A New Document Author Representation for Authorship Attribution

Adrián Pastor López-Monroy, Manuel Montes-y-Gómez, Luis Villaseñor-Pineda,
Jesús Ariel Carrasco-Ochoa, and José Fco. Martínez-Trinidad

National Institute for Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro #1, Tonantzintla, Puebla, México
{pastor, mmontesg, villasen, ariel, fmartine}@ccc.inaoep.mx

Abstract. This paper proposes a novel representation for Authorship Attribution (AA), based on Concise Semantic Analysis (CSA), which has been successfully used in Text Categorization (TC). Our approach for AA, called Document Author Representation (DAR), builds document vectors in a space of authors, calculating the relationship between textual features and authors. In order to evaluate our approach, we compare the proposed representation with conventional approaches and previous works using the c50 corpus. We found that DAR can be very useful in AA tasks, because it provides good performance on imbalanced data, getting comparable or better accuracy results.

Keywords: Authorship Attribution, Author Identification, Document Representation, Semantic Analysis, Text Classification

1 Introduction

Authorship Attribution (AA) consists in learning the writing style of one or more authors, in order to identify them automatically in future texts [2]. Today, the amount of information available on Internet is overwhelming, and much of it is plain text (e-mails, online forums, blogs, source code). In this context, several issues and applications related to AA have emerged, for example: ciber-bullying, plagiarism detection, spam filtering, computer forensics and fraud detection [2].

The AA task can be stated as a single-labeled multiclass classification problem, where authors represent the classes to discriminate. However, this task should not be approached exactly in the same way as thematic classification. There are important issues to consider; for example, in AA the most important textual features are non-thematic. The latter is because the main goal is to model the writing style of each author [7], in order to discriminate among them, even in the same context.

According to recent forums on AA [10], the most useful attributes to retain writing style are some specific words (like function words) and n-grams at the character level. For example, taking into account frequency and distribution of stopwords throughout a text, could help to identify its author [3]. On the other hand, n-grams at the character level could help to discover particular preferences for text structures [4]; to illustrate

this, consider a feature space of 3-grams, where a high frequency of the terms *ing* and *ed_* might discriminate authors that tend to write in gerund or in past tense.

The representation of documents is a key procedure for AA. Currently, different techniques based on words and character n-grams are being used. One of the most common approaches is the Bag of Terms (BOT) [2], which builds feature vectors of documents, taking each term in the vocabulary (e.g., words, n-grams, etc.) as an attribute. BOT like representations have been used for AA, but they have some drawbacks:

1. *They do not preserve any kind of relationship among terms or classes:* In this context, valuable information is being ignored, mainly because we believe that for stylistic features, it is useful to take into account relationships between authors and their vocabularies beyond the isolated word frequencies.
2. *They produce high dimensionality and high dispersion of the information:* both affect the quality of the representation and the performance of most machine learning algorithms; specially when there are large vocabularies, but few and imbalanced training data.

In this paper, we introduce a new method to represent and classify documents, in order to overcome these drawbacks for AA. Regarding the first one, we propose using the lexical richness of documents and relationships among terms, documents and authors to improve the representativeness. In this way, we are interested in relationships between authors and their terms, in order to define how a document is related to its author. In this context, let us call them second-order attributes, because they are calculated from the attributes that are extracted for BOT. These second-order attributes are few, but they are rich in representativeness; which faces the second problem.

In summary, we propose a new Document Author Representation (DAR) for AA. The main idea behind DAR is to build document vectors in a space of authors; therefore the dimensionality will be limited by the number of authors. Moreover, we propose the use of the vocabulary richness in documents; following the idea authors tend to write their documents with similar term repetition rates.

The rest of this paper is organized as follows: Section 2 shows related work, Section 3 introduces the DAR representation, Section 4 explains how we performed the experiments, Section 5 reports the results and Section 6 shows our conclusions.

