Improving Cross-Topic Authorship Attribution: The Role of Pre-Processing

Ilia Markov[1], Efstathios Stamatatos[2], and Grigori Sidorov[1]

[1] Center for Computing Research (CIC), Instituto Politécnico Nacional (IPN),
Mexico City, Mexico
markovilya@yahoo.com, sidorov@cic.ipn.mx,
[2] Department of Information and Communication Systems Engineering,
University of the Aegean, Karlovassi, Samos, Greece
stamatatos@aegean.gr

Abstract. The effectiveness of character n-gram features for representing the stylistic properties of a text has been demonstrated in various independent Authorship Attribution (AA) studies. Moreover, it has been shown that some categories of character n-grams perform better than others both under single and cross-topic AA conditions. In this work, we present an improved algorithm for cross-topic AA. We demonstrate that the effectiveness of character n-grams representation can be significantly enhanced by performing simple pre-processing steps and appropriately tuning the number of features, especially in cross-topic conditions.

Keywords: pre-processing, authorship attribution, cross-topic, character n-grams, machine learning

1 Introduction

Authorship Attribution (AA) is the task that aims at identifying the author of a text given a predefined set of candidate authors [1]. Practical applications of AA vary from electronic commerce and forensics, where part of the evidence refers to texts, to humanities research [2–5].

From the machine-learning perspective, AA can be viewed as a multi-class, single-label classification problem. In single-topic AA, there are no major differences in the thematic areas of training and test corpora, whereas in cross-topic AA, the thematic areas of training and test corpora are disjoint [6]. The latter better matches the requirements of a realistic scenario of forensic applications, when the available texts by the candidate authors can belong to totally different thematic areas than the texts under investigation.

Character n-grams have proved to be the best predictive feature type both under single and cross-topic AA conditions [7, 6]. A reasonable explanation is that these features capture 'a bit of everything', including lexical and syntactic information, punctuation and capitalization information related with the authors' personal style. They are sensitive to both the content and form of a text [8, 1, 9] while their higher frequency with respect to other feature types, e.g., words, make their probabilities estimation more accurate [10].

Recently, Sapkota *et al.* [11] showed that some categories of character n-grams perform better than others both for single and cross-topic settings. They claimed that a AA model trained on character n-grams that capture information about affixes and punctuation (morpho-syntactic and stylistic information) performs better than using all possible n-grams. Their results indicate that it is possible to improve basic character n-gram features without the need of extracting more complicated features.

In this paper, we present an approach that applies simple pre-processing steps, such as replacing digits, splitting punctuation marks, and replacing named entities, before extracting character n-gram features. We adopt the character n-gram categories proposed by Sapkota *et al.* [11] and examine how pre-processing steps affect their effectiveness. We evaluate the contribution of each step when applied separately and in combination. We further show that an appropriate tuning of the number of features is crucial and can further enhance AA performance, especially in cross-topic conditions.

The research questions addressed in this work are the following:

1. Can we improve the performance of AA by applying simple pre-processing steps? Which pre-processing steps are appropriate for both single and cross-topic AA settings?
2. Is it possible to enhance AA performance by selecting an appropriate feature set size using only the training corpora?
3. Is the conclusion reported in [11], that the best performing model is based solely on affix and punctuation n-grams, valid even after applying pre-processing steps? Is this conclusion valid when using different classification algorithms?

2 Related Work

Previous work in AA focuses mainly on the extraction of stylometric features that represent the personal style of authors [7, 12–15]. Several studies demonstrate the effectiveness of character n-grams in AA tasks [16, 17, 7, 18]. These features were also found robust in AA experiments under cross-topic conditions [6, 19] despite the fact that they also capture thematic information. They are also strongly associated with compression-based models that essentially exploit common character sequences [20, 21]. Character n-grams can be used either alone [22, 18] or combined with other stylometric features [23].

In most previous AA studies, training and test corpora share similar thematic properties [24, 22, 20, 18]. An early cross-topic study is described in [25] where email messages in different topic categories were used in training and test corpora. The *unmasking* method for author verification was successfully tested in cross-topic conditions [26]. A comparison of character n-grams and lexical features in cross-topic conditions is provided in [6].

Sapkota *et al.* [19] proposed to enrich the training corpus with multiple topics to enhance the performance of AA on another topic. The recent PAN evaluation campaign on author identification focused on cross-topic and cross-genre author verification [27]. As expected, the performance of AA models in cross-topic conditions is lower in comparison to single-topic conditions [6].

