Local Histograms of Character N-grams for Authorship Attribution

Hugo Jair Escalante
Graduate Program in Systems Eng.
Universidad Autónoma de Nuevo León,
San Nicolás de los Garza, NL, 66450, México
hugo.jair@gmail.com

Thamar Solorio
Dept. of Computer and Information Sciences
University of Alabama at Birmingham,
Birmingham, AL, 35294, USA
solorio@cis.uab.edu

Manuel Montes-y-Gómez
Computer Science Department, INAOE,
Tonantzintla, Puebla, 72840, México
Department of Computer and Information Sciences,
University of Alabama at Birmingham,
Birmingham, AL, 35294, USA
mmontesg@cis.uab.edu

Abstract

This paper proposes the use of local histograms (LH) over character n-grams for authorship attribution (AA). LHs are enriched histogram representations that preserve sequential information in documents; they have been successfully used for text categorization and document visualization using *word* histograms. In this work we explore the suitability of LHs over n-grams at the *character-level* for AA. We show that LHs are particularly helpful for AA, because they provide useful information for uncovering, to some extent, the writing style of authors. We report experimental results in AA data sets that confirm that LHs over character n-grams are more helpful for AA than the usual global histograms, yielding results far superior to state of the art approaches. We found that LHs are even more advantageous in challenging conditions, such as having imbalanced and small training sets. Our results motivate further research on the use of LHs for modeling the writing style of authors for related tasks, such as authorship verification and plagiarism detection.

1 Introduction

Authorship attribution (AA) is the task of deciding whom, from a set of candidates, is the author of a given document (Houvardas and Stamatatos, 2006; Luyckx and Daelemans, 2010; Stamatatos, 2009b). There is a broad field of application for AA methods, including spam filtering (de Vel et al., 2001), fraud detection, computer forensics (Lambers and Veenman, 2009), cyber bullying (Pillay and Solorio, 2010) and plagiarism detection (Stamatatos, 2009a). Therefore, the development of automated AA techniques has received much attention recently (Stamatatos, 2009b). The AA problem can be naturally posed as one of single-label multiclass classification, with as many classes as candidate authors. However, unlike usual text categorization tasks, where the core problem is modeling the thematic content of documents (Sebastiani, 2002), the goal in AA is modeling authors' writing style (Stamatatos, 2009b). Hence, document representations that reveal information about writing style are required to achieve good accuracy in AA.

Word and character based representations have been used in AA with some success so far (Houvardas and Stamatatos, 2006; Luyckx and Daelemans, 2010; Plakias and Stamatatos, 2008b). Such representations can capture style information through word or character usage, but they lack sequential information, which can reveal further stylistic information. In this paper, we study the use of richer document representations for the AA task. In particular, we consider local histograms over n-grams at the character-level obtained via the locally-weighted bag of words (LOWBOW) framework (Lebanon et al., 2007).

Under LOWBOW, a document is represented by a set of local histograms, computed across the whole document but smoothed by kernels centered on different document locations. In this way, document

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 288–298,
Portland, Oregon, June 19-24, 2011.

representations preserve both word/character usage and sequential information (i.e., information about the positions in which words or characters occur), which can be more helpful for modeling the writing style of authors. We report experimental results in an AA data set used in previous studies under several conditions (Houvardas and Stamatatos, 2006; Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a). Results confirm that local histograms of character n-grams are more helpful for AA than the usual global histograms of words or character n-grams (Luyckx and Daelemans, 2010); our results are superior to those reported in related works. We also show that local histograms over character n-grams are more helpful than local histograms over words, as originally proposed by (Lebanon et al., 2007). Further, we performed experiments with imbalanced and small training sets (i.e., under a realistic AA setting) using the aforementioned representations. We found that the LOWBOW-based representation resulted even more advantageous in these challenging conditions. The contributions of this work are as follows:

- We show that the LOWBOW framework can be helpful for AA, giving evidence that sequential information encoded in local histograms is useful for modeling the writing style of authors.
- We propose the use of local histograms over character-level n-grams for AA. We show that character-level representations, which have proved to be very effective for AA (Luyckx and Daelemans, 2010), can be further improved by adopting a local histogram formulation. Also, we empirically show that local histograms at the character-level are more helpful than local histograms at the word-level for AA.
- We study several kernels for a support vector machine AA classifier under the local histograms formulation. Our study confirms that the diffusion kernel (Lafferty and Lebanon, 2005) is the most effective among those we tried, although competitive performance can be obtained with simpler kernels.
- We report experimental results that are superior to state of the art approaches (Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a), with improvements ranging from $2\% - 6\%$ in balanced data sets and from $14\% - 30\%$ in imbalanced data sets.

