A Scalable Framework for Stylometric Analysis of Multi-author Documents

Raheem Sarwar[1], Chenyun Yu[1], Sarana Nutanong[1], Norawit Urailertprasert[2], Nattapol Vannaboot[2], and Thanawin Rakthanmanon[2,3]

[1] Department of Computer Science, City University of Hong Kong, HKSAR, China,
{rsarwar2-c,chenyunyu4-c}@my.cityu.edu.hk,s.nutanon@cityu.edu.hk
[2] Department of Computer Engineering, Kasetsart University, Thailand,
{norawit.u,nattapol.v}@ku.th,thanawin.r@ku.ac.th
[3] Vidyasirimedhi Institute of Science and Technology, Thailand

Abstract. *Stylometry* is a statistical technique used to analyze the variations in the author's writing styles and is typically applied to authorship attribution problems. In this investigation, we apply stylometry to authorship identification of multi-author documents (AIMD) task. We propose an AIMD technique called *Co-Authorship Graph (CAG)* which can be used to collaboratively attribute different portions of documents to different authors belonging to the same community. Based on CAG, we propose a novel AIMD solution which (i) significantly outperforms the existing state-of-the-art solution; (ii) can effectively handle a larger number of co-authors; and (iii) is capable of handling the case when some of the listed co-authors have not contributed to the document as a writer. We conducted an extensive experimental study to compare the proposed solution and the best existing AIMD method using real and synthetic datasets. We show that the proposed solution significantly outperforms existing state-of-the-art method.

Keywords: Stylometry, Authorship Identification, Co-Authorship Graph, Multi-author documents

1 Introduction

Authorship attribution (AA) aims to infer authorship information from documents [6]. Authorship attribution has several variations depending upon the type of information to be inferred. One of the extensively investigated variation is *authorship identification* [5]. "*Authorship identification* aims at identifying the true author of a disputed document from a set of candidate authors" [15]. The main idea of authorship identification is that, by computing stylometric feature from documents and building a classification model on them, we can distinguish between documents written by different authors [15].

One useful generalization of authorship identification problem that has received relatively little attention is *authorship identification of multi-author documents (AIMD)* [5]. The AIMD problem can be defined as follows. *Given a corpus*

of multi-author documents labeled with their authors, identify the authors of an anonymous multi-author document from a set of authors of a given corpus [16].

Existing authorship identification techniques designed to handle single-author documents are not applicable to multi-author documents [5]. This is because, single-author authorship attribution techniques rely on the assumption that every text sample (document) has only one single label (author). However, the AIMD problem requires the ability to (i) infer the writing style of each individual author from a corpus of multi-author documents; and (ii) make a multi-label prediction for each document.

One prominent application domain of AIMD is *bibliometrics*, in which AIMD can help improve the processes of measuring and analyzing the collaborative natures among a community of researchers [18]. Instead of attributing the entire paper to all the listed authors, one can use AIMD techniques to perform a more fine-grained analysis. Specifically, different parts of the same document can be attributed to different authors on the author list. Such an authorship identification capability can help the information retrieval system in the following ways: (i) scholarly search engines may implement an author specific search in which the researchers can look for text sample written by a particular author; and (ii) a researcher may wish to construct individual author profiles reflecting the contributions of each author in different scientific fields. In addition, AIMD techniques can also be used to identify researchers who had been involved actively in writing and mentors who are giving feedback and providing ideas. Another aspect of the AIMD is the peer-review system of the academic conferences where both the reviewers and the authors of the paper stay anonymous. This notion can be challenged by showing that it is possible for a reviewer to reveal the identity of the authors of scientific papers by using the AIMD framework.

Several existing studies [4, 9] on *authorship identification of multi-author documents (AIMD)* have shown some success on corpora consisting of scientific papers using the *citations* included in each paper. However, their success was achieved mostly in constrained scenarios, e.g., identifying the authors of papers sharing the self-citations. Along with the citation information, Payer et al. [16] have also made use of topic-information and some common stylometric features such as the frequencies of most common words.

The main difference between our work and a great majority of existing studies is that we make use of only the stylometric features. (see Section 2.2 for more details). Specifically, our features are topic-independent [19, 7]. Hence, unlike most of the existing studies, our solution is also applicable to corpora where citation information is not available and the documents have different topics.

In summary, existing AIMD studies have the following limitations. (i) The accuracy levels of existing AIMD techniques can still be greatly improved. For example, the state-of-the-art stylometry based technique [5] reports an accuracy level less than 30% on a corpus containing over 360 candidate authors. (ii) Existing techniques are adversely affected by an increase in the number of co-authors. For example, Dauber et al. [5] reported a drop in accuracy level from 25% to 16% as the number of authors had increased from 2 to 7. (iii) To the

best of our knowledge, existing AIMD techniques do not tackle the issue of *non-writing authors (NWA)* [5, 16]. However, NWAs do exist in real world scenarios. For example, in a scientific/engineering article, it is not necessary that all listed co-authors had contributed as *writers*.

In this investigation, we propose a solution to overcome the aforementioned limitations. The main challenge of AIMD is the lack of "*ground truth*" information. That is, most documents in the training set are associated with multiple authors. Hence, we need the ability to attribute different parts of the same document to different authors on the author list. In order to address this challenge, we propose a method which collaboratively learns individual writing styles from multiple co-authored documents called *Co-Authorship Graph (CAG)*.

