

Learning Stylometric Representations for Authorship Analysis

Steven H. H. Ding^{1b}, Benjamin C. M. Fung^{1b}, *Senior Member, IEEE*, Farkhund Iqbal, and William K. Cheung

Abstract—Authorship analysis (AA) is the study of unveiling the hidden properties of authors from textual data. It extracts an author's identity and sociolinguistic characteristics based on the reflected writing styles in the text. The process is essential for various areas, such as cybercrime investigation, psycholinguistics, political socialization, etc. However, most of the previous techniques critically depend on the manual feature engineering process. Consequently, the choice of feature set has been shown to be scenario- or dataset-dependent. In this paper, to mimic the human sentence composition process using a neural network approach, we propose to incorporate different categories of linguistic features into distributed representation of words in order to learn simultaneously the writing style representations based on unlabeled texts for AA. In particular, the proposed models allow topical, lexical, syntactical, and character-level feature vectors of each document to be extracted as stylometrics. We evaluate the performance of our approach on the problems of authorship characterization, authorship identification and authorship verification with the Twitter, blog, review, novel, and essay datasets. The experiments suggest that our proposed text representation outperforms the static stylometrics, dynamic n -grams, latent Dirichlet allocation, latent semantic analysis, distributed memory model of paragraph vectors, distributed bag of words version of paragraph vector, word2vec representations, and other baselines.

Index Terms—Authorship analysis (AA), computational linguistics, representation learning, text mining.

I. INTRODUCTION

THE PREVALENCE of the computer information system, personal computational devices, and the globalizing Internet have fundamentally transformed our daily lives and reshaped the way we generate and digest information. Countless pieces of textual snippets and documents are

generated every millisecond: This is the era of infobesity. Authorship analysis (AA) is one of the critical approaches to turn the burden of a vast amount of data into practical, useful knowledge. By looking into the reflected linguistic trails, AA is a study to unveil an underlying author's identity and sociolinguistic characteristics.

Studies of AA backed up by statistical or computational methods has a long history starting from 19th century [1], [2]. It has been a successful line of research [3]. Many customized approaches focusing on different subproblems and scenarios have been proposed [2]. Research problems in AA can be broadly categorized into three types.

- 1) *Authorship identification* (i.e., identify the most plausible author of an anonymous text snippet given a set of candidates [4]–[6]).
- 2) *Authorship verification* (i.e., verify whether or not a given candidate is the actual author of the given text [7]).
- 3) *Authorship characterization* (i.e., infer the sociolinguistic characteristics of the author of the given text [8]).

Both problems of authorship identification and authorship characterization can be formulated as a text classification problem. For the authorship identification problem, the classification label is the identity of the anonymous text snippet. For the authorship characterization problem, the label can be the hidden properties of the anonymous author, such as age and gender.

Regardless of the studied authorship problems, the existing solutions in previous AA studies typically consist of three major processes, as shown in the upper flowchart of Fig. 1: 1) feature engineering; 2) solution design; and 3) experimental evaluation. In the first process, a set of features are manually chosen by the researchers to represent each unit of textual data as a numeric vector. In the second process, a classification model is carefully adopted or designed. At the end, the entire solution is evaluated based on specific datasets. Representative solutions are [9]–[11]. Exceptions are few recent applications of the topic models [12]–[14] and text embedding models [15]–[17] that actually combine the first two processes into one. Still, the three-processes-based studies on AA problems dominate [7], [8], [18]. In the latest PAN2016 authorship characterization competition [8], 17 out of 22 approaches follow the three-processes-based solution. The other five approaches involve topical models.

To assist the feature selection process for AA, various feature selection algorithms have been proposed in the literature of AA [5], [19]–[21]. Some algorithms select features for representing a document by considering each feature individually

Manuscript received February 23, 2017; revised July 29, 2017 and October 2, 2017; accepted October 8, 2017. This work was supported in part by NSERC Discovery under Grant 356065-2013, in part by the Canada Research Chairs Program under Grant 950-230623, and in part by the Research Incentive through Zayed University, Abu Dhabi, UAE, under Grant RIF13059. This paper was recommended by Associate Editor M. Last. (*Corresponding author: Steven H. H. Ding.*)

S. H. H. Ding is with the School of Information Studies, McGill University, Montreal, QC H3A 1X1, Canada (e-mail: steven.h.ding@mail.mcgill.ca).

B. C. M. Fung is with the School of Information Studies, McGill University, Montreal, QC H3A 1X1, Canada, and also with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: ben.fung@mcgill.ca).

F. Iqbal is with the College of Technological Innovation, Zayed University, Abu Dhabi, UAE (e-mail: farkhund.iqbal@zu.ac.ae).

W. K. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: william@staff.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCYB.2017.2766189

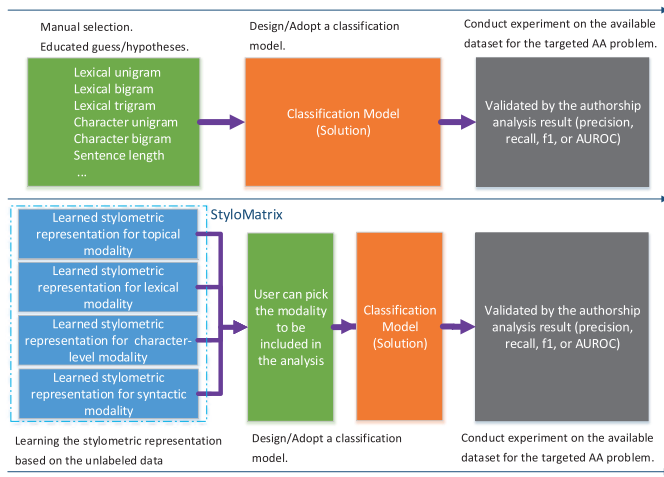


Fig. 1. Overview of the traditional solution and the proposed solution for AA.

with respective to their discriminant power [19], [21], while some algorithms include the classification or verification performance in the loop for feature selection [20] at the expense of longer computation. In addition, the representation learning approach has been proposed for text modeling [22] and classification [23], where the features are learned directly from the data in an unsupervised fashion. Inspired by the recent success of the representation learning approach in a variety of recognition tasks [24], we raise a new research question for AA. Given the *unlabeled* textual data, can we automatically come up with a vectorized numeric representation of the writing style?

In this paper, we present a stylistic representation learning approach for AA. Refer to the lower flowchart in Fig. 1. The goal is to learn an effective vector representation of writing style of different linguistic modalities in AA study. Following the previous work [5], [25], [26], we use the concept *linguistic modalities* to denote the categories of linguistic features [25]. We broadly categorize them into four modalities: 1) the *topical modality*; 2) the *lexical modality*; 3) the *character-level modality*; and 4) the *syntactic modality*. It is noted that the term “modality” used here is different from the term “multimodality” in machine learning. The former one denotes a category of linguistic features, and the latter denotes a combination of different ways in which information is presented, such as text, image, rating, etc. Also, we use the term *representation* and *embedding* to describe the vectorized representation of feature. In the first stage, we learn the stylistic representation for different linguistic modalities based on the unlabeled textual data. In the second stage, an authorship analyst can select the modality according to his or her needs. If the scenario requires the least interference from the topic-related information, the analyst can discard the topical modality, or more strictly, both topical and lexical modalities. One of the advantages for traditional feature engineering process is that an analyst can pick only relevant features to be included in the authorship studies. Our design inherits the flexibility of the original hand-crafted stylistic features while it enables the representation to be learned from the available data.

To the best of our knowledge, this is the very first work attempting to automate the feature engineering and discover the stylistic representations for AA. Specifically, our major contributions are summarized as follows.

- 1) We propose a joint learning model that can learn simultaneously the distributed word representation as well as the topical bias and lexical bias representations of each document based on unlabeled texts. The learned topical vector representation of a document captures the global topical context, while the learned lexical representation of a document captures the personal bias in choosing words under the given global topic.
- 2) We propose to learn the character-level and syntactic-level representations of each document. The former captures the morphological and phonemes bias of an author when he/she is composing a lexical token while the latter captures the syntactic/grammatical bias of an author when he/she is putting words together to construct a sentence.
- 3) We evaluate the effectiveness of the learned representations as stylometrics via extensive experiments and show its superiority over the state-of-the-art representations and algorithms for authorship verification, authorship identification and authorship characterization tasks using a number of benchmark datasets.

In practice, this paper suggests a different solution flow for AA. Based on the learned representations of the writing style corresponding to different linguistic modalities, the user/researcher can pick the modalities based on their needs and interests in the context of the AA problem. For example, political socialization researchers are interested in content, so they may choose topical modality. In contrast, cybercrime investigators would prefer avoiding topic-related features since given a harassment letter, the candidate authors may not have previously written anything on this topic. Our models are open-source.¹

The rest of this paper is organized as follows. Section II describes the related work. Section III elaborates our learning models. Section IV elaborates our evaluation on the authorship verification problem with the PAN2014 dataset. Section V studies the effect of the hyper-parameters choice and shows our experiment on the problem of authorship identification. Section VI presents our evaluation on the problem of authorship characterization. More relevant works are situated throughout the discussion in this paper. Finally, Section VII concludes this paper.

II. RELATED WORK

Stylometric features are the linguistic marks that quantify the linguistic characteristics [2], [3]. Various features have been proposed for the problems in AA. They can be categorized into dynamic features and static features based on how they are constructed [27]. Static features do not change over different datasets. They include context-free manually crafted styles such as sentence length [28], usage of functions words [1], [29], word-length distribution [30], [31],

¹Available at: <https://github.com/McGill-DMaS/StyloMatrix>.

vocabulary richness [32], [33], statistics over special characters and words [34], etc. In contrast, dynamic features are constructed based on the information of the dataset. They can be word n -grams, character n -grams, part-of-speech (POS) n -grams, and misspelled words, etc. Later, Seroussi *et al.* [14], [35] proposed to use topic models for the authorship attribution. These features can be also categorized according to their linguistic categories. Stamatatos [2] categorized them into lexical type, topical type, character type, syntactic type, semantic type, and application-specific type. Solorio *et al.* [25] and Sapkota *et al.* [26] used the word “modality” instead of the word “type” to describe a category. A modality denotes a single aspect of a given text snippet.

During the feature engineering process, given the available dataset and the application scenario, authorship analysts manually select a broad set of features based on the hypotheses or educated guesses, and then refine the selection according to experimental feedback. As demonstrated by previous research [5] and [19]–[21], the choice of the feature set (i.e., the feature selection method) is a crucial indicator of the prediction result, and it requires explicit knowledge in computational linguistics and tacit experiences in analyzing the textual data. Most of the existing studies in AA employ the filter-based approach [36] to select dynamic features. Abbasi and Chen [27] used information gain. Posadas-Durán *et al.* [37] used chi-square statistics. Savoy [19], [21] presented a comprehensive evaluation on filtering-based approaches. The study adopts different metrics such as document frequency, information gain, chi-square statistics, etc. It turns out that document frequency and information gain achieve the best result for authorship attribution. Zamani *et al.* [20] proposed a wrapper-based approach for the problem of authorship verification. They selected a distinct set of features for each author according to the performance on the training set. Layton [38] proposed to use an ensemble of classifiers that are built on different set of character n -grams for authorship verification. Besides of feature selection, [12]–[14] proposed to use latent variables in latent Dirichlet allocation (LDA) as document representation. References [39] and [40] are among the first studies that uses document representation learning for AA.

However, existing features suffer from several problems. First, all the features failed to separate the effect of topical preference and personal lexical preference. It is difficult to distinguish whether a specific lexical n -gram occurring in a sentence is mainly due to the holistic topics or the personal lexical preference. The LDA-based and latent semantic analysis (LSA)-based approaches also failed on this aspect. Second, the prevalent n -gram-based approaches failed to capture ordering information over long context and consider the semantic relationship between n -grams [22], [23]. Third, the effectiveness of the filtering metrics and the specific threshold are dataset and task dependent. Last, existing POS-based syntactic features failed to consider the tag dependency introduced by the POS tagger.