2 Related Work

AA can be faced as a multiclass classification task with as many classes as candidate authors. Standard classification methods have been applied to this problem, including support vector machine (SVM) classifiers (Houvardas and Stamatatos, 2006) and variants thereon (Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a), neural networks (Tearle et al., 2008), Bayesian classifiers (Coyotl-Morales et al., 2006), decision tree methods (Koppel et al., 2009) and similarity based techniques (Keselj et al., 2003; Lambers and Veenman, 2009; Stamatatos, 2009b; Koppel et al., 2009). In this work, we chose an SVM classifier as it has reported acceptable performance in AA and because it will allow us to directly compare results with previous work that has used this same classifier.

A broad diversity of features has been used to represent documents in AA (Stamatatos, 2009b). However, as in text categorization (Sebastiani, 2002), word-based and character-based features are among the most widely used features (Stamatatos, 2009b; Luyckx and Daelemans, 2010). With respect to word-based features, word histograms (i.e., the bag-of-words paradigm) are the most frequently used representations in AA (Zhao and Zobel, 2005; Argamon and Levitan, 2005; Stamatatos, 2009b). Some researchers have gone a step further and have attempted to capture sequential information by using n-grams at the word-level (Peng et al., 2004) or by discovering maximal frequent word sequences (Coyotl-Morales et al., 2006). Unfortunately, because of computational limitations, the latter methods cannot discover *enough* sequential information from documents (e.g., word n-grams are often restricted to $n \in \{1, 2, 3\}$, while full sequential information would be obtained with $n \in \{1 \dots D\}$ where D is the maximum number of words in a document).

With respect to character-based features, n-grams at the character level have been widely used in AA as well (Plakias and Stamatatos, 2008b; Peng et al., 2003; Luyckx and Daelemans, 2010). Peng et al. (2003) propose the use of language models at the n-gram character-level for AA, whereas Keselj et al. (2003) build author profiles based on a selection of frequent n-grams for each author. Stamatatos and co-workers have studied the impact of feature selection, with character n-grams, in AA (Houvardas and Stamatatos, 2006; Stamatatos, 2006a), ensemble learning with character n-grams (Stamatatos, 2006b) and novel classification techniques based

on characters at the n-gram level (Plakias and Stamatatos, 2008a).

Acceptable performance in AA has been reported with character n-gram representations. However, as with word-based features, character n-grams are unable to incorporate sequential information from documents in their original form (in terms of the positions in which the terms appear across a document). We believe that sequential clues can be helpful for AA because different authors are expected to use different character n-grams or words in different parts of the document. Accordingly, in this work we adopt the popular character-based and word-based representations, but we enrich them in a way that they incorporate sequential information via the LOWBOW framework. Hence, the proposed features preserve sequential information besides capturing character and word usage information. Our hypothesis is that the combination of sequential and frequency information can be particularly helpful for AA.

The LOWBOW framework has been mainly used for document visualization (Lebanon et al., 2007; Mao et al., 2007), where researchers have used information derived from local histograms for displaying a 2D representation of document's content. More recently, Chasanis et al. (2009) used the LOWBOW framework for segmenting movies into chapters and scenes. LOWBOW representations have also been applied to discourse segmentation (AMIDA, 2007) and have been suggested for text summarization (Das and Martins, 2007). However, to the best of our knowledge the use of the LOWBOW framework for AA has not been studied elsewhere. Actually, the only two references using this framework for text categorization are (Lebanon et al., 2007; AMIDA, 2007). The latter can be due to the fact that local histograms provide little gain over usual global histograms for thematic classification tasks. In this paper we show that LOWBOW representations provide important improvements over global histograms for AA; in particular, local histograms at the character-level achieve the highest performance in our experiments.

3 Background

This section describes preliminary information on document representations and pattern classification with SVMs.

3.1 Bag of words representations

In the bag of words (BOW) representation, documents are represented by histograms over the vocabulary[1] that was used to generate a collection of documents; that is, a document i is represented as:

$$\mathbf{d}_i = [x_{i,1}, \ldots, x_{i,|V|}] \quad (1)$$

where V is the vocabulary and $|V|$ is the number of elements in V, $\mathbf{d}_{i,j} = x_{i,j}$ is a weight that denotes the contribution of term j to the representation of document i; usually $x_{i,j}$ is related to the occurrence (binary weighting) or the weighted frequency of occurrence (e.g., the *tf-idf* weighting scheme) of the term j in document i.

3.2 Locally-weighted bag-of-words representation

Instead of using the BOW framework directly, we adopted the LOWBOW framework for document representation (Lebanon et al., 2007). The underlying idea in LOWBOW is to compute several local histograms per document, where these histograms are smoothed by a kernel function, see Figure 1. The parameters of the kernel specify the position of the kernel in the document (i.e., where the local histogram is centered) and its scale (i.e., to what extent it is smoothed). In this way the sequential information in the document is preserved together with term usage statistics.

