

# Learning Stylometric Representations for Authorship Analysis

Steven H. H. Ding, Benjamin C. M. Fung, *Senior Member, IEEE*, Farkhund Iqbal, and William K. Cheung

This is a postprint version. The official version was published in *IEEE Transactions on Cybernetics* in 2019.

**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 authorship analysis. 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, PV-DM, PV-DBOW, word2vec representations, and other baselines.

**Index Terms**—Authorship analysis, 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 authorship analysis backed up by statistical or computational methods has a long history starting from

S. H. H. Ding and B. C. M. Fung are with School of Information Studies, McGill University, 3661 Rue Peel, Montreal, QC H3A 1X1, Canada. Benjamin C. M. Fung completed part of the work during his visit at the Department of Computer Science, Hong Kong Baptist University. E-mails: steven.h.ding@mail.mcgill.ca, ben.fung@mcgill.ca

Farkhund Iqbal is with College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates. E-mail: Farkhund.Iqbal@zu.ac.ae William K. Cheung is with Department of Computer Science, Hong Kong Baptist University, Hong Kong. E-mail: william@staff.hkbu.edu.hk

The research is supported in part by the NSERC Discovery Grants (356065-2013), Canada Research Chairs Program (950-230623), and the Research Incentive Fund (RIF13059) from Zayed University, Abu Dhabi, UAE.

19th century [1], [2]. It has been a successful line of research [3]. Many customized approaches focusing on different sub-problems and scenarios have been proposed [2]. Research problems in authorship analysis can be broadly categorized into three types: *authorship identification* (i.e., identify the most plausible author of an anonymous text snippet given a set of candidates [4]–[6]), *authorship verification* (i.e., verify whether or not a given candidate is the actual author of the given text [7]), and *authorship characterization* (i.e., infer the sociolinguistic characteristics of the author of the given text [8]). Both the 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 Figure 1: feature engineering, solution design, and 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], [10], and [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 authorship analysis 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 5 approaches involve topical models.

To assist the feature selection process for authorship analysis, 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 with respect 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 authorship analysis. Given the *unlabeled*

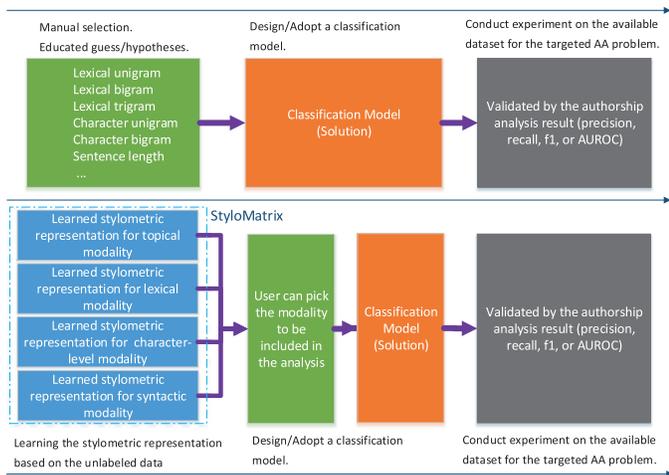


Fig. 1. Overview of the traditional solution and the proposed solution for authorship analysis.

textual data, can we automatically come up with a vectorized numeric representation of the writing style?

In this paper, we present a stylometric representation learning approach for authorship analysis (AA). Refer to the lower flowchart in Figure 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: the *topical modality*, the *lexical modality*, the *character-level modality*, and the *syntactic modality*. It is noted that the term “modality” used here is different from the term “multi-modality” 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 stylometric 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 the 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 stylometric 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 stylometric representations for authorship analysis. Specifically, our major contributions are summarized as follows:

- 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.

- 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.
- 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, our work suggests a different solution flow for authorship analysis. 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 authorship analysis 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<sup>1</sup>.

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 discussions in this paper. Finally Section VII concludes this paper.

## II. RELATED WORK

Stylometric features are the linguistic marks that quantify the linguistic characteristics [2] and [3]. Various features have been proposed for the problems in authorship analysis. 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], vocabulary richness [32], [33], and 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, [14], [35] propose to use topic models for the authorship attribution. These features can be also categorized according to their linguistic categories. [2] categorizes them into lexical type, topical type, character type, syntactic type, semantic type, and application-specific type. [25], [26] use the word ‘modality’ instead of the word

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

‘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], [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 authorship analysis employ the filter based approach [36] to select dynamic features. [27] uses information gain. [37] uses chi-square statistics. [19], [21] present a comprehensive evaluation on filtering-based approaches. The study adopts different metrics such as document frequency, information gain, and chi-square statistics, etc. It turns out that document frequency and information gain achieve the best result for authorship attribution. [20] proposes a wrapper-based approach for the problem of authorship verification. They select a distinct set of features for each author according to the performance on the training set. [38] proposes 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] propose to use latent variables in LDA as document representation. [39] and [40] are among the first studies that uses document representation learning for authorship analysis.

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 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. [23] proposes two similar neural network models to learn the vector representation of document: the PV-DM model and the 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 Figure 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  $\omega$  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 Part-of-Speech tag for token  $t_b$ . Given  $\mathbb{D}$ , the task is to learn four vector representations  $\vec{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$ ,  $\vec{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$ ,  $\vec{\theta}_\omega^{ch} \in \mathbb{R}^{\mathcal{D}(ch)}$ , and  $\vec{\theta}_\omega^{sy} \in \mathbb{R}^{\mathcal{D}(sy)}$ , 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 the topical modality and the 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  $\omega$  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: the topical bias, the local contextual bias, and the lexical bias.

- *Topical bias.* Based on the certain holistic topics to be conveyed through the text, the author is limited to a vague set

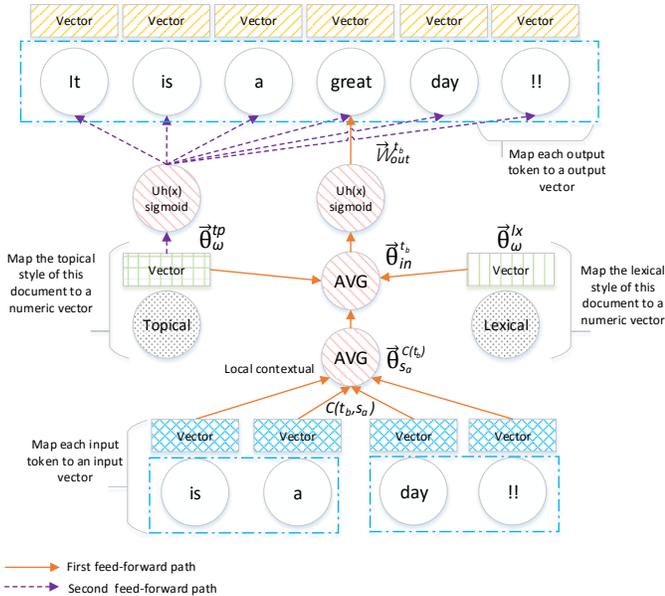


Fig. 2. The joint model for learning the stylistic representation of the topical and lexical modalities. The input word vectors are randomly initialized before training. The output word vectors are zeros before training.

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.

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

Figure 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 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:

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

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:  $\bar{w}_{in}^{t_b} \in \mathbb{R}^{dw}$  (the blue rectangles  $\blacksquare$  in Figure 2) and  $\bar{w}_{out}^{t_b} \in \mathbb{R}^{dw}$  (the yellow rectangles  $\blacktriangle$  in Figure 2) where  $dw$  denotes the dimensionality.  $\bar{w}_{in}^{t_b}$  is used to construct the input of contextual bias for the neural network, and  $\bar{w}_{out}^{t_b}$  is used for the multi-class 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:

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

The other two inputs to the model are the topical bias  $\bar{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$  and the lexical bias  $\bar{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$ . In order to have the model working properly, we need to set  $\mathcal{D}(lx)$ ,  $\mathcal{D}(tp)$ , and  $dw$  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  $\bar{\theta}_{in}^{t_b}$  since it is costly to have a fully connected layer.