To address the above issues, we leverage the concept of representation learning to model writing style. Instead of manually specifying the features, we propose three models to learn

stylometric representation directly from the unlabeled text. Learning writing style representation is different to learning general text representation that only captures general topic or sentiment. The learned style representation needs to capture the differences in word choice under similar topic, the preferences in using function words, the morphology bias in word spelling, and the differences in grammatical structure. Le and Mikolov [23] proposed two similar neural network models to learn the vector representation of document: 1) the distributed memory model of paragraph vectors (PV-DM) model and 2) the distributed bag of words version of paragraph vector (PV-DBOW) model. The PV-DM model predicts the word in the middle of the sliding window. The input of the PV-DM model is a document vector and the vectors of words inside sliding window except the word in the middle. The document vector captures the topic that is missing from the context (i.e., sliding window). The PV-DBOW model takes a document vector as input. It predicts each word in the sliding window. The learned document vectors are effective for the sentiment prediction task [23]. However, it is not clear what is captured by the learned vectors. We leverage and manipulate basic elements of these two models in order to separate the effect of topical and lexical preference on token level, model the morphology and phonemes bias, and capture the grammatical variations.

III. MINING STYLOMETRIC REPRESENTATIONS

In this section, we present the proposed models for learning the stylometric representations on unlabeled training data. To be consistent in terminology, *text dataset* refers to the union of available labeled and unlabeled text; *writing sample* are used to refer to the minimum unit of text data to be analyzed. A writing sample consists of a list of sentences, and a sentence consists of a sequence of lexical tokens. Each lexical token has its respective POS tag in the corresponding sentence.

This section corresponds to the first process of the lower flowchart in Fig. 1, where only unlabeled text data are available. In this process, we learn the representation of each chosen unit of text into four vectorized numeric representations, respectively, for four linguistic modalities. We formally define the stylometric feature learning problem as follows.

Definition 1 (Stylometric Representation Learning): The given text dataset is denoted by \mathbb{D} , and each document is formulated as $\omega \in \mathbb{D}$. A document ω consists of a list of ordered sentences $\mathcal{S}(\omega) = s[1 : a]$, where s_a represents one of them. Each sentence consists of an ordered list of lexical tokens $\mathcal{T}(s_a) = t[1 : b]$, where t_b represents the token at index b . $\mathcal{P}(t_b)$ denotes the POS tag for token t_b . Given \mathbb{D} , the task is to learn four vector representations $\vec{\theta}_\omega^{\text{tp}} \in \mathbb{R}^{\mathcal{D}(\text{tp})}$, $\vec{\theta}_\omega^{\text{lx}} \in \mathbb{R}^{\mathcal{D}(\text{lx})}$, $\vec{\theta}_\omega^{\text{ch}} \in \mathbb{R}^{\mathcal{D}(\text{ch})}$, and $\vec{\theta}_\omega^{\text{sy}} \in \mathbb{R}^{\mathcal{D}(\text{sy})}$, respectively, for topical modality tp, lexical modality lx, character-level modality ch, and syntactic modality sy for each document $\omega \in \mathbb{D}$. $\mathcal{D}(\cdot)$ denotes the dimensionality for a modality.

A. Joint Learning of Topical Modality and Lexical Modality

In this section, we are interested in both topical modality and lexical modality. The topical modality concerns the

differences of topics, and the lexical modality is concerned with the personal preference of the word choice.

1) *Joint Modeling of Topical and Lexical Modalities*: A text document ω can be considered to be generated by the author under a mixture effect of topical bias, contextual bias, and lexical bias. It is difficult to distinguish whether a lexical token occurring in a sentence is mainly due to the topics of the document or the author's lexical preference. In order to best separate the mixed effects of topical bias, contextual bias, and lexical bias, we propose a joint learning model in which a document is considered as a lexical token picking process, and the author picks tokens from her vocabulary in sequence to construct sentences in order to express her interests. We consider three factors in this token picking process: 1) the topical bias; 2) the local contextual bias; and 3) the lexical bias.

- 1) *Topical Bias*: Based on the certain holistic topics to be conveyed through the text, the author is limited to a vague set of possible thematic tokens. For example, if the previously picked tokens are mostly about Microsoft, then the author will have a higher chance of picking the word "Windows" in the rest of the document because they are probably under a similar topic. Given the topics of the document, the author's selection of the next token in a sentence is influenced by a relevant vocabulary.
- 2) *Local Contextual Bias*: Holistic topics and local contexts both influence how the next word is chosen in a sentence. For example, a document about Microsoft may consist of several parts that cover its different software products. Moreover, the context can be irrelevant to the topic. For example, a Web blog may have an opening about weather that has nothing to do with the topic of the text.
- 3) *Lexical Bias*: Given the topics and their related vocabularies, the author has different choices for picking the next token to convey a similar meaning. For example, if the author wants to talk about the good weather, she may pick the adjective "nice" to describe the word "day." Alternatively, the author can pick other words such as "great," "wonderful," or "fantastic," etc. The variation in choosing different words to convey a similar meaning is the lexical bias for an author to construct the document.

The word picking process is a sequence of individual decision problems influenced by the individual topical bias, contextual bias, and lexical bias; therefore, it is natural to jointly learn the topical representation and lexical representation in the same model. It has the advantage of modeling their joint effects simultaneously and at best of minimizing the interference between the learned representations.

2) *Proposed Joint Learning Model*: This section introduces our proposed joint learning model for the topical modality and lexical modality. The goal is to estimate $\vec{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$ and $\vec{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$ in Definition 1.

Fig. 2 depicts the model, which is a neural network with two feed-forward paths. The first feed-forward path simulates the word picking process under a mixture effect of topical bias, local contextual bias, and lexical bias. The second feed-forward path captures the overall topics of the document. These two feed-forward paths have different inputs but share

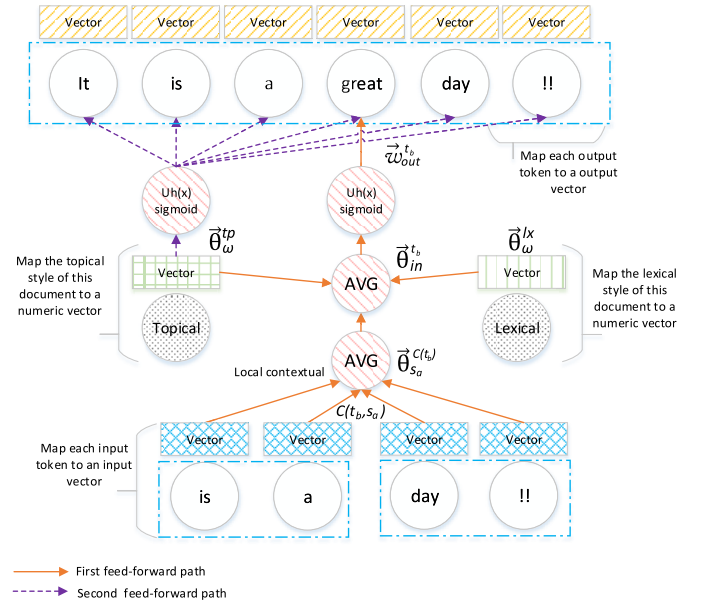


Fig. 2. Joint model for learning the stylistometric representation of the topical and lexical modalities. The input word vectors are randomly initialized before training. The output word vectors are zeros before training.

the same output vector space. The neural network updates the weights according to these two paths simultaneously at each training mini batch. The input to the whole neural network is the sliding window over a text sequence. The output of the first feed-forward path is the word in the middle of the sliding window. The output of the second feed-forward path is each of the words in the sliding window.

We start by describing the first feed-forward path. Recall that the contextual bias concerns the local information surrounding the token to be picked. We represent the vectorized local contextual bias surrounding token t_b in its corresponding sentence s_a as $\theta_{s_a}^{C(t_b)}$. The output is the prediction probability of the targeted word to be chosen by the author. The model tries to maximize the log probability for the first path

$$\arg \max \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \log \mathbf{P}(t_b | \underbrace{\vec{\theta}_\omega^{tp}}_{\text{topical}}, \underbrace{\vec{\theta}_\omega^{lx}}_{\text{lexical}}, \underbrace{\theta_{s_a}^{C(t_b)}}_{\text{contextual}}). \quad (1)$$

Similar to the other neural-network-based paragraph/word embedding learning models [22], [23], [41], this model maps each lexical token t_b into two vectors: $\vec{w}_{in}^{t_b} \in \mathbb{R}^{dw}$ (the blue rectangles in Fig. 2) and $\vec{w}_{out}^{t_b} \in \mathbb{R}^{dw}$ (the yellow rectangles in Fig. 2), where dw denotes the dimensionality. $\vec{w}_{in}^{t_b}$ is used to construct the input of contextual bias for the neural network, and $\vec{w}_{out}^{t_b}$ is used for the multiclass prediction output of the neural network. They are all model parameters to be estimated on the textual data.

The local context of a token is represented by its surrounding tokens in the window. Given a token t_b in a sentence s_a with a sliding window of size $\mathcal{W}(tp)$, the context of t_b is formulated as $\mathcal{C}(t_b, s_a) = \{t_{b-\mathcal{W}(tp)}, \dots, t_{b-1}, t_b, t_{b+1}, \dots, t_{b+\mathcal{W}(tp)}\}$. The contextual bias input to the neural network is defined as the average over the input mapped

vectors of $\mathcal{C}(t_b)$. We define $\langle \cdot \rangle$ as the vector element-wise average function

$$\theta_{s_a}^{\mathcal{C}(t_b)} = \left\langle \sum_t^{\mathcal{C}(t_b, s_a)} \tilde{w}_{in}^t \right\rangle. \quad (2)$$


The other two inputs to the model are the topical bias $\tilde{\theta}_\omega^{\text{tp}} \in \mathbb{R}^{\mathcal{D}(\text{tp})}$ and the lexical bias $\tilde{\theta}_\omega^{\text{lx}} \in \mathbb{R}^{\mathcal{D}(\text{lx})}$. In order to have the model working properly, we need to set $\mathcal{D}(\text{lx})$, $\mathcal{D}(\text{tp})$, and d_w equal to d_1 , where d_1 is the parameter of the whole model that indicates the dimensionality for the lexical modality representation, topical modality representation and contextual representation. With these three input vectors we further take their average as joint input vector $\theta_{in}^{t_b}$ since it is costly to have a fully connected layer

$$\tilde{\theta}_{in}^{t_b} = \left\langle \underbrace{\tilde{\theta}_\omega^{\text{tp}}}_{\text{topical}} + \underbrace{\tilde{\theta}_\omega^{\text{lx}}}_{\text{lexical}} + \underbrace{\tilde{\theta}_{s_a}^{\mathcal{C}(t_b)}}_{\text{contextual}} \right\rangle. \quad (3)$$

Example 1: Consider a simple sentence: $t_a = \text{"it is a great day !!"} in Fig. 2. For each token $\{t_b | b \in [1, 6]\}$ we pass forward the neural network. We take $b = 4$ and $t_b = \text{"great"}$ for example. The process is the same for other values of b . Given a window size of 2, which indicates two tokens on the left and two tokens on the right, we construct the local context as $\mathcal{C}(t_4, s_a) = \{t_2, t_3, t_5, t_6\} = \{\text{"is," "a," "day," "!!"}\}$. We map these tokens into their representations $\tilde{w}_{in}^{t_2}$, $\tilde{w}_{in}^{t_3}$, $\tilde{w}_{in}^{t_5}$, and $\tilde{w}_{in}^{t_6}$. With $\tilde{\theta}_\omega^{\text{tp}}$ and $\tilde{\theta}_\omega^{\text{lx}}$, we calculate $\tilde{\theta}_{in}^{t_4}$ using (3).$

Using a full soft-max layer to model (1) is costly and inefficient because of the large vocabulary V . Following recent development of an efficient word embedding learning approach [22], we use the negative sampling method to approximate the log probability:

$$\begin{aligned} \log \mathbf{P}(\tilde{w}_{out}^{t_b} | \tilde{\theta}_{in}^{t_b}) &\approx \log f(\tilde{w}_{out}^{t_b}, \tilde{\theta}_{in}^{t_b}) + \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} [\mathbb{I}[t \neq t_b]] \\ &\quad \times \log f(-1 \times \tilde{w}_{out}^t, \tilde{\theta}_{in}^{t_b}) \\ f(\tilde{w}_{out}^t, \tilde{\theta}_{in}^{t_b}) &= \text{Uh}\left((\tilde{w}_{out}^t)^T \times \tilde{\theta}_{in}^{t_b}\right). \end{aligned} \quad (4)$$


$\text{Uh}(\cdot)$ denotes the element-wise sigmoid function. It corresponds to the red circle  on the first feed-forward path in Fig. 2. $\mathbb{I}[\cdot]$ is an identity function. If the expression inside this function is evaluated to be true, then it outputs 1; otherwise 0. The negative sampling algorithm tries to distinguish the correct guess t_b with k randomly selected negative samples $\{t | t \neq t_b\}$ using $k+1$ logistic regressions. $\mathbb{E}_{t \sim P_n(t)}$ is a sampling function that samples a token v from the vocabulary V according to the noise distribution $P_n(t)$ of V .