Figure 1 illustrates the basic concept behind our CAG method. It shows four documents where each document contains three fragments. Each edge linking two fragments denotes that they are stylistically similar to each other. We initially assume that each fragment is associated with all listed authors. For example, the author list of three fragments $D_{1.1}$, $D_{1.2}$ and $D_{1.3}$ is $[A, B, C]$. That is, $D_{1.1}$, $D_{1.2}$, and $D_{1.3}$ could have been written by A, B, or C. The figure also shows that $D_{1.1}$ is stylistically similar to $D_{3.3}$ and $D_{4.2}$, which could have been written by $[C, D, A]$ and $[D, A, B]$, respectively. Since $D_{3.3}$ must have been written by one of the authors in $[A, B, C]$, we can see that A is the only author common to the three author lists. As a result, we can deduce that $D_{1.1}$ must have been written by A. Following the same principle, we can also deduce that the author of $D_{1.2}$ is B and the author of $D_{1.3}$ is C. The full result is given in the table on the right side of Fig. 1. In order to adopt the basic concept illustrated in Fig. 1 to a real-world

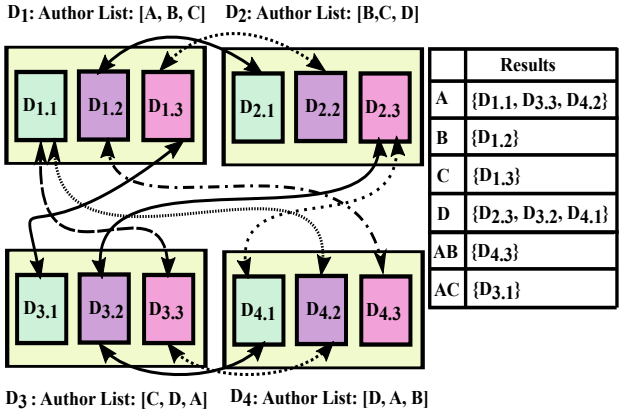


Fig. 1. Co-Authorship Graph

corpus we have to address the following issues. First, the intersection between multiple author lists (obtained from stylistically similar fragments) may not result with exactly 1 author. Second, the intersection between multiple author

lists may result with no author at all being identified. Third, a number of listed authors may *not* have contributed as writers. In this paper, we formulate an AIMD solution that can handled these stated issues in a real world corpus.

In order to demonstrate the effectiveness of our method, we apply it to one synthetic dataset and two real datasets. We also compare our method against the best-existing AIMD solution [5]. Results from our experimental studies show that our method outperforms the best existing technique in all three datasets. The contributions of this investigation can be summarized as follows.

- We propose an AIMD technique called *Co-Authorship Graph (CAG)* which can be used to collaboratively attribute text fragments in a set of documents to distinguish authors in the same community.
- Based on the CAG technique, we propose a novel AIMD solution which (i) significantly outperforms the existing state-of-the-art solution; (ii) can effectively handle a larger number of co-authors; and (iii) is capable of handling NWAs.
- We conducted an extensive experimental study comparing the proposed method and the best existing AIMD method [5] using real and synthetic datasets.

The rest of the paper is organized as follows. Section 2 reviews previous studies on authorship attribution for single- and multi-author documents. Section 3 presents the proposed solution. Section 4 reports results from our extensive experimental studies. Section 5 contains our concluding remarks.

2 Literature Review:

2.1 Stylometry

Stylometry is a statistical technique used to analyze variations in the writing styles of the authors. It has been used extensively in solving authorship attribution problems such as authorship identification and authorship profiling [15].

Stylometric features are stylistic markers/attributes of the writing style that can help discriminate between texts written by different authors. There are different types of stylometric features, e.g., lexical, structural, and syntactic features [16, 5, 7, 11, 14]. The lexical features are statistical measures of lexical variations such as word length distributions [11] and vocabulary richness [7]. Examples of lexical features are character-based and word-based measures of lexical variations [16]. Structural features are markers related to the layout of the text, e.g., the average number of words in a sentence or in a paragraph [11]. The examples of the syntactic features are part-of-speech tags and function words [14].

Payer et al. [16] proposed a solution for AIMD and applied it on a corpus of academic papers. They calculated a set of 10,727 features from the academic papers out of which 399 were stylometric and 2,374 content based, while the rest of the features 7,954 were based on citations. Later on, Dauber [5] proposed a solution for AIMD using the "Writeprints Limited features set" [1]. It includes content-specific, lexical, structural, syntactic and idiosyncratic features.

Comparison to our work. In this investigation, we use a set of 56 stylometric features which can be categorized into three types, namely, *syntactic, lexical* and *structural* features [7, 11, 14]. Specifically, we use 27 lexical [7, 11], 2 structural [11], and 27 syntactic features [14]. These features are explained in Appendix A. The features used in this investigation differ those adopted in existing studies in several ways. Unlike the existing feature sets [4, 9, 16, 5], our set of features contains only stylometric features. Specifically, these features are topic-independent [19, 7]. As a result, our solution is also applicable to corpora in which citations information is not available and the documents address different topics. Moreover, we use a set of 56 features which is smaller than feature sets used in existing AIMD studies [16, 5, 4, 9]. As a result, in comparison to existing studies, the proposed solution requires less storage and is computationally less expensive.