2 Related Work

One way to address the AA task is to consider it as a standard classification problem. In this way, it can be stated as a single-labeled multiclass classification problem, where authors represent the classes to discriminate. In this context, several standard methods have been used to face the identification of authors. For example, the Bag of Terms (BOT) using standard Support Vector Machines (SVM) has been widely used for AA [15]. BOT like representations build vectors using textual features; for example, taking each word in the corpus vocabulary as an attribute. In this way, BOT represents documents with feature vectors, and assigns a value to each feature [14]. This value could be from simple Boolean values (1 or 0) to complex frequencies computed from the analysis of the corpus. BOT representations have been used to identify authors of emails, spam

filtering and plagiarism detection [2]. However, one of the problems of BOT representations is that it does not maintain any order or relation among their terms or classes; which could give valuable information and improve the representativeness. Another problem with BOT like representations occurs in realistic scenarios where there are large vocabularies, but few training data and imbalanced classes for a set of authors [9]. The latter causes that BOT like representations tend to favor majority classes, when in fact each document can actually belong to any of the authors (e.g., in computer forensics where it is required to discriminate among a predefined set of suspected perpetrators) [9]. Moreover, BOT representations require huge computational resources to classify large sets of documents, which could be impractical in some situations (e.g., AA in webforums, where we can have hundreds of texts of some authors) [11].

In order to address the main drawbacks of BOT, other kinds of representations have been used in Authorship Attribution. For instance, S. Plankias and E. Stamatatos proposed the use of Second Order Tensors [3] for representing stylistic properties of texts. The main idea behind this representation, is to use tensors to place relevant features in the same neighborhood. The latter is accomplished because the tensor-based model takes into account associations between relevant features. In order to define feature relevance, they use the frequency of occurrence. In this way, each feature is associated with features in the same row and column. To handle tensors instead of vectors they used a generalization of SVM called Support Tensor Machines (STM) [13], and they evaluate the accuracy using 2500 n-grams as terms. This representation using tensors takes into account relationships between terms. However, it does not guarantee to solve the problem of dispersion of information and high dimensionality, which could affect the quality of the representation.

Considering the issues of the latter approaches, we focus in a representation with a low dimensional space, but high level of representativeness. Thus, our interest is a method to perform a simple but effective semantic analysis for the AA task. In our proposal we follow some ideas from Concise Semantic Analysis (CSA) [1] in order to achieve relationships among terms, documents and authors. Furthermore, we have considered stylistic factors such as vocabulary richness in documents, and we have introduced different functions to weigh terms and documents in order to simplify the semantic analysis and help the AA task. The CSA representation is a language independent technique designed for Text Categorization (TC), but it has not been used for the AA task. It extracts concepts from category labels and then it implements a concise interpretation of words and documents with very low computational cost. There are other techniques that could build semantic relationships and low dimensionality vectors. For example, latent semantic analysis [5] and explicit semantic analysis [6], which interpret elements of texts and their relation with a predefined set of concepts. Those techniques overcome the dimensionality problem, because the dimensionality is limited by the number of semantic elements (concepts). However, the problem with these techniques is that we usually have to interpret terms in a complex space of concepts [1], which results in a very high computational cost; moreover, those techniques are specially designed for (TC) and Information Retrieval (IR) [1].

3 Document Author Representation

We present an approach to Document Author Representation (DAR) following some ideas from CSA, but transporting them to the context of AA. We also weigh terms considering vocabulary richness and simple frequencies, allowing us to perform a simple semantic analysis for AA. DAR stores textual features of documents in a vector, where the problem of dimensionality is limited by the number of authors to classify. DAR is built in two steps: first we build term vectors in a space of authors, and second we build document vectors in a space of authors. The following sections explain these steps in detail.

3.1 Term Representation

The representation of terms is the first step towards the DAR. For this stage, it is necessary to construct a vector representation for each term. Terms are any textual unit used as document feature, for example, words, n-grams, phrases, etc. In order to clearly explain this section, terms are words.