Let $W_i = \{w_{i,1}, \ldots, w_{i,N_i}\}$, denote the terms (in order of appearance) in document i where N_i is the number of terms that appear in document i and $w_{i,j} \in V$ is the term appearing at position j; let $\mathbf{v}_i = \{v_{i,1}, \ldots, v_{i,N_i}\}$ be the set of indexes in the vocabulary V of the terms appearing in W_i, such that $v_{i,j}$ is the index in V of the term $w_{i,j}$; let $\mathbf{t} = [t_1, \ldots, t_{N_i}]$ be a set of (equally spaced) scalars that determine intervals, with $0 \leq t_j \leq 1$ and $\sum_{j=1}^{N_i} t_j = 1$, such that each t_j can be associated to a position in W_i. Given a kernel smoothing function $K^s_{\mu,\sigma} : [0,1] \rightarrow \mathbb{R}$ with location parameter μ and scale parameter σ, where $\sum_{j=1}^{k} K^s_{\mu,\sigma}(t_j) = 1$ and

[1] In the following we will refer to arbitrary vocabularies, which can be formed with terms from either words or character n-grams.

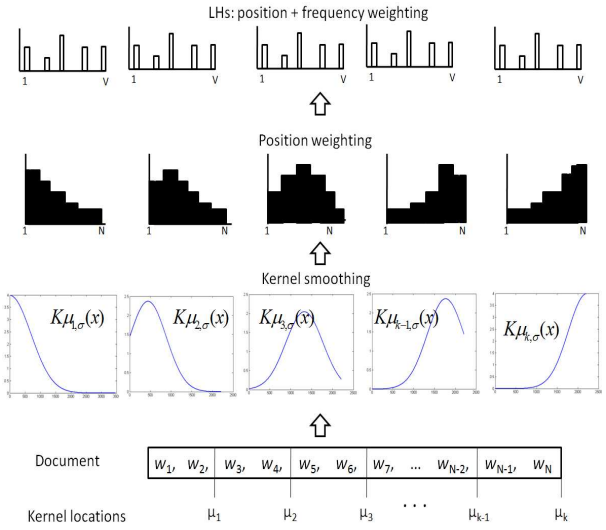

Figure 1: Diagram of the process for obtaining local histograms. Terms (w_i) appearing in different positions ($1, \ldots, N$) of the document are weighted according to the locations ($\mu_1, \ldots, \mu_k$) of the smoothing function $K_{\mu,\sigma}(x)$. Then, the term position weighting is combined with term frequency weighting for obtaining local histograms over the terms in the vocabulary ($1, \ldots, |V|$).

$\mu \in [0, 1]$. The LOWBOW framework computes a local histogram for each position $\mu_j \in \{\mu_1, \ldots, \mu_k\}$ as follows:

$$\mathbf{dl}^j_{i,\{v_{i,1},\ldots,v_{i,N_i}\}} = \mathbf{d}_{i,\{v_{i,1},\ldots,v_{i,N_i}\}} \times K^s_{\mu_j,\sigma}(\mathbf{t}) \quad (2)$$

where $\mathbf{dl}_{i,v_j:v_j \notin \mathbf{v}_i} = const$, a small constant value, and $\mathbf{d}_{i,j}$ is defined as above. Hence, a set $\mathbf{dl}_i^{\{1,\ldots,k\}}$ of k local histograms are computed for each document i. Each histogram $\mathbf{dl}_i^j$ carries information about the distribution of terms at a certain position μ_j of the document, where σ determines how the nearby terms to μ_j influence the local histogram j. Thus, sequential information of the document is considered throughout these local histograms. Note that when σ is small, most of the sequential information is preserved, as local histograms are calculated at very local scales; whereas when $\sigma \geq 1$, local histograms resemble the traditional BOW representation.

Under LOWBOW documents can be represented in two forms (Lebanon et al., 2007): as a single histogram $\mathbf{d}_i^L = const \times \sum_{j=1}^{k} \mathbf{dl}_i^j$ (hereafter LOWBOW histograms) or by the set of local histograms itself $\mathbf{dl}_i^{\{1,\ldots,k\}}$. We performed experiments with both forms of representation and considered words and n-grams at the character-level as terms (c.f. Section 5). Regarding the smoothing function, we considered the re-normalized Gaussian *pdf* restricted to [0, 1]:

$$K^s_{\mu,\sigma}(x) = \begin{cases} \frac{\mathcal{N}(x;\mu,\sigma)}{\phi(\frac{1-\mu}{\sigma}) - \phi(\frac{-\mu}{\sigma})} & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $\phi(x)$ is the cumulative distribution function for a Gaussian with mean 0 and standard deviation 1, evaluated at x, see (Lebanon et al., 2007) for further details.