3 Stylometric Features

3.1 Types of *n*-grams

In this paper, we adopt the character *n*-gram types introduced by Sapkota *et al.* [11]. However, we refine the original definitions for some of the categories of character *n*-grams in order to make them more accurate and complete. We also follow Sapkota *et al.* [11] and focus on character 3-grams. In more detail, there are 3 main types, and each one has sub-categories as explained below:

- **Affix character 3-grams**
  - **prefix** A 3-gram that covers the first 3 characters of a word that is at least 4 characters long.
  - **suffix** A 3-gram that covers the last 3 characters of a word that is at least 4 characters long.
  - **space-prefix** A 3-gram that begins with a space and does not contain punctuation.
  - **space-suffix** A 3-gram that ends with a space, does not contain punctuation, and whose first character is not a space.
- **Word character 3-grams**
  - **whole-word** A 3-gram that covers all characters of a word that is exactly 3 characters long.
  - **mid-word** A 3-gram that covers 3 characters of a word that is at least 5 characters long, and that covers neither the first nor the last character of the word.
  - **multi-word** A 3-gram that spans multiple words, identified by the presence of a space in the middle of the 3-gram.
- **Punctuation character 3-grams**
  - **beg-punct** A 3-gram whose first character is punctuation, but the middle character is not.
  - **mid-punct** A 3-gram whose middle character is punctuation.
  - **end-punct** A 3-gram whose last character is punctuation, but the first and the middle characters are not.

The advantage of our modified definitions is that each occurrence of a character 3-gram is unambiguously assigned to exactly one category. For example, we directly assign the 3-gram instance '␣a␣' to the *space-prefix* category, excluding it from the *space-suffix* category. Note that two instances of the same 3-gram can be assigned to different categories (e.g., in phrase *the mother*, the first instance of 3-gram *the* is assigned to *whole-word* and the second instance to *mid-word*).

Moreover, when using the original definitions by Sapkota *et al.* [11], we noticed that some n-grams do not fall into any of the categories (e.g., when two consecutive punctuation marks are in the beginning/end of a sentence). Our refined definitions do not exclude any n-gram.

As an example, let us consider the following sample sentence:

(1) *John said, “Tom can repair it for 12 euros.”*

The character 3-grams for the sample sentence (1) for each of the categories are following:

Table 1. Character 3-grams per category for the sample sentence (1) after applying the algorithm by Sapkota *et al.* [11].

SC	**Category**	***N*-grams**						
affix	*prefix*	Joh	sai	rep	eur			
	suffix	ohn	aid	air	ros			
	space-prefix	_sa	_ca	_re	_it	_fo	_12	_eu
	space-suffix	hn_	om_	an_	ir_	it_	or_	12_
word	*whole-word*	Tom	can	for				
	mid-word	epa	pai	uro				
	multi-word	n_s	m_c	n_r	r_i	t_f	r_1	2_e
punct	*beg-punct*	,_“	“To					
	mid-punct	d,_	_“T	s.”				
	end-punct	id,	os.					

3.2 Pre-processing steps

In this paper, we introduce simple pre-processing steps attempting to assist character n-gram features to capture more information related to personal style of the author and less information related to the theme of text. The pre-processing steps are applied before the extraction of n-grams and concern the following textual contents:

Digits (Ds) We replace each digit by 0 (e.g., 12,345 → 00,000) since the actual numbers do not carry stylistic information. However, their format (e.g., 1,000 vs. 10000 vs. 1k) reflects a stylistic choice of the author.

Punctuation marks (PMs) We split PMs in order to be able to capture their frequency separately and not just in combination with the adjacent words. For example, the character 3-grams *,_“*, *“To*, and *_“T* in Table 1 refer to the use of a quotation mark. The use of the same PM in a different context (suffix of previous word and prefix of next word) would produce completely different 3-grams. By splitting PMs from adjacent words we allow capturing the frequency of each PM as a separate 3-gram (e.g., *_“_*). We also add a space in the beginning and in the end of each line, as well as remove multiple whitespaces for the *mid-punct* category in order to be able to capture the frequency of all PMs.

Named entities (NEs) The use of NEs is strongly associated with the thematic area of texts. However, the patterns of their usage provide useful stylistic information. We replace all NE instances by the same symbol in order to keep information about their occurrence and remove information about the exact NEs.

