variable-length character n-grams [5] and frequent words [13] were implemented and verified with the results in the given papers. Then they were compared with the code features (extracted using our deep learning system) on the RCV1 corpus as shown in Figure 4. A variable size n-gram ($[1-5]$) feature set was fed to the SDAE and the extracted features were used as the input for the classification task to demonstrate the classification accuracy boost when using deep learning compared to previous work.

All experiments have been implemented using a 10-fold cross validation classification on the same corpus with frequency-based feature selection.

As shown in figure 4, code features have outperformed all previously used features by a high margin. In the next subsections, the effectiveness of feature selection and feature normalization on the classification accuracy will be studied.

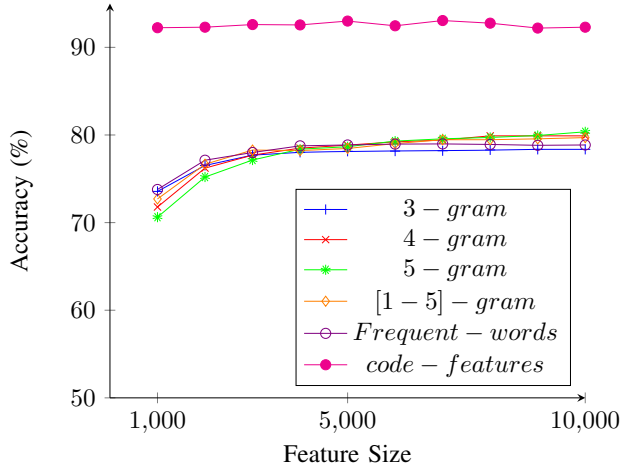

Fig. 4: The classification accuracy of author identification task using the feature sets proposed by previous studies vs. the feature set (code features) extracted using deep learning with the same corpus and classification settings.

E. Chi-square-based vs. Frequency-based feature selection

The usage of feature selection on the input to be fed to SDAE for feature extraction is investigated. Two feature selection methods were tested, frequency-based feature selection (referred to hereafter as *frequency*) and chi-square-based feature selection (referred to as *chi-square*).

Figure 5 shows that frequency-based feature selection performs better with relatively lower dimensional feature space whereas chi-square feature selection begins to performs better on higher dimensional feature space (above 8000 features).

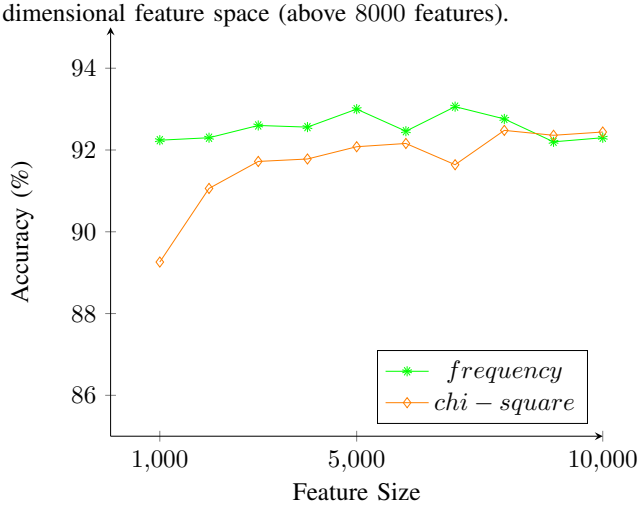

Fig. 5: Authorship identification accuracy comparison between the usage of frequency-based and chi-square-based feature selection methods on the SDAE input.

F. Linear decoder vs. Sigmoid decoder

A comparison between normalized input and non-normalized (frequencies) input are made where a linear decoder (referred to as *Linear*) is used with non-normalized input and sigmoid decoder (referred to as *Sigmoid*) is used with normalized input. Min-max normalization was used as a feature normalization method, while chi-square and frequency-based feature selection have been tested in both decoder settings.

Normalization of the input helps gradient descent to converge faster. As shown in figure 6, min-max normalization has achieved results higher than non-normalized input. In terms of feature selection, the observation of the previous section applies (i.e., chi-square feature selection outperforms frequency based only when large feature vectors were used). It has been shown that with input normalization, classification accuracy reaches up to 95.12% compared to 92.44% with non-normalized input.