3.3 Support vector machines

Support vector machines (SVMs) are pattern classification methods that aim to find an optimal separating hyperplane between examples from two different classes (Shawe-Taylor and Cristianini, 2004). Let $\{\mathbf{x}_i, y_i\}_N$ be pairs of training patterns-outputs, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y \in \{-1, 1\}$, with d the dimensionality of the problem. SVMs aim at learning a mapping from training instances to outputs. This is done by considering a linear function of the form: $f(\mathbf{x}) = W\mathbf{x} + b$, where parameters W and b are learned from training data. The particular linear function considered by SVMs is as follows:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b \quad (4)$$

that is, a linear function over (a subset of) training examples, where α_i is the weight associated with training example i (those for which $\alpha_i > 0$ are the so called support vectors) and y_i is the label associated with training example i, $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel[2] function that aims at mapping the input vectors, $(\mathbf{x}_i, \mathbf{x}_j)$, into the so called feature space, and b is a bias term. Intuitively, $K(\mathbf{x}_i, \mathbf{x}_j)$ evaluates how similar instances $\mathbf{x}_i$ and $\mathbf{x}_j$ are, thus the particular choice of kernel is problem dependent. The parameters in expression (4), namely $\alpha_{\{1,\ldots,N\}}$ and b, are learned by using exact optimization techniques (Shawe-Taylor and Cristianini, 2004).

[2]One should not confuse the kernel smoothing function, $K^s_{\mu,\sigma}(x)$, defined in Equation (3) with the Mercer kernel in Equation (4), as the former acts as a smoothing function and the latter acts as a similarity function.

4 Authorship Attribution with LOWBOW Representations

For AA we represent the training documents of each author using the framework described in Section 3.2, thus each document of each candidate author is either a LOWBOW histogram or a bag of local histograms (BOLH). Recall that LOWBOW histograms are an un-weighted sum of local histograms and hence can be considered a summary of term usage and sequential information; whereas the BOLH can be seen as term occurrence frequencies across different locations of the document.

For both types of representations we consider an SVM classifier under the one-vs-all formulation for facing the AA problem. We consider SVM as base classifier because this method has proved to be very effective in a large number of applications, including AA (Houvardas and Stamatatos, 2006; Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a); further, since SVMs are kernel-based methods, they allow us to use local histograms for AA by considering kernels that work over sets of histograms.

We build a multiclass SVM classifier by considering the pairs of patterns-outputs associated to documents-authors. Where each pattern can be either a LOWBOW histogram or the set of local histograms associated with the corresponding document, and the output associated to each pattern is a categorical random variable (outputs) that associates the representation of each document to its corresponding author $y_{1,\ldots,N} \in \{1,\ldots,C\}$, with C the number of candidate authors. For building the multiclass classifier we adopted the one-vs-all formulation, where C binary classifiers are built and where each classifier f_i discriminates among examples from class i (positive examples) and the rest $j : j \in \{1,\ldots,C\}, j \neq i$; despite being one of the simplest formulations, this approach has shown to obtain comparable and even superior performance to that obtained by more complex formulations (Rifkin and Klautau, 2004).

For AA using LOWBOW histograms, we consider a linear kernel since it has been successfully applied to a wide variety of problems (Shawe-Taylor and Cristianini, 2004), including AA (Houvardas and Stamatatos, 2006; Plakias and Stamatatos, 2008b). However, *standard* kernels cannot work for input spaces where each instance is described by a set of vectors. Therefore, usual kernels are not applicable for AA using BOLH. Instead, we rely on particular kernels defined for sets of vectors rather than for a single vector. Specifically, we consider kernels of the form (Rubner et al., 2001; Grauman, 2006):

$$K(P,Q) = \exp\big(-\frac{D(P,Q)^2}{\gamma}\big) \qquad (5)$$

where $D(P,Q)$ is the sum of the distances between the elements of the bag of local histograms associated to author P and the elements of the bag of histograms associated with author Q; γ is the scale parameter of K. Let $P = \{\mathbf{p}_1,\ldots,\mathbf{p}_k\}$ and $Q = \{\mathbf{q}_1,\ldots,\mathbf{q}_k\}$ be the elements of the bags of local histograms for instances P and Q, respectively, Table 1 presents the distance measures we consider for AA using local histograms.

Kernel	Distance
Diffusion	$D(P,Q) = \sum_{l=1}^{k} \arccos\left(\langle\sqrt{\mathbf{p}}_l \cdot \sqrt{\mathbf{q}}_l\rangle\right)$
EMD	$D(P,Q) = EMD(P,Q)$
Eucidean	$D(P,Q) = \sqrt{\sum_{l=1}^{k}(\mathbf{p}_l - \mathbf{q}_l).^2}$
χ^2	$D(P,Q) = \sqrt{\sum_{l=1}^{k} \frac{(\mathbf{p}_l - \mathbf{q}_l)^2}{(\mathbf{p}_l + \mathbf{q}_l)}}$

Table 1: Distance functions used to calculate the kernel defined in Equation (5).

