Frequent Subgraph Patterns for Author Verification Task

Daniel Castro-Castro[1*], Niusvel Acosta-Mendoza[2], Reynier Ortega Bueno[1], Rafael Muñoz[3], Yoan Gutierrez Vazquez[3]

[1]Center for Pattern Recognition and Data Mining (CERPAMID), calle Patricio Lumumba S/N, Rpto Quintero, Santiago de Cuba, Cuba
{daniel.castro, reynier.ortega}@cerpamid.co.cu
[2] Advanced Technologies Application Center (CENATAV), 7a # 21406 e/ 214 and 216, Siboney, Playa, CP: 12200, Havana, Cuba
nacosta@cenatav.co.cu
[3] Dpto. Lenguajes y Sistemas Informáticos. Universidad de Alicante (UA), Carretera San Vicente del Raspeig s/n, 03690, Alicante, España
{rafael, ygutierrez}@dlsi.ua.es

Abstract. The aim of the present work is modeling the authorship verification task, using graph-based representation of documents considering several linguistic features, and analyzing the authorship of unknown documents based on the number of patterns (subgraphs) extracted from the documents of a known author, between the unknown document. The classification is made based on the number of subgraph matched in the graph of the anonymous document considering, the use of maximum and minimum thresholds of overlap, adjusted from training collections. The final decision is made based on a majority vote over each linguistic graph representation. Experiments were performed over the Spanish Dataset of the PAN 2015 competition, where promising results were achieved. The representation based on Part of Speech was the most stable one, even though the test dataset was heterogeneous with respect to topic and genre.

Keywords: Subgraph pattern, author verification, linguistic features.

1 Introduction

From the beginning of the digital information, there has been a great interest in automatic methods for determining the authorship of anonymous documents based on clear evidence [18]. In this sense, finding elements that allow to know if a certain author wrote a document or not, have led different branches of science to seek methods for defining and finding an optimal solution [20] [3] [21]. In this context, Computational sciences have made remarkable contributions and innovations by using *stylometric* techniques to perform the analysis of digital texts for the authorship detection task [9] [11] [20].

* Corresponding author: Daniel Castro Castro, e-mail daniel.castro@cerpamid.co.cu, postal address calle 2da #20 %ByC AltaVista Santiago de Cuba, Cuba.

Authorship detection is a challenge task, and it is become more complex when the author´s documents are about multiple topics (Politics, Art, Finance, Sport, Health, etc.) and differs in the textual genre (novels, news, reviews, tweet, etc.) [20]. Particularly, the representation of linguistic information using a graph-based modeling would allow capturing linguistic patterns (e.g. frequent subgraphs) in several authored documents.

The main goal of this work is to address the problem of authorship verification by using a graph-based documents representation taking into account several linguistic layers. For that, nonlinear frequent patterns (subgraph) from the known author documents are discovered using graph mining techniques. Later, the authorship of the unknown document is resolved considering the overlapped between the frequent subgraph extracted from the author and the graph of the unknown document.

The remainder of this paper is organized as follow: In Section 2 we present a summarized state of the art about the most relevant work for automatic authorship detection. Afterwards, in Section 3, we introduce our proposal for authorship verification which relies on frequent subgraphs discovering. Subsequently, experiments and results achieved by our proposal over PAN 2015 dataset are discussed in Section 4. Finally, our conclusions and future work directions are remarked in Section 5.

2 State of the art

The Authorship Verification task aims at identify the author of an anonymous document. For that, there are developed algorithms which can learn the writing style of one or several authors and automatically identifying the authorship of new documents [20] [12]. In this task, there are two fundamental approaches: the *verification* of authorship, and the *identification* of authorship [20]. In the authorship verification, there are only samples of one author and the anonymous document to be classified; and in the authorship identification, there are samples of several authors and a document to be classified.

Document representation is a crucial step for developing robust and effective models for automatically classifying, in accurate way, the authorship of a given document. For that, several techniques for documents representation have been proposed in the literature [20], for example:

Bag of Words (BoW): In this representation, the text is seen as an unordered collection of words, where the frequency of occurrence of the words is more important than the position or relation between them.

