Application of the Distributed Document Representation in the Authorship Attribution Task for Small Corpora

Juan-Pablo Posadas-Durán · Helena Gómez-Adorno · Grigori Sidorov · Ildar Batyrshin · David Pinto · Liliana Chanona-Hernández

Abstract Distributed word representation in a vector space (word embeddings) is a novel technique that allows to represent words in terms of the elements in the neighborhood. Distributed representations can be extended to larger language structures like phrases, sentences, paragraphs, and documents. The capability to encode semantic information of texts and the ability to handle high dimensional data sets are the reasons why this representation is widely used in various natural language processing tasks such as text summarization, sentiment analysis, syntactic parsing, etc. In this paper, we propose to use the distributed representation at the document level to solve the task of the authorship attribution. The proposed method learns distributed vector representations at the document level and then uses the SVM classifier to perform the automatic authorship attribution. We also propose to use the word n-grams (instead of the words) as the input data type for learning the distributed representation model. We conducted experiments over six datasets used in the state of the art works and for the majority of the datasets we obtained comparable or better results. Our best results were obtained using the combination of words and n-grams of words as the input data types. Training data is relatively scarce, which did not affect the distributed representation.

Keywords Distributed Representation · Authorship Attribution · Author Identification · Embeddings · Word Embeddings · Stylometry · Machine Learning · SVM · Scarce Training Data

J. Posadas-Durán · L. Chanona-Hernández
Instituto Politécnico Nacional (IPN), Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco (ESIME-Zacatenco), Mexico City, Mexico
E-mail: jpposadas@gmail.com

H. Gómez-Adorno · G. Sidorov · I. Batyrshin
Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico City, Mexico
E-mail: helena.adorno@gmail.com, sidorov@cic.ipn.mx

D. Pinto
Benemérita Universidad Autónoma de Puebla (BUAP), Faculty of Computer Science, Puebla, Mexico

1 Introduction

Distributed word representation in a vector space, also known as word embeddings (Turian et al, 2010; Pennington et al, 2014), is a novel paradigm that is currently widely used in many Natural Language Processing (NLP) tasks. It aims to represent words in terms of fixed-length, continuous and dense feature vectors. A very popular model architecture for learning distributed word vector representations (Word2Vec) using a neural network was proposed in (Mikolov et al, 2013a,b). This technique captures semantic and syntactic word relations: similar words are close to each other in the vector space. For example, it was shown in (Mikolov et al, 2013c) that $vector[King] - vector[Man] + vector[Woman]$ results in the vector that is closest to the representation of the $vector[Queen]$. Another two well-known continuous representations of words are Latent Semantic Analysis (LSA) (Wiemer-Hastings et al, 2004) and Latent Dirichlet Allocation (LDA) (Trejo et al, 2015). Unlike LSA and LDA, the vector representations obtained by (Mikolov et al, 2013a,b) preserve linear regularities between words.

The success of this technique is explained by: (1) its capacity to encode semantic information of words in vectors, which makes it suitable for sentiment analysis (Socher et al, 2013b), syntactic parsing (Socher et al, 2013a), text summarization (Miranda et al, 2014) and

many other tasks in the field of Natural Language Processing; and (2) its ability to handle high dimensional data sets (from thousands to millions of instances). Distributed representations can be extended to model not only words, but also larger language structures like phrases, sentences and documents (Le and Mikolov, 2014).

In this paper, we explore the use of distributed document representation for the authorship attribution (AA) task. We propose to use the Doc2vec method (Le and Mikolov, 2014), which builds a distributed vector representation at the document level (distributed document representation) using an unsupervised approach. Further, traditional machine learning methods are used to perform the authorship attribution task. We evaluate the performance of this method on six different corpora usually used in the state of the art. In addition, we propose to use word n-grams (alone or in combination with bag of words) as features to build the distributed document representation instead of using only words, so the model can learn syntactic and grammatical patterns of an author.

In general, it is considered that for obtaining accurate distributed representations, it is needed a very large corpora for training (millions of instances), say, the whole Web was used for creation of the Word2Vec resource (Mikolov et al, 2013a,b). We experimented with relatively small corpora (hundreds of instances), but this did not affect negatively our method as compared with other methods.

The use of distributed documents representation has been rarely applied to the authorship attribution task and the previous results using this technique are discouraging Rhodes (2015). It is worth mentioning that in general this technique is seldom used in Natural Language Processing tasks with small corpora (less than a thousand of examples).

Thus, the aims of this paper are: (1) To evaluate the performance of the Doc2vec method over significantly small corpora (from 1 to 10 documents per author) in a complex classification task (authorship attribution), (2) To compare our results based on distributed representation with the results of previous works for the authorship attribution task, and (3) To analyze the effect of the use of features other than words as input data types for the Doc2vec method (such as word n-grams).

The rest of the paper is structured as follows. Section 2 presents a brief description of the authorship attribution task and the most successful methods for solving this problem. Section 3 describes the proposed method for the authorship attribution using the distributed representation of documents using words and word n-grams as input data types. The description of the data sets and the experimental settings are described in Section 4. The results obtained and the corresponding discussion is in Section 5. Finally, conclusions are presented in Section 6.

2 Related Work

The authorship attribution task, also known as the author identification task, consists in identifying the author of a given text. It can be viewed as a classification problem, when given a set of documents belonging to various authors and a set of documents with unknown authors, it is necessary to determine the corresponding authors of the documents with unknown authorship, in other words, to choose the class (the author) that corresponds to each unknown document. The authorship attribution task is an active research field with many relevant applications as plagiarism detection (Stamatatos, 2011; Sanchez-Perez et al, 2014), authorship identification (in suicide notes, ransom notes, threatening emails, to name some) (Chaski, 2005; Brocardo et al, 2013), and authorship profiling (Argamon et al, 2009).

In order to accomplish the authorship attribution task, it is necessary to identify the features or profiles corresponding to the target authors. The characterization of the writing style of an author remains an open problem, which is studied by the research discipline called stylometry. The features that capture the style of a writer are known as style markers and it is assumed that they are robust enough to model the style under different circumstances, i.e., the style is preserved in different genres and through time (Alzahrani et al, 2012).

The style markers can be classified depending on the information they use, namely, they can correspond to the following levels: character, lexical, syntactic, semantic, and formatting. The style markers at character and lexical levels do not need complex processing to be obtained and they show good results. Other style markers require complex processing and their results are usually considered to be lower (Stamatatos, 2009), but in a recent work (Sidorov et al, 2014) the use of syntactic n-grams that exploits the information of the dependency trees had obtained higher results than Part-of-Speech tags n-grams.