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

*Example 1:* Consider a simple sentence:  $t_a =$  “it is a great day !!” in Figure 2. For each token  $\{t_b | b \in [1, 6]\}$  we pass forward the neural network. We take  $b = 4$  and  $t_b =$  ‘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  $\vec{w}_{in}^{t_2}, \vec{w}_{in}^{t_3}, \vec{w}_{in}^{t_5}$  and  $\vec{w}_{in}^{t_6}$ . With  $\vec{\theta}_{\omega}^{tp}$  and  $\vec{\theta}_{\omega}^{bg}$ , we calculate  $\vec{\theta}_{in}^{t_4}$  using Equation 3. ■

Using a full soft-max layer to model Equation 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:

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

$\text{Uh}(\cdot)$  denotes the element-wise sigmoid function. It corresponds to the red circle  $\odot$  on the first feed-forward path in the Figure 2.  $\mathbb{1}[\cdot]$  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  $\vec{w}_{out}^{t_4}$ . Next we calculate  $\mathbf{P}(\vec{w}_{out}^{t_4} | \vec{\theta}_{in}^{t_4})$  using negative sampling (Equation 4). After that we calculate the gradients w.r.t.  $\vec{w}_{out}^{t_4}$  and  $\vec{\theta}_{in}^{t_4}$ . We update  $\vec{w}_{out}^{t_4}$  according to its gradient with a learning rate. We also update  $\vec{w}_{in}^{t_2}, \vec{w}_{in}^{t_3}, \vec{w}_{in}^{t_5}, \vec{w}_{in}^{t_6}$ , and  $\vec{\theta}_{\omega}^{tp}$  equally according to the gradient of  $\vec{\theta}_{in}^{t_4}$ . ■

The second feed-forward path of this model captures the topical bias reflected on the document  $\omega$ . 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 the left part of Figure 2) is a multi-class 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  $\vec{\theta}_{\omega}^{tp}$  of document  $\omega$  for each of its sentences  $\mathcal{S}(\omega)$ :

$$\arg \max_{\vec{\theta}_{\omega}^{tp}} \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \log \mathbf{P}(t_b | \underbrace{\vec{\theta}_{\omega}^{tp}}_{\text{topical}})$$

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

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

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  $\vec{\theta}_{\omega}^{tp}$ . Basically, the second feed-forward path of this model is an approximation to the full factorization of the document-term co-occurrence matrix.

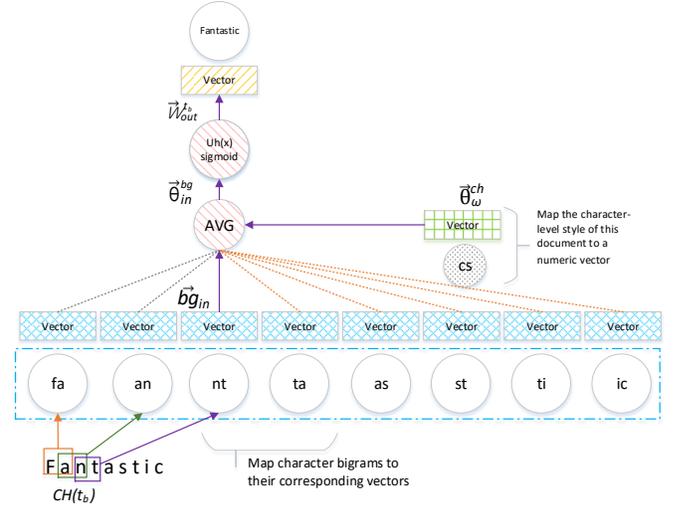


Fig. 3. The model for learning the representation of the character modality.

*Example 3:* Continue from Example 1. For the output of the second path, we map each token into a numeric vector  $\vec{w}_{out}^{t_b}$ , where  $t_b \in \{\text{'it', 'is', 'a', 'great', 'day', '!!'}\}$ . For each of the vectors we calculate  $\mathbf{P}(\vec{w}_{out}^{t_b} | \vec{\theta}_{\omega}^{tp})$  in Equation 5 using negative sampling. Then we calculate the derivatives for each  $\vec{w}_{out}^{t_b}$  and  $\vec{\theta}_{\omega}^{tp}$  and update them accordingly by multiplying the gradients with a pre-specified 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 authorship analysis [42]. After training the model on a given text dataset  $\mathbb{D}$ , we have a topical modality vector representation  $\vec{\theta}_{\omega}^{tp} \in \mathbb{R}^{d_1}$  and a lexical modality vector representation  $\vec{\theta}_{\omega}^{lx} \in \mathbb{R}^{d_1}$  for each document  $\omega \in \mathbb{D}$ . Also, for each lexical token  $t_b \in V$  we have a vectorized representation  $\vec{w}_{in}^{t_b} \in \mathbb{R}^{d_1}$ .

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

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 Figure 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.

### B. The 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 Figure 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  $\mathbb{D}$ :

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

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 4-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 multi-class 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 Figure 3, to predict its corresponding lexical token  $t_b$ . By using negative sampling, we maximize the following log probability:

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

$$\mathbf{P}(t_b | \underbrace{\vec{\theta}_\omega^{ch}}_{\text{char-level}}, bg) \approx \log f(\vec{w}_{out}^{t_b}, \vec{\theta}_{in}^{bg}) \quad (7)$$

$$+ \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} (\mathbb{1}[t \neq t_b]) \log f(-1 \times \vec{w}_{out}^t, \vec{\theta}_{in}^{bg})$$

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  $\vec{bg}_{in}$  and only propagate errors to  $\vec{\theta}_{\omega'}^{ch}$ .

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

The character modality in this work only captures the intra-word information. It only concerns with the morphology and phonemes biases in the processing of spelling lexical word.

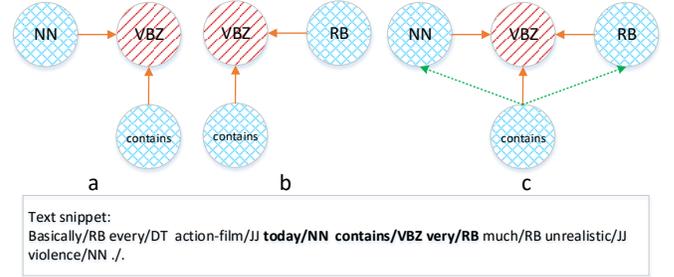


Fig. 4. Three typical inference structures for the Part-Of-Speech tagger. Solid lines indicate dependencies introduced by tagger.

The inter-word information is useful. It is captured by the lexical modality and the topical modality.

### C. The 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]:

- *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 by both the lexical token itself and the 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 Figure 4a.
- *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 the lexical token itself and the 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 Figure 4b.
- *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 the lexical token itself and the 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 Figure 4c.

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 stylometric 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 Figure 4c, 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.

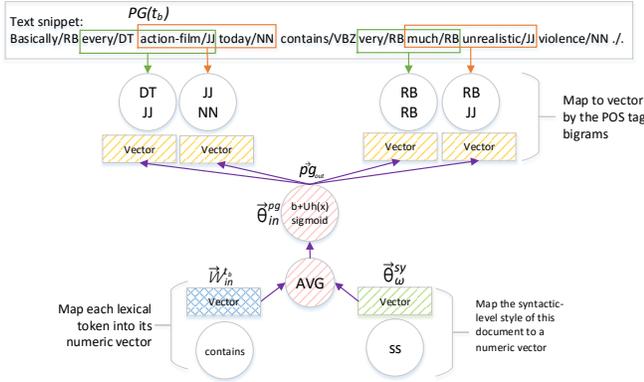


Fig. 5. The model for learning the representation of the syntactic modality.

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

$$\arg \max_{\omega} \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \sum_{n^b} \log \mathbf{P}(n^b | \underbrace{\bar{\theta}_{\omega}^{sy}}_{\text{syntactic}}, \bar{w}_{in}^{t_b})$$

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

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

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 Figure 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  $\mathcal{PG}(t_5) = \{\text{'DT JJ'}$ ,  $\text{'JJ NN'}$ ,  $\text{'RB RB'}$ ,  $\text{'RB JJ'}\}$  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  $\bar{n}_{in}^5$  and map  $t_5$  to its representation  $\bar{w}_{in}^{t_5}$ . With  $\bar{\theta}_{\omega}^{sy}$ , we calculate  $\bar{\theta}_{in}^n$  according to the first formula in Equation 8. In combination with  $\bar{n}_{in}^5$ , we calculate the forward log probability for  $\mathbf{P}(\bar{n}_{in}^5 | \bar{\theta}_{in}^n)$  in Equation 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  $\text{'JJ NN'}$  following the same procedure. After all the bigrams are processed, we move to the next token  $t_6$ . ■

TABLE I  
THE 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)

#### 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  $\omega_1$  and  $\omega_2$  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 stylometric representations  $\bar{\theta}_{\omega}^{tp} \in \mathbb{R}^{d_1}$ ,  $\bar{\theta}_{\omega}^{lx} \in \mathbb{R}^{d_1}$ ,  $\bar{\theta}_{\omega}^{ch} \in \mathbb{R}^{d_2}$ , and  $\bar{\theta}_{\omega}^{sy} \in \mathbb{R}^{d_3}$ , respectively, for the two anonymous documents  $\omega_1, \omega_2$ . The verification score is a simple cosine distance measure between the given two documents' stylometric representations. Formally, the solution outputs cosine similarity between two documents  $\omega_1$  and  $\omega_2$ :

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

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  $\bar{\theta}_{\omega}^v$  into a single one for each  $\omega$ . 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.