Diffusion, Euclidean, and χ^2 kernels compare local histograms one to one, which means that the local histograms calculated at the same locations are compared to each other. We believe that for AA this is advantageous as it is expected that an author uses similar terms at similar locations of the document. The Earth mover's distance (EMD), on the other hand, is an estimate of the optimal cost in taking local histograms from Q to local histograms in P (Rubner et al., 2001); that is, this measure computes the optimal matching distance between local histograms from different authors that are not necessarily computed at similar locations.

5 Experiments and Results

For our experiments we considered the data set used in (Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a). This corpus is a subset of the RCV1 collection (Lewis et al., 2004) and comprises

documents authored by 10 authors. All of the documents belong to the same topic. Since this data set has predefined training and testing partitions, our results are comparable to those obtained by other researchers. There are 50 documents per author for training and 50 documents per author for testing.

We performed experiments with LOWBOW[3] representations at word and character-level. For the experiments with words, we took the top 2,500 most common words used across the training documents and obtained LOWBOW representations. We used this setting in agreement with previous work on AA (Houvardas and Stamatatos, 2006). For our character n-gram experiments, we obtained LOWBOW representations for character 3-grams (only n-grams of size $n = 3$ were used) considering the $2,500$ most common n-grams. Again, this setting was adopted in agreement with previous work on AA with character n-grams (Houvardas and Stamatatos, 2006; Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a; Luyckx and Daelemans, 2010). All our experiments use the SVM implementation provided by Canu et al. (2005).

5.1 Experimental settings

In order to compare our methods to related works we adopted the following experimental setting. We perform experiments using all of the training documents per author, that is, a balanced corpus (we call this setting **BC**). Next we evaluate the performance of classifiers over reduced training sets. We tried balanced reduced data sets with: $1, 3, 5$ and 10 documents per author (we call this configuration **RBC**). Also, we experimented with reduced-imbalanced data sets using the same imbalance rates reported in (Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a): we tried settings $2-10$, $5-10$, and $10-20$, where, for example, setting 2-10 means that we use at least 2 and at most 10 documents per author (we call this setting **IRBC**). **BC** setting represents the AA problem under ideal conditions, whereas settings **RBC** and **IRBC** aim at emulating a more realistic scenario, where limited sample documents are available and the whole data set is highly imbalanced (Plakias and Stamatatos, 2008b).

[3]We used LOWBOW code of G. Lebanon and Y. Mao available from http://www.cc.gatech.edu/∼ymao8/lowbow.htm

5.2 Experimental results in balanced data

We first compare the performance of the LOWBOW histogram representation to that of the traditional BOW representation. Table 2 shows the accuracy (i.e., percentage of documents in the test set that were associated to its correct author) for the BOW and LOWBOW histogram representations when using words and character n-grams information. For LOWBOW histograms, we report results with three different configurations for μ. As in (Lebanon et al., 2007), we consider uniformly distributed locations and we varied the number of locations that were included in each setting. We denote with k the number of local histograms. In preliminary experiments we tried several other values for k, although we found that representative results can be obtained with the values we considered here.

Method	Parameters	Words	Characters
BOW	-	78.2%	75.0%
LOWBOW	$k = 2; \sigma = 0.2$	75.8%	72.0%
LOWBOW	$k = 5; \sigma = 0.2$	77.4%	75.2%
LOWBOW	$k = 20; \sigma = 0.2$	77.4%	75.0%

Table 2: Authorship attribution accuracy for the BOW representation and LOWBOW histograms. Column 2 shows the parameters we used for the LOWBOW histograms; columns 3 and 4 show results using words and character n-grams, respectively.

From Table 2 we can see that the BOW representation is very effective, outperforming most of the LOWBOW histogram configurations. Despite a small difference in performance, BOW is advantageous over LOWBOW histograms because it is simpler to compute and it does not rely on parameter selection. Recall that the LOWBOW histogram representations are obtained by the combination of several local histograms calculated at different locations of the document, hence, it seems that the raw sum of local histograms results in a loss of useful information for representing documents. The worse performance was obtained when $k = 2$ local histograms are considered (see row 3 in Table 2). This result is somewhat expected since the larger the number of local histograms, the more LOWBOW histograms approach the BOW formulation (Lebanon et al., 2007).

We now describe the AA performance obtained when using the BOLH formulation; these results

are shown in Table 3. Most of the results from this table are superior to those reported in Table 2, showing that bags of local histograms are a better way to exploit the LOWBOW framework for AA. As expected, different kernels yield different results. However, the diffusion kernel outperformed most of the results obtained with other kernels; confirming the results obtained by other researchers (Lebanon et al., 2007; Lafferty and Lebanon, 2005).

Kernel	Euc.	Diffusion	EMD	χ^2
		Words		
Setting-1	78.6%	81.0%	75.0%	75.4%
Setting-2	77.6%	82.0%	76.8%	77.2%
Setting-3	79.2%	80.8%	77.0%	79.0%
		Characters		
Setting-1	83.4%	82.8%	84.4%	83.8%
Setting-2	83.4%	84.2%	82.2%	84.6%
Setting-3	83.6%	**86.4%**	81.0%	85.2%

Table 3: Authorship attribution accuracy when using bags of local histograms and different kernels for word-based and character-based representations. The **BC** data set is used. Settings 1, 2 and 3 correspond to $k = 2, 5$ and 20, respectively.