Highly frequent words (HFWs) Usually highly frequent words are function words, e.g., prepositions, pronouns, etc. They are one of the most important stylometric features [9]. However, when a character n-gram representation is used, especially when n is low, it is not easy to capture patterns of their usage (combinations of certain HFWs with morphemes of previous or next words). To increase the ability of character 3-grams to capture such information, we replace each HFW by a distinct symbol.

As an illustrating example, the above pre-processing steps are applied to the sample sentence (1). NEs are replaced by symbol '#', HFWs *can* and *it* are replaced by symbols '%' and '$', respectively. The resulting sentence would be:

(2) *# said , “ # % repair $ for 00 euros . ”*

The character n-grams extracted from sample sentence (2) are shown in Table 2.

Table 2. Character 3-grams per category for the sample sentence (2) after applying our algorithm.

SC	**Category**	***N*-grams**
affix	*prefix*	sai rep eur
	suffix	aid air ros
	space-prefix	_sa _#_ _%_ _re _$_ _fo _00 _eu
	space-suffix	id_ ir_ or_ 00_ os_
word	*whole-word*	for
	mid-word	epa pai uro
	multi-word	#_s #_% %_r r_$ $_f r_0 0_e
punct	*beg-punct*	,_“ “_# ._”
	mid-punct	_,_ _“_ _._ _”_
	end-punct	d_, s_.

The proposed approach is more topic-neutral, since it does not depend on specific details that are not related to the personal style of authors. It is able to capture format of different numbers, dates, usage of NEs, the frequency of PMs and patterns of their usage, and patterns of HFWs usage. Finally, the number of features significantly decreases when applying our approach, as can be seen by comparing Tables 1 and 2 and as we show further in Section 5.[3]

4 Corpora and Experimental Settings

For the evaluation of our algorithm, we conducted experiments on both single-topic and cross-topic corpora. In more detail, we used CCAT_10, a subset of the Reuters Corpus Volume 1 [28], that includes 10 authors and 100 newswire stories per author on the same thematic area (corporate news). As in previous studies, we used the balanced training and test parts of this corpus [18, 11].

The cross-topic corpus used in this study is composed of texts published in *The Guardian* daily newspaper. It comprises opinion articles in four thematic areas (Politics, Society, World, U.K.) written by 13 authors [6]. The distribution of texts over the authors is not balanced and, following the practice of previous studies, at most ten documents per author were considered for each of the four topic categories [6, 11].

In order to be able to examine the contribution of each pre-processing step, we conducted our experiments using the same experimental settings as described in [11]. Thus, we used character 3-gram features and considered only the 3-grams that occur at least 5 times in the training corpus. We evaluate each model by measuring classification accuracy on the test corpus. For the cross-topic experiments, the results for each model correspond to the average accuracy over the 12 possible pairings of the 4 topics (training on one topic and testing on another). When the Society texts are used as training corpus, there are no training texts for one author. In that case, we removed all texts by that author from the test corpus.

To perform the pre-processing steps as described in the previous section, we used an improved version of Natural Language Toolkit[4] tokenizer, making sure that each PM is a separate token, and Stanford Named Entity Recognizer (NER) [29] in order to extract NEs, filtering out some erroneous detections. Different sets of highly frequent words were tested: 0, 50, 100, 150, and 200.

In order to examine whether different classifiers agree on the effectiveness of the proposed pre-processing steps, we compare the performance of two classifiers using their WEKA's [30] implementation: Support Vector Machines (SVM) and multinomial naive Bayes (MNB). These classification algorithms with default parameters are considered among the best for text categorization tasks [31, 32].[5]

[3] When large sets of HFWs are replaced by distinct symbols, the size of feature set increases.

[4] http://www.nltk.org [last access: 12.01.2017].

[5] We also examined naive Bayes classifier, which produced worse results but similar behaviour (not shown).

5 Experimental Results

5.1 Contribution of pre-processing steps

First, we re-implemented the method of Sapkota *et al.* [11] as described in their paper and applied it to the CCAT_10 and the Guardian corpora. Although the obtained results are very similar with the ones reported in [11], we were not able to reproduce the exact results. Correspondingly, we use the results of our own implementation of the algorithm by Sapkota *et al.* [11] as baseline for the proposed method.