2.2 Authorship Identification

From the viewpoint of the context of this investigation, existing studies on authorship identification can be categorized into two types (i) authorship identification of single-author documents (AISD); and (ii) authorship identification of multi-author documents (AIMD). The main idea behind AISD is to identify the true author of a disputed document from a set of candidate authors. Existing studies of AISD have reported good results [17, 3]. However, as already explained in the introduction section, existing authorship identification techniques designed to handle single-author documents are inapplicable to multi-author documents [5]. Since this investigation focuses on AIMD, we limit the discussion on AISD in interest of brevity.

AIMD. The AIMD problem can be defined as follows. *Given a corpus of multi-author documents labeled with their co-authors, identify the co-authors of an anonymous multi-author document from the authors in the given corpus [16].* Several existing studies [4, 9] on AIMD have shown some success for corpora that consist of scientific papers using only the *citations* made in each paper. However, their success was achieved mostly in a constrained scenario, e.g., identifying the authors of papers sharing the self-citations or in a specific domain such as Physics [9] or Machine Learning [4]. Specifically, Bradley et. al. [4] reported less than 71% accuracy while Hill el. al. [9] reported less than 50% accuracy. Later, Payer et al. [16] proposed a solution for AIMD and applied it on a corpus that consists of academic papers. Their method made use of citations-based features, stylometric features, and topic-based features. Hence, most of the existing solutions for AIMD are inapplicable to a corpus where the documents do not have citations such as novels or harassment letters, in addition their performance may turn worse when the corpus contains documents on multiple topics or may be performing topic classification. Our proposed solution is based purely on stylometric features. We do not make use of any other information such as topic information or citation information for the AIMD task. As a result, the feature set used in this investigation is topic-independent [19, 7].

There are several other variations of AIMD which are comparatively easier to implement than the aforementioned variation and have shown promising results. For example, one of the AIMD variations used a training set of single-author documents which makes this variation easy to tackle. However, this requirement may not be realistic in real-world scenarios in which the training sample themselves are also multi-author documents [8]. In addition, the study reported a drastic accuracy drop as the number of co-authors in one document increases, i.e., from 50% to 30% after increasing the number of co-authors from 2 to 3 [5]. Another variation of AIMD assumed that each co-author group had a sufficient number of writing samples for training [5]. Due to the combinatoric nature of collaborative patterns of researchers in a community, we consider this assumption to be unrealistic.

Since in the AIMD task, each document is associated with more than one author, where each author can be considered as a label, one can also consider this problem as a multi-label (ML) classification task. One of the popular ML classifiers is the *multi-label k-nearest neighbor (MLkNN) classifier* [22]. As with the regular kNN method, MLkNN identifies the k nearest neighbors with respect to a given test instance. To make a multi-label prediction, MLkNN derives statistical information from the label sets of identified kNNs, e.g., the number of neighbors for each label. Finally, it applies the *Maximum A Posteriori (MAP)* principle to determine the label set of the given test instance [22]. It can be seen that this multi-label classification task naturally fits the AIMD problem definition. However, existing AIMD studies have reported that transforming each multi-label sample into multiple single-label samples improves the classification accuracy [16, 5].

3 Proposed Solution

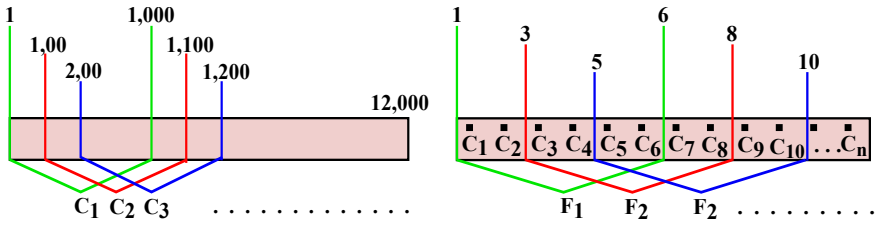
In this section, we show how the *collaborative authorship prediction* concept introduced in Section 1 can be realized. Our proposed solution consists of two preprocessing steps: *feature extraction* and *co-authorship graph training*. After the preprocessing steps, the trained data is used to make a multi-authorship prediction for any query document. Our design principle is based on the concept of *probabilistic multi-class, multi- label classification*. That is, each training sample is associated with multiple labels where each label is associated with a probability. Given a query sample Q, probabilistic labels of stylistically similar samples with respect to Q are used to derive a probabilistic prediction. In this way, we can accurately capture the multi-author nature in both training samples and test samples.

3.1 Preprocessing: Feature Extraction

In this subsection, we discuss the feature extraction process. Each document is represented as a collection of fragments where each fragment is represented as a set of points. Each point is calculated from 1,000 tokens (sequences of

characters separated by white spaces) using the stylometric feature described in Appendix A. In this way, authorship predictions are made per fragment and the prediction for an entire document is an aggregation over multiple fragments associated with the same document.

The main motivation of this "collection of point sets" representation is two-fold. First, a point set can capture how one's writing style varies within the same document. Second, different parts of the same document can be associated with different authors. Note that in order to obtain reliable stylometric information for each data point, the number of tokens for each data point should be set to at least 1,000 [20]. However, with this number of tokens, we can have only 12 data points for each 12,000-token document, which is insufficient for our analysis. Hence, we apply the sliding window method to generated data points from overlapping token sequences. This process is illustrated in Fig. 2(a) with a sliding window increment of 100 tokens and the window size of 1,000, which are the value we use in this paper. In this way, we can generate, 111 data points from a 12,000-token document. The same principle is also applied to fragments in order for us to obtain a sufficient number of fragments for our analysis as shown in Fig. 2(b) with a fragment sliding window increment of 2 data points and a fragment size of 6 data points. In this way, we can generate 53 fragments from a 12,000-token document.