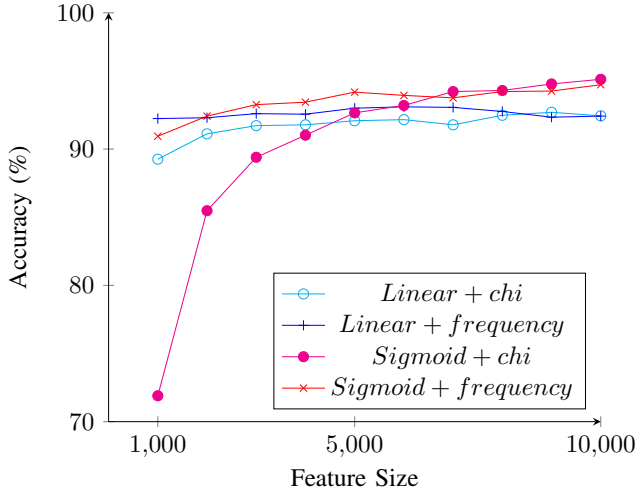

Fig. 6: Comparison between the usage of normalized input and non-normalized input for the proposed system, where sigmoid decoder is used with normalized input type and linear decoder is used with non-normalized (frequencies) input type.

VI. Conclusion

This paper has demonstrated that using deep learning (based on Stacked Denoising Autoencoders) can be successfully applied in author identification. The performance using the features extracted by the SDAE has been compared with the performance of the state-of-the art author identification techniques using the same corpus. The proposed system was able to outperform their results in terms of classification accuracy using a 10-fold cross validation settings.

It has been shown that chi-square-based feature selection outperforms frequency-based feature selection for relatively high feature space while the reverse is true for lower dimensional space. Using min-max normalization produces a higher accuracy than that produced without using normalization. Our system was able to reach classification accuracy up to 95.12% while the implemented feature sets used by related previous studies were only able to reach an accuracy around 80% for the same corpus.

It is to be noted that the robustness of the proposed system was verified under cross-topic and cross-genre data sets in our work [24]. However, the experimental results are not included here due to space limitations.

VII. Future Work

Our future work can be summarized in the following points:

- Investigate the usage of unlabelled data in a semi-supervised manner.
- Studying the sensitivity of the classification accuracy to the noise in SDAE
- Comparing the code features produced with pre-training only with the code features produced using pre-training and fine-tuning.
- Investigate the usage of convolution neural networks.
- Exploring more SDAE hyper parameters.

References

[1] Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, pages 237–249, 1887.

[2] Frederick Mosteller and David Wallace. Inference and disputed authorship: The federalist. 1964.

[3] Efstathios Stamatatos, Nikos Fakotakis, and Georgios Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35(2):193–214, 2001.

[4] David I Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

[5] John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 77–86. Springer, 2006.

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

[7] Marc'Aurelio Ranzato and Martin Szummer. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on Machine learning*, pages 792–799. ACM, 2008.

[8] Upul Bandara and Gamini Wijayarathna. Source code author identification with unsupervised feature learning. *Pattern Recognition Letters*, 34(3):330–334, 2013.

[9] Olivier De Vel. Mining e-mail authorship. In *Proc. Workshop on Text Mining, ACM International Conference on Knowledge Discovery and Data Mining (KDD2000)*, 2000.

[10] Efstathios Stamatatos et al. Ensemble-based author identification using character n-grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval*, pages 41–46, 2006.

[11] Efstathios Stamatatos. On the robustness of authorship attribution based on character n-gram features. *JL & Pol'y*, 21:421, 2012.

[12] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):378–393, 2006.

[13] Shlomo Argamon and Shlomo Levitan. Measuring the usefulness of function words for authorship attribution. In *Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, 2005.

[14] Harald Baayen, Hans Van Halteren, and Fiona Tweedie. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

[15] Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

[16] Upendra Sapkota, Thamar Solorio, Manuel Montes-y Gómez, Steven Bethard, and Paolo Rosso. Cross-topic authorship attribution: Will out-of-topic data help? In *COLING*, pages 1228–1237, 2014.

[17] Jiexun Li, Rong Zheng, and Hsinchun Chen. From fingerprint to writeprint. *Communications of the ACM*, 49(4):76–82, 2006.

[18] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[19] Jack Grieve. Quantitative authorship attribution: An evaluation of techniques. *Literary and linguistic computing*, 22(3):251–270, 2007.

[20] Neeraj Pradhan, Rachana Gogate, and Raghav Ramesh. Authorship identification of movie reviews. 2012.

[21] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Dmitry V Khmelev and William J Teahan. A repetition based measure for verification of text collections and for text categorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 104–110. ACM, 2003.

[24] Ahmed M. Mohsen. Author identification using deep learning. Master's thesis, Faculty of Engineering, Alexandria University, Egypt, 2016. unpublished thesis.