On average, the worse kernel was that based on the earth mover's distance (EMD), suggesting that the comparison of local histograms at different locations is not a fruitful approach (recall that this is the only kernel that compares local histograms at different locations). This result evidences that authors use similar word/character distributions at similar locations when writing different documents.

The best performance across settings and kernels was obtained with the diffusion kernel (in bold, column 3, row 9) (86.4%); that result is 8% higher than that obtained with the BOW representation and 9% better than the best configuration of LOWBOW histograms, see Table 2. Furthermore, that result is more than 5% higher than the best reported result in related work (80.8% as reported in (Plakias and Stamatatos, 2008b)). Therefore, the considered local histogram representations over character n-grams have proved to be very effective for AA.

One should note that, in general, better performance was obtained when using character-level rather than word-level information. This confirms the results already reported by other researchers that have used character-level and word-level information for AA (Houvardas and Stamatatos, 2006; Plakias and Stamatatos, 2008b; Plakias and Stamatatos, 2008a; Peng et al., 2003). We believe this can be attributed to the fact that character n-grams provide a representation for the document at a finer granularity, which can be better exploited with local histogram representations. Note that by considering 3-grams, words of length up to three are incorporated, and usually these words are function words (e.g., the, it, as, etc.), which are known to be indicative of writing style. Also, n-gram information is more dense in documents than word-level information. Hence, the local histograms are less sparse when using character-level information, which results in better AA performance.

				True author					
AC	AS	BL	DL	JM	JG	MM	MD	RS	TN
88	2	0	0	0	0	0	0	0	0
10	98	0	0	0	0	0	0	0	0
0	0	68	0	40	0	0	0	0	0
0	0	0	80	0	0	0	0	0	4
0	0	12	2	42	0	0	2	0	0
0	0	0	0	0	100	0	0	0	2
2	0	2	0	0	0	100	0	0	0
0	0	18	0	18	0	0	98	0	0
0	0	0	2	0	0	0	0	100	4
0	0	0	16	0	0	0	0	0	90

Table 4: Confusion matrix (in terms of percentages) for the best result in the **BC** corpus (i.e., last row, column 3 in Table 3). Columns show the true author for test documents and rows show the authors predicted by the SVM.

Table 4 shows the confusion matrix for the setting that reached the best results (i.e., column 3, last row in Table 3). From this table we can see that 8 out of the 10 authors were recognized with an accuracy higher or equal to 80%. For these authors sequential information seems to be particularly helpful. However, low recognition performance was obtained for authors BL (B. K. Lim) and JM (J. MacArtney). The SVM with BOW representation of character n-grams achieved recognition rates of 40% and 50% for BL and JM respectively. Thus, we can state that sequential information was indeed helpful for modeling BL writing style (improvement of 28%), although it is an author that resulted very difficult to model. On the other hand, local histograms were not very useful for identifying documents written by JM (made it worse by -8%). The largest improvement (38%) of local histograms over the BOW formulation was obtained for author TN (T. Nissen). This

result gives evidence that TN uses a similar distribution of words in similar locations across the documents he writes. These results are interesting, although we would like to perform a careful analysis of results in order to determine for what type of authors it would be beneficial to use local histograms, and what type of authors are better modeled with a standard BOW approach.

5.3 Experimental results in imbalanced data

In this section we report results with **RBC** and **IRBC** data sets, which aim to evaluate the performance of our methods in a realistic setting. For these experiments we compare the performance of the BOW, LOWBOW histogram and BOLH representations; for the latter, we considered the best setting as reported in Table 3 (i.e., an SVM with diffusion kernel and $k = 20$). Tables 5 and 6 show the AA performances when using word and character information, respectively.

We first analyze the results in the **RBC** data set (recall that for this data set we consider 1, 3, 5, 10, and 50, randomly selected documents per author). From Tables 5 and 6 we can see that BOW and LOWBOW histogram representations obtained similar performance to each other across the different training set sizes, which agree with results in Table 2 for the **BC** data sets. The best performance across the different configurations of the **RBC** data set was obtained with the BOLH formulation (row 6 in Tables 5 and 6). The improvements of local histograms over the BOW formulation vary across different settings and when using information at word-level and character-level. When using words (columns 2-6 in Table 5) the differences in performance are of 15.6%, 6.2%, 6.8%, 2.9%, 3.8% when using 1, 3, 5, 10 and 50 documents per author, respectively. Thus, it is evident that local histograms are more beneficial when less documents are considered. Here, the lack of information is compensated by the availability of several histograms per author.