(a) Chunk-Level Sliding Window (b) Fragment-Level Sliding Window

Fig. 2. Feature Extraction

3.2 Preprocessing: Co-authorship Graph Training

As stated in the introduction, the main challenge of the AIMD problem is that each document in the corpus at hand can be associated with multiple authors. Due to its combinatoric nature, the same list of authors may not be repeated in the corpus. In addition, some of the authors on the author list may not have contributed as writers to the document, making the AIMD problem more complicated. Hence, an AIMD predictive method must be able to infer the authorships of each document without relying on the absolute ground truth information.

In this investigation, we propose a novel AIMD solution based on the observation that *stylistically similar fragments should have been written by a similar*

group of authors. As a result, we propose a data structure called *Co-authorship Graph (CAG)* to capture the stylistic similarity between these fragments. We also propose an iterative algorithm which attempts to identify the true writer of each fragment.

The structure of the CAG construction process is given in Algorithm 1. Recall that after the feature extraction process, each document is represented as a collection of point sets (fragments), where each data point corresponds to one feature vector. The algorithm iterates through all fragments from all documents (Lines 4 to 9). CAG edges can be constructed by identifying k stylistically similar fragments for each document fragment. We use *modified Hausdorff distance (MHD)* [12] as the distance between two fragments. Specifically, the procedure GetKNN(F, Fragments) finds k fragments in "Fragments" with the smallest MHDs from F (Line 5). These neighbors are the graph's edges, while the distances (MHDs) are edge weights. We assume that each fragment F is associated with the list of document authors F.AuthorList which may include one or more non-writing authors (NWA). The *probability mass function* (PMF) over the author list is initialized by giving each author on the list the same probability (Lines 8 to 9). After iterating through *all fragments from all documents*, the CAG is returned (Line 10).

Algorithm 1 CAGConstruction

```
1: procedure CAG Construction
2:     Vertices ← []
3:     Edges ← []
4:     for F in Fragments do
5:         Neighbors ← GetKNN(F, Fragments)
6:         for N in Neighbors do
7:             Edges.Append((F, N))
8:         F.PMF ← GenerateUniformPMF(F.AuthorList)
9:         Vertices.Append(F)
10:    return G(Vertices, Edges)
```

We illustrate now, how a CAG can be constructed using the example given in Fig. 3. The example contains 4 documents and each document is associated with 4 listed authors as shown in the figure. In this example, we set the ground truth as follows. First, only the first three authors contributed as writers to the respective document, e.g., only authors A, B, and C wrote different parts of D_1, while author W is an NWA. Similarly, authors X, Y, and Z are NWAs of D_2, D_3, and D_4, respectively. Note that this ground truth information is hidden from the model.

Figure 3 also illustrates the initial PMF of each document fragment. Since *the ground truth regarding the non-writer authors is hidden from the model*. All fragments of all documents are associated initially with all listed authors with the equal probability. For example, the author PMFs of $D1.1$, $D1.2$, $D1.3$ are

uniform, e.g., $\{A : 0.25, B : 0.25, C : 0.25, W : 0.25\}$. The initial PMFs of the other fragments in the figures are derived in the same fashion.

After executing the CAG construction algorithm (Algorithm 1) we obtain the edges connecting stylistically similar fragments together. For example, according to the edges identified, we can see that the fragment $D_{1.1}$ is stylistically similar to $D_{3.3}$ and $D_{4.2}$. Similarly, $D_{1.2}$ is stylistically similar to $D_{2.1}$ and $D_{4.3}$. As can be seen, although all fragments in the same document are associated initially with the same authors with equal probability, they are connected to different sets of stylistically similar fragments with different author lists. Next, we will show that these differences can be used to collaboratively identify the author who had contributed as a writer of each fragment through Algorithm 2.

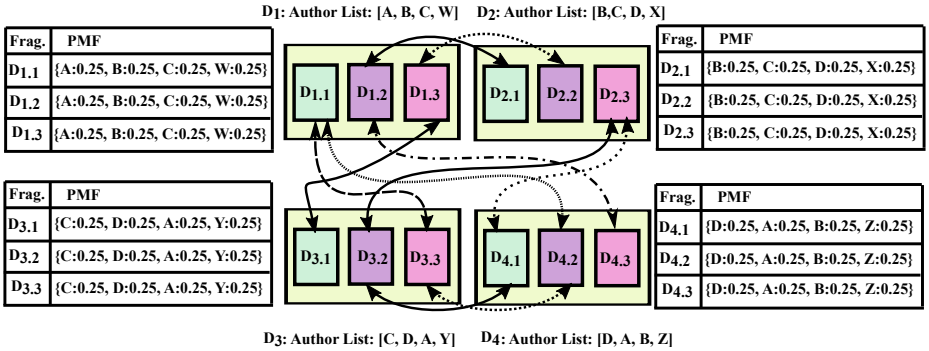


Fig. 3. Co-Authorship Graph: Each vertex represents a document fragment and each edge represents the 2 nearest neighbors of each fragments. The dotted and dashed patterns are used to only help distinguish overlapping crossing edges.Initial PMFs of all fragments are given in the corresponding tables.