Example 2: Continue from Example 1. We map t_4 into its output vector $\tilde{w}_{out}^{t_4}$. Next we calculate $\mathbf{P}(\tilde{w}_{out}^{t_4} | \tilde{\theta}_{in}^{t_4})$ using negative sampling (4). After that we calculate the gradients with respect to $\tilde{w}_{out}^{t_4}$ and $\tilde{\theta}_{in}^{t_4}$. We update $\tilde{w}_{out}^{t_4}$ according to its gradient with a learning rate. We also update $\tilde{w}_{in}^{t_2}$, $\tilde{w}_{in}^{t_3}$, $\tilde{w}_{in}^{t_5}$, $\tilde{w}_{in}^{t_6}$, and $\tilde{\theta}_\omega^{\text{lx}}$ equally according to the gradient of $\tilde{\theta}_{in}^{t_4}$.

The second feed-forward path of this model captures the topical bias reflected on the document ω . The topics reflected from the text can be interpreted as the union of effects of all

the local context in the sentence. Thus, the output of this path [see Fig. 2(left)] is a multiclass prediction of each word in the sentence s_a , which is denoted by $\mathcal{T}(s_a)$ in Definition 1. The goal is to maximize the log probability on $\tilde{\theta}_\omega^{\text{tp}}$ of document ω for each of its sentences $\mathcal{S}(\omega)$

$$\arg \max \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a}^{\mathcal{S}(\omega)} \sum_{t_b}^{\mathcal{T}(s_a)} \log \mathbf{P}\left(t_b | \underbrace{\tilde{\theta}_\omega^{\text{tp}}}_{\text{topical}}\right).$$

Similar to the first feed-forward path of this model, we map each lexical token at the output to a numeric vector $\tilde{w}_{out}^{t_b}$ (the yellow rectangles  in Fig. 2). By using negative sampling, we maximize the following log probability:

$$\begin{aligned} \log \mathbf{P}\left(t_b | \underbrace{\tilde{\theta}_\omega^{\text{tp}}}_{\text{topical}}\right) &\approx \log f(\tilde{w}_{out}^{t_b}, \tilde{\theta}_\omega^{\text{tp}}) + \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} \\ &\quad \times \left(\mathbb{I}[t \neq t_b] \log f(-1 \times \tilde{w}_{out}^t, \tilde{\theta}_\omega^{\text{tp}})\right). \end{aligned} \quad (5)$$

The total number of parameters is $(k+1) \times d_1$ for each t_b . Constant k is contributed by k negative samples, and constant 1 is contributed by the update of $\tilde{\theta}_\omega^{\text{tp}}$. Basically, the second feed-forward path of this model is an approximation to the full factorization of the document-term co-occurrence matrix.

Example 3: Continue from Example 1. For the output of the second path, we map each token into a numeric vector $\tilde{w}_{out}^{t_b}$, where $t_b \in \{\text{"it," "is," "a," "great," "day," "!!"}\}$. For each of the vectors we calculate $\mathbf{P}(\tilde{w}_{out}^{t_b} | \tilde{\theta}_\omega^{\text{tp}})$ in (5) using negative sampling. Then we calculate the derivatives for each $\tilde{w}_{out}^{t_b}$ and $\tilde{\theta}_\omega^{\text{tp}}$ and update them accordingly by multiplying the gradients with a prespecified learning rate.

In this model, we count punctuation marks as lexical tokens. Consequently, the information related to the punctuation marks is also included. Punctuation marks carry information of intonation in linguistics and are useful for AA [42]. After training the model on a given text dataset \mathbb{D} , we have a topical modality vector representation $\tilde{\theta}_\omega^{\text{tp}} \in \mathbb{R}^{d_1}$ and a lexical modality vector representation $\tilde{\theta}_\omega^{\text{lx}} \in \mathbb{R}^{d_1}$ for each document $\omega \in \mathbb{D}$. Also, for each lexical token $t_b \in V$ we have a vectorized representation $\tilde{w}_{in}^{t_b} \in \mathbb{R}^{d_1}$.

For an unseen document $\omega' \notin \mathbb{D}$ that does not belong to the training text data, we fix all the $\tilde{w}_{in}^{t_b} \in \mathbb{R}^{d_1}$ and $\tilde{w}_{out}^{t_b} \in \mathbb{R}^{d_1}$ in the trained model and only propagate errors to $\tilde{\theta}_{\omega'}^{\text{lx}} \in \mathbb{R}^{d_1}$ and $\tilde{\theta}_{\omega'}^{\text{tp}} \in \mathbb{R}^{d_1}$. At the end, we have both $\tilde{\theta}_{\omega'}^{\text{lx}}$ and $\tilde{\theta}_{\omega'}^{\text{tp}}$ for ω' .

The first feed-forward path corresponds to the PV-DM model in [23]. The second feed-forward path corresponds to the PV-DBOW model in [23]. The difference between this model and PV-DM/PV-DBOW is that we joint them by pushing the input of PV-DBOW to the input of PV-DM. The input of PV-DBOW (the topical vector in Fig. 2) captures the overall topic (i.e., word distribution) of the document. By pushing it to the input of PV-DM at each mini batch, the lexical vector captures what is missing from the topic and the current context or lexical preference, where people have different word choice under similar topic and similar context. Thus, it is very different from the PV-DBOW and PV-DM models.

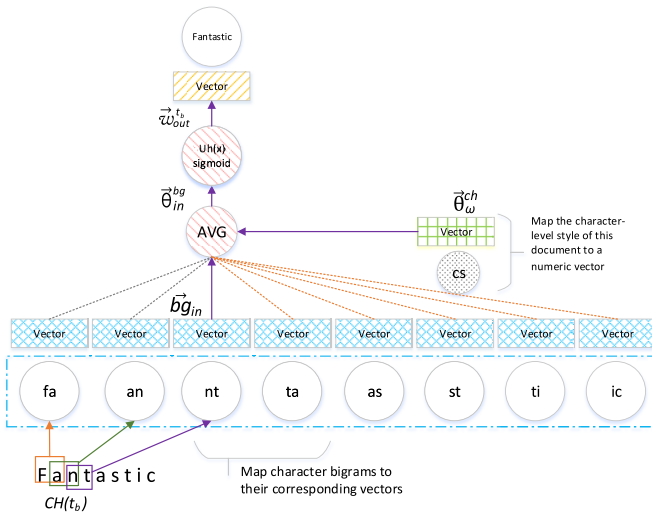


Fig. 3. Model for learning the representation of the character modality.

B. Character-Level Modality

We propose a neural-network-based model to learn the character modality representation on the plain text data. This model captures the morphological differences in constructing and spelling lexical tokens across different documents. Refer to Fig. 3. The input of this model is one of the character bigrams generated by a sliding window over a lexical token t_b with the character-level bias. The output of this model is the vectorized representation of the token t_b . The purpose is to learn $\vec{\theta}_\omega^{ch} \in \mathbb{R}^{\mathcal{D}^{(ch)}}$ for each document $\omega \in \mathbb{D}$ such that vector $\vec{\theta}_\omega^{ch}$ captures the morphological differences in constructing lexical tokens. Let $\mathcal{CH}(t_b) = bg[1 : c]$ denote the list of character bigrams of a given token t_b , and bg is one of them. The goal is to maximize the following log probability on \mathbb{D} :

$$\arg \max \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \sum_{bg} \log \mathbf{P} \left(t_b \mid \underbrace{\vec{\theta}_\omega^{ch}}_{\text{char-level}}, \vec{bg}_{in} \right).$$

We consider character bigram to increase the character-level vocabulary size. Increasing the length of character n -grams risks taking too much information from the lexical modality. For example, a character four-gram already matches a lot of exact words. Therefore, we only consider bigram at this stage. Similar to the previous lexical model, we map each lexical token t_b into a numeric vector $\vec{w}_{out}^{t_b}$, which is used to output a multiclass prediction. We also map each character bigram into a numeric vector \vec{bg}_{in} , which is used for the network input. Both are model parameters to be estimated. The input vectors of this model are $\vec{bg}_{in}^{b_i}$ and $\vec{\theta}_\omega^{ch}$. Both of them have the same dimensionality d_2 . After taking an average, it is fed into the neural network, as depicted in Fig. 3, to predict its corresponding lexical token t_b . By using negative sampling, we maximize the following log probability:

$$\vec{\theta}_{in}^{bg} = \left\langle \vec{\theta}_\omega^{ch}, \vec{bg}_{in} \right\rangle \quad (6)$$

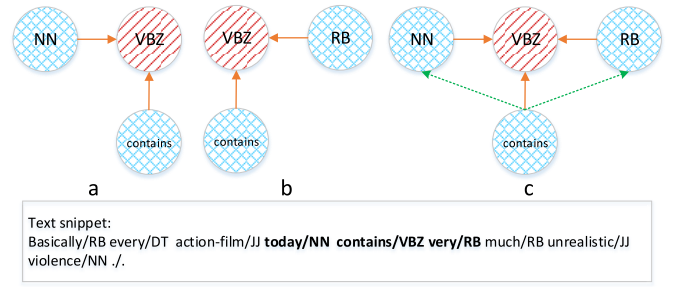


Fig. 4. Three typical inference structures for the POS tagger. Solid lines indicate dependencies introduced by tagger. (a) Left-to-right structure, (b) right-to-left structure, and (c) bidirectional structure.

$$\mathbf{P} \left(t_b \mid \underbrace{\vec{\theta}_\omega^{ch}}_{\text{char-level}}, bg \right) \approx \log f(\vec{w}_{out}^{t_b}, \vec{\theta}_{in}^{bg}) + \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} \times \left(\mathbb{I}[t \neq t_b] \log f(-1 \times \vec{w}_{out}^t, \vec{\theta}_{in}^{bg}) \right). \quad (7)$$

The number of parameters to be updated for each bigram bg of token t_b is $(k+2) \times d_2$. The constant k is contributed by the negative sampling function, and the constant 2 is contributed by $\vec{\theta}_\omega^{ch}$ and bg_{in} . To learn $\vec{\theta}_\omega^{ch}$, for $\omega' \notin \mathbb{D}$ we fix all $\vec{w}_{out}^{t_b}$ and bg_{in} and only propagate errors to $\vec{\theta}_{\omega'}^{ch}$.

Example 4: Consider a simple sentence: $t_a = \text{"Fantastic day !!"}$ in Fig. 3. For each token $\{t_b | b \in [1, 3]\}$ we extract its character bigrams. Suppose the word in the target is $t_1 = \text{"fantastic,"}$ and its bigrams are $\mathcal{CH}(t_1) = \{bg_c | c \in [1, 2, 3, 4, 5, 6, 7, 8]\} = \{\text{"fa," "an," "nt," "ta," "as," "st," "ti," "ic"}\}$. The process is the same for each word. Let us take a bigram $bg_1 = \text{"fa"}$ as an example. First, we map bg_1 to its representation \vec{bg}_{in} and map t_1 to its representation $\vec{w}_{out}^{t_1}$. With $\vec{\theta}_\omega^{ch}$, we calculate $\vec{\theta}_{in}^{bg}$ according to the first formula in (6). Then we calculate the forward log probability for $\mathbf{P}(\vec{w}_{out}^{t_1} | \vec{\theta}_{in}^{bg})$ in (7). We calculate the corresponding gradients and update the respective parameters. The training pass for bigram $bg_1 = \text{"fa"}$ is completed, and we move to the next bigram and following the sample procedure. After traversing all the bigrams we move to the next token $t_2 = \text{"day"}$.

The character modality in this paper only captures the intra-word information. It only concerns with the morphology and phonemes biases in the processing of spelling lexical word. The interword information is useful. It is captured by the lexical modality and the topical modality.