One of the former model, the typical bag-of-words model (Stamatatos, 2009), consists in representing a text as a set of words with their respective frequencies, i.e, words are features in the corresponding vector space model. In this model it is also assumed that their occurrence is independent of each other. For the authorship attribution problem, the most frequent words (function words or stop words) proved to have high accuracy. On

the other hand, the features based on word n-grams were proposed in order to take advantage of the contextual information and the results they obtained are slightly better than the bag-of-words model.

The most accurate style marker for the authorship attribution task are character n-grams (Stamatatos et al, 2001; Kešelj et al, 2003; Sidorov et al, 2014). These style markers show high accuracy for several benchmark corpora (Stamatatos, 2013; Plakias and Stamatatos, 2008; Escalante et al, 2011). However, their disadvantage is that they generate very sparse vectors of high dimensionality. Besides, they provide the text representations without any clear semantic interpretation.

In (Stamatatos, 2009), it is mentioned that the use of semantic features for the authorship attribution task usually improves the obtained results, however, very few attempts have been done to exploit high-level features for stylometric purposes. In this paper, we consider the usage of the distributed document representation for the authorship attribution task, which corresponds precisely to semantic features.

In the work (Segarra et al, 2013) the authors propose a Function Word Adjacency Networks (WANs), where the nodes are function words and the directed edges stand for the likelihood of finding a target function word in the ordered proximity of a source function word. In that work the authors report that the accuracy achieved by the WANs is higher than the one obtained by traditional methodologies that rely only on function word frequencies. Even though the WANs achieved high accuracy in very long texts as long as a play act or a novel, they only obtain reasonable rates for short texts such as newspaper opinion pieces if the number of candidate authors is small.

Recently, the use of distributed representation has shown great power in capturing the semantics of words, phrases and sentences, which benefits Natural Language Processing applications. In (Le and Mikolov, 2014), the authors present Doc2vec, an unsupervised algorithm that learns feature representations of fixed length from documents of variable length. The idea is to combine the meaning of words for construction of the meaning of documents using distributed memory model. The distributed representation obtained by the Doc2vec method outperforms both, bag-of-words and word n-grams models producing the new state of the art results for several text classification and sentiment analysis tasks.

In another related work (Li and Shindo, 2015), the authors present a supervised method called Compound RNN. It uses recursive and recurrent neural networks in order to learn the document distributed representation. This method is task-specific because it does not learn general representation of sentences as in the case of Doc2vec. Nevertheless, this method outperforms existing baselines in tasks such as binary classification, multi-class classification and regression.

Another interesting work using the Doc2vec and the skip-gram model for the task of discriminating similar languages is presented in (Franco-Salvador et al, 2015). The authors use the continuous skip-gram model (Word2vec) (Mikolov et al, 2013b) to generate distributed representations of words (i.e., n-dimensional vectors) and estimate the average of their dimensions in order to generate the distributed document representation. They also evaluate their classification model using the Doc2vec method for learning the document representation. For this task, the combination of word vectors (Word2vec) obtained better results in average as compared to the use of vectors generated directly from sentences (Doc2vec). Nevertheless, this conclusion is valid for the language variation identification task, but for the authorship attribution this approach is not applicable as we will show in the next sections.

There are few attempts of using distributed representations for the authorship attribution task. In (Kiros et al, 2014), the authors propose a framework for learning distributed representations of attributes. The attributes correspond to a wide variety of concepts, such as document indicators (to learn sentence vectors), language indicators (to learn distributed language representations), metadata and additional information about authors (to learn author profile, such as the age, gender and occupation). The framework is evaluated over several tasks: sentiment classification, cross-lingual document classification, and blog authorship attribution. For the authorship attribution task the methodology is evaluated over a corpus of blogs and the attributes are based only on the author metadata and not on the texts themselves. On the contrary, for our work we use only the textual information in order to automatically extract significant vector representations of documents.

A novel neural network architecture, namely convolutional neural networks (CNN), over word embeddings was presented in (Rhodes, 2015). It was evaluated over two datasets, a baseline developed by the author and the PAN 2012 dataset (Juola, 2012). This work is directly comparable to our work. For the representation of documents the author used the set of Google News word vectors trained via the skip-gram method and negative sampling presented in (Mikolov et al, 2013a,b). The author used the standard approach for convolutional models (simple concatenation operation) to encode sequences rather than words, instead of averaging the dimensions as it was done in (Franco-Salvador et al, 2015). The classification is performed via logis-

tic regression and the results show high accuracy over the baseline dataset, but the CNN architecture did not outperform the best method presented at PAN 2012 (Juola, 2012), while our method (presented below) obtains better results for this dataset.

In the next section we describe the method proposed to solve the authoship attribution task using a distributed representation at document level.

3 Authorship Attribution Using Distributed Representation of Documents

3.1 Distributed Representation of Documents

In order to learn the distributed representation of documents, i.e., the Document Vectors, we use the Doc2vec method inspired by previous research for learning word embeddings (Le and Mikolov, 2014).

First, we introduce the concept of distributed vector representations of words. For this purpose, we use the method proposed in (Mikolov et al, 2013a,b). The aim is to predict a word given the context surrounding the word. In this method, every word is mapped to a unique vector represented by a column in a matrix W. Formally, given a sequence of training words w_1, w_2, w_3,...,w_T, the training objective is to maximize the average of the log probability:

$$\frac{1}{T}\sum_{t=k}^{T-k} log\ p(w_t|w_{t-k},...,w_{t+k}).$$

The prediction task is performed by a multiclass classifier, such as Softmax:

$$p(w_t|w_{t-k},...,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}\ ,$$

where y_i is the $i-th$ element of the vector of class score y. The above formula can be interpreted as the (normalized) probability assigned to the correct label w_t given the training words $w_{t-k},...,w_{t+k}$.

Each of y_i is unnormalized log-probability for each output word i computed as:

$$y\ =\ b + Uh(w_{t-k},...,w_{t+k};W), \tag{1}$$

where U, b are the Softmax parameters. h is constructed by a concatenation or average of word vectors extracted from W.

For obtaining the log-probability for each word, a computationally efficient algorithm for the prediction task is used: the hierarchical Softmax (Mnih and Hinton, 2009; Mikolov et al, 2013b). The structure of the hierarchical Softmax is a binary Huffman tree (Bird and Wadler, 1988), where short codes are assigned to the frequent words. After the training converges, the words with similar meaning are mapped into a similar position in the vector space (Mikolov et al, 2013c).