The purpose of *Co-Authorship Graph (CAG)* training is to alter the PMF of each fragment in order to better reflect the true writer(s) of that fragment. Algorithm 2 shows how the PMF of each CAG vertex can be updated. The same algorithm is executed at each vertex in multiple iterations (called *supersteps*). In the algorithm, each vertex corresponds to a document fragment and each edge denotes the stylistic similarity between two fragments. Each vertex (fragment) keeps track of the top$-k$ most similar fragments as neighbors. The algorithm contains three main parts: *Receive*, *Compute* and *Send*.

- *Receive (Lines 5 to 10).* The vertex receives the PMFs from its neighbors.
- *Compute (Line 11).* The vertex PMF is updated as the weighted average of all neighbors' PMFs. These weights are obtained from the distances of the neighbors through the *Probabilistic k Nearest Neighbor* method with the *radial basis function (Gaussian) kernel* [10]. The total weight is assumed to have been normalized to 1.
- *Send (Lines 12 to 13).* The updated PMF is sent to the neighbors.

Algorithm 2 CAG Training

```
1: procedure UpdateCAGVertex
2:     NeighborPMFs ← []
3:     NeighborDistances ← []
4:     V ← ThisVertex
5:     for N in V.GetNeighbors() do
6:         PMF ← ReceivePMF(N)
7:         PMF ← RemoveNonAuthors(PMF, V.AuthorList)
8:         PMF ← Renormalize(PMF)
9:         NeighborPMFs.Append(PMF)
10:        NeighborDistances.Append(Distance(V, N))
11:    V.PMF ← ComputeWeightedAvg(NeighborPMFs, NeighborDistances)
12:    for N in V.GetNeighbors() do
13:        SendPMF(N, V.PMF)
```

At each superstep, the same process described in Algorithm 2 is repeated in all vertices and supersteps are repeated until all PMFs converges.

Consider now how Algorithm 2 operates in the context of the example given in Fig. 3. Consider the fragment $D_{1.1}$. The vertex receives 2 PMFs from its 2 neighbors $D_{3.3}$ and $D_{4.2}$ as $\{C : 0.25, D : 0.25, A : 0.25, Y : 0.25\}$ and $\{D : 0.25, A : 0.25, B : 0.25, Z : 0.25\}$, respectively (Line 4). Each PMF is compared against the author list $[A, B, C, W]$ to remove the authors that do not appear in the author list of D_1 (Line 5). In this case, D and Y are disregarded for $D_{3.3}$. Similarly, D and Z are disregarded for $D_{4.2}$. After re-normalization, we obtain $\{C : 0.5, A : 0.5\}$ and $\{A : 0.5, B : 0.5\}$ as the PMFs for $D_{3.3}$ and $D_{4.2}$, respectively. For ease of exposition, we assume that all 2 NNs have the same distance to its respective fragment and hence contributes to the fragment's PMF equally. As a result, the weighted average of the two PMFs is $\{A : 0.5, B : 0.25, C : 0.25\}$ after the first superstep.

Following the same process we obtain $\{A : 0.25, B : 0.5, C : 0.25\}$ for $D_{1.2}$, $\{A : 0.25, B : 0.25, C : 0.5\}$ for $D_{1.3}$, $\{B : 0.5, C : 0.25, D : 0.25\}$ for $D_{2.1}$, $\{B : 0.25, C : 0.5, D : 0.25\}$ for $D_{2.2}$, $\{B : 0.25, C : 0.25, D : 0.5\}$ for $D_{2.3}$, $\{C : 0.5, D : 0.25, A : 0.25\}$ for $D_{3.1}$, $\{C : 0.25, D : 0.5, A : 0.25\}$ for $D_{3.2}$, $\{C : 0.25, D : 0.25, A : 0.5\}$ for $D_{3.3}$, $\{D : 0.5, A : 0.25, B : 0.25\}$ for $D_{4.1}$, $\{D : 0.25, A : 0.5, B : 0.25\}$ for $D_{4.2}$, and $\{D : 0.25, A : 0.25, B : 0.5\}$ for $D_{4.3}$. We can see that all PMFs are becoming less uniform after only the first superstep.

For each document, the PMFs will converge to the following values.

- Document D_1: $\{A : 1\}$ for $D_{1.1}$, $\{B : 1\}$ for $D_{1.2}$, and $\{C : 1\}$ for $D_{1.3}$.
- Document D_2: $\{B : 1\}$ for $D_{2.1}$, $\{C : 1\}$ for $D_{2.2}$, and $\{D : 1\}$ for $D_{2.3}$.
- Document D_3: $\{C : 1\}$ for $D_{3.1}$, $\{D : 1\}$ for $D_{3.2}$, and $\{A : 1\}$ for $D_{3.3}$.
- Document D_4: $\{D : 1\}$ for $D_{4.1}$, $\{A : 1\}$ for $D_{4.2}$, and $\{B : 1\}$ for $D_{4.3}$.

As can be seen, the NWAs of each document are not included in the PMFs and the author lists of D_1,D_2, and D_3 are correctly identified as $[A, B, C]$, $[B, C, D]$, $[C, D, A]$, and $[D, A, B]$, respectively.