The main idea behind this first step is to capture the relation that each term maintains with each author. In other words, we compute a value that shows how a term t_j is used by each author a_i. Let $\{t_1, \ldots, t_m\}$ denote the vocabulary in the collection. Let $\{a_1, \ldots, a_n\}$ be the set of authors to be discriminated. For each term t_j in the vocabulary, we build a term vector $\mathbf{t_j} = \langle ta_{1j}, \ldots, ta_{nj} \rangle$, where ta_{ij} is a real value representing the relationship of the term t_j with the author a_i. For computing ta_{ij} we mainly take into account those documents that belong to the author a_i. However, documents of other authors are not completely ignored, because increasing the value for an author, de-emphasizes the value for other authors. The relationship of a term with an author considers the term frequency just in the documents of this author. Thus, high frequencies will show more preference for the term. Equation 1 follows the above idea and computes a relative weight as:

$$w_{ij} = \sum_{k:d_k \in A_i} \log_2 \left(1 + \frac{tf_{kj}}{len(d)}\right) \tag{1}$$

where A_i is the set of documents that belong to author a_i, tf_{kj} is the number of occurrences of the term t_j in the document d_k, and $len(d_k)$ is the length of the document d_k. Note that, the aim of the logarithmic function in equation 1 is to soften high frequency of terms.

As it can be seen, because of the sum of high frequencies, the weights could vary too much among terms. Therefore, it is convenient to apply a simple normalization for computing ta_{ij}. Note that, this normalization takes into account the weights computed for other authors, causing each weight being relative to all authors.

$$ta_{ij} = \frac{w_{ij}}{\sum_{i=1}^{n} w_{ij}} \tag{2}$$

3.2 Document Representation

In the previous step we calculated term vectors that represent relationships between terms and authors. The main idea in this second step is to build relationships between documents and authors; this is, our second-order attributes. We compute these from term vectors of the terms contained in the documents. For this, we get the terms of each document and add their term vectors. In this way, we will have documents represented as $\mathbf{d_k} = \langle da_{1k}, \ldots, da_{nk} \rangle$, where n is the total number of authors, and da_{ik} is a real value representing the relationship of the document d_k with the author a_i. Additionally, each term vector, before being added, is weighted by the frequency of the term t_j in the document d_k, normalized by the length of d_k. Finally, we multiply by the lexical richness of document d_k; in order to take into account the relative repetition rate of the context (see explanation about equation 4). Equation 3 shows the above ideas.

$$\boldsymbol{d}_k = richness(d_k) \sum_{t_j \epsilon D_k} \frac{tf_{kj}}{len(d_k)} \times \boldsymbol{t}_j \tag{3}$$

where D_k is the set of terms that belongs to document d_k. Furthermore we define:

$$richness(d_k) = \frac{1}{repetitiveness(d_k)} \tag{4}$$

Equation 4 attempts to capture more information about the lexical richness; following the idea that authors tend to maintain similar rates of repetition of their terms through their documents. Moreover, lexical richness let us to address the following situation; rich lexical documents usually have many terms with low frequencies (this is relative to the length of the document). We believe these low frequencies could not capture the real importance of author terms. The latter is because we speculate that an author with rich documents, pays more attention to select their terms; this means, there are important terms with relatively low repetition rates throughout the text. Thus, for the relevance of terms, we consider the lexical richness of the document that contains them. In this way, if the context is rich, then the terms were carefully selected and therefore their relevance will be higher.

In order to calculate the repetitiveness of a document we need a measure independent of the text length. For this reason, we have used the Yule's characteristic K, computed as [8] suggested. Equation 5 shows how the Yule's characteristic K is computed for each document:

$$repetitiveness(d_k) = 10^4 \left(\sum_{i=1}^{N} \frac{i^2 V(i,N)}{N^2} \right) - \frac{1}{N} \tag{5}$$

where N is the document length and $V(i,N)$ is the number of words occurring i times in the document.

3.3 DAR's time complexity analysis

The construction of term vectors in Section 3.1 is a summation. Therefore, its complexity is $O(dt)$, where d is the number of documents and t is the maximum number of different terms in a document. For the representation of a document, each term vector is added. Since each term is represented by a authors, the complexity of representing a document is $O(at)$. In this way $O(dta)$ is the complexity of representing all the documents in a data set.