##### A. PAN2014 authorship verification dataset

PAN provides a series of shared tasks on digital text forensics. the PAN2014 authorship verification dataset<sup>2</sup> consists of sub-datasets 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 don't know".

We preprocess the data by tokenization, detecting sentence boundaries, and parsing POS tags using the Stanford tagger [46] and OpenNlp tagger<sup>3</sup>. 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

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

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

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>

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:

- *Style*. It represents a document under 302 widely studied static stylistometric features in [4], [29]<sup>4</sup> (Table II).
- *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  $k \in \{500, 1000, 2000, 5000\}$ .
- *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  $k \in \{500, 1000, 2000, 5000\}$ .
- *Typed-n-gram*. The typed character  $n$ -gram approach proposed in [6]. Each  $n$ -gram is prefixed by its category.
- *LDA* and *LSA*. The Latent Dirichlet Allocation (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. Latent semantic analysis (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.
- *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.
- *PV-DBOW* and *PV-DM*. Two neural networks that learn document representation [23] discussed in Section II.
- Top 5 approaches reported in PAN2014 as well as the meta-classifier called *META-CLF-PAN14*.

These baselines cover both the 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.

- *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.
- *Character modality*. We pick  $d_2 = 300$ .
- *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  $\mathcal{W}(tp)$ ,  $d_2$ , and  $d_3$  will be further discussed in Section V. Evaluation results are reported based on the performance on the test dataset.

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

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]*	<b>0.998</b>	<b>0.744</b>	<b>0.887</b>	0.767	0.924	0.934	<b>0.881</b>
[Lexical]†	0.998	0.658	0.885	<b>0.799</b>	<b>0.949</b>	<b>0.937</b>	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	0.736	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	0.594	0.804	0.681	0.712	0.736	0.724
w2v-skipgram+cbow	0.896	0.641	0.503	0.675	0.848	0.781	0.724
w2v-skipgram	0.932	0.569	0.572	0.664	0.826	0.755	0.720
Frey et al. [51]†	0.896	0.640	0.442	0.651	0.875	0.812	0.719
w2v-cbow	0.906	0.601	0.723	0.612	0.679	0.774	0.716
Castillo et al. [52]†	0.838	0.612	0.521	0.689	0.832	0.775	0.711
Typed-n-gram [6]	0.861	0.669	0.549	0.628	0.686	0.734	0.688
LDA-100	0.781	0.575	0.515	0.607	0.803	0.804	0.681
LSA-200	0.784	0.520	0.390	0.499	0.900	0.606	0.617
LDA-200	0.503	0.646	0.714	0.588	0.520	0.629	0.600
Style+[500-info-ngram]	0.717	0.456	0.416	0.442	0.893	0.596	0.587
Style	0.574	0.524	0.490	0.678	0.594	0.642	0.584
LSA-500	0.559	0.516	0.490	0.678	0.592	0.635	0.578
Style+[1000-info-ngram]	0.503	0.646	0.502	0.588	0.520	0.629	0.565
Style+[5000-freq-ngram]	0.437	0.507	0.490	0.677	0.577	0.642	0.555
Style+[5000-info-ngram]	0.498	0.465	0.471	0.650	0.573	0.661	0.553
Style+[1500-freq-ngram]	0.451	0.459	0.511	0.652	0.574	0.612	0.543
Style+[1500-info-ngram]	0.368	0.471	0.490	0.678	0.566	0.647	0.537
Style+[2000-freq-ngram]	0.368	0.474	0.492	0.679	0.548	0.654	0.536
Style+[2000-info-ngram]	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-ngram]	0.424	0.465	0.468	0.644	0.548	0.628	0.530
Style+[1000-freq-ngram]	0.391	0.464	0.469	0.641	0.539	0.614	0.520
Style+[500-freq-ngram]	0.360	0.458	0.462	0.644	0.520	0.592	0.506

### 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 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 non-formal 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

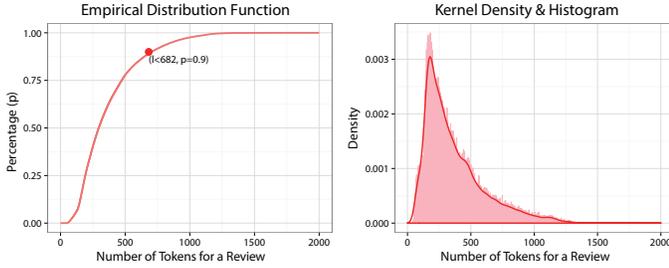


Fig. 6. Empirical distribution, kernel density and histogram on the review length for the IMDB62 review dataset.

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 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  $\omega$ :  $\hat{\theta}_{\omega}^{tp} \in \mathbb{R}^{d_1}$ ,  $\hat{\theta}_{\omega}^{lx} \in \mathbb{R}^{d_1}$ ,  $\hat{\theta}_{\omega}^{ch} \in \mathbb{R}^{d_2}$ , and  $\hat{\theta}_{\omega}^{sy} \in \mathbb{R}^{d_3}$ . 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. The IMDB62 review dataset

The IMDB62 dataset<sup>5</sup> 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<sup>6</sup>. Each user wrote 1,000 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. Figure 6 shows the empirical distribution, kernel density and histogram on the reviews' length. Following the same experimental setup in [14], [35], [54], we conduct a stratified 10-fold cross validation experiment and report the best performance on accuracy.

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

### B. The 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 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], [54] as well as available baselines in previous experiments.

- *Token SVM*. A SVM model trained on normalized token frequency features [14].
- *AT-P*. A probabilistic attribution model AT-P [14] built on the top of an author-topic (AT) model in [55]. It generates each document according to the topic distribution of its observed author [14].

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

<sup>6</sup><http://www.imdb.com>

TABLE VI

(A) PERFORMANCE ON THE IMDB62 DATASET WITH MICRO F-MEASURE.  
 (B) PERFORMANCE ON THE IMDB62 DATASET WITH ACCURACY. ENTRIES WITH † ARE CITED PERFORMANCE. [-] INDICATES RANGE.

Model	Accuracy
[Lexical+Topical]*	<b>0.972</b>
SCAP [56]†	0.948
Typed- $n$ -gram [6]	0.936
Modality [Topical]*	0.930
CNN-char [54]†	0.917
w2v-skigram+cbow	0.915
LSA	0.907
CNN-word-char [54]†	0.903
PV-DBOW+PV-DM	0.900
CNN-word-word-char [54]†	0.884
Static+1000- $n$ -gram	0.869
CNN-word [54]†	0.843
SVM+Stems [54]†	0.839
CNN-word-word [54]†	0.820
Imposters (KOP) [57]	0.769
[Lexical] *	0.742
[Character]*	0.733
LDA+Hellinger-S†	0.720
LDA	0.665
[Syntactic] *	0.601
[Syntactic] *	0.601

Model	Accuracy
[Lexical+Topical]*	<b>0.972</b>
Typed- $n$ -gram [6]	0.937
[Topical]*	0.930
Token SVM [14]†	0.925
DADT-P [14]†	0.918
w2v-skigram+cbow	0.916
LSA	0.909
PV-DBOW+PV-DM	0.900
AT-P [14]†	0.896
Static+1000- $n$ -gram	0.870
LDA+Hellinger-S [35]†	[0.80, 0.85]
Imposters (KOP)†	[0.70, 0.75]
[Lexical]*	0.742
[Character]*	0.733
LDA+Hellinger-M [35]†	< 0.70
LDA	0.677
[Syntactic]*	0.601

- *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.
- *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.
- *LDA+Hellinger-M*. This model is the same as the previous except that it does not merge documents. It uses averaged Hellinger distance over samples of a given author.
- *KOP*. A character  $n$ -gram approach proposed by [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.
- *SVM+Stems*. A SVM classifier with stemmed words [54]. Words are weighted with tf-idf and scaled to unit variance.
- *SCAP*. A source code authorship profiling approach proposed by [56] used in [54]. It uses the intersection of the most frequent character  $n$ -gram to score a candidate author.
- *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 [58] and used in [54].
- *CNN-word-word* [54]. Similar to CNN-word, but the input has an updatable word embedding and a non-updatable word embedding from pre-trained GloVe model [59].
- *CNN-char* [54]. Similar to CNN-word, but the input is an updatable character embedding channel.
- *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.
- *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

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

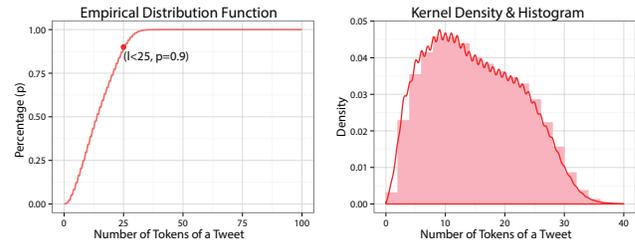


Fig. 7. Empirical distribution, kernel density and histogram on the tweets length for the ICWSM2012 Twitter dataset.