3.3 Multi-authorship Prediction

In this subsection, we explain how we can make a multi-authorship prediction for a query document $\mathcal{Q}$ using the trained document fragments obtained from the two preprocessing steps. Algorithm 3 provides the structure of this process. The query document $\mathcal{Q}$ is decomposed into multiple query fragments. For each query fragment Q (Lines 4 to 11), we find the k nearest neighbors using the same GetKNN() function introduced in the CAG construction step (cf. Algorithm 1). In a fashion similar to that in the CAG training process (cf. Algorithm 2), the PMFs of the neighboring fragments and their distances with respect to Q are used to compute the weighted average to make a single prediction. After obtaining the PMFs of all query fragments (Line 12), we compute the average PMF to make a final prediction for the entire document.

Algorithm 3 Authorship Identification

```
 1: procedure Multi-AuthorshipPrediction
 2:     FragmentPMFs ← []
 3:     QueryFragments ← GetDocumentFragments(Q)
 4:     for Q in QueryFragments do
 5:         Neighbors ← GetKNN(Q, Fragments)
 6:         NeighborPMFs ← []
 7:         for N in Neighbors do
 8:             NeighborPMFs.Append(PMF)
 9:             NeighborDistances.Append(Distance(Q, N))
10:         Q.PMF ← ComputeWeightedAvg(NeighborPMFs, NeighborDistances)
11:         FragmentPMFs.Append(Q.PMF)
12:     return GetDocumentPMF(FragmentPMFs)
```

According to Fig. 3, given that there is a query document Q_1 with $Q_{1.1}$ and $Q_{1.2}$ as its fragments. We assume that $D_{1.1}$ and $D_{3.3}$ are identified as the 2 NNs of $Q_{1.1}$, $D_{1.3}$ and $D_{3.1}$ are identified as the 2 NNs of $Q_{1.2}$. We can then obtain $Q_{1.1}$ and $Q_{1.2}$ predictions as the following PMFs: $\{A : 1.0\}$ and $\{C : 1.0\}$, respectively. As a result, the document prediction for Q_1 is $\{A : 0.5, C : 0.5\}$, i.e., A and C are the authors of Q_1.

4 Performance Evaluation

In this section, we report results from our experimental studies. We compare the performance of the proposed solution against the best existing stylometry-based method for *authorship identification in multi-author documents (AIMD)* [5] and its improved version.

Competitive methods. The competitive method presented in *stylometric authorship attribution of collaborative documents (AICD)* [5] is based on a linear *support vector machine (SVM)* classifier. For the training documents, AICD

makes use of copy transformation in which m single-label samples is created from each training sample associated with m labels. In this way, we can associate each single-label sample to one label at a time [21]. As for the features, AICD extracts the *"Writeprints Limited Features Set"* [1] from multi-author documents using the JStylo tool [13]. The output from the linear SVM classifier is converted into a probabilistic distribution. AICD uses the most probable m authors as their result, where m is the given number of co-authors.

Furthermore, we formulate an improved variant of the AICD, named as I-AICD in this paper. In I-AICD, we use the sliding window method to generate chunks of 1,000 tokens and follow the same procedure as used in AICD. To this end, we aggregate our chunk level predictions by having each chunk vote for its most likely author. As for both of the techniques mentioned above, the 5-fold cross-validation is used for evaluation.

4.1 Experimental Setup

In this subsection, we describe the datasets used in this investigation along with the performance measures and parameters settings. One synthetic dataset and two real datasets are used to evaluate the three methods.

Synthetic dataset. To generate a corpus of multi-author documents, we retrieved a collection of 23,096 single-author documents written by a set of 8,698 authors from online Project Gutenberg[4]. We first found a set A of authors such that each author $a_i \in A$ had 15 or more single-author documents, where each document had at least 6,000 tokens. Assume that D_a is a document written by m authors in A. The document D_a is generated by randomly selecting m authors $\{a_1, ..., a_m\}$ from A. For each author a_i, we obtained a text sample of $\mathcal{L}/m$ tokens where $\mathcal{L}$ is the synthetic document length. In this way, each author in the same document has the same number of tokens. Note that once a single-author document had been used in a multi-author document, it was never used again in any other document to avoid any possible training-testing sample contamination. Furthermore, each co-author set $\{a_1, ..., a_m\}$ was unique.

Real Datasets. As for real datasets, we retrieved two sets of research papers from `arXiv.org`: (i) Computer Sciences; and (ii) Social Sciences. Specifically, we sampled a set of papers from the real word datasets such that each author had his/her name appear in 5 papers. As can be seen from Table 1, for the Computer Science papers, we got a resulting dataset of 1,957 papers from a set of 707 authors. As for Social Sciences papers, we got a dataset of 616 papers from a set of 300 authors.

Parameter Settings. We tested different values for each parameter in order to find the most appropriate value. In the interest of conciseness, we display only the final results of this test. For synthetic dataset, the size of each synthetic document($\mathcal{L}$) was fixed at 12,000 tokens. The chunk size was set at 1,000 tokens and the fragment size at 6,000. Chunk-level and fragment-level sliding window

[4] `https://www.gutenberg.org`

Table 1. Statistics of the Datasets

	Synthetic Dataset	Real Dataset (Computer Science)	Real Dataset (Social Sciences)
#Authors	1,360	707	300
#Documents	3,600	1,957	616
#Tokens	43,200,000	22,139,274	15,613,718

increments were set to 100 and 2,000 tokens, respectively. The k value of 10 was used for the top-k retrieval.