4 Experimental methodology

For evaluating the proposed representation, we used a subset of 10 authors of the c50 corpus. This corpus subset was originally used by S. Plakias and E. Stamatatos [3]. The c50 corpus consist of texts from the Reuters Corpus Volume 1 (RCV1) [16]. The c50 corpus has 50 authors with documents that belongs to the CCAT category (about corporate and industrial news). The same category is used in order to reduce the topic factor and in this way focus the evaluation in AA. Furthermore, each author has 50 documents to train and 50 documents to test. The experiments we have conducted using this corpus are similar to those reported in [3].

First of all, with the purpose of simulating realistic scenarios [2], we have built different training sets. Three of them are balanced, with 50, 20 and 10 training documents per author, and the other three are imbalanced with 2:10, 5:10 and 10:20 (where $a : b$ means, minimum a and maximum b documents per author).

We have performed three different experiments. In the first and second ones we evaluated the classification accuracy using two of the most effective terms in AA; words and n-grams at the character level [2]. In this way, we compare DAR against BOT, which is a conventional approach used in AA. Moreover, we also performed experiments to compare DAR against Tensor Space Models [3], which has been evaluated using the c50 corpus. In the third experiment, DAR is built based on a simple selection of attributes in order to get better results. Summarizing, we compare DAR against the following methods:

- Bag of Terms (using words and character n-grams) classifying with SVM and 1NN. We used SVM because it has shown to be effective for AA [14], and we used 1NN because it lets us to show how it improves its performance when DAR is used.
- Tensor Space Models (using character n-grams), classifying with Support Tensor Machines (STM) [3].

DAR representation benefits lazy algorithms of Machine Learning [1], because it produces very dense vectors with very low dimensionality; thus, the classification is performed in a very fast way, finding the most likely vector and performing the prediction. For this reason, we have chosen the 1 Nearest Neighbor algorithm using the Euclidean distance function to classify documents using DAR. In this context, we also compared our results using SVM, because SVM and BOT representation are conventional approaches for AA [2]. To build document representations we use a word approach with stemming and 3-grams at the character level.

5 Experiments and Results

We have chosen the most relevant experiments that show interesting properties of DAR. DAR representation was constructed as described in Section 3. Furthermore, each experiment is the average of ten runs of DAR, in order to have enough data to perform statistical tests. In this way, we have applied the Wilcoxon signed-rank test to each result, getting a 95% statistical confidence in our results. We denote in bold the best outcomes.

5.1 Experiment 1. DAR using words

Table 1 shows the results of the first experiment using 2500 words with stemming. Note that we are maintaining stopwords, in order to capture stylistic information about how authors use them. It can be seen how DAR outperforms the BOT representation when the data is imbalanced (a realistic scenario). We believe this is because relations captured in DAR are representing documents from a different perspective beyond the independent words.

Model	Instances per author					
	Balanced			Imbalanced		
	50	10	5	10:20	5:10	2:10
BOT - SVM	**79.6**	**71.6**	**65.8**	55.2	56.4	42.8
DAR - SVM	70.0	62.3	57.7	**61.2**	**59.6**	**46.1**
BOT - 1NN	37.0	49.6	36.6	30.8	39.4	34.2
DAR - 1NN	**70.8**	**65.5**	**61.1**	**66.2**	**62.0**	**53.3**

Table 1. Classification accuracy in the c50 corpus. We compared DAR against SVM using the 2500 most frequent words.

5.2 Experiment 2. DAR using character 3-grams

Table 2 shows the results of the second experiment, in this experiment we compared BOT and DAR using the 2500 most frequent character 3-grams. Moreover, we are comparing DAR against STM using the same methodology as the authors of [3] follow in their experiments; therefore, we can say that the results are directly comparable. Results in Table 2 shows how DAR outperforms BOT in most imbalanced datasets; we believe this is because character 3-grams are features with more stylistic information. This experiment also allows us to see that in most of the datasets DAR (specially DAR-1NN) is better than BOT-SVM and TSM-STM when running under the same conditions.