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.

#### D. Performance comparison

[14], [35] use accuracy while [54] use micro f-measure for evaluation. Table VI(b) compares our models with baselines from [14], [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 the lexical modality and the 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. The Twitter characterization dataset

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

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] *	<b>0.7887</b>	0.8308	<b>0.9318</b>	<b>0.8504</b>
Modality [Topical] *	0.7606	<b>0.8423</b>	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- <i>n</i> -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

automatically and manually inspected [60]. This dataset consists of three categories of labels for 1,170 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 pre-tokenize 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, and the friends’ tweets, etc. Since we focus on writing style, we omit this information as well as those re-tweeted tweets.

- *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.
- *Age*. The label is either *18-23* or *25-30*, which is generated by birthday related tweets, e.g., “Happy birthday to me”.
- *Political orientation*. The labels can be either *Democrat* or *Republican*, collected from the *wefollow* directory [60].

Figure 7 shows the empirical distribution, kernel density and histogram on the tweets’ length. 90 percent of the tweets have less than 25 tokens and most have a length of around 10 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 10-fold cross validation on the Twitter dataset and measure the accuracy.

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:

TABLE IX

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

	EN	EN	ES	ES	Avg
	Gender	Age	Gender	Age	
w2v-skipgram+cbow	0.599	<b>0.670</b>	<b>0.654</b>	0.671	<b>0.648</b>
[Lexical+Topical]*	0.598	0.649	0.654	<b>0.679</b>	0.645
PV-DBOW+PV-DM	<b>0.605</b>	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 <sup>nd</sup> ] [64]	0.588	0.631	0.582	0.625	0.607
[PAN16 1 <sup>st</sup> ] [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

- *target user info* [60]. A SVM-based model trained on the token-based text features and the socio-linguistic features.
- *all info* [60] 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  $\mathcal{W}(tp) = 8$  based on our observation in the previous experiment. Vector size 400 is a typical value suggested by [23]. For other baselines in 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 a 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 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

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- $n$ -gram	5:19
w2v-skigram+cbow	5:26
LSA	1:25
LDA	12:11
Static+ $n$ -gram	4:04
Vollenbroek et al. [65] [PAN16 1st]	11:44
Modaresi et al. [64] [PAN16 2nd]	3:39

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. The 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: 10s (13-17), 20s (23-27), and 30s (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 8,160 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-10 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  $\mathcal{W}(tp) = 6$  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 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.

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)	●	●	○	●	●	●	○

### 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 skigram+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. CONCLUSIONS AND FUTURE DIRECTIONS

In this article, we present our three models for learning the vectorized stylistic representations of different linguistic modalities for authorship analysis. To the best of our knowledge, it is the very first work introducing the problem of stylistic representation learning into the authorship analysis 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 authorship analysis 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 work 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 article.

## REFERENCES

- [1] F. Mosteller and D. Wallace, "Inference and disputed authorship: The federalist," 1964.
- [2] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, 2009.
- [3] M. Brennan, S. Afroz, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Transactions on Information and System Security (TISSEC)*, vol. 15, no. 3, 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," *Information Science*, vol. 231, 2013.
- [5] S. H. H. Ding, B. C. M. Fung, and M. Debbabi, "A visualizable evidence-driven approach for authorship attribution," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 3, 2015.
- [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 *Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL Human Language Technologies*, 2015.
- [7] E. Stamatatos, W. D. amd Ben Verhoeven, P. Juola, A. López-López, M. Potthast, and B. Stein, in *Proceedings of the Working Notes Papers of the CLEF 2015 Evaluation Labs*, 2015.
- [8] F. Rangel Pardo, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein, in *Proceedings of the Working Notes Papers of the CLEF 2016 Evaluation Labs*, 2016.
- [9] J. D. Burger, J. C. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on twitter," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.
- [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 *Proceedings of the 19th Iberoamerican Congress, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP)*, 2014.
- [12] N. Pratanwanich and P. Liò, "Who wrote this? textual modeling with authorship attribution in big data," in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDM)*, 2014.
- [13] J. Savoy, "Authorship attribution based on a probabilistic topic model," *Inf. Process. Manage.*, vol. 49, no. 1, 2013.
- [14] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with topic models," *Computational Linguistics*, vol. 40, no. 2, 2014.
- [15] P. Shrestha, S. Sierra, F. A. González, P. Rosso, M. Montes-y Gómez, and T. Solorio, "Convolutional neural networks for authorship attribution of short texts," in *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2017.
- [16] Z. Ge, Y. Sun, and M. J. Smith, "Authorship attribution using a neural network language model," in *Proceedings of the 2016 AAAI Conference*, 2016.
- [17] Y. Sari, A. Vlachos, and M. Stevenson, "Continuous n-gram representations for authorship attribution," in *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2017.
- [18] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, and W. Daelemans, "Overview of the 2nd author profiling task at pan 2014," in *Proceedings of the Conference and Labs of the Evaluation Forum (Working Notes)*, 2014.
- [19] J. Savoy, "Authorship attribution based on specific vocabulary," *ACM Transaction on Information System (TOIS)*, vol. 30, no. 2, 2012.
- [20] H. Zamani, H. N. Esfahani, P. Babaie, S. Abnar, M. Dehghani, and A. Shakery, "Authorship identification using dynamic selection of features from probabilistic feature set," in *Proceedings of the International Conference on Information Access Evaluation. Multilinguality, Multimodality, and Interaction*. Springer, 2014.
- [21] J. Savoy, "Feature selections for authorship attribution," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, 2013.
- [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," *CoRR*, vol. abs/1405.4053, 2014.
- [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 *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 2011.
- [26] U. Sapkota, T. Solorio, M. Montes-y Gómez, and P. Rosso, "The use of orthogonal similarity relations in the prediction of authorship," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2013.
- [27] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 2, 2008.
- [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, no. 3/4, 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," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, 2006.
- [30] T. C. Mendenhall, "The characteristic curves of composition," *Science*, 1887.
- [31] O. De Vel, "Mining e-mail authorship," in *Proc. Workshop on Text Mining, ACM International Conference on Knowledge Discovery and Data Mining (KDD2000)*, 2000.
- [32] C. U. Yule, *The statistical study of literary vocabulary*. Cambridge University Press, 1944.
- [33] F. J. Tweedie and R. H. Baayen, "How variable may a constant be? measures of lexical richness in perspective," *Computers and the Humanities*, vol. 32, no. 5, 1998.
- [34] L. W. Juan Soler-Company, "Authorship attribution using syntactic dependencies," *Frontiers in Artificial Intelligence and Applications*, vol. 288, 2017.
- [35] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with latent dirichlet allocation," in *Proceedings of the fifteenth conference on computational natural language learning*. Association for Computational Linguistics, 2011.
- [36] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature extraction*. Springer, 2006.
- [37] I. Markov, H. G. Adorno, I. Batyrshin, and A. Gelbukh, "Syntactic n-grams as features for the author profiling task," 2015.
- [38] R. Layton, "A simple local n-gram ensemble for authorship verification," in *Proceedings of the Working Notes for CLEF Conference*, 2014.
- [39] J.-P. Posadas-Durán, H. Gómez-Adorno, G. Sidorov, I. Batyrshin, D. Pinto, and L. Chanona-Hernández, "Application of the distributed document representation in the authorship attribution task for small corpora," *Soft Computing*, vol. 21, 2017.
- [40] H. Gómez-Adorno, I. Markov, G. Sidorov, J.-P. Posadas-Durán, M. A. Sanchez-Perez, and L. Chanona-Hernandez, "Improving feature representation based on a neural network for author profiling in social media texts," *Computational intelligence and neuroscience*, vol. 2016, 2016.
- [41] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Springer, 2006.
- [42] R. Torney, P. Vamplew, and J. Yearwood, "Using psycholinguistic features for profiling first language of authors," *JASIST*, vol. 63, no. 6, 2012.
- [43] M. A. Boukhaled and J. Ganascia, "Probabilistic anomaly detection method for authorship verification," in *Proceedings of the 2nd International Conference on Statistical Language and Speech Processing*, 2014.
- [44] G. Baron, "Influence of data discretization on efficiency of bayesian classifier for authorship attribution," in *Proceedings of the 18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES)*, 2014.
- [45] T. Qian, B. Liu, M. Zhong, and G. He, "Co-training on authorship attribution with very fewlabeled examples: methods vs. views," in *Proceedings of the 37th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.
- [46] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003.
- [47] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, 2006.
- [48] M. Khonji and Y. Iraqi, "A slightly-modified gi-based author-verifier with lots of features (asgalf)," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2014.