C. Syntactic Modality

Instead of using the typical POS n -grams as syntactic feature [43]–[45], we seek alternative to maximize the degree of variations that we can gain from the POS tags. First, we look into the state-of-the-art tagger models. Suppose we have a sentence s_a with its tokens $t_b \in \mathcal{T}(s_a)$. Recall that $\mathcal{P}(t_b)$ denotes the POS tag for the token t_b in the sentence. Refer to Definition 1. To assign a tag $\mathcal{P}(t_b)$ to a token t_b , there are three typical structures [46].

- 1) *Left-to-Right Structure:* This structure tries to maximize $\mathbf{P}(\mathcal{P}(t_b) | t_b, \mathcal{P}(t_{b-1}))$. The tag for token t_b is determined

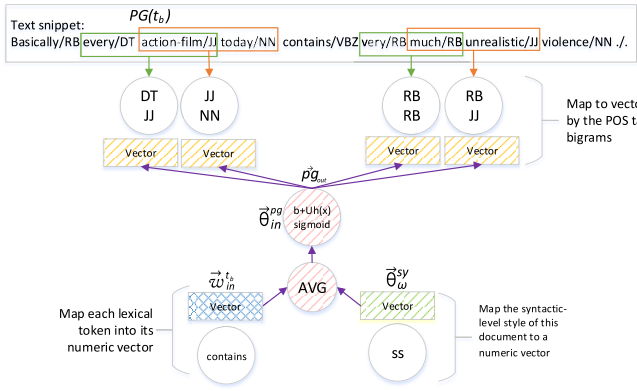


Fig. 5. Model for learning the representation of the syntactic modality.

by both lexical token itself and previous tag $\mathcal{P}(t_{b-1})$. Strong dependencies exist between $\mathcal{P}(t_{b-1})$ and $\mathcal{P}(t_b)$ and between $\mathcal{P}(t_b)$ and t_b [see Fig. 4(a)].

- 2) *Right-to-Left Structure*: This structure tries to maximize $\mathbf{P}(\mathcal{P}(t_b)|t_b, \mathcal{P}(t_{b+1}))$. The tag for token t_b is determined by both lexical token itself and next tag $\mathcal{P}(t_{b+1})$. Strong dependencies exist between $\mathcal{P}(t_{b+1})$ and $\mathcal{P}(t_b)$ and between $\mathcal{P}(t_b)$ and t_b [see Fig. 4(b)].
- 3) *Bidirectional Structure*: This structure combines the previous two. It maximizes $\mathbf{P}(\mathcal{P}(t_b)|t_b, \mathcal{P}(t_{b+1}), \mathcal{P}(t_{b-1}))$. The tag for token t_b is determined by both lexical token itself and surrounding tags $\mathcal{P}(t_{b+1})$ and $\mathcal{P}(t_{b-1})$. Strong dependencies exist between $\mathcal{P}(t_{b+1})$ and $\mathcal{P}(t_b)$, between $\mathcal{P}(t_b)$ and $\mathcal{P}(t_{b-1})$, and between $\mathcal{P}(t_b)$ and t_b [see Fig. 4(c)].

For all of these three structures, there exists a strong dependency between contiguous POS tags, as well as between the actual lexical token and its tag. Using POS tags n -grams as a stylistic feature is less effective than using character n -grams and lexical n -grams because the strong dependencies between contiguous POS tags introduced by the POS taggers are shared between different documents.

Therefore, we seek another way that has fewer dependencies introduced by the POS tagger. In Fig. 4(c), strong dependencies introduced by the tagger are shown as solid lines. We select two weak dependency links from t_b to $\mathcal{P}(t_{b+1})$ and from t_b to $\mathcal{P}(t_{b-1})$, as indicated by the dashed lines. The tagger only introduces indirect dependencies on these two paths. Thus, these two paths have more variations across different documents than the others, as indicated by solid lines. Formally, our model tries to maximize $\mathbf{P}(\mathcal{P}(t_{b-1}), \mathcal{P}(t_{b+1})|t_b)$, which is different from the typical structures for the taggers.

The number of unique POS tags is quite limited, so we use the bigrams of POS tags (see Fig. 5). Let $\mathcal{P}_2(t_b)$ be a POS tag bigram $[\mathcal{P}(t_b), \mathcal{P}(t_{b+1})]$, and $n^b \in \mathcal{P}\mathcal{G}(t_b) = \{\mathcal{P}_2(t_{b-3}), \mathcal{P}_2(t_{b-2}), \mathcal{P}_2(t_{b+1}), \mathcal{P}_2(t_{b+2})\}$ be the neighbor POS bigrams of token t_b . The goal is to maximize

$$\arg \max \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \sum_{n^b} \log \mathbf{P} \left(n^b \mid \underbrace{\vec{\theta}_{\omega}^{sy}}_{\text{syntactic}}, \vec{w}_{in}^{t_b} \right).$$

Similar to the previous models, this model maps each lexical token t_b into a numeric vector $\vec{w}_{in}^{t_b}$, and each of its neighbor POS bigrams maps into a numeric vector \vec{n}_{out}^b . The input of the model, denoted by $\vec{\theta}_{in}^n$, is the average of $\vec{w}_{in}^{t_b}$ and $\vec{\theta}_{\omega}^{sy}$, and the prediction is one of the target token t_b 's neighbor POS tag bigrams, as shown in Fig. 5. $\vec{w}_{in}^{t_b}$ and $\vec{\theta}_{\omega}^{sy}$ share the same dimensionality d_3 . By using negative sampling, we maximize the following log probability:

$$\begin{aligned} \vec{\theta}_{in}^n = \left(\vec{\theta}_{\omega}^{sy}, \vec{w}_{in}^{t_b} \right) \mathbf{P} \left(n^b \mid \underbrace{\vec{\theta}_{\omega}^{sy}}_{\text{syntactic}}, t_b \right) &\approx \log f \left(\vec{n}_{out}^b, \vec{\theta}_{in}^n \right) \\ + \sum_{i=1}^k \mathbb{E}_{n \sim P_n(n^b)} \llbracket n \neq n^b \rrbracket \log f \left(-1 \times \vec{n}_{out}^b, \vec{\theta}_{in}^n \right) \end{aligned} \quad (8)$$

where $P_n(n^b)$ denotes the negative sampling function for V_n .

Example 5: Consider a sentence and its corresponding sequence of POS tags in Fig. 5. For each token $\{t_b | b \in [1, 10]\}$ we extract its POS neighbor bigrams. Suppose the word in target is $t_5 = \text{"contains,"}$ and its POS neighbor bigrams are $\mathcal{P}\mathcal{G}(t_5) = \{\text{"DT JJ," "JJ NN," "RB RB," "RB JJ"}\}$ given a window size of 2. The process is the same for other lexical tokens. Let us take one of its (t_5 's) POS neighbor bigrams $n^5 = \text{DT JJ}$ as an example. First we map n^5 to its vectorized representation \vec{n}_{in}^5 and map t_5 to its representation $\vec{w}_{in}^{t_5}$. With $\vec{\theta}_{\omega}^{sy}$, we calculate $\vec{\theta}_{in}^n$ according to the first formula in (8). In combination with \vec{n}_{in}^5 , we calculate the forward log probability for $\mathbf{P}(\vec{n}_{in}^5 | \vec{\theta}_{in}^n)$ in (8). Then we calculate the corresponding gradients and update the respective parameters. The training pass for bigram $n^5 = \text{DT JJ}$ is completed, and we move to the next bigram JJ NN following the same procedure. After all the bigrams are processed, we move to the next token t_6 .

IV. EVALUATION ON AUTHORSHIP VERIFICATION

In this section, we evaluate the proposed models on the authorship verification problem. The problem is to verify whether or not two anonymous text documents ω_1 and ω_2 are written by the same author. We first train the three models mentioned in Section III on the unlabeled text data, and then we estimate the stylistic representations $\vec{\theta}_{\omega}^{tp} \in \mathbb{R}^{d_1}$, $\vec{\theta}_{\omega}^{lx} \in \mathbb{R}^{d_1}$, $\vec{\theta}_{\omega}^{ch} \in \mathbb{R}^{d_2}$, and $\vec{\theta}_{\omega}^{sy} \in \mathbb{R}^{d_3}$, respectively, for the two anonymous documents ω_1, ω_2 . The verification score is a simple cosine distance measure between the given two documents' stylistic representations. Formally, the solution outputs cosine similarity between two documents ω_1 and ω_2

$$\mathcal{Q}(\omega_1, \omega_2) = \text{cosine} \left(\vec{\theta}_{\omega_1}^v, \vec{\theta}_{\omega_2}^v \right) \quad v \in \{tp, lx, ch, sy\} \quad (9)$$

where v denotes the selected modality. It could be tp topical modality, lx lexical modality, ch character-level modality, sy syntactic modality, or their combinations. If more than one modality is selected, we concatenate their $\vec{\theta}_{\omega}^v$ into a single one for each ω . We use the area under receiver operating characteristic curve (AUROC) [47] as evaluation metric. It is a well-known evaluation measure for binary classifiers.

The AUROC measure captures the overall performance of the classifier when the threshold is varied.

TABLE I
PAN2014 AUTHORSHIP VERIFICATION DATASET. THE NUMBER IN
ROUND BRACKETS IS THE STANDARD DEVIATION

Training	#Problems	#Docs	#Tokens	Tokens per doc
Dutch-Essays	96	192	123,713	644 (551)
Dutch-Reviews	100	200	25,416	127 (66)
English-Essays	200	400	694,477	1,736 (1372)
English-Novels	100	200	723,412	3,617 (3973)
Greek-Articles	100	200	616,497	3,082 (2283)
Spanish-Articles	100	200	767,916	3,839 (2639)
Testing	#Problems	#Docs	#Tokens	Tokens per doc
Dutch-Essays	96	192	128,179	667.59 (522)
Dutch-Reviews	100	200	26,169	130.85 (81)
English-Essays	200	400	671,056	1,677 (1352)
English-Novels	200	400	2,831,531	7,078 (5091)
Greek-Articles	100	200	646,361	3,231 (2395)
Spanish-Articles	100	200	755,929	3,779 (2622)

TABLE II
CATEGORIES OF BASELINE STATIC FEATURES

Features	Count	Example
Lexical	105	Ratio of digits and vocabulary richness, etc.
Function words	150	Occurrence of <i>after</i>
Punctuation marks	9	Occurrences of punctuation !
Structural	15	Presence/absence of greetings
Domain-specific	13	Occurrences of word <i>contract</i> , and <i>time</i> , etc.
Gender-preferential	10	Ratio of words ending with <i>ful</i>

A. PAN2014 Authorship Verification Dataset

PAN provides a series of shared tasks on digital text forensics. the PAN2014 authorship verification dataset² consists of subdatasets of different languages and different types (see Table I). Each dataset consists of a number of verification problems. Each problem consists of a set of known documents, an unknown document and a label. It can be either true, which indicates that the same author wrote the known documents and the unknown document, or false, vice versa. The solution can produce an answer “I do not know.”

We preprocess the data by tokenization, detecting sentence boundaries, and parsing POS tags using the Stanford tagger [46] and OpenNlp tagger.³ We merge all known documents of a problem into a single one since they are written by the same author. As our approach does not require labeled data, we strip all the ground-truth labels for training. We tune the hyper-parameters for the proposed models and all the baseline models by cross-validating on each dataset of training datasets.

B. Baselines

We choose several most relevant approaches as baselines.

- 1) *Style*: It represents a document under 302 widely studied static stylometric features in [4] and [29]⁴ (Table II).
- 2) *Style+[k-freq-ngram]*: It adds $3 \times k$ dynamic features to the previous baseline. We select k lexical n -grams, k character n -grams, and k POS n -grams ($n \in 1, 2, 3$) by occurring frequency. We rank each group separately for $k \in \{500, 1000, 2000, 5000\}$.

²PAN2014 authorship verification. Available at <http://pan.webis.de/clef14/pan14-web/author-identification.html>.

³Available at <http://opennlp.apache.org>.

⁴Full list of features is available at <http://dmas.lab.mcgill.ca/fung/pub/Stylometric.pdf>.