Experiments in Table 1 and 2 show how DAR helps the 1NN over the SVM algorithm. We think this is because DAR produces very small dense vectors, which are easily compared, by 1NN, using the Euclidean distance function. However, it is important to highlight that this is not the best setting of DAR, since, as shown in Figure 1, DAR can be improved by simply using a frequency threshold.

Model	Instances per author					
	Balanced			Imbalanced		
	50	10	5	10:20	5:10	2:10
BOT - SVM	**80.8**	**64.4**	48.8	**64.2**	62.4	51.0
DAR - SVM	72.1	63.1	**56.6**	62.1	**63.9**	**53.2**
BOT - 1NN	36.4	50.3	38.6	33.8	41.4	36.2
DAR - 1NN	**76.0**	**67.3**	**62.7**	**66.9**	**65.6**	**55.1**
TSM - STM	*78.0*	*67.8*	*53.4*	*63.0*	*62.6*	*50.0*

Table 2. Classification accuracy in the C50 corpus. We compared DAR against SVM and STM using the 2500 most frequent character 3-grams.

5.3 Experiment 3. DAR using a frequency threshold

Figure 1 shows the results of the third experiment and an interesting property of DAR. Here, we can see how DAR could be improved by carefully selecting the selected attributes. In these experiments we selected only those attributes having frequencies equal or greater than 3, 5, 7 and 10, in the whole training data of each experiment. The main idea was to select attributes with higher information in the training data set, in order to improve the quality of the representation. Figure 1 shows how DAR can be significantly improved by this simple selection. In general, the best setting for DAR was a frequency threshold of 5.

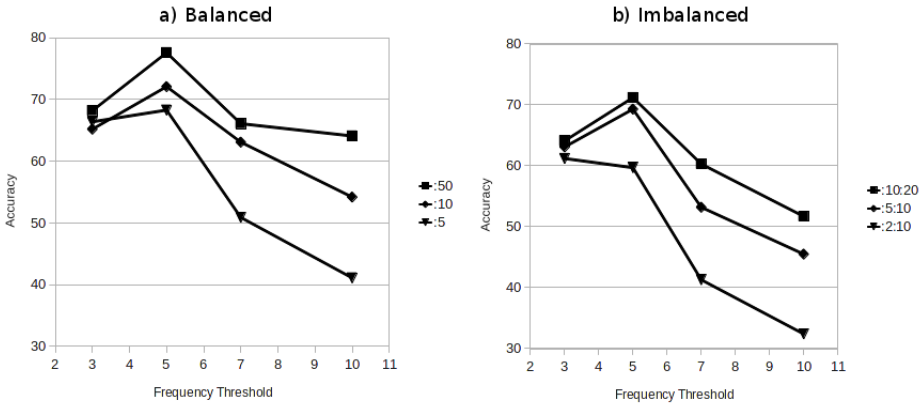

Fig. 1. Classification accuracy using words with different frequency thresholds in the balanced (fig. a) and Imbalanced (fig. b) training datasets. Each line represents an experiment with a specific setting.

5.4 Discussion of the Results

The latter experiments show how DAR outperforms the conventional approach using words and character 3-grams. Furthermore, in contrast to a fixed number of terms for experiments, we have showed how a frequency threshold can significantly improve the performance of DAR.

These results demonstrate the performance of our proposal. Note that, especially when the corpus is imbalanced or with little training data, DAR outperforms all other classifiers. We also showed how DAR provides better performance than conventional approaches and the STM method, which are reported in the state of the art. Furthermore, analyzing Tables 1 and 2 we can realize how other methods reduce their accuracy rates when data are scarce or when classes are imbalanced; on the other hand, DAR seems to be less sensitive to small and imbalanced training sets.