- [49] E. Moreau, A. Jayapal, and C. Vogel, "Author verification: Exploring a large set of parameters using a genetic algorithm," 2014.
- [50] C. Mayor, J. Gutierrez, A. Toledo, R. Martinez, P. Ledesma, G. Fuentes, and I. Meza, "A single author style representation for the author verification task," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2014.
- [51] J. Frery, C. Langeron, and M. Juganaru-Mathieu, "Ujm at clef in author verification based on optimized classification trees," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2014.
- [52] E. Castillo, O. Cervantes, D. Vilaríño, D. Pinto, and S. León, "Unsupervised method for the authorship identification task," 2014.
- [53] S. Harvey, "Author verification using ppm with parts of speech tagging," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2014.
- [54] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv preprint arXiv:1609.06686*, 2016.
- [55] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 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," *International Journal of Digital Evidence*, vol. 6, no. 1, 2007.
- [57] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Language Resources and Evaluation*, vol. 45, no. 1, 2011.
- [58] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [59] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, vol. 14, 2014, pp. 1532–43.
- [60] F. Al Zamal, W. Liu, and D. Ruths, "Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors," in *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM)*, 2012.
- [61] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters." Association for Computational Linguistics, 2013.
- [62] A. P. López-Monroy, M. Montes-Y-Gomez, H. J. Escalante, L. V. Pineda, and E. Villatoro-Tello, "Inaoc's participation at pan'13: Author profiling task notebook for pan at clef 2013," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [63] K. Santosh, R. Bansal, M. Shekhar, and V. Varma, "Author profiling: Predicting age and gender from blogsnotebook for pan at clef 2013," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [64] P. Modaresi, M. Liebeck, and S. Conrad, "Exploring the effects of cross-genre machine learning for author profiling in pan 2016," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2016.
- [65] M. B. op Vollenbroek, T. Carlotto, T. Kreutz, M. Medvedeva, C. Pool, J. Bjerva, H. Haagsma, and M. Nissim, "Gronup: Groningen user profiling," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2016.
- [66] F. L. Cruz, R. Haro, and F. J. Ortega, "Italica at pan 2013: An ensemble learning approach to author profilingnotebook for pan at clef 2013," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [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 *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [68] W.-y. Lim, J. Goh, and V. L. Thing, "Content-centric age and gender profiling notebook for pan at clef 2013," 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 *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [70] M. Meina, K. Brodzinska, B. Celmer, M. Czoków, M. Patera, J. Pezacki, and M. Wilk, "Ensemble-based classification for author profiling using various features," in *Proceedings of the International Conference on CLEF, Notebook for PAN*, 2013.
- [71] R. Kern, "Grammar checker features for author identification and author profiling notebook for pan at clef 2013," 2013.
- [72] A. Pavan, A. Mogadala, and V. Varma, "Author profiling using lda and maximum entropynotebook for pan at clef 2013," in *In Former et al*, 2013.
- [73] L. Gillam, "Readability for author profiling? notebook for pan at clef 2013," 2013.
- [74] F. Rangel, P. Rosso, M. Moshe Koppel, E. Stamatatos, and G. Inches, "Overview of the author profiling task at pan 2013," in *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation*, 2013.
- [75] B. Verhoeven, I. Škrjanec, and S. Pollak, "Gender profiling for slovene twitter communication: The influence of gender marking, content and style," in *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, 2017.



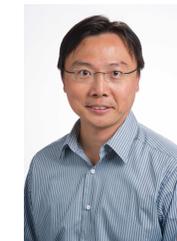
**Steven H. H. Ding** is a Ph.D. Candidate in the School of Information Studies at McGill University. He is affiliated with the Data Mining and Security Lab. His research focuses on developing novel data mining and machine learning techniques driven by the needs and challenges of applications in cybersecurity and cybercrime investigation. His research in authorship analysis has been published in the top security journal ACM TISSEC 2015 and reported by McGill Headway. His work on assembly clone search has been published in the top data mining conference ACM SIGKDD 2016, and the resulting search engine won the Hex-Rays plug-in contest award in 2015.



**Benjamin C. M. Fung** is a Canada Research Chair in Data Mining for Cybersecurity, an Associate Professor in the School of Information Studies, an Associate Member in the School of Computer Science at McGill University, and a Co-curator of Cybersecurity in the World Economic Forum (WEF). He received a Ph.D. degree in computing science from Simon Fraser University in 2007. 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 in software engineering. He is a senior member of the IEEE.



**Farkhund Iqbal** holds the position of Associate Professor and Graduate Program Coordinator in the College of Technological Innovation, Zayed University, United Arab Emirates. He holds a Master (2005) and a Ph.D. degree (2011) from Concordia University, Canada. 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 has more than 80 papers published in high impact factor journals and conferences. He is an Affiliate Member in School of Information Studies, McGill university, Canada and an Adjunct Professor in Faculty of Business and IT, University of Ontario Institute of Technology, Canada. He is the recipient of several prestigious awards and research grants. He has served as a chair and 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, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 1999. He is currently the Head and Associate Professor of the Department of Computer Science, Hong Kong Baptist University. His research interests include collaborative information filtering, social network analysis and mining, and data mining applications in healthcare. Dr. Cheung 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. Since 2002, he has been on the Editorial Board of the IEEE Intelligent Informatics Bulletin.

SUPPLEMENTARY MATERIALS A  
EXTENDED TECHNICAL DETAILS

This technical report extends the original model description by adding more details on the derivation of our proposed neural network models. Table I lists all the involved symbols in this report.

*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  $\omega$  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 Part-of-Speech tag for token  $t_b$ . Given  $\mathbb{D}$ , the task is to learn four vector representations  $\vec{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$ ,  $\vec{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$ ,  $\vec{\theta}_\omega^{ch} \in \mathbb{R}^{\mathcal{D}(ch)}$ , and  $\vec{\theta}_\omega^{sy} \in \mathbb{R}^{\mathcal{D}(sy)}$ , 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. ■

TABLE I  
SYMBOL DESCRIPTION.

Symbol	Description
$\mathbb{D}$	The given text data set without any labels
$\omega$	A document of the data set.
$\mathcal{S}(\omega)$	Sentences of a document.
$s_a$	A sentence.
$\mathcal{T}_{s_a}$	Tokens of a sentence.
$t_b$	A token at position $b$ of a sentence.
$\vec{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$	Topical modality representation of a document.
$\vec{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$	Lexical modality representation of a document.
$\vec{\theta}_\omega^{ch} \in \mathbb{R}^{\mathcal{D}(ch)}$	Character-level modality representation of a document.
$\vec{\theta}_\omega^{sy} \in \mathbb{R}^{\mathcal{D}(sy)}$	Syntactic modality representation of a document.
$\theta_{s_a}^C(t_b)$	Contextual bias for word $t_b$ in sentence $s_a$
$\mathcal{W}(tp)$	Sliding window size for constructing the contextual bias.
$\mathcal{C}(t_b, s_a)$	A sliding window on $s_a$ where $t_b$ is in the middle.
$\vec{w}_{out}^t$	The mapped output vector for word $t$ .
$\vec{w}_{in}^t$	The mapped input vector for word $t$ .
$\mathcal{CH}(t_b)$	The character bigrams of token $t_b$ .
$bg$	A character bigram.
$\vec{bg}_{in}$	The mapped input vector for character bigram $bg$ .
$\mathcal{P}(t_b)$	The POS tag for the token $t_b$
$\mathcal{P}_2(t_b)$	The concatenation of $\mathcal{P}(t_b)$ and $\mathcal{P}(t_b + 1)$ .
$\mathcal{PG}(t_b)$	The neighbor POS bigrams of word $t_b$ .
$n^b$	One of the neighbor POS bigrams of token $t_b$ .
$\vec{n}_{out}^b$	The mapped output vector for POS bigram $n^b$ .