In order to learn paragraph vectors the same method is followed. Paragraph vectors are asked to contribute to the prediction task of the next word given many contexts sampled from the document in the same manner in which the words vector are asked to contribute to a prediction task about the next word in a sentence. The word or paragraph vectors are initialized randomly, but in the end they capture semantics as an indirect result of the prediction task. Finally, vectors for documents are obtained in similar way (Mikolov et al, 2013c).

In the Document Vector model, every document is mapped to a unique vector represented by a column in matrix D in the same manner: every word is mapped into a unique vector represented by a column in matrix W. In the Document Vector model the equation 1 is changed in order to construct h from W and D.

There are two models for distributed representation of documents: Distributed Memory (DM) and Distributed Bag of Words (DBOW). In case of DM, any paragraph is a vector of feature values, which are used to predict the vectors of the context paragraphs (the previous or the following paragraphs). DBOW ignores the context words (or n-grams in our case) in the input, but predict words randomly sampled from the paragraphs in the output. So, in each iteration of stochastic gradient descent it samples a text window and perform a classification task given the paragraph Vector.

3.2 Authorship Attribution as a Classification Problem

There are two types of the authorship attribution: closed, when the set of authors is predefined, and open, when a new author (absent in the training corpus) can be presented to the method. In the rest of the paper we will deal with the closed authorship attribution.

The task of closed authorship attribution can be viewed as a multiclass classification problem defined as follows: given a set of known authors $\mathbf{A} = \{A_1, A_2, \ldots, A_i\}$ and a set of texts examples (from the texts written by these authors) $\mathbf{T} = \{t_1^1, t_2^1, \ldots, t_n^1, \ldots, t_j^i\}$, where the element t_j^i corresponds to the j example of the author A_i, the problem is to build a classifier F that assigns each element of a set of texts of unknown authorship $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ to exactly one known author, i.e., $F : \mathbf{X} \rightarrow \mathbf{A}$.

Our approach for the authorship attribution task is based on machine learning, composed of two phases: the training phase and the testing phase. At the training phase we learn the distributed document representation of the documents in the set $\mathbf{T}$, i.e., $\mathbf{V}$ =

$\{v_1^1, v_2^1, \ldots, t_j^i\}$, where $v_j^i = \{f_1, f_2, \ldots, f_n\}$ is the vector representation of the text t_j^i. The vector representation is obtained using an implementation of the Doc2vec method (Le and Mikolov, 2014) in GENSIM [1]. The Doc2vec method builds a model for obtaining distributed representations of documents, it offers two possible approaches to build the model: the DM and the DBOW as mentioned before. Previous research on the task of sentiment analysis report better results when both representations are concatenated, so in our proposal the final document vector is composed of the concatenation of the representations obtained by the DM and the DBOW models.

A classifier is trained with the vector representations of the documents using two well-known method: SVM and Logistic Regression. We selected these classifiers because they are robust with sparse data and can optimally handle vectors with high dimensionality.

At the testing phase, we obtain the vector representations of texts of the unknown authorship. It is performed by concatenating the representations obtained with the Doc2vec models used at the training phase. Finally, the classifier assigns the author (class) to each document using these distributed representation (vectors) as features.

4 Experimental Settings

4.1 Description of the Datasets

There were great advances in the authorship attribution task over the last two decades and various datasets have been used to test the performance of the methods proposed in the state of the art. Earlier works centered on problems of the domain of humanities commonly related to attribution of the disputed documents (Mosteller and Wallace (1963); Matthews and Merriam (1993)) or revealing anonymous authors (Holmes, 1998). These datasets consist of historical documents or literary documents of large length (more than 1,000 words) and with a reduced number of candidate authors (not more than three authors).

In recent years, an interest in more controlled dataset has emerged. The datasets are controlled in three aspects: the genre of the documents, the topics contained in the documents and the length of the documents. Recent datasets (specifically compiled for the authorship attribution task) now include the phenomena of cross-genre and cross-topic documents. Also, these documents are gathered from newspapers or from social media like blogs or twitter. It implies a significant reduction in the length of documents and the inclusion of social media language (hashtags, emoticons, slang words). The controlled datasets represent more suitable scenarios for the development of authorship attribution methods in contrast with the earlier datasets and there were different attempts to build a unified benchmark to test and compare such methods (Juola, 2004; Argamon and Juola, 2011; Juola, 2012).

Six different datasets related to the closed authorship attribution problem were used to evaluate our proposal. The collected datasets focus on the English language and vary in the following aspects: the number of known authors, the size of training data, genres (novels, reviews, news) and topics.

The PAN/CLEF evaluation lab[2] is an important forum for advances in plagiarism, authorship and social software misuse, where challenging benchmarks for these topics are proposed and participants are asked to present novel methods for evaluating them. The PAN/CLEF 2012 was the latest edition which included the closed authorship attribution task as a part of its authorship section. There were three balanced benchmarks named Problem A, Problem C, and Problem I, conformed by fragments of novels written by English-speaking authors. We use these three benchmarks to evaluate of our proposal. In further sections we will refer to the benchmark of Problem A as PAN A, to the Problem C as PAN C and to the Problem I as PAN I.

Table 1 presents the structure of the benchmark of each PAN subproblem. For PAN A, three authors are known and the training data consists of two examples for each one, while the testing data consists of two examples of each author to be classified (six examples in total). The length of the examples is between 1,800 and 6,060 words. For Problem C, the number of known authors is increased to eight with two examples per author, while the testing data is reduced to one example of each author to be classified. The length of the examples is increase up to about 13,000 words. Finally, for Problem I, there are fourteen known authors with two examples for each one as the training data, while the testing data includes one example for each author, but the samples correspond to complete novels with the length that varies from 40,000 to 170,000 words (Juola, 2012).

The PAN/CLEF 2012 closed authorship attribution benchmarks were chosen to test our proposal because they allow us to explore three important issues discussed in the state of the art. First, it is known that with a bigger number of authors the task becomes more dif-

[1] `https://radimrehurek.com/gensim/`

[2] `http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/`

Table 1 PAN/CLEF 2012 benchmark for closed authorship attribution

	PAN A	PAN C	PAN I
Authors	3	8	14
Examples per author	2	2	2
Test samples per author	2	1	1
Size (in thousands)	1.8 to 6	at most 13	40 to 170

ficult, in this sense the benchmarks let us explore with three different sizes of known authors starting from a simple case with just three authors to a more challenging scenario with fourteen authors. Second, the benchmarks propose different lengths of text samples letting us to experiment over the required amount of data to create a reliable model for the author's style depending on the number of known authors. Third, the benchmark for Problem I offers a scenario with different topics since the text samples correspond to complete novels.