6 Conclusions

We have explored a new alternative to represent documents for AA. To the best of our knowledge, this is the first time that a CSA technique has been explored for the AA task. We found that representations such as DAR, can store relevant information for AA keeping good classification rates, even when the corpus is imbalanced, which is a realistic scenario. We think that this is due to the relations among terms, authors and the context vocabulary richness, which can preserve the writing style in the final representation. We also report experimental results against conventional approaches in imbalanced data, and better results compared against the STM method. In this way, we have showed the high quality of DAR representativeness. We further believe that DAR is a feasible and stable representation, which can be used in AA to discover a new set of attributes (let us call them second-order attributes) that represent the relations between documents and authors.

As future work, we are interested in exploring the use of this second-order attributes in other conditions; for example, we will evaluate DAR increasing the number of authors or using different document lengths (e.g., with short documents), or in different domains (e.g., AA in blogs, webforums, etc.). The main idea is to use DAR as a complement to other document representations. For example, we could build variations of the state of the art representations, in order to extend them with the DAR attributes. In particular, we are interested in second-order attributes, in order to describe how a document is related to authors in a specific feature space. In this way, we could use different feature spaces (or different views) to perform AA. For example, DAR could be used within ensembles of classifiers, which analyze different text features to develop voting methods.

In conclusion, we have studied a successful representation for AA that has the potential to be used in different ways, specially because it produces attributes with high level of representativeness in low dimensional dense vectors with low computing cost.

Acknowledgments. This work was done under partial support of CONACyT-Mexico (project grants 106013, 134186 and 106443, and scholarship 243957).

References

1. Zhixing, L., Zhongyang, X., Yufang, Z., Chunyong, L., Kuan., L.: Fast text categorization using concise semantic analysis. Pattern Recognition Letters. 32(3), 441–448 (2010)
2. Stamatatos, E.: A survey on modern authorship attribution methods. Journal of the American Society for Information Science and Technology. 60(3), 538–556 (2009)
3. Plakias, S., Stamatatos., E.: Tensor space models for authorship attribution. In Proc. of the 5th Hellenic Conference on Artificial Intelligence (SETN'08). LNCS, vol. 5138, pp. 239–249. Syros, Greece (2008)
4. Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C. E., Howald, B.S.: Identifying authorship by byte-level n-grams: the source code author profile (SCAP). Int. Journal of Digital Evidence. 6(1) (2007)
5. Deerwester, S.: Indexing by latent semantic analysis. Journal of the American Society for Information Science. 41 (6), 391–407 (1990)
6. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. Journal of Artificial Intelligence Research. 34, 443–498 (2009)
7. Schler, J., Koppel, M., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for Information Science. 60(1), 9–26 (2009)
8. Miranda-García, A., Calle-Martín, J.: Yule's k characteristic K revisited. Language Resources and Evaluation. 39(4), 287–294 (2005)
9. Stamatatos, E.: Author identification: Using text sampling to handle the class imbalance problem. Information Processing and Management. 44(2), 790–799 (2008)
10. Argamon, S., Juola, P.: Overview of the international authorship identification competition at PAN-2011. Notebook for PAN at CLEF 2011.
11. Solorio, T., Pillay, S., Raghavan, S., Montes-y-Gómez, M.: Modality specific meta features for authorship attribution in web forum posts. In Proceedings of the 5th International Joint Conference on Natural Language Processing, 156–164 (2011)
12. Abbasi, A., Chen, H.: Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems. 26 (2), Article 7 (2008)
13. Cai, D., He, X., Wen, J.R., Han, J., Ma, W.Y.: Support tensor machines for text categorization. Technical report, UIUCDCS-R-2006-2714, University of Illinois at Urbana-Champaign (2006)
14. Pavelec, D., Justino, E., Batista, L. V., Oliveira, L. S.: Author identification using writer-dependent and writer-independent strategies. In Proceedings of the 2008 ACM Symposium on Applied Computing - SAC08. 414–418 (2008)
15. Houvardas, J., Stamatatos, E.: N-gram feature selection for author identification. In Proceedings of the 12th International Conference on Artificial Intelligence. LNCS, vol. 4183, pp. 77–86 (2006)
16. Lewis, D., Yang, Y., Rose, T., Li. F.: RCV1: a new benchmark collection for text categorization research. Journal of Machine Learning Research. 5, 361–397 (2004)