- 3) *Style+[k-info-ngram]*: This approach is the same as the previous except that the n -grams are selected by the information gain. Information gain is calculated by using document id as label. We pick $k \in \{500, 1000, 2000, 5000\}$.
- 4) *Typed-n-gram*: The typed character n -gram approach proposed in [6]. Each n -gram is prefixed by its category.
- 5) *LDA and LSA*: The LDA learns latent semantic topics between the documents and the words by Gibbs sampling. It represents a document as a distribution over the latent topics. LSA learns a latent representation between document and word by factorizing the document-to-word occurring matrix. A document is represented as weights over k singular values.
- 6) *w2v-skipgram and w2v-cbow*: Two neural networks that learn the vector representations of words in a corpus [22]. We take the word vectors' average as a document vector.
- 7) *PV-DBOW and PV-DM*: Two neural networks that learn document representation [23] discussed in Section II.
- 8) Top five approaches reported in PAN2014 as well as the meta-classifier called *META-CLF-PAN14*.

These baselines cover both recent development in text embedding learning and authorship verification. We use cosine as document distance for all baselines. Following the same procedure, we train our models on the training set and choose the hyper-parameters by cross validation with training labels.

- 1) *Topical and Lexical Modality*: We select $d_1 = 200$ and a window size of 2 for datasets other than the Dutch Review. We set $d_1 = 300$ and a window size of 16 for the Dutch Review. In our interpretation, the authors talk about similar topic in a longer context than the other corpus.
- 2) *Character Modality*: We pick $d_2 = 300$.
- 3) *Syntactic Modality*: We pick $d_3 = 500$ for the Spanish Article and $d_3 = 300$ for the others.

The effect of choosing d_1 , window size $\mathcal{W}(\text{tp})$, d_2 , and d_3 will be further discussed in Section V. Evaluation results are reported based on the performance on the test dataset.

C. Performance Comparison

As indicated in Table III, our proposed *modality* models achieve the highest AUROC score on this problem. Specifically, on average the first-rated model is the joint learning model for lexical modality and the topical modality. This model also outperforms all the others on the English Essay dataset and the Dutch Essay dataset. The runner-up is the lexical modality representation that is learned in the joint learning model. It achieves the best performance on the Dutch Essay dataset, English Novel dataset, Greek Article dataset, and Spanish Article dataset. Character-level modality outperforms all the aforementioned baselines except that it has a comparable performance to the PV-DBOW+DM model. The syntactic modality does not perform as well as the lexical, topical, and character-level modalities; however, it still achieves better AUROC than the LSA, LDA, and other n -gram

TABLE III
PERFORMANCE COMPARISON FOR THE AUTHORSHIP VERIFICATION
PROBLEM ON THE PAN2014 DATASET. ENTRIES WITH* ARE THE
PERFORMANCE OF OUR PROPOSED APPROACHES. ENTRIES
WITH † ARE CITED PERFORMANCE

Approach	Dutch Essay	Dutch Review	English Essay	English Novel	Greek Article	Spanish Article	Avg.
[Lexical+Topical]*	0.998	0.744	0.887	0.767	0.924	0.934	0.881
[Lexical]†	0.998	0.658	0.885	0.799	0.949	0.937	0.871
PV-DBOW+PV-DM	0.979	0.670	0.847	0.738	0.934	0.859	0.838
[Character]†	0.960	0.642	0.854	0.758	0.889	0.911	0.836
META-CLF-PAN14†	0.957	0.737	0.781	0.732	0.836	0.898	0.824
PV-DBOW	0.985	0.656	0.848	0.711	0.868	0.870	0.823
[Topical]*	0.969	0.695	0.818	0.629	0.773	0.897	0.797
PV-DM	0.959	0.600	0.828	0.711	0.876	0.829	0.801
Khonji et al. [48]†	0.913	0.732	0.599	0.750	0.889	0.898	0.798
LSA-100	0.918	0.652	0.665	0.702	0.805	0.751	0.749
Moreau et al. [49]†	0.907	0.635	0.620	0.597	0.800	0.845	0.734
[Syntactic]†	0.819	0.504	0.804	0.681	0.712	0.736	0.724
w2v-skigram+cbow	0.896	0.641	0.503	0.675	0.848	0.781	0.724
Mayor et al. [50]†	0.932	0.569	0.572	0.664	0.826	0.755	0.720
w2v-skipgram	0.896	0.640	0.442	0.651	0.875	0.812	0.719
Frery et al. [51]†	0.906	0.601	0.723	0.612	0.679	0.774	0.716
w2v-cbow	0.838	0.612	0.521	0.689	0.832	0.775	0.711
Castillo et al. [52]†	0.861	0.669	0.549	0.628	0.686	0.734	0.688
Typed- <i>n</i> -gram [6]	0.781	0.575	0.515	0.607	0.803	0.804	0.681
LSA-100	0.784	0.520	0.390	0.499	0.900	0.606	0.617
LSA-200	0.503	0.646	0.714	0.388	0.520	0.629	0.600
LDA-200	0.717	0.456	0.416	0.442	0.893	0.596	0.587
Style+[500-info-gram]	0.574	0.524	0.490	0.678	0.594	0.642	0.584
Style	0.559	0.516	0.490	0.678	0.592	0.635	0.578
LSA-500	0.503	0.646	0.502	0.388	0.520	0.629	0.565
Style+[1000-info-gram]	0.437	0.507	0.490	0.677	0.577	0.642	0.555
Style+[5000-freq-gram]	0.498	0.465	0.471	0.650	0.573	0.661	0.553
Style+[5000-info-gram]	0.451	0.439	0.511	0.652	0.574	0.612	0.543
Style+[1500-info-gram]	0.368	0.471	0.490	0.678	0.566	0.647	0.537
Style+[2000-info-gram]	0.368	0.474	0.492	0.679	0.548	0.654	0.536
Style+[2000-freq-gram]	0.446	0.462	0.470	0.644	0.555	0.636	0.536
LDA-500	0.412	0.451	0.432	0.647	0.688	0.572	0.534
Style+[1500-freq-gram]	0.424	0.465	0.468	0.644	0.548	0.628	0.530
Style+[1000-freq-gram]	0.391	0.464	0.469	0.641	0.539	0.614	0.520
Style+[500-freq-gram]	0.360	0.458	0.462	0.644	0.520	0.592	0.506

approaches. It is noted that THE syntactic modality outperforms the other POS-tags-based approach, such as [53] and *n*-gram approaches, that involve POS tags.

Our proposed models perform better than LSA and LDA, and the LSA approaches outperform the LDA approaches. Our model jointly considers the effect of document-to-word relationship and word-to-word relationship. In contrast, LSA and LDA only consider the relationship between document and word. PV-DBOW and PV-DM outperform LSA and LDA. The neural-network-based models perform better than the others.

The *w2v*-related approaches, which learn document embedding by averaging the word embedding, do not perform as well as our proposed approaches and the PV-DM-related approaches that directly learn the document embedding. We also see that the overall performance on the formal writings is better than that on the nonformal writing. The overall performance on datasets that have more text is better than those that has less text, which is consistent with our expectation and the observation in our previous work [5]. The exception is the English Novel dataset. It has more text data but performs not as good as the English Essay dataset. Regarding the selection of *n*-grams, in this experiment information gain outperforms frequency when given the same *n*. It contradicts the observation reported in the survey [2]. The information-gain-based feature selection method mostly outperforms the frequency-based measure for the authorship verification problem.

V. EVALUATION ON AUTHORSHIP IDENTIFICATION

In this section, we experiment on the authorship identification problem. We study effect of different choices of the hyper-parameters and compare the performance between the proposed models and the relevant ones following the same

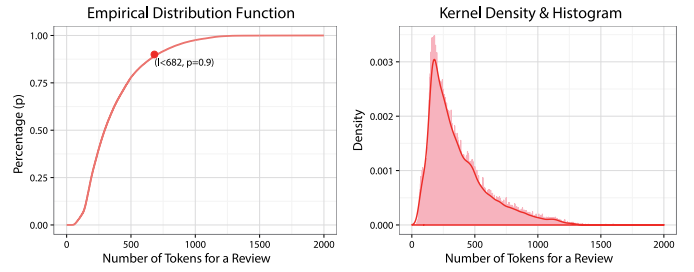


Fig. 6. Empirical distribution, kernel density, and histogram on the review length for the IMDB62 review dataset.

experimental setups. The problem is to identify the actual author of a given anonymous text snippet from a group of known candidate authors. Each known candidate author has a set of written text samples. It is a classification problem where the class label is the author name. To solve the problem, we treat all the text as unlabeled documents, and apply the proposed three models to learn vector representations for each document. Then we have four vector representations for each document ω : $\vec{\theta}_{\omega}^{tp} \in \mathbb{R}^{d_1}$, $\vec{\theta}_{\omega}^{lx} \in \mathbb{R}^{d_1}$, $\vec{\theta}_{\omega}^{ch} \in \mathbb{R}^{d_2}$, and $\vec{\theta}_{\omega}^{sy} \in \mathbb{R}^{d_3}$. We train a simple logistic regression model on the representations of a chosen modality and use it to classify the unknown document. The logistic regression needs labeled sampled to be trained. However, the underlying representation learning model does not rely on the labeled information.

A. IMDB62 Review Dataset

The IMDB62 dataset⁵ has been used by recent research [14], [35], [54] and enable a direct comparison between our proposed approaches and the state-of-the-art solutions. It contains 62 000 movie reviews by 62 prolific users from the movie review database IMDb.⁶ Each user wrote 1000 reviews. It is less formal than most datasets in the previous experiment. It contains spelling and grammatical errors. The authorship identification problem on this dataset is formulated as a 62-class classification problem. Fig. 6 shows the empirical distribution, kernel density and histogram on the reviews' length. Following the same experimental setup in [14], [35], and [54], we conduct a stratified tenfold cross validation experiment and report the best performance on accuracy.

B. Effect of Choosing Different Hyper-Parameter

There are four hyper-parameters for the aforementioned models. d_1 , d_2 , and d_3 , respectively, denote the vector size of the topical-lexical model, the character model, and the syntactic model. $\mathcal{W}(tp)$ is only for the topical-lexical model, which denotes the size of the sliding window for context.

Table IV shows the accuracy with varying $\mathcal{W}(tp) \in \{2, 4, 6, 8\}$. The overall performance increases as the sliding window size increases. When $\mathcal{W}(tp) = 8$ the topical-lexical model achieves the best performance. The topical modality of the joint learning model follows a similar trend. However, it is not significantly increased. The lexical modality follows a

⁵Available at <http://www.csse.monash.edu.au/research/umnl/data/>.

⁶<http://www.imdb.com>

TABLE IV

PERFORMANCE OF THE PROPOSED TOPICAL-LEXICAL MODEL WITH RESPECT TO DIFFERENT WINDOWS SIZE. VECTOR SIZE $d_1 = 200$

Windows size $\mathcal{W}(tp)$	= 2	= 4	= 6	= 8
Topical+Lexical	0.9032	0.9305	0.9310	0.9338
Topical	0.8327	0.8358	0.8367	0.8372
Lexical	0.7369	0.6682	0.6527	0.6470

TABLE V

PERFORMANCE OF THE PROPOSED MODELS WITH RESPECT TO DIFFERENT SIZE OF DIMENSION. $d_1 = d_2 = d_3 = d$. $\mathcal{W}(tp)$ IS SET TO 2

Vector size	$d=200$	$d=300$	$d=400$	$d=500$	$d=600$
Topical+Lexical	0.9032	0.9209	0.9310	0.9379	0.9436
Topical	0.8327	0.8665	0.8779	0.8927	0.9028
Lexical	0.7369	0.7793	0.7979	0.8005	0.8037
Character	0.7185	0.7283	0.7330	0.7348	0.7298
Syntactic	0.3894	0.5104	0.5352	0.5828	0.6009

reverse trend. Its performance decreases as the windows size increases. This table shows that, as the sliding window size increases, even though the performance of lexical modality decreases, but the performance of the combination increases.