Moreover, following the practice of Sapkota *et al.* [11] we examine three cases according to what kind of n-gram categories are used:
(1) **all-untyped** – where the categories of n-grams are ignored. Any distinct n-gram is a different feature.
(2) **all-typed** – where n-grams of all available categories (**affix+punct+word**) are considered. Instances of the same n-gram may refer to different features.
(3) **affix+punct** – where the n-grams of the **word** category are excluded.

Table 3 shows the performance of the baseline method and the contribution of each proposed pre-processing step separately, as well as their combinations on the CCAT_10 corpus. For the sake of brevity, we do not present all possible combinations, but only the most representative ones. In most cases, the pre-processing steps reduce the effectiveness of the AA models. In more detail, replacing NEs seems to be the least effective step. This can be explained by the thematic-specificity of this corpus. Each author tends to write news stories about specific topics, and this is consistent in both training and test corpora. NEs are strongly associated with thematic choices. The most useful combination of pre-processing steps for this corpus is the replacement of digits that manages to slightly improve the accuracy in most cases using either of the classification algorithms. SVM classifier seems better able to cope with this corpus.

The corresponding evaluation results on the Guardian corpus can be seen at Table 4. Here, most pre-processing steps significantly enhance the performance of AA models. In most cases, the best combination of steps is to replace digits and NEs, split PMs and not to replace HFWs. Note also, that the feature set size for this combination is significantly lower with respect to the baseline. In cross-topic conditions, the proposed approach provides a more robust reduced set of features that are not affected that much by topic shifts. Moreover, the MNB classifier provides much better results for this corpus, and it better handles the **all-untyped** features.

The main conclusion of Sapkota *et al.* [11] that models using **affix+punct** features are better than models trained on all the features is also valid in most cases of our experiments even when applying the proposed pre-processing steps. In addition, in the case of the cross-topic corpus, the highest improvement in accuracy is achieved for the **affix+punct** features when using both SVM and MNB classifiers (4.7% and 5.9% respectively).

To demonstrate the effectiveness of character n-gram features, we conducted experiments using the Bag-of-Words (BoW) approach, obtaining accuracy of

Table 3. Accuracy results on the CCAT_10 corpus after applying the proposed preprocessing steps. Accuracy (Acc, %) and the number of features (N) are reported for each step. The "+" columns show the difference of each step and each combination with the baseline. The best accuracy and improvement for each model are in bold; in the case when the accuracies are equal, we chose the one obtained with a smaller set of features.

Approach				**all-typed**			**affix+punct**			**all-untyped**		
D	PM	NE	HFW	Acc	+	N	Acc	+	N	Acc	+	N
Baseline				78.0		10,859	78.8		6,296	78.2		9,258
✓			0	77.8	–0.2	9,761	**79.6**	**0.8**	**5,503**	78.2	0.0	8,143
✓	✓		0	77.4	–0.6	8,430	77.4	–1.4	4,171	**78.2**	**0.0**	**6,648**
✓		✓	0	76.0	–2.0	7,606	75.4	–3.4	4,187	76.2	–2.0	6,364
✓	✓	✓	0	76.4	–1.6	6,651	75.8	–3.0	3,087	77.2	–1.0	5,239
✓			50	77.4	–0.6	12,457	78.4	–0.4	5,860	76.8	–1.4	10,902
✓	✓		50	76.6	–1.4	11,005	75.4	–3.4	4,416	76.4	–1.8	9,250
✓	✓	✓	50	75.2	–2.8	8,890	74.0	–4.8	3,296	75.8	–2.4	7,510
✓			100	78.0	0.0	13,687	78.6	–0.2	6,041	77.4	–0.8	12,360
✓	✓		100	77.2	–0.8	12,433	75.0	–3.8	4,570	77.4	–0.8	10,702
✓	✓	✓	100	74.6	–3.4	10,088	73.4	–5.4	3,405	74.8	–3.4	8,733
✓			150	**78.4**	**0.4**	**14,863**	78.2	–0.6	6,167	77.4	–0.8	13,931
✓	✓		150	78.0	0.0	13,520	76.4	–2.4	4,717	77.4	–0.8	11,804
✓	✓	✓	150	75.0	–3.0	11,021	73.2	–5.6	3,519	75.2	–3.0	9,682
✓			200	78.4	0.4	15,749	78.0	–0.8	6,359	78.0	–0.2	14,314
✓	✓		200	77.6	–0.4	14,260	75.2	–3.6	4,843	77.6	–0.6	12,557
✓	✓	✓	200	75.0	–3.0	11,704	72.4	–6.4	3,620	74.8	–3.4	10,382