When using character n-grams (columns 2-6 in Table 6) the corresponding differences in performance are of 5.4%, 6.4%, 6.4%, 6% and 11.4%, when using 1, 3, 5, 10, and 50 documents per author, respectively. In this case, the larger improvement was obtained when 50 documents per author are available; nevertheless, one should note that results using character-level information are, in general, significantly better than those obtained with word-level information; hence, improvements are expected to be smaller.

When we compare the results of the BOLH formulation with the best reported results elsewhere (c.f. last row 6 in Tables 5 and 6) (Plakias and Stamatatos, 2008b), we found that the improvements range from 14% to 30.2% when using character n-grams and from 1.2% to 26% when using words. The differences in performance are larger when less information is used (e.g., when 5 documents are used for training) and we believe the differences would be even larger if results for 1 and 3 documents were available. These are very positive results; for example, we can obtain almost 71% of accuracy, using local histograms of character n-grams when a single document is available per author (recall that we have used all of the test samples for evaluating the performance of our methods).

We now analyze the performance of the different methods when using the **IRBC** data set (columns 7-9 in Tables 5 and 6). The same pattern as before can be observed in experimental results for these data sets as well: BOW and LOWBOW histograms obtained comparable performance to each other and the BOLH formulation performed the best. The BOLH formulation outperforms state of the art approaches by a considerable margin that ranges from 10% to 27%. Again, better results were obtained when using character n-grams for the local histograms. With respect to **RBC** data sets, the BOLH at the character-level resulted very robust to the reduction of training set size and the highly imbalanced data.

Summarizing, the results obtained in **RBC** and **IRBC** data sets show that the use of local histograms is advantageous under challenging conditions. An SVM under the BOLH representation is less sensitive to the number of training examples available and to the imbalance of data than an SVM using the BOW representation. Our hypothesis for this behavior is that local histograms can be thought of as expanding training instances, because for each training instance in the BOW formulation we have $k-$training instances under BOLH. The benefits of such expansion become more notorious as the number of available documents per author decreases.

WORDS								
Data set	**Balanced**					**Imbalanced**		
Setting	*1-doc*	*3-docs*	*5-docs*	*10-docs*	*50-docs*	*2-10*	*5-10*	*10-20*
BOW	36.8%	57.1%	62.4%	69.9%	78.2%	62.3%	67.2%	71.2%
LOWBOW	37.9%	55.6%	60.5%	69.3%	77.4%	61.1%	67.4%	71.5%
Diffusion kernel	52.4%	63.3%	69.2%	72.8%	82.0%	66.6%	70.7%	74.1%
Reference	-	-	53.4%	67.8%	80.8%	49.2%	59.8%	63.0%

Table 5: AA accuracy in **RBC** (columns 2-6) and **IRBC** (columns 7-9) data sets when using words as terms. We report results for the BOW, LOWBOW histogram and BOLH representations. For reference (last row), we also include the best result reported in (Plakias and Stamatatos, 2008b), when available, for each configuration.

CHARACTER N-GRAMS								
Data set	**Balanced**					**Imbalanced**		
Setting	*1-doc*	*3-docs*	*5-docs*	*10-docs*	*50-docs*	*2-10*	*5-10*	*10-20*
BOW	65.3%	71.9%	74.2%	76.2%	75.0%	70.1%	73.4%	73.1%
LOWBOW	61.9%	71.6%	74.5%	73.8%	75.0%	70.8%	72.8%	72.1%
Diffusion kernel	70.7%	78.3%	80.6%	82.2%	86.4%	77.8%	80.5%	82.2%
Reference	-	-	50.4%	67.8%	76.6%	49.2%	59.8%	63.0%

Table 6: AA accuracy in the **RBC** and **IRBC** data sets when using character n-grams as terms.

6 Conclusions

We have described the use of local histograms (LH) over character n-grams for AA. LHs are enriched histogram representations that preserve sequential information in documents (in terms of the positions of terms in documents); we explored the suitability of LHs over n-grams at the *character-level* for AA. We showed evidence supporting our hypothesis that LHs are very helpful for AA; we believe that this is due to the fact that LOWBOW representations can uncover, to some extent, the writing preferences of authors. Our experimental results showed that LHs outperform traditional bag-of-words formulations and state of the art techniques in balanced, imbalanced, and reduced data sets. The improvements were larger in reduced and imbalanced data sets, which is a very positive result as in real AA applications one often faces highly imbalanced and small sample issues. Our results are promising and motivate further research on the use and extension of the LOWBOW framework for related tasks (e.g. authorship verification and plagiarism detection).

As future work we would like to explore the use of LOWBOW representations for profile-based AA and related tasks. Also, we would like to develop model selection strategies for learning what combination of hyperparameters works better for modeling each author.

Acknowledgments

We thank E. Stamatatos for making his data set available. Also, we are grateful for the thoughtful comments of L. A. Barrón and those of the anonymous reviewers. This work was partially supported by CONACYT under project grants 61335, and CB-2009-134186, and by UAB faculty development grant 3110841.