Other corpus used to test our proposal was The Reuters Corpus Volume 1 (RCV1) (Lewis et al, 2004). It consists of a collection of newswire stories written in English that cover four main topics: corporate/industrial (CCAT), economics (ECAT), government/social (GCAT) and markets (MCAT). Although it was not compiled for authorship attribution task, it has been adapted to this task in previous works. For example, in (Stamatatos, 2008; Plakias and Stamatatos, 2008) the 10 most prolific authors were chosen from the CCAT category and then 50 examples per author for training and 50 examples for testing were selected randomly with no overlapping between training and testing sets. In further sections we will reference to this corpus as RCV1-10.

In (Houvardas and Stamatatos, 2006), the authors proposed another adaptation of the RCV1 corpus for the authorship attribution task. They choose the 50 most prolific authors from the Reuters Corpus, keeping 50 examples per author for training and 50 examples per author for testing with no overlapping between them. We will refer to this corpus as RCV1-50.

The RCV1-10 and RCV1-50 datasets are both balanced over different authors and have their genre fixed to news. The main category of the news in both cases is fixed to corporate/industrial, but there are many subtopics covered in the news and the length of the texts is short (from 2 KBytes to 8 KBytes). These corpora resembles a more realistic scenario, when the amount of texts is limited and the number of candidate authors is large.

Another benchmark we used was presented in (Stamatatos, 2013). It consists of a collection of articles published in *The Guardian* newspaper from 1999 to 2009. The articles belong to 13 authors and are grouped into five categories (*Politics*, *Society*, *World*, *UK*, and *Book reviews*), discarding those articles whose content covers more than one category, i.e., each article only belongs to one category.

The articles were downloaded from the online repository and preprocessed to obtain their plain text versions. The number of examples of each author over different categories is not balanced. It corresponds to the production of each author through the period mentioned before. The complete distribution of the corpus is presented in the original article (Stamatatos, 2013). In our paper, following prior work (Sapkota et al, 2015), we choose at most ten documents per author for each of the five categories. The new distribution is shown in table 2.

Table 2 *The Guardian* corpus for closed authorship attribution, with at most 10 documents per author

Author	Politics	Society	World	UK	Reviews
CB	10	4	10	10	10
GM	6	3	10	3	0
HY	8	6	10	5	3
JF	9	1	10	10	2
MK	7	0	10	3	2
MR	8	10	10	10	4
NC	10	2	9	7	5
PP	10	1	10	10	10
PT	10	10	10	5	4
RH	10	4	3	10	10
SH	10	5	5	6	2
WH	10	6	10	5	7
ZW	4	10	10	6	4
Totals	112	62	117	90	63

The Guardian corpus offers the opportunity to explore a scenario with different topics under the same genre with the exception of the category "Books reviews", which is considered as another genre. It is assumed that each category represents a topic, which is different enough from the other categories. In contrast with the previously described benchmarks, *The Guardian* is a cross-topic, cross-genre and unbalance benchmark, representing in this way a very challenging scenario.

5 Experimental Results and Discussion

We conducted experiments over the different datasets described in the previous section, using a machine learn-

ing approach. The experiments consisted in training a classifier with the corresponding training set using the distributed document representation and then evaluating the classifier with the testing set using the same representation.

We report our results in terms of accuracy obtained for each dataset using word n-grams (n=1,2,3,4,5) as input data type for the Doc2vec method, in order to obtain the document distributed representation. The accuracy represents the percentage of instances which are correctly classified and it is defined in terms of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

We used the Doc2vec (Le and Mikolov, 2014) method available in the freely downloadable GENSIM module in order to implement our proposal. The implementation of the Doc2vec method requires the following three parameters: the number of features to be returned (length of the vector), the size of the window that captures the neighborhood and the minimum frequency of words to be considered into the model. The values of these parameters depend on the used corpus. At our best knowledge, there is no previous work in this specific application area at the time we performed the experiments, however in a previous work related to opinion classification task (Mikolov et al, 2013b), it was reported a representation of 300 features, a window size equal to 10 and minimum frequency of 5. Therefore based on this previous work we conduct a grid search over the following fixed ranges: number of features $[50, 350]$, size of window $[3, 19]$ and minimum frequency $[3, 4]$.

The Doc2vec module uses stochastic gradient descend and back-propagation algorithms for generating the model, however, these algorithm use random elements that do not guarantee their reproducibility. In order to ensure the reproducibility of our experiments, the values of the following parameters are fixed (following the recommendations of the user manual[3]): the value of threshold for configuring which higher-frequency words are randomly downsampled is set to 1e-3, negative sampling is set to 5, the seed of the random number generator is set to 1 and the number of threads is set to 1 (not multithreading is used). The rest of parameters are set with default values.

With the purpose of increasing the efficiency of the Doc2vec method, it is recommended to train the model several times over the unlabeled corpus but exchanging the order of entry of the documents. In this work we propose the use of nine different configurations. In order to ensure the reproducibility of the experiments, instead of using a random number generator we propose a set of nine rules to perform the changes to the order in which the documents are input into the Doc2vec method. Consider the list of all unlabeled documents of the corpus $\mathbf{T} = [d_1,\ d_2,\ \ldots,\ d_i]$, we generate new lists of documents with different arrangements as follows:

1. First rule: invert the order of the elements in the set, i.e., $\mathbf{T} = [d_i,\ d_{i-1},\ \ldots,\ d_1]$.
2. Second rule: select first the documents with an odd index in ascending order and then documents with even index, i.e., $\mathbf{T} = [d_1,\ d_3,\ \ldots,\ d_2,\ d_4,\ \ldots]$.
3. Third rule: select first the documents with an even index in ascending order and then documents with odd index, i.e., $\mathbf{T} = [d_2,\ d_4,\ \ldots,\ d_1,\ d_3,\ \ldots]$.
4. Fourth rule: for each document with an odd index i, exchange it with the document of index $i+1$, i.e., $\mathbf{T} = [d_2,\ d_1, d_4,\ d_3, \ldots]$.
5. Fifth rule: shift each element two elements to the left in a circular way, i.e., if m is the index of the last element of the corpus $\mathbf{T}$ then $\mathbf{T} = [d_{m-1},\ d_m,\ d_1,\ d_2, \ldots]$.
6. Sixth rule: for each document with index i exchange it with the document whose index is $i + 3$.
7. Seventh rule: for each document with an index i whose value is a multiple of three exchange it with the document next to it $(i + 1)$.
8. Eighth rule: for each document with an index i whose value is a multiple of four exchange it with the document next to it $(i + 1)$.
9. Nineth rule: for each document with an index i whose value is multiple of three exchange it with the document whose index is $i + 2$.