Evaluation Measures. Two types of measures were used in this experiment. (i) **Accuracy (A):** The accuracy indicates the discrepancy of a prediction with respect to the ground truth, which was defined as the number of correctly predicted authors divided by the size of the true co-author set. (ii) **Guess-one (G):** A document was considered correct if the prediction contains at least one of the true authors.

4.2 Experimental Results

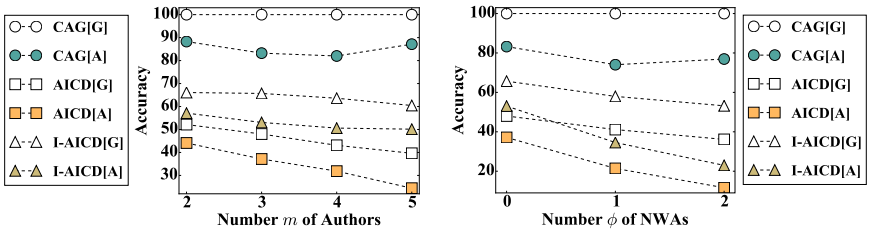
Our experimental studies were designed to verify whether our proposed method can handle (i) a larger number of co-authors (than those used in existing studies); and (ii) non-writing authors (NWAs). To control the number m of co-authors and the number ϕ of NWAs, these studies were conducted on our synthetic dataset. In addition, we also conducted experimental studies on two real datasets to show that our method can handle real-world corpora. Results from the studies are reported as follows.

Effect of Number m of Authors. We studied the effect of number m of co-authors on accuracy by varying m between 2 and 5. The number ϕ of NWAs was set to the default value of 0. These m values were chosen because they conformed with the numbers of co-authors in the real datasets used in these experimental studies. Moreover, the bibliometric analysis of different disciplines shows that mostly the average number of authors per paper are 5 or less [2]. Figure 4(a) shows that our method was the best performer, while the improved variant of the competitive method *I-AICD* performed slightly better than *AICD*. Furthermore, the performance gap between our proposed method and *I-AICD* increases as the number m of co-authors increases. Our method can handle a larger number of authors better than the two competitive methods. We can also see that our method had maintained the perfect *guess-one* accuracy in all cases.

Effect of Number ϕ of Non-writing Authors We study the effect of number ϕ of non-writing authors (NWA) on the accuracy as we vary ϕ from 0 to 2. The number m of co-authors is set to the default value of 3. As can be seen from Table 4(b), including non-writing (NWA) authors into the list of actual authors negatively affects the prediction accuracy. Specifically, the accuracy level drops from 83.24 to 74.05 as we increase the value of ϕ from 0 to 1, while further increasing ϕ to 2 has no significant effect on the accuracy. The figure also shows that our method continues to be the best performer in this study, while *I-AICD*

performs substantially better than *AICD*. Since *AICD* and *I-AICD* are not designed to handle NWAs, the accuracy levels of the two methods drastically drop as the ϕ is increased from 0 to 2. We can also see that our method had maintained the perfect *guess-one* accuracy in all cases.

Real Datasets. We evaluated the proposed method on real datasets. Note that unlike the synthetic dataset, the real datasets do not contain the ground truth regarding the number of NWAs of each document. As a result, for accuracy measurements, we assumed that all listed authors are assumed to be the writing authors. This assumption makes the measured accuracies of all methods lower than their actual values. However, it allows us to compare the three methods using real-world data. As can be seen from Table 2, the proposed method significantly outperformed the two competitive methods. Note that due to the unknown NWAs in the corpora, the accuracy level of our method reported here was lower than those of the synthetic datasets. We can also see that our method maintained the perfect *guess-one* accuracy in all cases.



(a) Effect of the number m of Authors (b) Effect of the number ϕ of NWAs

Fig. 4. Comparison of CAG Performance against competitors: Method[G] denotes Guess-one Accuracy and method[A] denotes the accuracy.

Table 2. Real Dataset Results

	Computer Science		Social Science	
Method	Accuracy	Guess-one	Accuracy	Guess-one
CAG	72.17	100	42.46	100
I-AICD	26.02	48.21	29.41	54.49
AICD	16.46	31.26	21.31	40.15

5 Conclusions

We have presented a solution for authorship identification of multi-author documents. The crux of our solution lies in the ability to probabilistically attribute

different parts (fragments) of the same documents to different subsets of co-authors. Specifically, we have proposed a data structure called the *Co-Authorship Graph (CAG)* to capture stylistic similarity between pairs of fragments across the entire document corpus. We have also formulated a CAG training algorithm to learn the true writer(s) of each fragment. We evaluated the proposed solution using one synthetic dataset and two real datasets. Our experimental results have shown that our method had (i) significantly outperformed the best existing solution; (ii) could effectively handle a larger number of co-authors; and (iii) could handle non-writer authors (NWAs).