Graphs: In this representation, a diagram of vertices interconnected by edges is used, where each vertex represents an specific knowledge and the edges are the way to represent the relations between vertices [5].

Once the computational representation of the linguistic features has been performed, text classification methods can be used to categorize an anonymous document. The textual features are usually extracted from linguistic layers, which form small structured units within documents without a well-defined structure. These linguistic layers are: the *layer of phonemes*; the *character layer*; the *lexical layer*; the *syntactic layer*; and the *semantic layer* [20] [6]. The phonemes layer includes features based on phonemes that

can be extracted from documents by means of dictionaries (e.g. the International Phonetic Alphabet). The character layer includes features based on prefixes, suffixes or n-grams of characters. The lexical layer consider features based on auxiliary words. The syntactic layer includes features based on syntax as components or positions. The semantic layer consider features based on semantics such as homonyms or synonyms or even word/s relations extracted from ontologies.

Most of the reported works are based on BoW, resulting the linguistic features employed the main difference among them. This representation reported good execution times and good results for verification and identification in problems that deal with different topics or authors with diverse writing style, even when the documents of an author were not homogeneous according to the topic or genre [20]. However, the BoW model is difficult to use in problems where the documents samples of different authors have the same topic and they use topic related words in a similar way. This fact is because the features in the BoW vectors do not maintain a relationship with each other, affecting the differentiation among authors.

Hence, in recent years several works have proposed the representation of texts based on graphs also using different types of textual features [5] [15] [19]. This representation maintains the relationships between the different textual features achieving greater accuracy in collections of authors with very similar linguistic features.

An interesting approach using graph-based document representation was presented by Castillo et al., 2012. In this work, the authors represented each paragraph in the document as a graph. Specifically, vertices are the lemmas of the words into the paragraph and the edges are established from one vertex to another adjacent in the sentence. In this representation, the edges contain the morphological label of the words, with the purpose of looking for patterns in the subgraphs. The proposal of [19] use the method based on the profile[1] (a prototype); however, the authors of this work consider that it would be interesting to use the instance-based method, representing a graph for each training document, and not a single graph with the concatenation of all the training documents of an author. This work also employs the concept of integrated syntactic graphs, where all textual features are integrated into a single graph.

Reviewing the state-of-the-art we identified that there are few approaches for authorship verification task that used graph-based representation of the documents. The use of a graph-based representation, from linguistic features in different layers of natural language analysis, would allow the effective identification of the author of an anonymous document or discard the authorship of it.

The main contribution of this work corresponds to the implementation of an Authorship Verification algorithm, using prototype-based representation where the features are the frequent subgraphs extracted from the set of documents of the author. In addition, a strategy for the optimization of the parameters of our method is also detailed (cf. Section 3.4).

---

[1] The profile-based paradigm attempts to capture the style of authors by computing a single representation for all texts written by the same author, a so-called author profile.

3 Frequent Subgraph Pattern Classification

The graph-based representation of documents, allows analyzing linguistic patterns to determine the writing style of authors based on its digital signatures representation [4]. Besides, with graph representation it is possible to capture linguistic features, including relationships between them, of patterns that could not be obtained by using a BoW representation [5]. Thus, we propose a method based on a graph representation for the task of Author Verification.
The proposed method comprises the following steps:

1. Representation of the document using a graph-based approach.
2. Extracting frequent subgraph from author known documents, and building the author prototype.
3. Decision rule based on subgraph overlapping (matching).