References

AMIDA. 2007. Augmented multi-party interaction with distance access. Available from `http://www.amidaproject.org/`, AMIDA Report.

S. Argamon and S. Levitan. 2005. Measuring the usefulness of function words for authorship attribution. In *Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, Victoria, BC, Canada.

S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. 2005. SVM and kernel methods Matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France.

V. Chasanis, A. Kalogeratos, and A. Likas. 2009. Movie segmentation into scenes and chapters using locally weighted bag of visual words. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 35:1–35:7, Santorini, Fira, Greece. ACM Press.

R. M. Coyotl-Morales, L. Villaseñor-Pineda, M. Montes-y-Gómez, and P. Rosso. 2006. Authorship attribution using word sequences. In *Proceedings of 11th*

Iberoamerican Congress on Pattern Recognition, volume 4225 of *LNCS*, pages 844–852, Cancun, Mexico. Springer.

D. Das and A. Martins. 2007. A survey on automatic text summarization. Available from: `http://www.cs.cmu.edu/~nasmith/LS2/das-martins.07.pdf`, Literature Survey for the Language and Statistics II course at Carnegie Mellon University.

O. de Vel, A. Anderson, M. Corney, and G. Mohay. 2001. Multitopic email authorship attribution forensics. In *Proceedings of the ACM Conference on Computer Security - Workshop on Data Mining for Security Applications*, Philadelphia, PA, USA.

K. Grauman. 2006. *Matching Sets of Features for Efficient Retrieval and Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

J. Houvardas and E. Stamatatos. 2006. N-gram feature selection for author identification. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, volume 4183 of *LNCS*, pages 77–86, Varna, Bulgaria. Springer.

V. Keselj, F. Peng, N. Cercone, and C. Thomas. 2003. N-gram-based author profiles for authorship attribution. In *Proceedings of the Pacific Association for Computational Linguistics*, pages 255–264, Halifax, Canada.

M. Koppel, J. Schler, and S. Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60:9–26.

J. Lafferty and G. Lebanon. 2005. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163.

M. Lambers and C. J. Veenman. 2009. Forensic authorship attribution using compression distances to prototypes. In *Computational Forensics, Lecture Notes in Computer Science, Volume 5718. ISBN 978-3-642-03520-3. Springer Berlin Heidelberg, 2009, p. 13*, volume 5718 of *LNCS*, pages 13–24. Springer.

G. Lebanon, Y. Mao, and J. Dillon. 2007. The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, 8:2405–2441.

D. Lewis, T. Yang, and F. Rose. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

K. Luyckx and W. Daelemans. 2010. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, pages 1–21, August.

Y. Mao, J. Dillon, and G. Lebanon. 2007. Sequential document visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1208–1215.

F. Peng, D. Shuurmans, V. Keselj, and S. Wang. 2003. Language independent authorship attribution using character level language models. In *Proceedings of the 10th conference of the European chapter of the Association for Computational Linguistics*, volume 1, pages 267–274, Budapest, Hungary.

F. Peng, D. Shuurmans, and S. Wang. 2004. Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval Journal*, 7(1):317–345.

S. R. Pillay and T. Solorio. 2010. Authorship attribution of web forum posts. In *Proceedings of the eCrime Researchers Summit (eCrime), 2010*, pages 1–7, Dallas, TX, USA. IEEE.

S. Plakias and E. Stamatatos. 2008a. Author identification using a tensor space representation. In *Proceedings of the 18th European Conference on Artificial Intelligence*, volume 178, pages 833–834, Patras, Greece. IOS Press.

S. Plakias and E. Stamatatos. 2008b. Tensor space models for authorship attribution. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications*, volume 5138 of *LNCS*, pages 239–249, Syros, Greece. Springer.

R. Rifkin and A. Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141.

Y. Rubner, C. Tomasi, J. Leonidas, and J. Guibas. 2001. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

J. Shawe-Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

E. Stamatatos. 2006a. Authorship attribution based on feature set subspacing ensembles. *International Journal on Artificial Intelligence Tools*, 15(5):823–838.

E. Stamatatos. 2006b. Ensemble-based author identification using character n-grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval*, pages 41–46, Riva del Garda, Italy.

E. Stamatatos. 2009a. Intrinsic plagiarism detection using character n-gram profiles. In *Proceedings of the 3rd International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN'09*, pages 38–46, Donostia-San Sebastian, Spain.

E. Stamatatos. 2009b. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

M. Tearle, K. Taylor, and H. Demuth. 2008. An algorithm for automated authorship attribution using neural networks. *Literary and Linguist Computing*, 23(4):425–442.

Y. Zhao and J. Zobel. 2005. Effective and scalable authorship attribution using function words. In *Proceedings of 2nd Asian Information Retrieval Symposium*, volume 3689 of *LNCS*, pages 174–189, Jeju Island, Korea. Springer.