Table V shows the accuracy with varying d_1 , d_2 , and d_3 . We set $d_1 = d_2 = d_3 = d$ and report the cross-validation accuracy on the dataset. We pick $d \in \{200, 300, 400, 500, 600\}$. As the vector size increases, the performance of the proposed models increases. Except the character modality. It reaches its best performance when $d_2 = 500$. Based on these two experiments, we pick $d_1 = 700$, $\mathcal{W}(tp) = 8$, $d_2 = 500$, and $d_3 = 600$ as our hyper-parameter on the IMDB62 dataset. We pick $d_1 = 700$ since we still see an obvious increase of accuracy when we increase d_1 from 500 to 600. Even though increasing the vector size beyond 600 and sliding window size beyond 8 can promote accuracy, we stay with $d_1 = 700$, $\mathcal{W}(tp) = 8$ since it already achieves the best results compared the baselines.

C. Baselines

We choose to compare our proposed models and all the methods reported in [14], [35], and [54] as well as available baselines in previous experiments.

- 1) *Token SVM*: An support vector machine (SVM) model trained on normalized token frequency features [14].
- 2) *Author-Topic (AT)-P*: A probabilistic attribution model AT-P [14] built on the top of an AT model in [55]. It generates each document according to the topic distribution of its observed author [14].
- 3) *DADT-P*: It is a combination of LDA and AT [14]. The model draws two disjoint set of words according to document topic and author topic. It separates words that discriminate documents and words that discriminate authors.
- 4) *LDA+Hellinger-S*: It merges writing samples of a candidate author into a profile [35]. After applying LDA, the Hellinger distance is used as the similarity between the anonymous document and an author profile.
- 5) *LDA+Hellinger-M*: This model is the same as the previous except that it does not merge documents. It uses

TABLE VI

PERFORMANCE ON THE IMDB62 DATASET WITH (a) MICRO f -MEASURE AND (b) ACCURACY. ENTRIES WITH \dagger ARE CITED PERFORMANCE. [-] INDICATES RANGE

(a)	(b)
[Lexical+Topical]* 0.972	Model Accuracy
SCAP [56] \dagger 0.948	[Lexical+Topical]* 0.972
Typed- n -gram [6] 0.936	Typed- n -gram [6] 0.937
Modality [Topical]* 0.930	[Topical]* 0.930
CNN-char [54] \dagger 0.917	Token SVM [14] \dagger 0.925
w2v-skigram+cbow 0.915	DADT-P [14] \dagger 0.918
LSA 0.907	w2v-skigram+cbow 0.916
CNN-word-char [54] \dagger 0.903	LSA 0.909
PV-DBOW+PV-DM 0.900	PV-DBOW+PV-DM 0.900
CNN-word-word-char [54] \dagger 0.884	AT-P [14] \dagger 0.896
Static+1000- n -gram 0.869	Static+1000- n -gram 0.870
CNN-word [54] \dagger 0.843	LDA+Hellinger-S [35] \dagger [0.80, 0.85]
SVM+Stems [54] \dagger 0.839	Imposters (KOP) \dagger [0.70, 0.75]
CNN-word-word [54] \dagger 0.820	[Lexical]* 0.742
Imposters (KOP) [57] 0.769	[Character]* 0.733
[Lexical]* 0.742	LDA+Hellinger-M [35] \dagger < 0.70
[Character]* 0.733	LDA 0.677
LDA+Hellinger-S \dagger 0.720	[Syntactic]* 0.601
LDA 0.665	
[Syntactic]* 0.601	
[Syntactic]* 0.601	

averaged Hellinger distance over samples of a given author.

- 6) *KOP*: A character n -gram approach proposed by Koppel *et al.* [57]. It evaluates a fraction of features to attribute the author, and repeats this process several times. A candidate's score is the portion of times being attributed as actual author.
- 7) *SVM+Stems*: An SVM classifier with stemmed words [54]. Words are weighted with tf-idf and scaled to unit variance.
- 8) *SCAP*: A source code authorship profiling approach proposed by Frantzeskou *et al.* [56] used in [54]. It uses the intersection of the most frequent character n -gram to score a candidate author.
- 9) *CNN-Word*: A convolutional layer with max-pooling is applied on the top of the concatenated word embeddings. A fully connected layer with dropout and soft-max predicts the author. It is proposed by Collobert *et al.* [58] and used in [54].
- 10) *CNN-Word-Word* [54]: Similar to CNN-word, but the input has an updatable word embedding and a nonupdatable word embedding from pretrained GloVe model [59].
- 11) *CNN-Char* [54]: Similar to CNN-word, but the input is an updatable character embedding channel.
- 12) *CNN-Word-Char* [54]: Similar to CNN-word, but the input has an updatable word embedding channel and a updatable character embedding channel as input.
- 13) *CNN-Word-Word-Char* [54]: It is a combination of CNN-word-word and CNN-char.

Table VI(a) compares our proposed models with baseline methods from [54] with respect to the micro f -measure. Still, the combination of the lexical modality and the topical modality performs the best. The topical-lexical combination as well as the topical modality along outperforms different variations of the convolutional neural network that contains more parameters. Similar to Table VI(b), the lexical modality, character modality and syntactic modality do not perform well.

TABLE VII
SUMMARY OF THE ICWSM TWITTER CHARACTERIZATION DATASET

Label type	Label	Users	Valid tweets	Tokens
Gender	Female	192	115,746	1,366,699
	Male	192	127,368	1,475,018
Age	(18 - 23)	194	104,686	1,473,512
	(25 - 30)	192	71,883	1,122,247
Political orientation	Republican	200	147,423	2,545,947
	Democrat	200	170,822	2,957,180

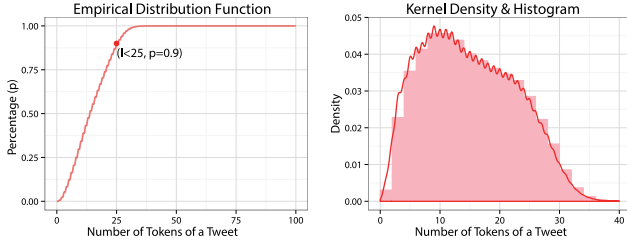


Fig. 7. Empirical distribution, kernel density, and histogram on the tweets length for the ICWSM2012 Twitter dataset.

D. Performance Comparison

Seroussi *et al.* [14], [35] used accuracy while Ruder *et al.* [54] used micro f -measure for evaluation. Table VI(b) compares our models with baselines from [14] and [35] with respect to the accuracy. Baseline performance are not concretely mentioned in [35]. We can only estimate a tie inclusive range of the accuracy from the diagram. The combination of the lexical modality and the topical modality performs the best, and the runner-up is the typed n -gram. Topical modality performs closely to the typed n -gram. Both lexical modality and character modality along do not perform as well as other state-of-the-art LDA-based methods such as DADT-P and AT-P. We also notice that the Token SVM as well as LDA-based models perform very well on this dataset, and we suspect that there is a strong topical correlation between reviews written by same author. If such a correlation exists, the lower accuracy achieved by the lexical modality and the character modality show that they carry less topical information than the topical modality.

VI. EVALUATION ON AUTHORSHIP CHARACTERIZATION

We evaluate the proposed models on the authorship characterization task, which is to identify the socio-linguistic characteristics of the author based on text. It has two paradigms. Instance-based paradigm assumes each document of an author is independent. Profile-based paradigm considers all documents by an author as a single one. This problem is mostly formulated as a document classification problem where labels can be age, gender, and political orientation, etc. We first learn the stylometric representations for all the documents. Then a logistic regression model is trained on vectors with known labels. Finally, the classifier predicts labels for the testing data.

A. Twitter Characterization Dataset

ICWSM2012 is a publicly available Tweets dataset with labels [60]. The labels in this dataset are generated

TABLE VIII
PERFORMANCE COMPARISON FOR THE AUTHORSHIP CHARACTERIZATION PROBLEM ON THE ICWSM2012 TWITTER DATASET. ENTRIES WITH † ARE CITED PERFORMANCE

Approach	Age	Gender	Political Orientation	Average
Modality [Lexical+Topical] *	0.7887	0.8308	0.9318	0.8504
Modality [Topical] *	0.7606	0.8423	0.9205	0.8411
Modality [Lexical] *	0.7782	0.8154	0.9148	0.8361
[60] (all info)†	0.7720	0.8020	0.9150	0.8297
Modality [Character] *	0.7711	0.7846	0.9034	0.8197
[60] (target user only)†	0.7510	0.7950	0.8900	0.8120
Static+[5000-freq-ngram]	0.7606	0.7615	0.8580	0.7934
Static+[2000-freq-ngram]	0.7500	0.7731	0.8523	0.7918
Static+[1500-freq-ngram]	0.7782	0.7308	0.8352	0.7814
PV-DBOW+PV-DM	0.7323	0.7346	0.8693	0.7787
w2v-skipgram	0.7112	0.7692	0.8380	0.7728
w2v-cbow+skipgram	0.7253	0.7692	0.8238	0.7728
w2v-cbow	0.7218	0.7807	0.8096	0.7707
Static+[1000-freq-ngram]	0.7465	0.7385	0.8097	0.7649
Static+[0500-freq-ngram]	0.7641	0.7192	0.8011	0.7615
LSA-k=200	0.6937	0.7577	0.8097	0.7537
LSA-k=100	0.6937	0.7538	0.8097	0.7524
LSA-k=500	0.7007	0.7500	0.8040	0.7516
Typed- n -gram [6]	0.6780	0.773	0.764	0.739
PV-DBOW	0.6936	0.6653	0.8409	0.7333
PV-DM	0.6901	0.6653	0.8409	0.7321
Static+[0200-freq-ngram]	0.7570	0.7000	0.7301	0.7290
LDA-k=500	0.6338	0.7423	0.8040	0.7267
LDA-k=100	0.6303	0.7462	0.7869	0.7211
Static+[1500-info-ngram]	0.7254	0.7000	0.6847	0.7034
Static+[5000-info-ngram]	0.6866	0.7231	0.6960	0.7019
Static+[2000-info-ngram]	0.7289	0.6962	0.6790	0.7014
Static	0.6904	0.7324	0.6769	0.6999
Static+[0500-info-ngram]	0.7394	0.6692	0.6847	0.6978
Static+[1000-info-ngram]	0.7113	0.7000	0.6818	0.6977
LDA-k=200	0.5986	0.7269	0.7585	0.6947
Modality [syntactic] *	0.6303	0.6654	0.6364	0.6440

semiautomatically and manually inspected [60]. This dataset consists of three categories of labels for 1170 Twitter users: age, gender, and political orientation (see Table VII). Due to the limitation of Twitter’s policy, the actual content of tweets were not included in the dataset; however, the available users’ IDs as well as the tweet IDs enable us to retrieve the tweets. We preprocess the dataset by removing all the non-ASCII characters and replace all the URLs with a special lexical token. We pretokenize the tweets and parse POS tags using the tagger from [61]. In this dataset there is other social-network-based information, such as the target user’s friends, the friends’ tweets, etc. Since we focus on writing style, we omit this information as well as those retweeted tweets.

- 1) *Gender*: The label is either *male* or *female*. The labels are generated based on the Twitter user’s name with a name-gender database, and are manually inspected.
- 2) *Age*: The label is either 18–23 or 25–30, which is generated by birthday related tweets, e.g., “happy birthday to me.”
- 3) *Political Orientation*: The labels can be either *Democrat* or *Republican*, collected from the *wefollow* directory [60].

Fig. 7 shows the empirical distribution, kernel density and histogram on the tweets’ length. The 90% of the tweets have less than 25 tokens and most have a length of around ten tokens. We combine all tweets of a single user into a single document and treat each tweet as an individual sentence. Following the same setup in [60], we conduct a tenfold cross validation on the Twitter dataset and measure the accuracy.