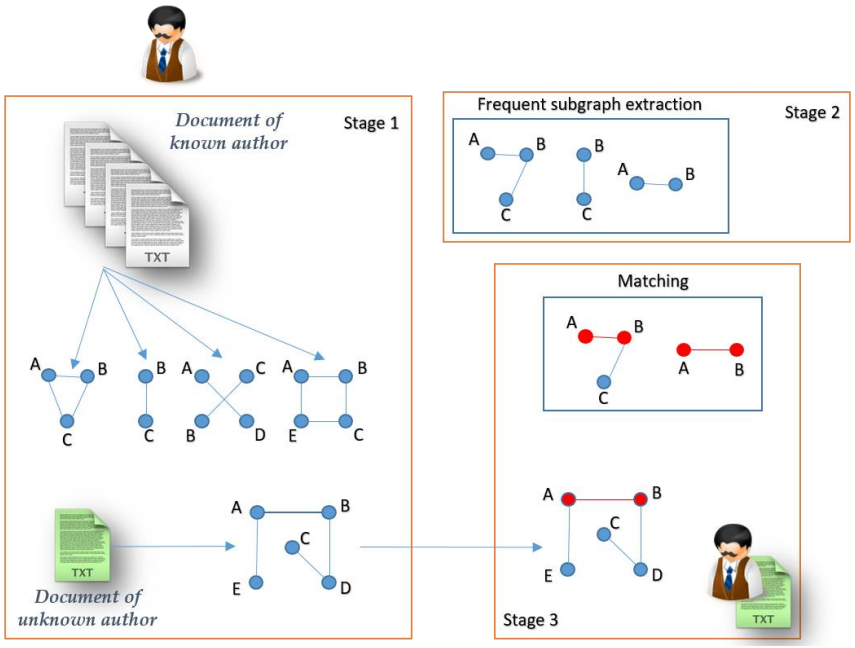

Fig. 1. Architecture of our authorship verification method based on frequent subgraphs. The letters represent features of one of the linguistic layer-representation.

Figure 1 shows our proposal architecture, where three main stages are highlighted: one for the graph representation of the textual information for documents considering a specific feature type (Character, Lexical or Syntactic); the second stage is for the frequent subgraphs extraction, which is used as for representing the writing style of the author; and a third stage for identifying the unknown documents (step 3 of the method).

In the first step of the method (stage one of the architecture and first step of the previous algorithm), it is necessary to represent all the authored documents using graph representation. To do that, we proceed to use natural language processing tools to obtain all the possible linguistic features from the known documents. In Section 3.1, we explain the different feature types that are extracted from the contents of the documents, which are used for build a graph for each document. In this representation, the vertices are the linguistic features and the edges represent the adjacency of these vertices in the context of the sentence, including, as edge label, the frequency in which this adjacency occurs.

For the second stage, a subgraph mining algorithm is used (see Section 3.2), in order to identify the writing style patterns. These patterns should allow to identify writing features with a small dimension of the vectors used in traditional BoW approach. From the mining subgraphs stage, we get a prototype for each author composed by the set of subgraphs mined from the correspondent author documents. Our method is evaluated for each feature type in the representation and without combining the obtained results. Each of the subgraphs would be a feature in the obtained prototype.

Finally, in the classification phase, we have the graph of the anonymous document, where we analyze, for each layer-representation, how many of the author subgraphs appear in the graph of the document to be classified. This information allows to evaluate the usefulness of the subgraphs in the Authorship Verification task. A pseudocode of our method is presented below.

Author Verification method (frequent subgraph)

Input: $D_A = \{D_{a1},...,D_{ai}\}$, D_U, P_A, P_B

Output: <True> or <False> or <Unknown>

1. **foreach** D_{ai} in D_A
   GD_{ai} = Build graph representation(D_{ai})
2. $SubG_A$ = gdFil($\{GD_A\}$)
3. GD_U = Build graph representation(D_U)
4. M = Matching(GD_U, $SubG_A$)
5. Answer
   a) If $M > P_A$, *Answer* = <True>
   b) If $M < P_B$, *Answer* = <False>
   c) If $M > P_B$ and $M < P_A$, *Answer* = <Unknown>

D_A is the set of known documents, D_U is the unknown document, and (P_A, P_B) the thresholds used to give an answer. The first step is the construction of a graph for each document in the set D_A, as a result we obtain a set of graph representation GD_A. In the second step is obtained all the frequent subgraph using a frequent subgraph mining algorithm and the result is a prototype object $SubG_A$ composed by all de subgraph extracted. Next we build the graph of D_U and in the four step is executed the Matching function in order to get the number of $SubG_A$ presented in the graph GD_U. The number of subgraph matched M, are then compared to the input parameters thresholds to decide the answer of the method.

3.1 Linguistic Features for the Graph Representation

We consider six types of linguistic features of those reported in the literature, grouped in the following three dimension of linguistic features.