### A. Joint learning of topical modality and lexical modality

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.

Figure 1 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 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.

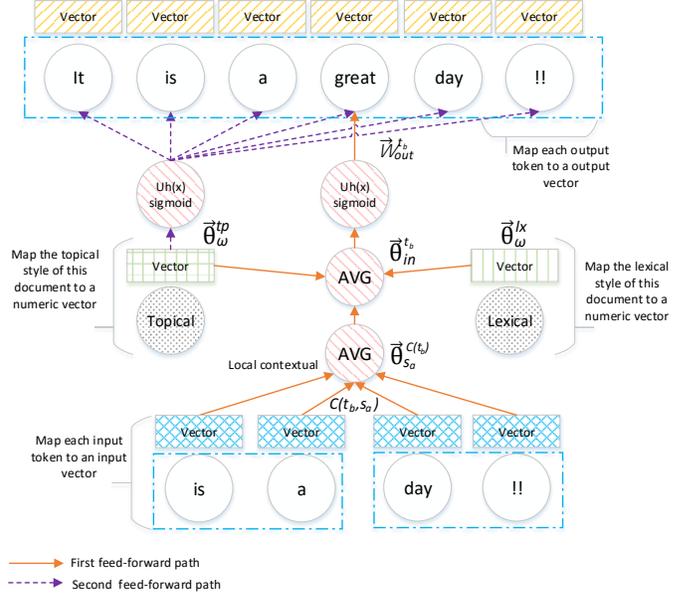


Fig. 1. The joint model for learning the stylometric representation of the topical and lexical modalities. The input word vectors are randomly initialized before training. The output word vectors are zeros before training.

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 [1]–[3], this model maps each lexical token  $t_b$  into two vectors:  $\vec{w}_{in}^{t_b} \in \mathbb{R}^{dw}$  (the blue rectangles in Figure 1) and  $\vec{w}_{out}^{t_b} \in \mathbb{R}^{dw}$  (the yellow rectangles in Figure 1) 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 multi-class 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  $\mathcal{C}(t_b, s_a) = \{t_{b-\mathcal{W}(tp)}, \dots, t_{b-1}, t_b, t_{b+1}, \dots, t_{b+\mathcal{W}(tp)}\}$ , where  $\mathcal{C}(t_b, s_a) \subseteq \mathcal{T}(s_a)$ .

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:

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

The other two inputs to the model are the topical bias  $\vec{\theta}_\omega^{tp} \in \mathbb{R}^{\mathcal{D}(tp)}$  and the lexical bias  $\vec{\theta}_\omega^{lx} \in \mathbb{R}^{\mathcal{D}(lx)}$ . In order to have the model working properly, we need to set  $\mathcal{D}(lx)$ ,  $\mathcal{D}(tp)$ , and  $dw$  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.

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

*Example 1:* Consider a simple sentence:  $t_a = \text{"it is a great day !!"}$  in Figure 1. 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  $\mathcal{C}(t_4, s_a) = \{t_2, t_3, t_5, t_6\} = \{\text{'is'}, \text{'a'}, \text{'day'}, \text{'!!'}\}$ . We map these tokens into their representations  $\bar{w}_{in}^{t_2}, \bar{w}_{in}^{t_3}, \bar{w}_{in}^{t_5}$  and  $\bar{w}_{in}^{t_6}$ . With  $\bar{\theta}_{\omega}^{tp}$  and  $\bar{\theta}_{\omega}^{lx}$ , we calculate  $\bar{\theta}_{in}^{t_4}$  using Equation 3. ■

Suppose that we use the typical soft-max multi-class output layer. The first feed-forward path of this model captures the probability of picking a word  $t_b$  based on the joint bias input  $\bar{\theta}_{in}^{t_b}$  as follows:

$$\begin{aligned} \mathbf{P}(t_b | \underbrace{\bar{\theta}_{\omega}^{tp}}_{\text{topical}}, \underbrace{\bar{\theta}_{\omega}^{lx}}_{\text{lexical}}, \underbrace{\bar{\theta}_{s_a}^{C(t_b)}}_{\text{contextual}}) &= \mathbf{P}(\bar{w}_{out}^{t_b} | \bar{\theta}_{in}^{t_b}) = \frac{f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b})}{\sum_V f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b})} \\ f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b}) &= Uh((\bar{w}_{out}^{t_b})^T \times \bar{\theta}_{in}^{t_b}) \end{aligned} \quad (4)$$

$V$  denotes the whole vocabulary constructed upon the text dataset  $\mathbb{D}$ .  $Uh(\cdot)$  denotes the element-wise sigmoid function. It corresponds to the red circle  on the first feed-forward path in the Figure 1. This function scales the output to the range of  $[0, 1]$ , so its output can be interpreted as probability.  $\bar{w}_{out}^{t_b}$  is the mapped out vector for the lexical token  $t_b$ .

By substituting the log probability in Equation 1 with Equation 4 and taking derivatives respectively on  $\bar{w}_{out}^{t_b}$  and  $\bar{\theta}_{in}^{t_b}$ , we have the gradients to be updated for each  $t_b$  at each mini-batch in the back propagation algorithm that is used to train this model:

$$\begin{aligned} \frac{\partial}{\partial \bar{w}_{out}^{t_b}} J(\theta)_1 &= (\llbracket t == t_b \rrbracket - \mathbf{P}(\bar{w}_{out}^{t_b} | \bar{\theta}_{in}^{t_b})) \times \bar{\theta}_{in}^{t_b} \\ \frac{\partial}{\partial \bar{\theta}_{in}^{t_b}} J(\theta)_1 &= \bar{w}_{out}^{t_b} - \sum_t \mathbf{P}(\bar{w}_{out}^{t_b} | \bar{\theta}_{in}^{t_b}) \times \bar{w}_{out}^{t_b} \end{aligned} \quad (5)$$

$\llbracket \cdot \rrbracket$  is an identity function. If the expression inside this function is evaluated to be true, then it outputs 1; otherwise 0. For example,  $\llbracket 1+2 == 3 \rrbracket = 1$  and  $\llbracket 1+1 == 3 \rrbracket = 0$ . Using a full soft-max layer to model Equation 1 is costly and inefficient because of the large vocabulary  $V$ . Following recent development of an efficient word embedding learning approach [2], we use the negative sampling method to approximate the log probability:

$$\begin{aligned} \log \mathbf{P}(\bar{w}_{out}^{t_b} | \bar{\theta}_{in}^{t_b}) &\approx \log f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b}) \\ &+ \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} [\llbracket t \neq t_b \rrbracket] \log f(-1 \times \bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b}) \\ f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b}) &= Uh((\bar{w}_{out}^{t_b})^T \times \bar{\theta}_{in}^{t_b}) \end{aligned} \quad (6)$$

$Uh(\cdot)$  denotes the element-wise sigmoid function. It corresponds to the red circle  on the first feed-forward path in the Figure 1.  $\llbracket \cdot \rrbracket$  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  $V$ . By taking derivatives, respectively, on  $\bar{w}_{out}^{t_b}$  and  $\bar{\theta}_{in}^{t_b}$ , we have the gradients to be updated:

$$\begin{aligned} \frac{\partial}{\partial \bar{w}_{out}^{t_b}} J(\theta) &= \llbracket t = t_b \rrbracket - f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b}) \times \bar{\theta}_{in}^{t_b} \\ \frac{\partial}{\partial \bar{\theta}_{in}^{t_b}} J(\theta) &= \sum_i^k \mathbb{E}_{t \sim P_n(t)} (\llbracket t = t_b \rrbracket - f(\bar{w}_{out}^{t_b}, \bar{\theta}_{in}^{t_b})) \times \bar{w}_{out}^{t_b} \end{aligned}$$

We do not propagate the errors from the first feed-forward path to the topical modality  $\bar{\theta}_{\omega}^{tp}$  since the topical bias is determined by the holistic distribution of vocabulary and is not determined by the specific token selection on the local level. We update  $\bar{\theta}_{\omega}^{tp}$  in the second feed-forward path.