TABLE IX

PERFORMANCE ON THE PAN2013 AUTHORSHIP CHARACTERIZATION DATASET ACCURACY. ENTRIES WITH † ARE CITED PERFORMANCE

	EN		ES		
	Gender	Age	Gender	Age	Avg
w2v-skigram+cbow	0.599	0.670	0.654	0.671	0.648
[Lexical+Topical]*	0.598	0.649	0.654	0.679	0.645
PV-DBOW+PV-DM	0.605	0.651	0.649	0.669	0.643
[Topical]*	0.589	0.637	0.648	0.677	0.638
[Character]*	0.591	0.644	0.640	0.660	0.634
López-Monroy et al. [62]†	0.569	0.657	0.630	0.656	0.628
[Lexical]*	0.592	0.634	0.627	0.649	0.626
Santosh et al. [63]†	0.565	0.641	0.647	0.643	0.624
Static+1000-ngram	0.572	0.656	0.612	0.641	0.620
LSA-800	0.588	0.631	0.582	0.625	0.607
[PAN16 2 nd] [64]	0.588	0.631	0.582	0.625	0.607
[PAN16 1 st] [65]	0.594	0.570	0.615	0.623	0.600
LDA-800	0.589	0.640	0.579	0.590	0.600
Cruz et al. [66]†	0.546	0.597	0.617	0.622	0.596
Ladra et al. [67]†	0.561	0.612	0.614	0.573	0.590
[Syntactic]*	0.560	0.605	0.554	0.608	0.582
Lim et al. [68]†	0.567	0.610	0.547	0.571	0.574
Typed- <i>n</i> -gram	0.593	0.432	0.607	0.645	0.569
Modaresi et al. [64]†	0.593	0.432	0.607	0.645	0.569
Flekova et al. [69]†	0.534	0.529	0.610	0.597	0.568
Meina et al. [70]†	0.592	0.649	0.529	0.493	0.566
Kern et al. [71]†	0.527	0.569	0.571	0.538	0.551
Pavan et al. [72]†	0.500	0.606	0.500	0.564	0.543
Gillam et al. [73]†	0.541	0.603	0.478	0.538	0.540

1) *Baselines*: We inherit the same set of baselines used in the authorship verification experiment, except for those studies reported in PAN2014 [18]. The baselines are used with a logistic classifier. We also include two baselines.

1) *Target User Info* [60]: An SVM-based model trained on the token-based text features and the socio-linguistic features.

2) *All Info* [60]: It is the same SVM-based model with additional social-network features.

The runtime of cross-validation is prohibitively expensive due to the large number of records. We did not hard tune the hyper-parameter on this dataset. Instead, we heuristically pick $d_1 = 400$, $d_2 = 400$, $d_3 = 400$, and $\mathcal{W}(\text{tp}) = 8$ based on our observation in the previous experiment. Vector size 400 is a typical value suggested by Le and Mikolov [23]. For other baselines in the previous section we use their default hyper-parameter.

2) *Performance Comparison*: Table VIII shows that the lexical+topical modality achieves the highest accuracy value. The runner-up is the topical modality. The character-level modality does not perform as well as the other two. The lexical+topical modality and the character-level modality also outperform the PV-DM-related models, w2v-related models, and other dynamic *n*-gram-based models. Unlike the results for the authorship verification problem, the w2v-related baselines perform fairly well. They achieve a higher accuracy value than PV-DM, PV-DBOW, LSA, and LDA.

We notice that the *target user only* approach and the *all info* approach [60] have more advantages over the proposed models and baselines. First, they use an SVM model that typically outperforms a logistic regression model given the same data. Second, our approaches only consider the stylometric information reflected from the text. Other socio-linguistic, behavioral, and social-network-related information is discarded. However, these two baselines achieve a lower accuracy value than our proposed joint model for lexical and topical modality.

TABLE X

TRAINING AND TESTING TIME FOR THE PAN2013 AUTHORSHIP PROFILING DATASET

Model	Hours:Minutes
Lexical+Topical	7:55
PV-DBOW+DM	6:00
Typed- <i>n</i> -gram	5:19
w2v-skigram+cbow	5:26
LSA	1:25
LDA	12:11
Static- <i>n</i> -gram	4:04
Vollenbroek et al. [65] [PAN16 1st]	11:44
Modaresi et al. [64] [PAN16 2nd]	3:39

Table VIII also shows that the proposed syntactic representation learning model does not perform well on the ICWSM2012 dataset, which is different from the previous authorship verification problem. This is because the tweet text data are relatively more casual than essay and novel, which does not introduce much variation in the grammatical bias. Moreover, it is difficult to determine the correct POS tags for tweets. Regarding the feature selection measure, the frequency-based approach outperforms the information-gain-based approach. Even the top-100 frequency-ranked *n*-grams outperform top-1500 information-gain-ranked *n*-grams, which is different from the result in previous verification experiment. Such a difference further confirms our argument that feature selection metrics are scenario-dependent. Even the feature set is dynamically constructed based on a different dataset, the measurement for the selection process is data-dependent. A language model over text is better.

B. PAN2013 Authorship Characterization Dataset

Additionally we benchmark the PAN2013 blog post dataset [74]. It contains an English (EN) dataset and a Spanish (ES) dataset. Each dataset consists of a list of blog post, and each blog post is labeled with the age and the gender of the actual author. The age of the author falls into: 10 s (13–17), 20 s (23–27), and 30 s (33–47). The gender of the author falls into: male and female. Each dataset comes with a training set and a testing set. This dataset covers a wide spectrum of topics. There are total 236 600 authors in the training set and 25 440 authors in the testing set for English. There are 75 900 authors in the training set and 8160 authors in the testing dataset for Spanish. More than 80% Spanish blogs have about 15 words.

In this experiment, we compare our proposed models with the top ten models reported in [74], top two models from PAN2016 competition [8], and available baselines from previous experiments. Hyper-parameters tuning using cross-validation is again infeasible due to the large size of the dataset. Instead, we heuristically set $d_1 = 400$, $d_2 = 300$, $d_3 = 500$, and $\mathcal{W}(\text{tp}) = 6$ by considering the size of the dataset and the length of text samples. We run the proposed models on the blog posts and use a logistic model for classification. Following the setup in [74] we use the classification accuracy as performance measure. Table IX compares the proposed models and the baselines. The Lexical+Topical model, the topical model, and the character model all perform better than the top two models from PAN2016 and the best methods

TABLE XI
WILCOXON SIGNED-RANK TEST OVER ALL THE DATASETS. ○, ●, AND ●, RESPECTIVELY, INDICATE $p > 0.05$, $p \leq 0.05$, AND $p \leq 0.01$. (*) INDICATES THE AVERAGED PERFORMANCE

	Lexical +Topical	DBOW +DM	Typed n -gram	skigram +cbow	LSA	LDA	Static n -gram
Lexical+Topical (.822)	○	●	●	●	●	●	●
DBOW+DM (.782)	●	○	●	○	●	●	●
Typed- n -gram (.697)	●	●	○	●	○	○	○
skigram+cbow (.739)	●	○	●	○	○	●	●
LSA (.733)	●	●	○	○	○	●	○
LDA (.641)	●	●	○	●	●	○	○
Static+ n -gram (.661)	●	●	○	●	○	○	○

reported in [74]. The skigram+cbow model achieves the highest average score. However, there is only a slight difference among skigram+cbow, Lexical+Topical, and PV-DBOW+DM models. In general, text representation learning methods outperform n -gram-based dynamic approaches. The performance on the Spanish dataset is better than the English dataset, which is out of our expectation. A potential interpretation is that Spanish has more expressed gender marks than English [75]. We also report the runtime information in Table X.

C. Overall Comparison

We further collect the results for above experiments and conducted a Wilcoxon signed-rank test for different baselines across different dataset (see Table XI). The difference between the proposed approach and the relevant baselines is significant ($p < 0.01$). PV-DBOW+PV-DM model performs close to the skipgram+cbow model. LSA is generally better than LDA. These approach generally outperforms dynamic n -grams ($p < 0.05$). In all the experiments, the topical and lexical models perform generally well. In our interpretation, the topical and lexical factors play a significant role in determining the author's identity and characteristics for these datasets. For example, the n -gram-based approaches work very well in the IMDB dataset. The PAN2014 dataset has some cross-topic problems. Therefore, n -gram-based approach does not perform very well. In the future we will explore cross-domain datasets.

VII. CONCLUSION

In this paper, we present our three models for learning the vectorized stylometric representations of different linguistic modalities for AA. To the best of our knowledge, it is the very first work introducing the problem of stylometric representation learning into the AA field. By using the proposed models, guided by the selected linguistic modality, we attempt to mitigate the issues related to the feature engineering process in current authorship study. Our experiments on the publicly available benchmark datasets for the authorship verification problem, the authorship identification problem, and the authorship characterization problem, demonstrate that our proposed models are effective and robust on different datasets and AA problems.

We find that the proposed models work well for prolific authors. For short text its performance will degrade. Our future research will focus on exploring better models to capture writing styles. A recurrent neural network is more suitable for

capturing the contextual relationship over long text. For learning the syntactic modality representation, a recursive neural network that operates on the fully parsed syntactic tree will be more suitable for the nature of grammatical variations than the current one. Moreover, this paper only focuses on capturing the variations of writings in Indo-European languages. Additional changes need to be applied for text in other languages where the word boundary is absent.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and the editor for the thorough reviews and valuable comments, which significantly improve the quality of this paper.

REFERENCES

- [1] F. Mosteller and D. L. Wallace, *Inference and Disputed Authorship: The Federalist*. London, U.K.: Addison-Wesley, 1964.
- [2] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009.
- [3] M. Brennan, S. Afroz, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Trans. Inf. Syst. Security*, vol. 15, no. 3, p. 12, 2012.
- [4] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "A unified data mining solution for authorship analysis in anonymous textual communications," *Inf. Sci.*, vol. 231, pp. 98–112, May 2013.
- [5] S. H. H. Ding, B. C. M. Fung, and M. Debbabi, "A visualizable evidence-driven approach for authorship attribution," *ACM Trans. Inf. Syst. Security*, vol. 17, no. 3, 2015, Art. no. 12.
- [6] U. Sapkota, S. Bethard, M. Montes-Y-Gómez, and T. Solorio, "Not all character n -grams are created equal: A study in authorship attribution," in *Proc. Annu. Conf. North Amer. Chapter ACL Human Lang. Technol.*, 2015, pp. 93–102.
- [7] E. Stamatatos *et al.*, "Overview of the author identification task at PAN 2015," in *Proc. Working Notes Papers CLEF Eval. Labs*, 2015.
- [8] F. R. Pardo *et al.*, "Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations," in *Proc. Working Notes Papers CLEF Eval. Labs*, 2016.
- [9] J. D. Burger, J. C. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on twitter," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Edinburgh, U.K., 2011, pp. 1301–1309.
- [10] S. Nirakhi and R. V. Dharaskar, "Comparative study of authorship identification techniques for cyber forensics analysis," *CoRR*, vol. abs/1401.6118, 2014.
- [11] T. Cavalcante, A. Rocha, and A. Carvalho, "Large-scale micro-blog authorship attribution: Beyond simple feature engineering," in *Proc. 19th Iberoamer. Congr. Progr. Pattern Recognit. Image Anal. Comput. Vis. Appl. (CIARP)*, 2014, pp. 399–407.
- [12] N. Pratanwanich and P. Liò, "Who wrote this? Textual modeling with authorship attribution in big data," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDM)*, Shenzhen, China, 2014, pp. 645–652.
- [13] J. Savoy, "Authorship attribution based on a probabilistic topic model," *Inf. Process. Manage.*, vol. 49, no. 1, pp. 341–354, 2013.
- [14] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with topic models," *Comput. Linguistics*, vol. 40, no. 2, pp. 269–310, 2014.
- [15] P. Shrestha *et al.*, "Convolutional neural networks for authorship attribution of short texts," in *Proc. Eur. Chapter Assoc. Comput. Linguistics*, Valencia, Spain, 2017, pp. 669–674.
- [16] Z. Ge, Y. Sun, and M. J. T. Smith, "Authorship attribution using a neural network language model," in *Proc. AAAI Conf.*, Phoenix, AZ, USA, 2016, pp. 4212–4213.
- [17] Y. Sari, A. Vlachos, and M. Stevenson, "Continuous n -gram representations for authorship attribution," in *Proc. Eur. Chapter Assoc. Comput. Linguistics*, Valencia, Spain, 2017, pp. 267–273.
- [18] F. Rangel *et al.*, "Overview of the 2nd author profiling task at PAN 2014," in *Proc. Conf. Labs Eval. Forum (Working Notes)*, 2014, pp. 898–827.
- [19] J. Savoy, "Authorship attribution based on specific vocabulary," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, 2012, Art. no. 12.