1. **Character Layer**:
   - ***N-grams of characters***: all sequences of *N* characters, without deletion of elements in the text. For the experiments, we evaluate several values of *N* and the best results were achieved for *N* = 3 (**3GC**) and *N* = 4 (**4GC**).
   - ***N-grams of k-size Prefixes***: a representation is constructed by taking only the *N* character sequences of size *k* at the beginning of words. We tested with different *N* and *k* values and used for experiments *N* = 1 and *k* = 3 (**3P**).
   - ***N-grams of k-size Suffixes***: this is similar to the previous representation, but taking the *N* character sequences of size k at the end of each word. We tested with different *N* and *k* values and used for experiments *N* = 1 and *k* = 3 (**3S**).

The features of this layer are simple to compute and this approach do not require effective tools for deep natural language processing.

2. **Lexical Layer**:
   - ***N-grams of words***: sequences of *N* consecutive words of a tokenized text. We construct the representations using *N* = 1. *N* was taken with 1 after testing with *N* from 1 to 5 and obtaining the best results with *N* = 1 (**W**).

Similar to character layer, lexical features can be obtained employing simple tools such as text tokenizers and are used to get writing patterns through the use of words, consecutive sequences of words, among others.

3. **Grammatical Layer**:
   - ***N-grams of Part of Speech tags (PoS)*:** sequences of *N* consecutive PoS tags after a text tagger execution. We construct the representations using *N* = 1. *N* was taken with 1 after testing with *N* from 1 to 3 and obtaining the best results with *N* = 1.

The features of this layer are a bit more complex, depending on the tagger tools and are language dependent. Require more time to be calculated and are used to determine writing patterns through the use of grammatical categories and lemmatization of words. For example, if we have a document with these two sentences “El pueblo ha sido feliz” and “El público ha sido comprensivo”, the graph representation using *N*-grams of word (*N* = 1) will produce a graph like the one illustrated in Figure 3.

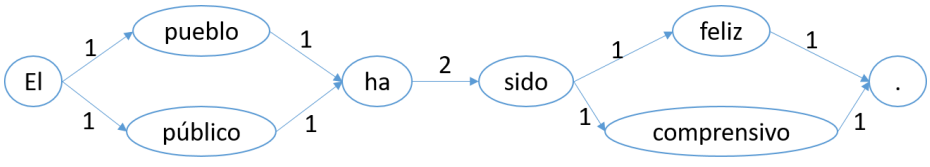

Fig. 2. *N*-gram word graph representation of a document.

3.2 Frequent Subgraph Mining Algorithm

Having the representation in graphs of each known document of the author, we proceed to the extraction of the frequent subgraphs, considering as frequent, those subgraphs that occur in at least two of the author's documents.

The method gdFil [7] is an algorithm for mining Frequent Subgraphs (FSs) in simple graph collections. This algorithm is based on a pattern growth approach where the FSs are calculated through the Depth-First Search (DFS). In this algorithm, several pruning was introduced for decreasing the generation of subgraph candidates, accelerating the mining process. Besides, the DFS structure is used for efficiently representing FSs and speeding up the calculation of the pattern supports avoiding all the sub-isomorphism tests. The DFSE structure allows to maintain the occurrences of FSs in each graph of the collection, which avoids the exhaustive occurrences searching. Another thing to mention is that gdFil uses the canonical form based on DFS trees for representing isomorphic graphs. In this way, duplicate candidate generation is avoided and the sub-isomorphism test problem is transformed, which is a NP-Hard problem, into a problem of simple chain comparisons.

gdFil begins by removing all the vertices and edges that are not frequent in the given graph collection. This action can be applied because this algorithm is based on the descending closure property, which says that a non-frequent subgraph cannot produce a frequent one. Later, gdFil recursively extends all FSs, beginning with the frequent edges, by adding a new edge at a time. These candidates are represented by a DFS tree in the DFSE structure. This extension process is performed on the candidate subgraphs that meet the support threshold and as long as there is a frequent edge that has not been extended.

It is important to highlight that; gdFil is one of the most efficient algorithms of the state-of-the-art [7].