*Example 2:* Continue from Example 1. We map  $t_4$  into its output vector  $\bar{w}_{out}^{t_4}$ . Next we calculate  $\mathbf{P}(\bar{w}_{out}^{t_4} | \bar{\theta}_{in}^{t_4})$  using negative sampling (Equation 6). After that we calculate the gradients w.r.t.  $\bar{w}_{out}^{t_4}$  and  $\bar{\theta}_{in}^{t_4}$ . We update  $\bar{w}_{out}^{t_4}$  according to its gradient with a learning rate. We also update  $\bar{w}_{in}^{t_2}, \bar{w}_{in}^{t_3}, \bar{w}_{in}^{t_5}, \bar{w}_{in}^{t_6}$ , and  $\bar{\theta}_{\omega}^{lx}$  equally according to the gradient of  $\bar{\theta}_{in}^{t_4}$ . ■

The second feed-forward path of this model captures the topical bias reflected on the document  $\omega$ . 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 the left part of Figure 1) is a multi-class 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  $\bar{\theta}_{\omega}^{tp}$  of document  $\omega$  for each of its sentences  $\mathcal{S}(\omega)$ :

$$\arg \max_{\bar{\theta}_{\omega}^{tp}} \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a} \sum_{t_b} \log \mathbf{P}(t_b | \underbrace{\bar{\theta}_{\omega}^{tp}}_{\text{topical}})$$

Similar to the first feed-forward path of this model, we map each lexical token at the output to a numeric vector  $\bar{w}_{out}^{t_b}$  (the yellow rectangles  in Figure 1). By using negative sampling, we maximize the following log probability:

Suppose that we use the typical soft-max multi-class output layer. The second feed-forward path of this model captures the probability of picking a word  $t_b$  based on the topics  $\bar{\theta}_{\omega}^{tp}$  as follows:

$$\mathbf{P}(t_b | \underbrace{\vec{\theta}_\omega^{tp}}_{\text{topical}}) = \mathbf{P}(\vec{w}_{out}^{tb} | \vec{\theta}_\omega^{tp}) = \frac{f(\vec{w}_{out}^{tb}, \vec{\theta}_\omega^{tp})}{\sum_t^V f(\vec{w}_{out}^t, \vec{\theta}_\omega^{tp})} \quad (7)$$

$$f(\vec{w}_{out}^t, \vec{\theta}_\omega^{tp}) = \text{Uh}((\vec{w}_{out}^t)^T \times \vec{\theta}_\omega^{tp})$$

The total number of parameters to be estimated is  $(|V| + 1) \times d_1$ . However, the term  $|V|$  is too large. Similar to the first feed-forward path of this model, we use the  $k$  negative sampling approach to approximate the log probability:

$$\log \mathbf{P}(t_b | \underbrace{\vec{\theta}_\omega^{tp}}_{\text{topical}}) \approx \log f(\vec{w}_{out}^{tb}, \vec{\theta}_\omega^{tp}) + \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} (\llbracket t \neq t_b \rrbracket \log f(-1 \times \vec{w}_{out}^t, \vec{\theta}_\omega^{tp})) \quad (8)$$

By taking the derivatives respectively over  $\vec{w}_{out}^{tb}$  and  $\vec{\theta}_\omega^{tp}$ , we have the derivatives to be updated for each  $t_b$ :

$$\frac{\partial}{\partial \vec{w}_{out}^{tb}} J(\theta) = \llbracket t = t_b \rrbracket - f(\vec{w}_{out}^{tb}, \vec{\theta}_\omega^{tp}) \times \vec{\theta}_\omega^{tp}$$

$$\frac{\partial}{\partial \vec{\theta}_\omega^{tp}} J(\theta) = \sum_i^k \mathbb{E}_{t \sim P_n(t)} (\llbracket t = t_b \rrbracket - f(\vec{w}_{out}^t, \vec{\theta}_\omega^{tp})) \times \vec{w}_{out}^t$$

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  $\vec{\theta}_\omega^{tp}$ . 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  $\vec{w}_{out}^{tb}$ , where  $t_b \in \{\text{'it', 'is', 'a', 'great', 'day', '!!'}\}$ . For each of the vectors we calculate  $\mathbf{P}(\vec{w}_{out}^{tb} | \vec{\theta}_\omega^{tp})$  in Equation 8 using negative sampling. Then we calculate the derivatives for each  $\vec{w}_{out}^{tb}$  and  $\vec{\theta}_\omega^{tp}$  and update them accordingly by multiplying the gradients with a pre-specified 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 authorship analysis [4]. After training the model on a given text dataset  $\mathbb{D}$ , we have a topical modality vector representation  $\vec{\theta}_\omega^{tp} \in \mathbb{R}^{d_1}$  and a lexical modality vector representation  $\vec{\theta}_\omega^{lx} \in \mathbb{R}^{d_1}$  for each document  $\omega \in \mathbb{D}$ . Also, for each lexical token  $t_b \in V$  we have a vectorized representation  $\vec{w}_{in}^{tb} \in \mathbb{R}^{d_1}$ .

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

The first feed-forward path corresponds to the *PV-DM* model in [3]. The second feed-forward path corresponds to the *PV-DBOW* model in [3]. 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 Figure 1) captures what would be the overall topic (i.e., word distribution) of the document. By pushing it to the input of *PV-DM* at each mini batch, the

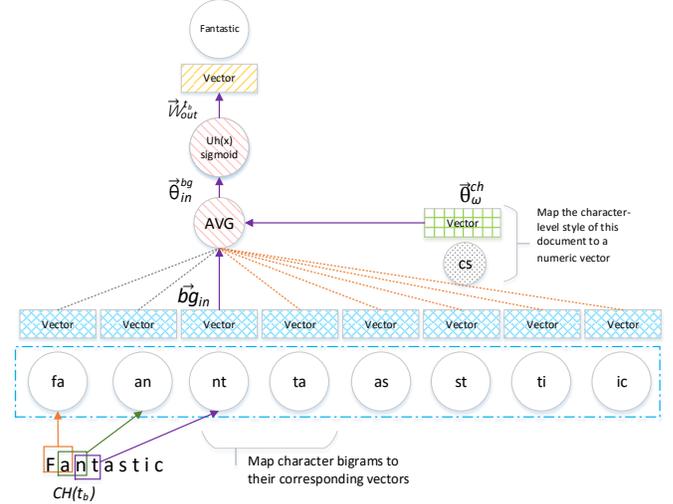


Fig. 2. The model for learning the representation of the character modality.

lexical vector captures what is missing from the topic and the current context or lexical preference, where people have different word choice even under similar topic and similar context. Thus, it is very different from the *PV-DBOW* and *PV-DM* models.

### B. The 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 Figure 2. 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  $\vec{\theta}_\omega^{ch} \in \mathbb{R}^{D(ch)}$  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  $\mathbb{D}$ :

$$\arg \max_{\mathbb{D}} \frac{1}{|\mathbb{D}|} \sum_{\omega} \sum_{s_a}^{S(\omega)} \sum_{t_b}^{\mathcal{T}(s_a)} \sum_{bg}^{\mathcal{CH}(t_b)} \log \mathbf{P}(t_b | \underbrace{\vec{\theta}_\omega^{ch}}_{\text{char-level}}, \vec{bg}_{in})$$

We use a character bigram instead of unigram to increase the character-level vocabulary size. Similar to the previous lexical model, we map each lexical token  $t_b$  into a numeric vector  $\vec{w}_{out}^{tb}$ , which is used to output a multi-class 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  $\vec{bg}_{in}$  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 Figure 2, to predict its corresponding lexical token  $t_b$ . Suppose that we use the typical soft-max multi-class output layer:

$$\vec{\theta}_{in}^{bg} = \langle \vec{\theta}_{\omega}^{ch}, \vec{bg}_{in} \rangle$$

$$\mathbf{P}(t_b | \underbrace{\vec{\theta}_{\omega}^{ch}}_{\text{char-level}}, \vec{bg}_{in}) = \mathbf{P}(\vec{w}_{out}^{t_b} | \vec{\theta}_{in}^{bg}) = \frac{f(\vec{w}_{out}^{t_b}, \vec{\theta}_{in}^{bg})}{\sum_t f(\vec{w}_{out}^t, \vec{\theta}_{in}^{bg})} \quad (9)$$

$$f(\vec{w}_{out}^t, \vec{\theta}_{in}^{bg}) = \text{Uh}((\vec{w}_{out}^t)^T \times \vec{\theta}_{in}^{bg})$$