(a) SVM classifier

Approach				**all-typed**			**affix+punct**			**all-untyped**		
D	PM	NE	HFW	Acc	+	N	Acc	+	N	Acc	+	N
Baseline				73.4		10,859	**75.4**		**6,296**	74.2		9,258
✓			0	73.8	0.4	9,761	75.0	–0.4	5,503	74.4	0.2	8,143
✓	✓		0	73.6	0.2	8,430	74.0	–1.4	4,171	73.2	–1.0	6,648
✓		✓	0	71.6	–1.8	7,606	72.6	–2.8	4,187	70.8	–3.4	6,364
✓	✓	✓	0	70.2	–3.2	6,651	71.8	–3.6	3,087	70.8	–3.4	5,239
✓			50	73.2	–0.2	12,457	74.4	–1.0	5,860	74.4	0.2	10,902
✓	✓		50	73.6	0.2	11,005	74.0	–1.4	4,416	73.6	–0.6	9,250
✓	✓	✓	50	70.6	–2.8	8,890	71.4	–4.0	3,296	70.6	–3.6	7,510
✓			100	74.6	1.2	13,687	75.0	–0.4	6,041	73.6	–0.6	12,360
✓	✓		100	74.6	1.2	12,433	74.4	–1.0	4,570	73.6	–0.6	10,702
✓	✓	✓	100	70.4	–3.0	10,088	71.0	–4.4	3,405	69.8	–4.4	8,733
✓			150	**75.0**	**1.6**	**14,863**	75.2	–0.2	6,167	**74.8**	**0.6**	**13,931**
✓	✓		150	75.0	1.6	13,520	74.2	–1.2	4,717	73.8	–0.4	11,804
✓	✓	✓	150	70.4	–3.0	11,021	71.2	–4.2	3,519	69.8	–4.4	9,682
✓			200	74.6	1.2	15,749	74.4	–1.0	6,359	74.8	0.6	14,314
✓	✓		200	73.8	0.4	14,260	74.2	–1.2	4,843	74.2	0.0	12,557
✓	✓	✓	200	70.2	–3.2	11,704	71.8	–3.6	3,620	70.4	–3.8	10,382

(b) MNB classifier

76.2% and 73.6% on the CCAT_10 test corpus, and 46.0% and 55.0% on the Guardian test corpus using SVM and MNB classifiers respectively. Character 3-gram features outperformed the BoW approach on both corpora for both classifiers by 1.8%–6.5%; see Tables 3 and 4.

Table 4. Accuracy results on the Guardian corpus after applying the proposed preprocessing steps (following the notations of Table 3).

Approach				**all-typed**			**affix+punct**			**all-untyped**		
D	PM	NE	HFW	Acc	+	N	Acc	+	N	Acc	+	N
Baseline				50.0		6,903	52.3		3,779	52.5		5,728
✓			0	50.9	0.9	6,841	52.4	0.1	3,725	52.4	–0.1	5,656
✓	✓		0	50.4	0.4	6,267	52.9	0.6	3,151	52.3	–0.2	4,985
✓		✓	0	**54.1**	**4.1**	**6,202**	56.2	3.9	3,347	54.4	1.9	5,121
✓	✓	✓	0	53.9	3.9	5,629	56.7	4.4	2,775	**55.8**	**3.3**	**4,443**
✓	✓	✓	50	52.3	2.3	7,411	56.5	4.2	2,978	52.6	0.1	6,251
✓	✓	✓	100	50.8	0.8	8,056	56.8	4.5	3,070	50.4	–2.1	6,924
✓	✓	✓	150	51.1	1.1	8,325	**57.0**	**4.7**	**3,150**	51.1	–1.4	7,210
✓	✓	✓	200	49.4	–0.6	8,451	56.1	3.8	3,219	49.7	–2.8	7,346