3.3 Prototype Construction base on Frequent Subgraphs

A prototype is obtained for each author, conformed by the set of subgraphs extracted from his known documents, and for the classification phase it is considered only that the subgraph exists.

In our case, two subgraphs with the same vertices and edges, but with different frequencies in the edges are considered as equal subgraphs when we look for them to be present in the anonymous document.

For the future, a weight could be considered for each subgraph where the number of its occurrences in the sample documents and the frequency of each edge are evaluated.

3.4 Authorship Verification using Subgraph Matching

For the authorship verification task, we have several sample documents of one author, and the idea is to decide the authorship of an anonymous document using only the samples provided by him.

From the graph representation built starting by the known documents, a prototype object using the mined FSs is computed and these subgraphs are matched with the graph of the anonymous document. For those cases in which the anonymous document corresponds to the author we computed a parameter called *Parameter A* (the number of matching between the prototype of this problems and the graph of the anonymous document), so the Problems A are those of set of authors were the anonymous document was written by them. On the other hand, Problems B are those were the authors did not write the anonymous document, and the number of matching between the prototype of these problems and the graph of the anonymous document it is called *Parameter B*.

Those parameters (A and B) are the thresholds used to decide when the author wrote the anonymous document. When we do the matching, if the number of matched subgraphs are greater than *Parameter A*, then it is considered that the author wrote the document. If the number of matched subgraphs are less than the *Parameter B*, then the author did not write it. When the matched number of subgraphs are between the two parameters the answer is Abstention, it cannot decide a *True* or *False* answer.

A final answer can be taken evaluating the answer that appears by majority, taking the response of each representation. We consider a simple majority vote for the final decision. When the number of *True* answer are equal than the number of *False*, then it is considered an *Abstention*, because there is no majority for each of the expected answer.

4 Experiments and results

The aim of our experiments is to assess the document graph-based representation, as well as the usefulness of the linguistic patterns captured by the frequent subgraph extraction algorithm, which allow us to make a decision about the authorship of unknown documents.

4.1 Dataset

In order to verify the effectiveness of our model, we evaluated it on the authorship verification dataset provided by PAN 2015 Evaluation Forum [21]. This dataset covers four languages: Dutch, English, Greek, and Spanish. The author's documents can be distinct according to topic and genre.

The evaluation needs to be done with author's dataset in which for each author there are more than one document, in order to characterize the written style based in the number of subgraph mined using several graph-based document representations. To achieve this purpose, we used the Spanish dataset provided in the PAN 2015 Evaluation Forum.

The Spanish dataset present 4 of "known" documents by a single author and a "questioned" document, the task is to determine whether the questioned document was written by the same author who wrote the known document set. Notice that, in this dataset the genre and/or topic may differ significantly between the known and unknown documents [21]. From the 100 authors, in 50 of them, the anonymous document was written

by him (Problem A) and, in the remaining, the anonymous document was not written by the known author (Problem B). In Table 1 statistics about Spanish subset of the PAN 2015 dataset is presented.

Table 1. PAN'2015 Spanish dataset distribution.

Collection	Type	Problems (# authors)	Documents	Avg. known documents	Avg. words document
Train	mixed	100	500	4.0	954
Test	mixed	100	500	4.0	946

4.2 Evaluation measure

The used evaluation measure is the accuracy (c@1), which is one of the measures applied in the PAN'2015 Author Verification task [21] and proposed by [14].

$$c@1 = (1/n) * (nc + (nu * nc/n)),$$

Where *n* is the number of authors, *nc* is the number of correct answers, and *nu* is the number of unanswered problems. This measure considers as a correct answer when for the unknown document, the answer is *True* for Problem A, and *False* for Problem B.

4.3 Analysis of the Results Achieved over the PAN'2015 Dataset

As it was explained in section 3.4, our method depends on two parameters used as thresholds for decision. Table 2 resumes the used thresholds obtained in the Train part of the dataset and the accuracy achieved for each representation considering the Test dataset for evaluation.

Table 2. Thresholds used for the A and B problems in the collection.