The experiments were carried out using two different classifiers: Support Vector Machines (SVM) and Linear Regression (LR). For most of the cases, Linear Regression obtained the best results, thus only the results obtained with this classifier are reported.

In order to build the distributed representation model, the Doc2vec method receives as input the plain text documents of the training set without any label (there is no specification of the class to which they belong) and the unlabeled text documents from the testing set.

For preprocessing, we only performed a standard tokenization process. Different representations of the texts were used as input data types for the Doc2vec method in order to evaluate the quality of different distributed representation outputs. In particular we represented the texts in terms of word unigrams, bigrams, trigrams, four-grams and five-grams.

[3] `https://radimrehurek.com/gensim/models/Doc2vec.html`

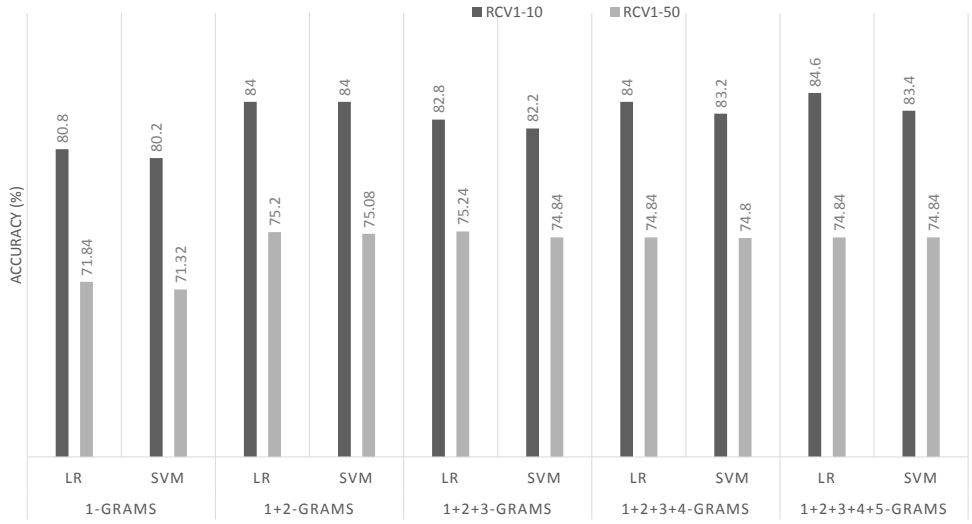

Figure 1 Accuracy obtained by the classifiers using the different input data types in the RCV1 corpora

All PAN/CLEF datasets for closed authorship attribution were divided into the training and testing set. In the experiments, the accuracy obtained by the classifiers were similar. Table 3 shows the results achieved only by the LR classifier. The first five rows of table 3 correspond to the different representations obtained with the Doc2vec method using n-grams as input data types, and the remaining rows correspond to the results obtained at the competition. In the table 3 appears the rank along with the name that the team used in the evaluation, note that a team could send more than one answer. There were 20 teams that submitted their approach to the competition. Our approach achieve results that are better comparable to the best participating systems. For more details please refer to the overview of the competition (Juola, 2012).

The experiments on the RCV1-10 and RCV1-50 were performed using the same proposed input data types for the Doc2vec method. The corpora were divided into training and testing set as described in previous section 4.1 and all the collected examples for each author were used (50 texts per author in training phase and 50 texts per author in testing phase). The accuracy obtained by the SVM and LR classifiers is showed in the Figure 1. The performance achieved by the LR classifer is comparable with the one obtained by the SVM classifier, but LR classifier obtained the higher accuracy. Table 4 presents in the first five rows the accuracy obtained with our proposed representation using the LR classifier, the rest of the rows present the accuracy obtained by previous works: local histograms of character n-grams (Escalante et al, 2011), tensor space models (Plakias and Stamatatos, 2008), character and word n-grams (Stamatatos, 2008)typed character n-grams (Sapkota et al, 2015) and n-gram feature selection (Houvardas and Stamatatos, 2006).

Table 3 Results for closed authorship attribution PAN datasets (A, C and I)

Name	A	C	I
D2V words	**100**	**100**	85.71
D2V words+2-grams	**100**	**100**	**100**
D2V words+2+3-grams	**100**	**100**	**100**
D2V words+2+3+4-grams	**100**	**100**	**100**
D2V words+2+3+4+5-grams	**100**	**100**	**100**
Brainsignals	**100**	**100**	**92.85**
Sapkota	**100**	**100**	**92.85**
Lip6 1	100	100	85.71
EVL Lab	100	87.50	85.71
de Graaff 1	100	87.50	71.42
Bar I U	100	75.00	85.71
Lip 6 2	100	75.00	**85.71**
Lip 6 3	100	62.50	78.57
CLLE-ERSS 3	100	37.50	92.85
CLLE-ERSS 4	100	37.50	92.85
CLLE-ERSS 1	100	35.00	85.71
Zech terms	83.33	62.50	64.28
Vilarino 2	83.33	62.5	57.14
Ruseti	66.66	75.00	85.71
CLLE-ERSS 2	66.66	25.00	85.71
Vilarino 1	66.66	25.00	71.42
Zech stats	66.66	25.00	35.71
Surrey	66.66	12.5	50.00
de Graaff 2	50.00	50.00	50
Zech stylo	33.33	25.00	35.71

For the corpus RCV1-10 the proposed method does not outperform the best accuracy reported by (Escalante et al, 2011) but it performs better than the rest of reported works. Experiments conducted in the RCV1-50 corpus outperform the accuracy reported in