Again, there are  $O(V)$  parameters to be updated for each pass of the neural network, which is not efficient. Thus, we use the negative sampling approach to approximate the log probability:

$$\vec{\theta}_{in}^{bg} = \langle \vec{\theta}_{\omega}^{ch}, \vec{bg}_{in} \rangle \quad (10)$$

$$\mathbf{P}(t_b | \underbrace{\vec{\theta}_{\omega}^{ch}}_{\text{char-level}}, bg) \approx \log f(\vec{w}_{out}^{t_b}, \vec{\theta}_{in}^{bg})$$

$$+ \sum_{i=1}^k \mathbb{E}_{t \sim P_n(t_b)} (\llbracket t \neq t_b \rrbracket \log f(-1 \times \vec{w}_{out}^t, \vec{\theta}_{in}^{bg})) \quad (11)$$

Similar to the previous model, we have the following derivatives by using negative sampling:

$$\frac{\partial}{\partial \vec{w}_{out}^t} J(\theta) = (\llbracket t = t_b \rrbracket - f(\vec{w}_{out}^t, \vec{\theta}_{in}^{bg})) \times \vec{\theta}_{in}^{bg}$$

$$\frac{\partial}{\partial \vec{\theta}_{in}^{bg}} J(\theta) = \sum_i^k \mathbb{E}_{t \sim P_n(t)} (\llbracket t = t_b \rrbracket - f(\vec{w}_{out}^t, \vec{\theta}_{in}^{bg})) \times \vec{w}_{out}^t$$

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  $\vec{bg}_{in}$  and only propagate errors to  $\vec{\theta}_{\omega'}^{ch}$ .

*Example 4:* Consider a simple sentence:  $t_a =$  ‘‘Fantastic day !!’’ in Figure 2. For each token  $\{t_b | b \in [1, 3]\}$  we extract its character bigrams. Suppose the word in the target is  $t_1 =$  ‘fantastic’, and its bigrams are  $\mathcal{CH}(t_1) = \{bg_c | c \in [1, 2, 3, 4, 5, 6, 7, 8]\} = \{\text{‘fa’}, \text{‘an’}, \text{‘nt’}, \text{‘ta’}, \text{‘as’}, \text{‘st’}, \text{‘ti’}, \text{‘ic’}\}$ . The process is the same for each word. Let us take a bigram  $bg_1 =$  ‘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 Equation 10. Then we calculate the forward log probability for  $\mathbf{P}(\vec{w}_{out}^{t_1} | \vec{\theta}_{in}^{bg})$  in Equation 11. We calculate the corresponding gradients and update the respective parameters. The training pass for bigram  $bg_1 =$  ‘fa’ is completed, and we move to the next bigram ‘an’ following the sample procedure. After traversing all the bigrams we move to the next token  $t_2 =$  ‘day’. ■

The character modality in this work only captures the intra-word information. It only concerns with the morphology and phonemes biases in the processing of spelling lexical word. The inter-word information is useful. It is captured by the lexical modality and the topical modality. This model can be extended with inter-word information by using the current character  $n$ -gram to predict the surrounding words.

### C. The syntactic modality

The number of unique POS tags is quite limited, so we use the bigrams of POS tags. See Figure 3. Let

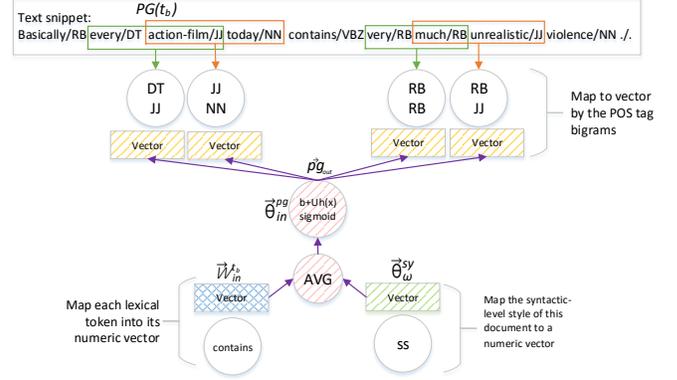


Fig. 3. The model for learning the representation of the syntactic modality.  $\mathcal{P}_2(t_b)$  be a POS tag bigram  $[\mathcal{P}(t_b), \mathcal{P}(t_{b+1})]$ , and  $n^b \in \mathcal{PG}(t_b) = \{\mathcal{P}_2(t_{b-3}), \mathcal{P}_2(t_{b-2}), \mathcal{P}_2(t_{b+1}), \mathcal{P}_2(t_{b+2})\}$  be the neighbor POS bigrams of token  $t_b$ . The goal is to maximize:

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

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 Figure 3.  $\vec{w}_{in}^{t_b}$  and  $\vec{\theta}_{\omega}^{sy}$  share the same dimensionality  $d_3$ . The prediction can be implemented as a soft-max layer:

$$\vec{\theta}_{in}^n = \langle \vec{\theta}_{\omega}^{sy}, \vec{w}_{in}^{t_b} \rangle$$

$$\mathbf{P}(n^b | \underbrace{\vec{\theta}_{\omega}^{sy}}_{\text{syntactic}}, t_b) = \mathbf{P}(\vec{n}_{out}^b | \vec{\theta}_{in}^n) = \frac{f(\vec{n}_{out}^b, \vec{\theta}_{in}^n)}{\sum_{n^b} f(\vec{n}_{out}^b, \vec{\theta}_{in}^n)} \quad (12)$$

$$f(\vec{n}_{out}^b, \vec{\theta}_{in}^n) = \text{Uh}((\vec{n}_{out}^b)^T \times \vec{\theta}_{in}^n)$$

where  $V_n$  denotes the union of all distinct POS bigrams, and the number of parameters to be updated for each  $n^b$  is bounded by  $V_n$ , which is around a few hundreds. It is still computationally feasible to directly use the soft-max layer. It is possible to use the negative sampling as well:

$$\vec{\theta}_{in}^n = \langle \vec{\theta}_{\omega}^{sy}, \vec{w}_{in}^{t_b} \rangle$$

$$\mathbf{P}(n^b | \underbrace{\vec{\theta}_{\omega}^{sy}}_{\text{syntactic}}, t_b) \approx \log f(\vec{n}_{out}^b, \vec{\theta}_{in}^n)$$

$$+ \sum_{i=1}^k \mathbb{E}_{n \sim P_n(n^b)} (\llbracket n \neq n^b \rrbracket \log f(-1 \times \vec{n}_{out}^b, \vec{\theta}_{in}^n)) \quad (13)$$

where  $P_n(n^b)$  denotes the negative sampling function for  $V_n$ . Accordingly, we have the following derivatives for back propagation:

$$\frac{\partial}{\partial \vec{n}_{out}^b} J(\theta) = (\llbracket n = n^b \rrbracket - f(\vec{n}_{out}^b, \vec{\theta}_{in}^n)) \times \vec{\theta}_{in}^n$$

$$\frac{\partial}{\partial \vec{\theta}_{in}^n} J(\theta) = \sum_i^k \mathbb{E}_{n \sim P_n(n^b)} (\llbracket n = n^b \rrbracket - f(\vec{n}_{out}^b, \vec{\theta}_{in}^n)) \times \vec{n}_{out}^b$$

At the end of the training, we have  $\vec{\theta}_\omega^{sy}$  for each document  $\omega \in \mathbb{D}$ . To estimate  $\vec{\theta}_{\omega'}^{sy}$  for  $\omega' \notin \mathbb{D}$ , we fix all  $\vec{w}_{in}^{t_b}$  and  $\vec{n}_{out}$  and only propagate errors to  $\vec{\theta}_{\omega'}^{sy}$ .

*Example 5:* Consider a sentence and its corresponding sequence of POS tags in Figure 3. 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  $\mathcal{PG}(t_5) = \{\text{'DT JJ'}$ ,  $\text{'JJ NN'}$ ,  $\text{'RB RB'}$ ,  $\text{'RB JJ'}\}$  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 Equation 13. 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 Equation 13. 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  $\text{'JJ NN'}$  following the same procedure. After all the bigrams are processed, we move to the next token  $t_6$ . ■

## REFERENCES

- [1] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Springer, 2006.
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.
- [3] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014.
- [4] R. Torney, P. Vamplew, and J. Yearwood, "Using psycholinguistic features for profiling first language of authors," *JASIST*, vol. 63, no. 6, 2012.