Representation	Parameter A	Parameter B	$c@1$
3GC	27.7	23.6	0.54
4GC	21.08	19.58	0.5
3P	0.96	0.7	0.36
3S	4.08	2.7	0.63
W	3.84	3.1	0.52
PoS	5.08	4.5	0.59

With these experiments we want to see the impact of the thresholds on the accuracy result for each graph representation. The **PoS** and **3S** graph-based representation achieved good results, because these representations are less sensible when the samples of an author are heterogeneous according to the topic and genre. The greater difference between thresholds A and B, the better the result, because the documents that must be positive or exceed the A threshold or will not be smaller than the B. However, those that correspond to negative responses will not exceed threshold B or they will not be

greater than threshold A. In both situations, few errors would be generated. This behavior can be observed in the thresholds and results achieved with the 3GC and 3S representation.

In the PAN'2015 Authorship Verification task, the organizers proposed two baselines (Baseline PAN 2013 and Baseline PAN 2015) methods with the purpose to decide how effective could be a participating system. The proposals that do not outperform the Baseline-PAN-2013 (the best algorithm presented in PAN'2013 edition, but executed with the data of PAN'2015) are considered not good. The proposals that obtain better results than Baseline-PAN-2015 (an ensemble of all proposals) are considered very good, and adequate those between Baseline-PAN-2013 and Baseline-PAN-2015.

The Baseline-PAN-2013 results was 0.56; our proposal with PoS graph-based representation achieved 0.59 and Baseline-PAN-2015 achieved 0.8. The lowest result was 0.34 and the media, considering all the participants, was 0.62. As it can be observed, the result using only the PoS graph-based representation was considered good, and using **3S** linguistic graph representation, it was obtained even better results.

In Table 3, our results for the PAN'2015 Spanish Test dataset evaluation are summarized, employing the majority voting schema. The *Subgraph voting (all)* is the proposal using a voting with all representations and the *Subgraph voting (3)* is using the 3GC, W and PoS representations. The combination of the answers using all representations obtains good results, even when the accuracy for five of the six representations are lesser than 0.61. Using the *Subgraph voting (3) proposal*, the results were better than those achieved by using the configuration with *all* and the main difference was in the number of abstention. There are two difference, the first one correspond to an odd number of representation and those lead us to less cases where the *True* and *False* answer were equal. The second difference is that it was not used the 3P representation that achieved the worst result on the *Test* dataset.

Bagnall [1] presented a proposal using a multi-headed recurrent neural network and the rest of the works with best results used Machine-learning approximation employing Random Forest and Support Vector Machines. It is important to notice, that our approach is focused in a verification task considering only one author and it can capture stylistic subgraphs formed by words, their characters and PoS, mainly of auxiliary words and so, less influenced by the topic or the genre of the known documents.

Table 3. Comparison of our results with the participating systems in PAN 2015 Verification Task, Spanish datset.

Rank	**Approaches**	**c@1**	**Rank**	**Approaches**	**c@1**
1	Bartoli et al. [2]	0,830	9	Posadas-Durán et al [16]	0,68
2	PAN14-BASELINE-2	0,830	…		
3	Bagnall. [1]	0,814	14	gdFil voting (all)	0,61
4	PAN 15-ENSEMBLE	0,8	…		
…			18	PAN13-BASELINE	0,56
7	Hürlimann et al, [8]	0,73	…		
8	**gdFil voting (3)**	**0,71**	23	Nikolov et al. [13]	0,34

4.4 Evaluation Analysis

It is important to analyze the different answers given by our proposal for each linguistic graph representation (cf. Table 4). The answers marked as *Correct* are those when the anonymous document corresponds to the analyzed author and our proposal was able to identify it, and when the anonymous document do not correspond to the author, our proposal say *False*. The *Abstention* answer is when the algorithm does not say *True* or *False*.

It can be observed that with the 3GC representation, less errors are achieved and a major number of *Abstention* in comparisson with the rest representation, and that could be a desired results in real scenarios. In the case of PoS and 3S representation, it have the greater number of *Correct* answers, and we can conclude that the desired results are those in which the least amount of errors occur, even when there is a large number of abstentions.