(a) SVM classifier

Approach				**all-typed**			**affix+punct**			**all-untyped**		
D	PM	NE	HFW	Acc	+	N	Acc	+	N	Acc	+	N
Baseline				56.6		6,903	58.4		3,779	56.9		5,728
✓			0	57.3	0.7	6,841	58.0	–0.4	3,725	57.1	0.2	5,656
✓	✓		0	59.5	2.9	6,267	61.6	3.2	3,151	60.2	3.3	4,985
✓		✓	0	58.0	1.4	6,202	58.9	0.5	3,347	58.5	1.6	5,121
✓	✓	✓	0	**60.8**	**4.2**	**5,629**	**64.3**	**5.9**	**2,775**	**61.9**	**5.0**	**4,443**
✓	✓	✓	50	59.1	2.5	7,411	63.8	5.4	2,978	59.5	2.6	6,251
✓	✓	✓	100	58.5	1.9	8,056	63.0	4.6	3,070	58.3	1.4	6,924
✓	✓	✓	150	58.1	1.5	8,325	62.2	3.8	3,150	57.8	0.9	7,210
✓	✓	✓	200	57.4	0.8	8,451	63.4	5.0	3,219	57.3	0.4	7,346

(b) MNB classifier

5.2 Frequency threshold selection

So far, all character n-grams with at least five occurrences in the training corpus were considered, similar to Sapkota *et al.* [11]. However, the appropriate tuning of feature set size has proved to be of great importance in cross-topic AA [6]. In this study, we attempt to select the most appropriate frequency threshold based on grid search. In more detail, we examine the following frequency threshold values: 5, 10, 20, 50, 100, 150, 200, 300, 500 and select the one that provides the best 10-fold cross-validation result on the training corpus. In the Guardian corpus, we use the average 10-fold cross-validation accuracy over the 4 training corpora.

In this experiment, we used the best combination of pre-processing steps for each corpus, as described in the previous section. For CCAT_10, the pre-processing combination was the replacement of digits. According to 10-fold cross-validation on the training corpus, the selected frequency threshold in all cases was 100 or less. This managed to slightly improve the results on the test corpus by approximately 1% with respect to a fixed frequency threshold of 5 (detailed results are omitted due to lack of space).

Table 5. Accuracy (%) variation with respect to the minimum feature frequency, where 10FCV – 10-fold cross-validation results on the training corpus; test – on the test corpus. The selected settings according to maximum 10-fold cross-validation result on the training corpus are in boldface; the top accuracies in test corpus are in italics.

min. feature	**all-typed**			**affix+punct**			**all-untyped**		
frequency	10FCV	test	N	10FCV	test	N	10FCV	test	N
5 (baseline)	67.9	53.9	5,629	71.7	56.7	2,775	68.4	55.8	4,443
10	69.1	55.9	4,372	73.2	59.6	2,144	71.3	57.1	3,573
20	71.5	59.8	3,249	**75.1**	**62.8**	**1,582**	73.0	60.1	2,779
50	73.1	61.5	1,956	73.4	65.2	964	72.7	62.1	1,821
100	**74.5**	**61.6**	**1,183**	74.9	***66.5***	602	71.0	61.2	1,176
150	74.1	60.9	809	74.4	65.0	436	73.2	***62.6***	856
200	74.2	62.7	604	74.2	65.9	341	**75.0**	**62.2**	**661**
300	74.4	***63.8***	386	73.5	65.2	238	73.8	62.4	437
500	67.5	60.9	205	68.9	63.3	141	70.0	60.9	227

(a) SVM classifier

min. feature	**all-typed**			**affix+punct**			**all-untyped**		
frequency	10FCV	test	N	10FCV	test	N	10FCV	test	N
5 (baseline)	71.7	60.8	5,629	72.6	64.3	2,775	71.6	61.9	4,443
10	73.3	63.6	4,370	74.5	67.3	2,144	73.2	64.8	3,573
20	75.6	66.4	3,249	77.6	68.6	1,582	75.1	66.7	2,779
50	76.4	66.6	1,956	77.8	70.3	964	75.4	67.1	1,821
100	**77.0**	**67.9**	**1,183**	**78.8**	**72.3**	**602**	76.2	67.6	1,176
150	76.9	68.8	809	77.1	72.3	436	77.5	69.0	856
200	76.7	***70.5***	604	78.3	***73.2***	341	**78.1**	**69.6**	**661**
300	76.4	70.4	386	76.7	72.9	238	77.4	***70.1***	437
500	73.6	69.2	205	77.4	71.3	141	73.5	68.1	227

(b) MNB classifier

For the cross-topic experiments, we applied the combination of pre-processing steps that are useful in this corpus: replacement of digits, NEs, and splitting PMs. Table 5 shows the performance results (both 10-fold cross-validation accuracy on the training corpus and the corresponding results on the test corpus) for different frequency threshold values using either SVM or MNB classifiers. We compare the obtained results with the fixed threshold of 5 used in the previous experiments, as well as by Sapkota *et al.* [11].