References

1. Abbasi, A., Chen, H.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans. Inf. Syst. 26(2), 7:1–7:29 (2008)
2. Akhavan, P., Ebrahim, N.A., Fetrati, M.A., Pezeshkan, A.: Major trends in knowledge management research: a bibliometric study. Scientometrics 107(3), 1249–1264 (2016)
3. Baron, G.: Influence of data discretization on efficiency of bayesian classifier for authorship attribution. Procedia Computer Science 35, 1112–1121 (2014)
4. Bradley, J.K., Kelley, P.G., Roth, A.: Author identification from citations. Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep (2008)
5. Dauber, E., Overdorf, R., Greenstadt, R.: Stylometric authorship attribution of collaborative documents. In: First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings. pp. 115–135 (2017)
6. Giannella, C.: An improved algorithm for unsupervised decomposition of a multi-author document. JASIST 67(2), 400–411 (2016)
7. Grieve, J.: Quantitative authorship attribution: An evaluation of techniques. LLC 22(3), 251–270 (2007)
8. Hassan, S.U., Sarwar, R., Muazzam, A.: Tapping into intra-and international collaborations of the organization of islamic cooperation states across science and technology disciplines. Science and Public Policy 43(5), 690–701 (2016)
9. Hill, S., Provost, F.: The myth of the double-blind review?: author identification using only citations. Acm Sigkdd Explorations Newsletter 5(2), 179–184 (2003)
10. Holmes, C., Adams, N.: A probabilistic nearest neighbour method for statistical pattern recognition. J R Stat Soc Series B Stat Methodol 64(2), 295–306 (2002)
11. Li, J., Zheng, R., Chen, H.: From fingerprint to writeprint. Commun. ACM 49(4), 76–82 (2006)
12. Lipikorn, R., Shimizu, A., Kobatake, H.: A modified hausdorff distance for object matching. In: Pattern Recognition. vol. 1, pp. 566–568 (1994)
13. McDonald, A.W.E., Afroz, S., Caliskan, A., Stolerman, A., Greenstadt, R.: Use fewer instances of the letter "i": Toward writing style anonymization. In: Privacy Enhancing Technologies - 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings. pp. 299–318 (2012)
14. Mosteller, F., Wallace, D.: Inference and disputed authorship: The federalist (1964)
15. Nutanong, S., Yu, C., Sarwar, R., Xu, P., Chow, D.: A scalable framework for stylometric analysis query processing. In: ICDM (2016)

16. Payer, M., Huang, L., Gong, N.Z., Borgolte, K., Frank, M.: What you submit is who you are: A multimodal approach for deanonymizing scientific publications. IEEE Trans. Information Forensics and Security 10(1), 200–212 (2015)
17. Ramnial, H., Panchoo, S., Pudaruth, S.: Authorship attribution using stylometry and machine learning techniques. In: IJISTA, pp. 113–125 (2016)
18. Rexha, A., Klampfl, S., Kröll, M., Kern, R.: Towards a more fine grained analysis of scientific authorship: Predicting the number of authors using stylometric features. In: Proceedings of the Third Workshop on BIR co-located with the 38th (ECIR 2016), Padova, Italy, March 20, 2016. pp. 26–31 (2016)
19. Sboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., Moloshnikov, I.: Machine learning models of text categorization by author gender using topic-independent features. Procedia Computer Science 101, 135–142 (2016)
20. Stamatatos, E.: A survey of modern authorship attribution methods. JASIST 60(3), 538–556 (2009)
21. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. IJDWM 3(3), 1–13 (2007)
22. Zhang, M., Zhou, Z.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)

Appendix A Stylometric Features

The stylometric features used in this investigation are shown in Table 3. For features 5 to 12, N represents count of words and V represents count of distinct words. For Features 6 and 9, V_i represents the count of words that occur i times.

Table 3. List of Stylometric Features

Lexical Features		
1. N: Total #words	2. V: Total #distinct words	3. Average word length
4. S.D. of word lengths	5. $\frac{V}{N}$	6. $VR(K) = \frac{10^4(\sum i^2 V_i - N)}{N^2}$
7. $VR(R) = \frac{V}{\sqrt{N}}$	8. $VR(C) = \frac{\log V}{\log N}$	9. $VR(H) = \frac{(100 \log N)}{(1 - V_1)/V}$
10. $VR(S) = \frac{V_2}{V}$	11. $VR(k) = \frac{logV}{\log(\log N)}$	12. $VR(LN) = \frac{(1-V^2)}{V^2(\log N)}$
13. Entropy of word freq. ditri.	14. Total number of chars	15. Freq. of alpha chars
16. Freq. of uppercase chars	17. Freq. of lowercase chars	18. Freq. of numeric chars
19. Freq. of special chars	20. Freq. of white spaces	21. Freq. of punctuations
22. Alpha char ratio	23. Uppercase char ratio	24. Lowercase char ration
25. Numeric char ratio	26. Special char ratio	27. White spaces ratio
Syntactic Features		
28. Freq. of nouns	29. Freq. of proper nouns	30. Freq. of pronouns
31. Freq. of ordinal adjs.	32. Freq. of comparative adjs.	33. Freq. of superlative adjs.
34. Freq. of advs.	35. Freq. of comparative advs.	36. Freq. of superlative advbs.
37. Freq. of modal auxiliaries	38. Freq. of bases form verbs	39. Freq. of past verbs
40. Freq. of present part. verbs	41. Freq. of past part. verbs	42. Freq. of particles
43. Freq. of wh-words	44. Freq. of conjunctions	45. Freq. of numerical words
46. Freq. of determiners	47. Freq. of existential theres	48. Freq. of existential to
49. Freq. of prepositions	50. Freq. of genitive markers	51. Freq. of quotations
52. Freq. of commas	53. Freq. of terminators	54. Freq. of symbols
Structural Features		
55. Total number of sentence	56. Avg. #words per sentence	