Table 4. The answers obtained by using each linguistic graph representation

Collection	Answer	3GC	4GC	3P	3S	W	PoS
gdFIL Test	Correct	39	39	24	59	41	**51**
	Abstention	39	27	49	6	27	15
	Not correct	**22**	34	27	35	32	34

In our graph-based representation, only the frequency of the co-occurrence of two or more adjacent features is considered, but we are not considering the frequency of the individual features. Another aspect to distinguish is that the subgraph mined are mainly formed by stop words used in the documents and this may be the result of the variety in the samples documents in terms of topic and gender, and this are an expected result.

5 Conclusions and Future Work

The graph-based representation of documents, allow the analysis of non-linear linguistic patterns to determine the writing style of authors based on his digital documents. In addition, with the graph representation it is possible to capture important patterns based on the relations of features that could not be analyzed using the BoW representation, considering that BoW representations assumes the independence between features.

The presented proposal builds a prototype object to represent the documents of an author based on frequent subgraphs. Several linguistic features graph representations were implemented and the results achieved by using character, word and PoS representations can be considered as encouraging results and allows us to consider future works analyzing different strategies.

The combination of answer by a majority-voting scheme achieved very good results. The representation based on PoS tagging features was the most stable one, even when the evaluation collection was heterogeneous considering the topic and the document genre. It is also important to notice that the dimensionality of the prototype of each

author is shorter than a traditional BoW approach and the subgraph mined strategy is considered also as a feature selection method.

The results reflected using prefix representation are low and merits more effort in the analysis of the used samples. The main difficult with prefix representation is that there are few subgraphs mined and very few overlapping with unknown documents. Include in the representations prototype the weight and importance of each of the subgraphs, taking into account the size of the subgraph and the frequency of the edges. In addition, consideration should be given to a representation based on the canonical forms of the graphs to improve the identification of the frequent patterns in the documents to be classified. In this way, we could improve the effectiveness of the proposed scheme.

The collection with which we evaluate has as a characteristic, that all the authors present differences in their samples based on the genre and topic. This made the task difficult, but is precisely this scenario the one that could be presented in real applications, for example, in forensic document analysis.

We will as future work, to use a dataset of emails, which presents 5000 emails distributed in 50 authors, in which different topics are approached between authors and between the samples of the same author. Another interesting scenario to experiment correspond to the dataset of Spanish News proposed by [17] in which we could prove our proposal in collections of text written by authors of different nationalities.

References

1. Bagnall, D. (2015). Author identification using multi-headed recurrent neural networks. In L. Cappellato, N. Ferro, J. Gareth, & E. San Juan (Eds.), CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers (p. 11). Toulouse: CEUR-WS.org.
2. Bartoli, A., Dagri, A., De Lorenzo, A., Medvet, E., & Tarlao, F. (2015). An author verification approach based on differential features. In L. Cappellato, N. Ferro, G. J. F. Jones, & E. SanJuan (Eds.), CEUR Workshop Proceedings (Vol. 1391, pp. 1–7). Toulouse, France: CEUR-WS.org. https://doi.org/10.1007/s00256-005-0933-8
3. Castillo, E., Vilariño, D., Pinto, D., Olmos, I., González, J. A., & Carrillo, M. (2012). Graph-based and Lexical-Syntactic Approaches for the Authorship Attribution Task - Notebook for PAN at CLEF 2012. In P. Forner, J. Karlgren, & C. Womser (Eds.), Working Notes Papers of the CLEF 2012 Evaluation Labs (Vol. 1178, pp. 1–7). Rome, Italy: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-1178
4. Castillo, E., Cervantes, O., Vilariño, D., & Pinto, D. (2015). Author Attribution Using A Graph Based Representation. In 25. International Conference on Electronics, Communications and Computers, CONIELECOMP (pp. 135–142). Cholula, Puebla, Mexico: IEEE. https://doi.org/10.1109/CONIELECOMP.2015.7086940
5. Castillo, E., Cervantes, O., & Puebla, D. (2017). Text Analysis Using Different Graph-Based Representations. Computación y Sistemas, 21(4), 581–599. https://doi.org/10.13053/CyS-21-4-2551
6. Castro, D., Adame, Y., Pelaez, M., & Muñoz, R. (2017). Authorship Verification, Neighborhood-based Classification | Verificación de autoría, clasificación por vecindad. Computacion y Sistemas, 21(2). https://doi.org/10.1017/CBO9781107415324.004