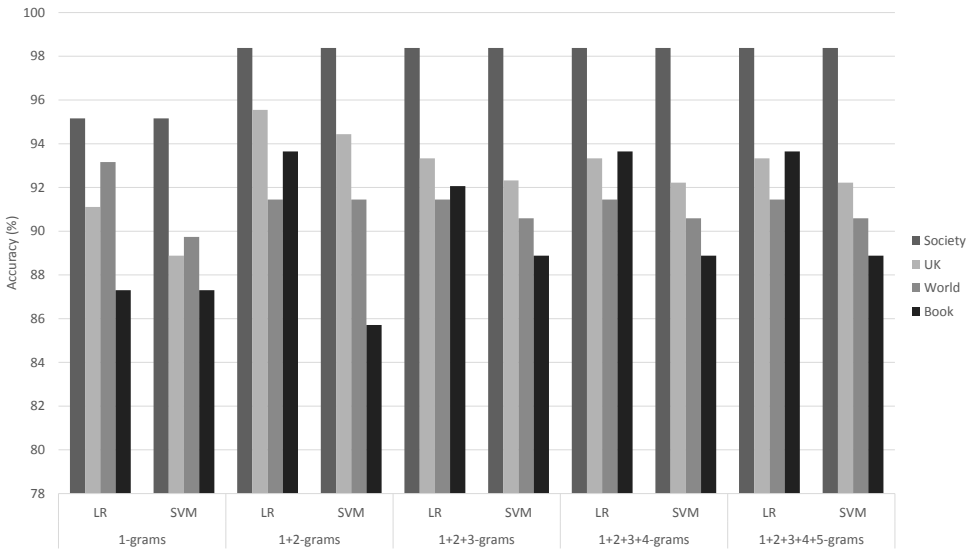

Figure 2 Accuracy obtained by the different input data types using *Politics* as training set

Table 4 Results for RCV1 corpora

Name	RCV1-10	RCV1-50
D2V words	80.80	71.84
D2V words+2-grams	84.00	75.20
D2V words+2+3-grams	82.80	**75.24**
D2V words+2+3+4-grams	84.00	74.84
D2V words+2+3+4+5-grams	84.60	74.84
Local histograms	**86.40**	–
Tensor space models	80.80	–
Character and word n-grams	79.40	–
Enhanced typed char n-grams	79.6	–
Typed character n-grams	78.80	–
N-gram feature selection	–	74.04

the state of the art. The proposed method showed to be efficient in corpora with a high number of authors. Although, our methodology obtained lower performance than the best approach in the literature in the RCV1-10 corpus it overcomes the results for all the other evaluated corpora in this work. In the work of (Potthast et al, 2016) the authors performed the reimplementation of 15 authorship attribution methodologies and concluded that very few of them achieve consistent results across different corpora. In this work, we show the consistency of our approach on 6 different dataset, evaluating a wide range of testing scenarios.

For *The Guardian* corpus we used the same test scheme established in previous work (Stamatatos, 2013):

- Training phase: from the five different categories in *The Guardian* corpus (*Politics*, *Society*, *World*, *UK*, and *Book reviews*), the *Politics* category is selected as the training set and at most ten text per author are used in the training phase of the classifier. Following this guideline, the total number of texts used in the training phase (n_{tr}) is 112.
- Testing phase: the trained classifier is tested over the rest of the different categories, having a total of four pairs (*Politics-Society*, *Politics-UK*, *Politics-World*, *Politics-Book reviews*) and also at most 10 texts per author are selected.

The accuracy obtained by the SVM and LR classifiers is shown in the Figure 2. The performance achieved by the LR classifier is comparable with the one obtained by the SVM classifier, but LR classifier obtained the higher accuracy. In the following tables related with the corpus *The Guardian* only the accuracy obtained by the LR classifier is reported. Table 5 shows the results obtained for each pair of categories along with the total number of texts used in each category and the results obtained in previous work (Char 3-grams) (Stamatatos, 2013).

Next we describe a different test scheme over *The Guardian* corpus proposed in (Sapkota et al, 2015). Consider the set of all authors $A = \{A_1, A_2, \ldots, A_i\}$ and the set of all categories $C = \{C_{Politics},\ C_{Society},\ C_{UK},\ C_{World}, C_{BookReviews}\}$ where each element represents all the texts in a category, i.e., $C_{label} = \left\{t_1^1, \ldots t_i^j\right\}$ with t_i^j is the i^{th} text of the author A_j in the category C_{label}. Like the previously explained scheme, at most 10 text per author are selected in each category. Given the set S that contains all the possible 2-tuples made by the elements of the set C:
$S = \{(C_{Politics},\ C_{Society}),\ (C_{Politics},\ C_{UK}),\ \ldots\}$
with cardinality equal to 20, for each $s_k \in S$ build

Table 5 Results for *The Guardian* corpus considering *Politics* as the training and the rest of the categories as the testing sets

Name	Society	UK	World	Books reviews	Average Acc.
D2V words	95.16	91.11	**93.16**	87.30	91.68
D2V words+2-grams	**98.38**	**95.55**	91.45	**93.65**	**94.75**
D2V words+2+3-grams	**98.38**	93.33	91.45	92.06	93.80
D2V words+2+3+4-grams	**98.38**	93.33	91.45	**93.65**	94.20
D2V words+2+3+4+5-grams	**98.38**	93.33	91.45	**93.65**	94.20
Char 3-grams (from Stamatatos (2013))	$\simeq$ 91.00	$\simeq$ 88.00	$\simeq$ 82.50	$\simeq$ 79.50	$\simeq$ 85.50

Table 6 Results for *The Guardian* corpus considering *Society* as the training and the rest of the categories as the testing sets

Name	Politics	UK	World	Books reviews	Average Acc.
D2V words	64.28	55.55	60.68	**57.14**	59.41
D2V words+2-grams	**65.17**	**61.11**	**63.24**	53.96	**60.87**
D2V words+2+3-grams	**65.17**	60.00	**63.24**	53.96	60.59
D2V words+2+3+4-grams	64.28	60.00	**63.24**	53.96	60.37
D2V words+2+3+4+5-grams	64.28	60.00	**63.24**	53.96	60.37
Char 3-grams (baseline)	52.67	45.55	47.00	30.15	43.84

Table 7 Results for *The Guardian* corpus considering *UK* as the training and the rest of the categories as the testing sets

Name	Politics	Society	World	Books reviews	Average Acc.
D2V words	78.57	90.32	78.63	82.53	82.51
D2V words+2-grams	**87.50**	93.54	**83.76**	88.88	88.42
D2V words+2+3-grams	**87.50**	**95.16**	82.90	**90.47**	**89.00**
D2V words+2+3+4-grams	**87.50**	**95.16**	82.90	88.88	88.61
D2V words+2+3+4+5-grams	**87.50**	**95.16**	82.90	88.88	88.61
Char 3-grams (baseline)	69.64	70.96	60.68	61.90	65.79

a classifier F_k using the first element of the 2-tuple s_k as the training set and obtain the accuracy of the classifier Acc_k using the second element as the testing set. Finally report the average accuracy obtained by all the classifiers F.