- [20] H. Zamani *et al.*, "Authorship identification using dynamic selection of features from probabilistic feature set," in *Proc. Int. Conf. Inf. Access Eval. Multilinguality Multimodality Interact.*, Sheffield, U.K., 2014, pp. 128–140.
- [21] J. Savoy, "Feature selections for authorship attribution," in *Proc. 28th Annu. ACM Symp. Appl. Comput. (SAC)*, Coimbra, Portugal, 2013, pp. 939–941.
- [22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.
- [23] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, 2014, pp. 1188–1196.
- [24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [25] T. Solorio, S. Pillay, S. Raghavan, and M. Montes-Y-Gómez, "Modality specific meta features for authorship attribution in Web forum posts," in *Proc. 5th Int. Joint Conf. Nat. Lang. Process.*, 2011, pp. 156–164.
- [26] U. Sapkota, T. Solorio, M. Montes-Y-Gómez, and P. Rosso, "The use of orthogonal similarity relations in the prediction of authorship," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguist.*, Samos, Greece, 2013, pp. 463–475.
- [27] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, 2008, Art. no. 7.
- [28] G. U. Yule, "On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship," *Biometrika*, vol. 30, nos. 3–4, pp. 363–390, 1939.
- [29] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, no. 3, pp. 378–393, 2006.
- [30] T. C. Mendenhall, "The characteristic curves of composition," *Science*, vol. 9, no. 214, pp. 237–246, 1887.
- [31] O. Y. De Vel, "Mining e-mail authorship," in *Proc. Workshop Text Min. ACM Int. Conf. Knowl. Disc. Data Min. (KDD)*, Boston, MA, USA, 2000, pp. 21–27.
- [32] C. U. Yule, *The Statistical Study of Literary Vocabulary*. Cambridge, U.K.: Cambridge Univ. Press, 1944.
- [33] F. J. Tweedie and R. H. Baayen, "How variable may a constant be? Measures of lexical richness in perspective," *Comput. Humanities*, vol. 32, no. 5, pp. 323–352, 1998.
- [34] L. W. Juan, "Authorship attribution using syntactic dependencies," in *Frontiers in Artificial Intelligence and Applications*, vol. 288. Amsterdam, The Netherlands: IOS Press, 2017.
- [35] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with latent Dirichlet allocation," in *Proc. 15th Conf. Comput. Nat. Lang. Learn.*, Portland, OR, USA, 2011, pp. 181–189.
- [36] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature Extraction*. Heidelberg, Germany: Springer, 2006, pp. 1–25.
- [37] J.-P. Posadas-Durán *et al.*, "Syntactic n -grams as features for the author profiling task," in *Proc. Conf. Labs Eval. Forum Working Notes (CLEF)*, 2015. [Online]. Available: <http://ceur-ws.org/Vol-1391/>
- [38] R. Layton, "A simple local n -gram ensemble for authorship verification," in *Proc. Working Notes CLEF Conf.*, 2014, pp. 1073–1078.
- [39] J.-P. Posadas-Durán *et al.*, "Application of the distributed document representation in the authorship attribution task for small corpora," *Soft Comput.*, vol. 21, no. 3, pp. 627–639, 2017.
- [40] H. Gómez-Adorno *et al.*, "Improving feature representation based on a neural network for author profiling in social media texts," *Comput. Intell. Neurosci.*, vol. 2016, Oct. 2016, Art. no. 2.
- [41] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Berlin, Germany: Springer, 2006, pp. 137–186.
- [42] R. Torney, P. Vamplew, and J. L. Yearwood, "Using psycholinguistic features for profiling first language of authors," *J. Assoc. Inf. Sci. Technol.*, vol. 63, no. 6, pp. 1256–1269, 2012.
- [43] M. A. Boukhaled and J.-G. Ganascia, "Probabilistic anomaly detection method for authorship verification," in *Proc. 2nd Int. Conf. Stat. Lang. Speech Process.*, Grenoble, France, 2014, pp. 211–219.
- [44] G. Baron, "Influence of data discretization on efficiency of Bayesian classifier for authorship attribution," *Proc. Comput. Sci.*, vol. 35, pp. 1112–1121, 2014.
- [45] T. Qian, B. Liu, M. Zhong, and G. He, "Co-training on authorship attribution with very few labeled examples: Methods vs. views," in *Proc. 37th Int. ACM Conf. Res. Devel. Inf. Retrieval (SIGIR)*, Gold Coast, QLD, Australia, 2014, pp. 903–906.
- [46] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, vol. 1. Edmonton, AB, Canada, 2003, pp. 173–180.
- [47] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
- [48] M. Khonji and Y. Iraqi, "A slightly-modified gi-based author-verifier with lots of features (ASGALF)," in *Proc. Int. Conf. CLEF Notebook PAN*, 2014, pp. 977–983.
- [49] E. Moreau, A. Jayapal, and C. Vogel, "Author verification: Exploring a large set of parameters using a genetic algorithm," in *Proc. Int. Conf. CLEF Notebook PAN*, 2014, pp. 1079–1083.
- [50] C. Mayor *et al.*, "A single author style representation for the author verification task," in *Proc. Int. Conf. CLEF Notebook PAN*, 2014, pp. 1079–1083.
- [51] J. Frery, C. Langeron, and M. Juganaru-Mathieu, "UJM at CLEF in author verification based on optimized classification trees," in *Proc. Int. Conf. CLEF Notebook PAN*, 2014, pp. 1042–1048.
- [52] E. Castillo, O. Cervantes, D. Vilaríño, D. Pinto, and S. León, "Unsupervised method for the authorship identification task," in *Proc. Int. Conf. CLEF Notebook PAN*, 2014, pp. 1035–1041.
- [53] S. Harvey, "Author verification using PPM with parts of speech tagging," in *Proc. Int. Conf. CLEF Notebook PAN*, Sheffield, U.K., 2014, pp. 1063–1068.
- [54] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *CoRR*, vol. abs/1609.06686, 2016.
- [55] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proc. 20th Conf. Uncertainty Artif. Intell.*, Banff, AB, Canada, 2004, pp. 487–494.
- [56] G. Frantzeskou, E. Stamatatos, S. Gritzalis, C. E. Chaski, and B. S. Howald, "Identifying authorship by byte-level n -grams: The source code author profile (SCAP) method," *Int. J. Digit. Evidence*, vol. 6, no. 1, pp. 1–18, 2007.
- [57] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resources Eval.*, vol. 45, no. 1, pp. 83–94, 2011.
- [58] R. Collobert *et al.*, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Feb. 2011.
- [59] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, vol. 14. Doha, Qatar, 2014, pp. 1532–1543.
- [60] F. Al Zamil, W. Liu, and D. Ruths, "Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors," in *Proc. 5th Int. Conf. Weblogs Soc. Media (ICWSM)*, 2012, pp. 387–390.
- [61] O. Owoputi *et al.*, "Improved part-of-speech tagging for online conversational text with word clusters," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Atlanta, GA, USA, 2013, pp. 380–391.
- [62] A. P. López-Monroy, M. Montes-Y-Gómez, H. J. Escalante, L. V. Pineda, and E. Villatoro-Tello, "INAOE's participation at PAN'13: Author profiling task notebook for PAN at CLEF 2013," in *Proc. Int. Conf. CLEF Notebook PAN*, Valencia, Spain, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [63] K. Santosh, R. Bansal, M. Shekhar, and V. Varma, "Author profiling: Predicting age and gender from blogs notebook for PAN at CLEF 2013," in *Proc. Int. Conf. CLEF Notebook PAN*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [64] P. Modaresi, M. Liebeck, and S. Conrad, "Exploring the effects of cross-genre machine learning for author profiling in PAN 2016," in *Proc. Int. Conf. CLEF Notebook PAN*, 2016, pp. 970–977.
- [65] M. B. O. Vollenbroek *et al.*, "Gronup: Groningen user profiling," in *Proc. Int. Conf. CLEF Notebook PAN*, 2016, pp. 846–857.
- [66] F. L. Cruz, R. R. Haro, and F. J. Ortega, "Italica at PAN 2013: An ensemble learning approach to author profiling notebook for PAN at CLEF 2013," in *Proc. Int. Conf. CLEF Notebook PAN*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [67] S. Ladra, F. Claude, and R. Konow, "Submission to the author profiling task from the university of A Coruña, Spain, the university of Waterloo, Canada, and Roberto Konow, Chile," in *Proc. Int. Conf. CLEF Notebook PAN*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [68] W.-Y. Lim, J. Goh, and V. L. Thing, "Content-centric age and gender profiling notebook for PAN at CLEF 2013," in *Proc. Int. Conf. CLEF Notebook PAN*, 2013.
- [69] L. Flekova and I. Gurevych, "Can we hide in the Web? Large scale simultaneous age and gender author profiling in social media," in *Proc. Int. Conf. CLEF Notebook PAN*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>

- [70] M. Meina *et al.*, “Ensemble-based classification for author profiling using various features,” in *Proc. Int. Conf. CLEF Notebook PAN*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [71] R. Kern, “Grammar checker features for author identification and author profiling notebook for PAN at CLEF 2013,” in *Proc. Int. Conf. CLEF Notebook PAN*, 2013.
- [72] A. Pavan, A. Mogadala, and V. Varma, “Author profiling using LDA and maximum entropy notebook for PAN at CLEF 2013,” in *Proc. Working Notes CLEF Conf.*, Valencia, Spain, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [73] L. Gillam, “Readability for author profiling? Notebook for PAN at CLEF 2013,” in *Proc. Int. Conf. CLEF Notebook PAN*, 2013.
- [74] F. Rangel, P. Rosso, M. M. Koppel, E. Stamatatos, and G. Inches, “Overview of the author profiling task at PAN 2013,” in *Proc. Conf. Multilingual Multimodal Inf. Access Eval.*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/>
- [75] B. Verhoeven, I. Škrjanec, and S. Pollak, “Gender profiling for slovene Twitter communication: The influence of gender marking, content and style,” in *Proc. 6th Workshop Balto Slavic Nat. Lang. Process.*, Valencia, Spain, 2017, pp. 119–125.



Steven H. H. Ding is currently pursuing the Ph.D. degree with the School of Information Studies, McGill University, Montreal, QC, Canada.

He is affiliated with the Data Mining and Security Laboratory, McGill University. His research in authorship analysis has been published in the top security journal ACM TISSEC 2015 and reported by McGill Headway and assembly clone search has been published in the top data mining conference ACM SIGKDD 2016. His current research interests

include developing novel data mining and machine learning techniques driven by the needs and challenges of applications in cybersecurity and cybercrime investigation.

Dr. Ding was a recipient of the Hex-Rays Plug-In Contest Award in 2015.



Benjamin C. M. Fung (M'09–SM'13) received the Ph.D. degree in computing science from Simon Fraser University, Burnaby, BC, Canada, in 2007.

He is a Canada Research Chair of data mining for cybersecurity, an Associate Professor with the School of Information Studies, an Associate Member with the School of Computer Science, McGill University, Montreal, QC, Canada, and a Co-Curator of cybersecurity with World Economic Forum, Cologny, Switzerland. He has over 100 refereed publications that span the research forums of

data mining, privacy protection, cyber forensics, services computing, and building engineering. His data mining works in crime investigation and authorship analysis have been reported by media worldwide.

Dr. Fung is a Licensed Professional Engineer of software engineering.



Farkhund Iqbal received the master's and Ph.D. degrees from Concordia University, Montreal, QC, Canada, in 2005 and 2011, respectively.

He holds the position of Associate Professor and the Graduate Program Coordinator with the College of Technological Innovation, Zayed University, Abu Dhabi, UAE. He uses machine learning and big data techniques to solve problems in healthcare, cybersecurity, and cybercrime investigation in the context of smart and safe city. He is an Affiliate Member with the School of Information Studies,

McGill university, Montreal, and an Adjunct Professor with the Faculty of Business and IT, University of Ontario Institute of Technology, Oshawa, ON, Canada. He has over 80 papers published in high impact factor journals and conferences.

Dr. Iqbal was a recipient of several prestigious awards and research grants. He has served as the Chair and a TPC Member of several IEEE/ACM conferences. He is a member of several professional organization, including ACM and IEEE Digital Society.



William K. Cheung received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 1999.

He is currently the Head and an Associate Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include collaborative information filtering, social network analysis and mining,

and data mining applications in healthcare.

Dr. Cheung has been on the Editorial Board of the IEEE Intelligent Informatics Bulletin since 2002. He has served as the Co-Chair and a Program Committee Member for a number of international conferences, as well as a Guest Editor of journals on areas, including artificial intelligence, Web intelligence, data mining, Web services, e-commerce technologies, and health informatics.