7. Gago-Alonso, A., Carrasco-Ochoa, J. A., Medina-Pagola, J. E., & Martínez-Trinidad, J. F. (2010). Full duplicate candidate pruning for frequent connected subgraph mining. Integrated Computer-Aided Engineering, 17(3), 211–225. https://doi.org/10.3233/ICA-2010-0342
8. Hürlimann, M., Weck, B., Berg, E. Van Den, Šuster, S., & Nissim, M. (2015). GLAD : Groningen Lightweight Authorship Detection. In L. Cappellato, N. Ferro, G. J. F. Jones, & E. SanJuan (Eds.), Working Notes for CLEF 2015 Conference (pp. 1–12). Toulouse, France: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-1391/141-CR.pdf
9. Juola, P. (2006). Authorship attribution. Foundations and Trends in Information Retrieval, 1(3), 233–334. https://doi.org/10.1007/BF01830689
10. Juola, P. (2012). An Overview of the Traditional Authorship Attribution Subtask Notebook for PAN at CLEF 2012. In P. Forner, U. Karlgren, & C. Womser-Hacker (Eds.), CLEF 2012 Evaluation Labs and Workshop, Online Working Notes (pp. 37–41). Rome, Italy: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-1178/CLEF2012wn-PAN-Juola2012.pdf
11. Koppel, M., Schler, J., & Argamon, S. (2009). Computational Methods in Authorship Attribution. Journal of the American Society for Information Science and Technology, 60(1), 9–26. https://doi.org/10.1002/asi.20961
12. López-Monroy, A. P., Montes-Y-Gómez, M., Villaseñor-Pineda, L., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2012). A new document author representation for authorship attribution. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7329 LNCS, 283–292. https://doi.org/10.1007/978-3-642-31149-9_29
13. Nikolov, S., Tabakova, D., Savov, S., Kiprov, Y., Nakov, P. (2015). SU@PAN'2015: Experiments in Author Veri cation. In: Cappellato, L., Ferro, N., Gareth, J., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs.
14. Peñas, A., & Rodrigo, A. (2011). A Simple Measure to Assess Non-Response. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 1415–1424.
15. Pinto, D., Gómez-Adorno, H., Vilariño, D., & Singh, V. K. (2014). A graph-based multi-level linguistic representation for document understanding. Pattern Recognition Letters, 41(1), 93–102. https://doi.org/10.1016/j.patrec.2013.12.004
16. Posadas-Durán, J.P., Sidorov, G., Batyrshin, I., Mirasol-Meléndez, E. (2015). Author Verification Using Syntactic N-grams. In: Cappellato, L., Ferro, N., Gareth, J., San Juan, E. (eds.) Working Notes Papers of the CLEF 2015 Evaluation Labs.
17. Sanchez-Perez, M. A., Markov, I., Gómez-Adorno, H., & Sidorov, G. (2017). Comparison of character n-grams and lexical features on author, gender, and language variety identification on the same Spanish news corpus. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10456 LNCS, 145–151. https://doi.org/10.1007/978-3-319-65813-1_15
18. Sarwar, R., Li, Q., Rakthanmanon, T., & Nutanong, S. (2018). A scalable framework for cross-lingual authorship identification. Information Sciences, 465, 323–339. https://doi.org/10.1016/j.ins.2018.07.009
19. Sidorov, G., Pinto, D., Markov, I., & Gómez-Adorno, H. (2015). A Graph Based Authorship Identification Approach. In L. Cappellato, N. Ferro, G. J. F. Jones, & E. SanJuan (Eds.), Working Notes for CLEF 2015 Conference (pp. 1–6). Toulouse, France: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-1391/135-CR.pdf
20. Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, 60(3), 538–556. https://doi.org/10.1002/asi.21001
21. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B. (2015). Overview of the Author Identification Task at PAN 2015. In L. Cappellato,

N. Ferro, G. J. F. Jones, & E. SanJuan (Eds.), Working Notes for CLEF 2015 Conference.
Toulouse, France: CEUR-WS.org. ISSN 1613-0073.