The experiments were conducted over the all possible pairs of categories using the proposed input data types (words, words + words2-grams, words + words 2-grams + words 3-grams, words + words 2-grams + words 3-grams + words 4-grams and words + words 2-grams + words 3-grams + words 4-grams + words 5-grams) and the implementations of LibSVM and LR as classifiers. In the following tables the best result using the LR classifier is shown, note that no experiment was performed using the same category for both training and testing. For the experiments mentioned above, the char 3-grams approach (Stamatatos, 2013) is used as a baseline and the results obtained by its implementation in each pairing categories are shown along with the results of our proposed method in the following tables.

Table 6 shows the accuracy obtained when using *Society* category for training over the different pairs, the Table 7 shows the accuracy obtained when using *UK* category for training over the different pairs, the Table 8 shows the accuracy obtained when using *World* category for training over the different pairs and the Table 9 shows the accuracy obtained when using Books reviews category for training over the different pairs. The results presented in these tables indicate that the Doc2vec method obtain better distributed documents representation when the input data type is based on word 1-grams + 2-grams in most cases. For some cases the addition of words 3-grams, 4-grams or 5-grams also improve the results, but, based on our experiment we believe that the use of word 1-grams and 2-grams is enough. Specially considering that the more information is passed to the Doc 2 vec algorithm, the longer it takes to learn the distributed representation.

Table 10 shows the average accuracy obtained by the proposed input data types over the different pairs, the accuracy obtained by the baseline (Char 3-grams (Stamatatos, 2013)) and the accuracy reported in previous works: Typed character n-grams (Sapkota et al, 2015) Our best result is obtained when using the

Table 8 Results for *The Guardian* corpus considering *World* as the training and the rest of the categories as the testing sets

Name	Politics	Society	UK	Books reviews	Average Acc.
D2V words	78.57	87.09	80.00	**77.77**	80.85
D2V words+2-grams	83.92	**93.54**	**83.33**	**77.77**	**84.64**
D2V words+2+3-grams	**86.60**	**93.54**	**83.33**	74.60	84.51
D2V words+2+3+4-grams	85.71	**93.54**	82.22	74.60	84.01
D2V words+2+3+4+5-grams	85.71	**93.54**	82.22	74.60	84.01
Char 3-grams (baseline)	76.78	72.58	63.33	57.14	67.45

Table 9 Results for *The Guardian* corpus considering *Books reviews* as the training and the rest of the categories as the testing sets

Name	Politics	Society	UK	World	Average Acc.
D2V words	56.25	51.61	62.22	**51.28**	55.34
D2V words+2-grams	**60.71**	**59.67**	**62.22**	48.71	**57.82**
D2V words+2+3-grams	58.92	58.06	60.00	47.00	55.99
D2V words+2+3+4-grams	58.92	58.06	60.00	47.00	55.99
D2V words+2+3+4+5-grams	58.92	58.06	60.00	47.00	55.99
Char 3-grams (baseline)	43.75	24.19	36.66	32.47	34.26

Table 10 Results for *The Guardian* corpus averaging accuracy in each category

Name	Politics	Society	UK	World	Books reviews	Average Acc.
D2V words	91.68	59.41	82.51	80.85	55.34	73.95
D2V word+2-grams	**94.75**	**60.87**	88.42	**84.64**	**57.82**	**77.30**
D2V word+2+3-grams	93.80	60.59	**89.00**	84.51	55.99	76.77
D2V word+2+3+4-grams	94.20	60.37	88.61	84.01	55.99	65.43
D2V word+2+3+4+5-grams	94.20	60.37	88.61	84.01	55.99	65.43
Char 3-grams (baseline)	85.50	43.84	65.79	67.45	34.26	59.36
Typed char n-grams (Sapkota et al (2015))	–	–	–	–	–	57.00

distributed document representation with words + 2-grams as input data type. Nevertheless, the distributed document representation (with the different input data types) obtains better results than previous works on this corpus with this test scheme. From these results we can draw several conclusions, the distributed representations features for the AA problem is robust across unbalanced corpus, avoiding the use of specific techniques for this kind of corpus, such as re-sampling (Cleofas-Sánchez et al, 2016). It is also efficient in the cross-topic scenario, overcoming results obtained by methodologies of the state of the art.

6 Conclusions

In this paper we present a novel approach that uses the Doc2vec method for learning distributed document representation and supervised machine learning methods to perform the authorship attribution task. In addition of using words as the standard input data type for building the distributed representations we propose a new input data type based on word n-grams ($n = 2, 3, 4, 5$). The inclusion of word n-grams let us gain further insight into the efficiency of the model to learn syntactic and grammatical patterns of an author.

The experiments conducted over different datasets show that the use of Doc2vec method for learning distributed document representations, in most of the cases, outperforms the results achieved by the previous works. The experiments show that building distributed representation models of each proposed input data types (word n-grams with $n = 2, 3, 4, 5$) independently and then combining them into a single vector obtained the best results (most of the time) over the different datasets. The representation using words and bigrams of words resulted to be the more informative combination than the other proposed input data types.

The datasets used in our experiments offered a variety of scenarios to test our proposal, the most interesting one was *The Guardian* corpus because it gave us

the possibility to evaluate our proposal over cross-topic and cross-genre texts with the small length (around 2KBytes and 13KBytes). The results obtained for this dataset (see Table 5 and Table 10) outperformed the results reported by previous works where the use of character and word n-grams yielded the best performance.

The proposed method applied to other datasets also achieved good results, in the PAN/CLEF 2012 datasets our proposal obtained results comparable to the best teams at the workshop in task A and C. In the case of the task I our proposed method outperforms the best accuracy reported in the workshop (see Table 3). For the RCV1-50 corpus the results of our proposal also outperforms the results obtained by previous works (see Table 4). In the RCV1-10 corpus, our error was 1.8% higher than the best result reported in previous works (see Table 4).

In general, the use of the Doc2vec representation offers a robust way to represent the style within the documents and therefore allows to obtain competitive results in the authorship attribution task, outperforming the well known character and word n-grams in the majority of our evaluation corpora.