In general, any frequency threshold higher than the baseline produces better results. The best settings found by 10-fold cross-validation on the training set do not correspond to the best possible results on the test set. However, they provide a near-optimal estimation, regardless of the classifier. It is also remarkable that the settings that achieve the best performance correspond to relatively high frequency thresholds (about 100–200), much higher than the ones found for the CCAT_10 corpus. This means that low frequency features should be avoided under cross-topic conditions, since they provide confusing information to the classifiers. Note that these high values of frequency threshold drastically reduced feature set sizes (around 80% reduction in most of the cases). The selection of an appropriate frequency threshold, using only the training data, allowed us to improve the accuracy in cross-topic AA almost by around 10% for each of the models. The increase in performance is even higher if we compare the result of this experiment with the original approach of Sapkota *et al.* [11].

6 Conclusions

It is well-known in AA research that character n-grams provide very effective features. They are able to capture many nuances of writing style, and they are very simple to be extracted from any text in any language. However, it is not clear how thematic information can be appropriately reduced when a character n-gram representation is used. In this paper, we showed that it is possible to notably enhance the performance of AA under realistic cross-topic conditions by performing simple pre-processing steps that discard topic-dependent information from texts. It seems that the replacement of digits, punctuation marks splitting, and the replacement of named entities before the extraction of character n-grams improve the results in cross-topic AA when these steps applied separately or even better when they are combined.

On the other hand, the replacement of highly frequent words with distinct symbols does not seem to be helpful. When applied to a single-topic corpus, where authors tend to deal with specific topics, and therefore, they can be distinguished by a combination of their personal style and thematic preferences, the proposed pre-processing steps do not seem so effective.

We also showed that the appropriate selection of the dimensionality of the representation is crucial for cross-topic AA, and that it is possible to significantly improve the accuracy results by fine tuning the frequency threshold based on the training data. In cross-topic conditions, high frequency threshold values were found the most effective. It indicates that least frequent n-grams, associated with topic-specific information, should be avoided. Our approach improves the cross-topic AA accuracy by more than 10% over the baseline for the examined classifiers, while drastically reducing the size of the feature set by 80%.

Our experiments confirmed the conclusion by Sapkota *et al.* [11] that the model trained on affix and punctuation character n-grams is more effective than the models trained on all the features. This is consistent regardless of the particular learning algorithm, with or without performing pre-processing steps. It is also interesting that based on features of **affix+punct** we achieved the best increase in AA performance in cross-topic conditions.

Another interesting observation is that MNB classifier performs better than SVM under cross-topic conditions, whereas SVM is better for single-topic conditions. Further investigation is required to verify this conclusion.

One of the directions for future work would be to conduct experiments using longer character n-grams in single and cross-topic conditions and select an appropriate n-gram order. It would also be interesting to examine the effect of the proposed method to word level features, such as syntactic n-grams [33]. Moreover, the combination of different feature types should be examined since this usually improves the performance of the attribution models [23, 34]. In addition, the robustness of our approach under cross-genre conditions, when training and test corpora belong to different genres (e.g., scientific papers and e-mail messages) will be tested.

Acknowledgments

This work was partially supported by the Mexican Government (CONACYT projects 240844, SNI, COFAA-IPN, SIP-IPN 20161947, 20161958, 20162204, 20162064, 20171813, 20171344, and 20172008).

References

1. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society For Information Science and Technology **60** (2009) 538–556
2. Abbasi, A., Chen, H.: Applying authorship analysis to extremist-group Web forum messages. IEEE Intelligent Systems **20** (2005) 67–75
3. Chaski, C.E.: Who's at the keyboard? Authorship attribution in digital evidence investigations. International Journal of Digital Evidence **4** (2005) 1–13
4. Coulthard, M.: On admissible linguistic evidence. Journal of Law & Policy **21** (2013) 441–466
5. Koppel, M., Seidman, S.: Automatically identifying pseudepigraphic texts. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, (EMNLP '13). (2013) 1449–1454
6. Stamatatos, E.: On the robustness of authorship attribution based on character n-gram features. Journal of Law & Policy **21** (2013) 427–439
7. Luyckx, K., Daelemans, W.: Authorship attribution and verification with many authors and limited data. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08). (2008) 513–520
8. Houvardas, J., Stamatatos, E.: N-gram feature selection for authorship identification. In: Proceedings of Artificial Intelligence: Methodologies, Systems, and Applications (AIMSA '06). (2006) 77–86

9. Kestemont, M.: Function words in authorship attribution. From black magic to theory? In: Proceedings of the 3rd Workshop on Computational Linguistics for Literature (EACL '14). (2014) 59–66
10. Daelemans, W.: Explanation in computational stylometry. In: Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing '13). (2013) 451–462
11. Sapkota, U., Bethard, S., Montes-y-Gómez, M., Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. In: Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL: Human Language Technologies (NAACL-HLT '15). (2015) 93–102
12. Hedegaard, S., Simonsen, J.G.: Lost in translation: Authorship attribution using frame semantics. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT '11). (2011) 65–70
13. Schwartz, R., Tsur, O., Rappoport, A., Koppel, M.: Authorship attribution of micro-messages. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP '13). (2013) 1880–1891
14. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications **41** (2014) 853–860
15. Gómez-Adorno, H., Sidorov, G., Pinto, D., Markov, I.: A graph based authorship identification approach. In: Working Notes Papers of the CLEF 2015 Evaluation Labs (CLEF '15). Volume 1391., CEUR (2015)
16. Grieve, J.: Quantitative authorship attribution: An evaluation of techniques. Literary and Linguistic Computing **22** (2007) 251–270
17. Stamatatos, E.: Author identification using imbalanced and limited training texts. In: Proceedings of the 18th International Conference on Database and Expert Systems Applications (DEXA '07). (2007) 237–241
18. Escalante, H.J., Solorio, T., Montes-y-Gómez, M.: 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 (HLT '11). (2011) 288–298
19. Sapkota, U., Solorio, T., Montes-y-Gómez, M., Bethard, S., Rosso, P.: Cross-topic authorship attribution: Will out-of-topic data help? In: Proceedings of the 25th International Conference on Computational Linguistics (COLING '14). (2014) 1228–1237
20. Khmelev, D.V., Teahan, W.J.: A repetition based measure for verification of text collections and for text categorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '03). (2003) 104–110
21. Marton, Y., Wu, N., Hellerstein, L.: On compression-based text classification. In: Proceedings of the 27th European conference on Advances in Information Retrieval Research (ECIR '05). (2005) 300–314
22. Peng, F., Schuurmans, D., Keselj, V., Wang, S.: Language independent authorship attribution with character level n-grams. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL '03). (2003) 267–274
23. Qian, T., Liu, B., Chen, L., Peng, Z.: Tri-training for authorship attribution with limited training data. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14). (2014) 345–351

24. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic text categorization in terms of genre and author. Computational Linguistics **26** (2000) 471–495
25. de Vel, O.Y., Anderson, A., Corney, M., Mohay, G.M.: Mining email content for author identification forensics. SIGMOD Record **30** (2001) 55–64
26. Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. The Journal of Machine Learning Research **8** (2007) 1261–1276
27. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B.: Overview of the author identification task at PAN 2015. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum. (2015)
28. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research **5** (2004) 361–397
29. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05). (2005) 363–370
30. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations **11** (2009) 10–18
31. Gómez-Adorno, H., Markov, I., Sidorov, G., Posadas-Durán, J., Sanchez-Perez, M.A., Chanona-Hernandez, L.: Improving feature representation based on a neural network for author profiling in social media texts. Computational Intelligence and Neuroscience **2016** (2016) 13 pages
32. Kibriya, A.M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive Bayes for text categorization revisited. In: Proceedings of the 17th Australian joint conference on Advances in Artificial Intelligence (AI '04). (2005) 488–499
33. Sidorov, G., Gómez-Adorno, H., Markov, I., Pinto, D., Loya, N.: Computing text similarity using tree edit distance. In: Proceedings of the Annual Conference of the North American Fuzzy Information processing Society (NAFIPS '15) and 5th World Conference on Soft Computing. (2015) 1–4
34. Markov, I., Gómez-Adorno, H., Sidorov, G., Gelbukh, A.: Adapting cross-genre author profiling to language and corpus. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. Volume 1609 of CEUR Workshop Proceedings., CLEF and CEUR-WS.org (2016) 947–955