Acknowledgements

This work was done under partial support of the Mexican Government (CONACYT PROJECT 240844, SNI, COFAA - IPN, SIP - IPN 20151406, 20144274).

Compliance with Ethical Standards

Conflict of Interest: The authors report no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alzahrani S, Salim N, Abraham A (2012) Understanding plagiarism linguistic patterns, textual features, and detection methods. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42(2):133 –149

Argamon S, Juola P (2011) Overview of the international authorship identification competition at pan-2011. In: CLEF (Notebook Papers/Labs/Workshop)

Argamon S, Koppel M, Pennebaker JW, Schler J (2009) Automatically profiling the author of an anonymous text. Communications of the ACM 52(2):119–123

Bird R, Wadler P (1988) Introduction to functional programming, vol 1. Prentice Hall New York

Brocardo ML, Traore I, Saad S, Woungang I (2013) Authorship verification for short messages using stylometry. In: Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on, IEEE, pp 1–6

Chaski CE (2005) Who's at the keyboard? authorship attribution in digital evidence investigations. International Journal of Digital Evidence 4(1):1–13

Cleofas-Sánchez L, Sánchez J, García V, Valdovinos R (2016) Associative learning on imbalanced environments: An empirical study. Expert Systems with Applications 54:387–397

Escalante HJ, Solorio T, Montes-y Gómez M (2011) Local histograms of character n-grams for authorship attribution. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pp 288–298

Franco-Salvador M, Rosso P, Rangel F (2015) Distributed representations of words and documents for discriminating similar languages. In: Proceeding of the RANLP Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)

Holmes DI (1998) The evolution of stylometry in humanities scholarship. Literary and linguistic computing 13(3):111–117

Houvardas J, Stamatatos E (2006) Stamatatos e.: N-gram feature selection for authorship identification. In: In: 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications, Springer, pp 77–86

Juola P (2004) Ad-hoc authorship attribution competition. In: Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, pp 175–176

Juola P (2012) An overview of the traditional authorship attribution subtask. In: CLEF (Online Working Notes/Labs/Workshop)

Kešelj V, Peng F, Cercone N, Thomas C (2003) N-gram-based author profiles for authorship attribution. In: Proceedings of the conference pacific association for computational linguistics, PACLING, vol 3, pp 255–264

Kiros R, Zemel RS, Salakhutdinov RR (2014) A multiplicative model for learning distributed text-based attribute representations. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, De-

cember 8-13 2014, Montreal, Quebec, Canada, pp 2348–2356

Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pp 1188–1196

Lewis DD, Yang Y, Rose TG, Li F (2004) Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research 5:361–397

Li R, Shindo H (2015) Distributed document representation for document classification. In: Cao T, Lim EP, Zhou ZH, Ho TB, Cheung D, Motoda H (eds) Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol 9077, Springer International Publishing, pp 212–225

Matthews R, Merriam T (1993) Neural computation in stylometry i: An application to the works of shakespeare and fletcher. Literary and Linguistic Computing 8(4):203–209

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. CoRR abs/1301.3781, URL `http://arxiv.org/abs/1301.3781`

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pp 3111–3119

Mikolov T, tau Yih W, Zweig G (2013c) Linguistic regularities in continuous space word representations. In: NAACL HLT, Atlanta, Georgia, June 9, 14, pp 746–751

Miranda S, Gelbukh A, Sidorov G (2014) Generating summaries by means of synthesis of conceptual graphs. Revista Signos 47(86):463

Mnih A, Hinton GE (2009) A scalable hierarchical distributed language model. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in Neural Information Processing Systems 21, Curran Associates, Inc., pp 1081–1088

Mosteller F, Wallace DL (1963) Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. Journal of the American Statistical Association 58(302):275–309

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp 1532–1543

Plakias S, Stamatatos E (2008) Tensor space models for authorship identification. In: Darzentas J, Vouros G, Vosinakis S, Arnellos A (eds) Artificial Intelligence: Theories, Models and Applications, Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol 5138, pp 239–249

Potthast M, Braun S, Buz T, Duffhauss F, Friedrich F, Gülzow JM, Köhler J, Lötzsch W, Müller F, Müller ME, Paßmann R, Reinke B, Rettenmeier L, Rometsch T, Sommer T, Träger M, Wilhelm S, Stein B, Stamatatos E, Hagen M (2016) Who wrote the web? revisiting influential author identification research applicable to information retrieval. In: Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings, pp 393–407

Rhodes D (2015) Author attribution with cnns. Tech. rep., CS224, Stanford University

Sanchez-Perez MA, Sidorov G, Gelbukh AF (2014) A winning approach to text alignment for text reuse detection at pan 2014. In: CLEF (Working Notes), pp 1004–1011

Sapkota U, Bethard S, Montes-y Gómez M, Solorio T (2015) Not all character n-grams are created equal: A study in authorship attribution. In: Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pp 93–102

Segarra S, Eisen M, Ribeiro A (2013) Authorship attribution using function words adjacency networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013, pp 5563–5567

Sidorov G, Velasquez F, Stamatatos E, Gelbukh A, Chanona-Hernández L (2014) Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications 41(3):853–860

Socher R, Bauer J, Manning CD, Ng AY (2013a) Parsing with compositional vector grammars. In: Proceedings of the ACL conference

Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C (2013b) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol 1631, p 1642

Stamatatos E (2008) Author identification: Using text sampling to handle the class imbalance problem. Information Processing & Management 44(2):790–799

Stamatatos E (2009) A survey of modern authorship attribution methods. Journal of the American Society

for Information Science and Technology 60(3):538 – 556

Stamatatos E (2011) Plagiarism detection using stopword n-grams. Journal of the American Society for Information Science and Technology 62(12):2512–2527

Stamatatos E (2013) On the robustness of authorship attribution based on character n-gram features. Journal of Law and Policy 21(2)

Stamatatos E, Fakotakis N, Kokkinakis G (2001) Computer-based authorship attribution without lexical measures. Computers and the Humanities 35(2):193–214

Trejo JVC, Sidorov G, Miranda-Jiménez S, Ibarra MAM, Martínez RC (2015) Latent dirichlet allocation complement in the vector space model for multi-label text classification. IJCOPI 6(1):7–19

Turian J, Ratinov L, Bengio Y (2010) Word representations: A simple and general method for semisupervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp 384–394

Wiemer-Hastings P, Wiemer-Hastings K, Graesser A (2004) Latent semantic analysis. In: Proceedings of the 16th international joint conference on Artificial intelligence, pp 1–14