Authorship Attribution using Syntactic Dependencies

Authorship Attribution using Syntactic Dependencies

Juan SOLER-COMPANY [a,1] and Leo WANNER [a,b,2]
[a] *NLP Group, Pompeu Fabra University*
[b] *Catalan Institute for Research and Advanced Studies (ICREA)*

Abstract. Authorship attribution deals with the prediction of the author of a (usually written) discourse. This is of high relevance to a number of applications, including plagiarism detection, authenticity verification and deception detection. So far, most of the state-of-the-art approaches to author attribution rely mainly upon lexical and token (sequence) distribution features. But this means to neglect numerous linguistic studies that clearly indicate the high relevance of syntactic features to the characterization of a personal style of an author. We show in an experiment with 26 authors that indeed the use of syntactic features helps us to achieve a $>$77% an accuracy.

Keywords. Author Profiling, Author Identification, Text Classification

1. Introduction

It is an attractive idea to be able to identify, confirm or characterize the author of a text or of a text collection. There is a large range of potential practical applications: from forensic investigation to marketing studies. If we focus on forensics, these research fields could be used to detect potential pedophiles in chat rooms, crime confessor authenticity verification or to even monitor terrorist activity in forum/chat/email/dark web data. In the field of marketing studies, author profiling can be used to analyze customer content and feedback data. Even in literary studies the question of authorship is an issue; see, e.g., [8] for an analysis of Shakespeare's and Fletcher's works from this perspective.

One of the crucial tasks in author identification is feature engineering: the process of selecting the features that describe best the style of an individual author and thus help distinguish this author from others. Some of the commonly used features in the state-of-the-art are word n-gram frequency, function words usage, word length distribution, digit frequency, etc. Nearly all of these features draw upon specific tokens and their distribution within a writing. However, these features are not sufficient to capture a writing style in its entireness. Thus, it is well-known from linguistics and philology that syntax is an important aspect of the writing style of an author [3]. But only a small number of works incorporate syntactic features into author identification experiments, and when they do they usually draw upon information in constituency parse trees (e.g., frequency of NPs

[1] Email: juan.soler@upf.edu
[2] Email: leo.wanner@upf.edu

containing a noun in singular, …); see, e.g., [10]. Punctuation marks, function words, passive count or Parts-of-speech are also included in this feature category in some cases, see e.g., [17].

In what follows, we introduce a set of features mainly of syntactic dependency and structural nature and prove that these features are able to distinguish between 26 different authors with high accuracy. Using several state-of-the-art approaches as baselines, we furthermore show that these features lead to a better performance.

The rest of the paper is organized as follows. Section 2 reviews briefly the related work in the field, Section 3 introduces the set of features we use in our experiments, Section 4 presents the experimental setup, Section 5 contains the results we obtain with our features and their discussion and Section 6 draws some conclusions and outlines future work.

2. Related Work

One of the first proposals that implemented data-driven author identification was [9], who attempted to clarify the authorship of the Federalist Papers drawing upon function words and Naïve Bayes classification. Years after, this problem was retaken by [6] using high-frequency words and genetic algorithms. Character-based and word-based features such as word or character n-grams are used frequently in the field; see, e.g., [11]. Function words are also used to distinguish between authors, see [19].

Some approaches try to avoid features of the kinds mentioned above. For instance, [12] use context-free grammars, [14] focus on style markers and [1] use the frequencies of syntactic rewrite rules to distinguish between authors.

A combination of shallow linguistic (trigrams, function word frequencies) and deep linguistic features (context-free grammar production frequencies and features derived from semantic graphs) are used in [5]. Syntactic n-grams[3] are used to distinguish between three literary authors using 39 documents retrieved from the Gutenberg project[4] in [13].

An overview of an annual authorship attribution shared task can be found in [16]. In the task, texts of four languages and different genres are given. The goal is, given a set of documents by the same author along with one document with unknown author, to determine if this document was written by the same author or not. More information on the current state-of-the-art in authorship attribution can be found in reviews such as [15].

3. Capturing the Style

As already mentioned in Sections 1 and 2, there is a tendency in the field of author identification to use content-dependent features such as word or character n-grams. However, the use of these features is not scalable and significantly depends on the domain of the used corpora.

On the other side, theoretical studies argue that syntax is an important means to model style. For instance, in [4] the authors identify, e.g., (adjectival) premodification,

[3] Syntactic n-grams are defined as linearized paths of length n in syntactic trees.

[4] https://www.gutenberg.org

(PP) postmodification, subordination, coordination, etc. as style-influencing elements. Therefore, in our approach, syntactic features play an important role. More generally, we rely mainly on linguistic structure rather than on lexis. The features that we extracted and used to predict authors can be divided into character-, word-, sentence-, dictionary-, and syntax-based features. The first four have been extracted using Python and its Natural Language Toolkit (NLTK). To obtain dependency trees from which the syntactic features are then extracted, [2]' statistical dependency parser is used.

The advantage of this approach is that given the assumption that there is an underlying unconscious choice of style by the authors in small details such as the usage of punctuation marks or the used syntactic dependencies between words, our model will be effective even if the author tries to change its writing.

Character-based features capture the usage of punctuation marks: commas, periods, parenthesis, exclamation and question marks, hyphens, colons, semi-colons, quotations and other symbols such as the percentage sign, the ampersand, the plus sign and the dollar sign. These features are calculated as the ratio of the number of apparitions of said characters in each instance and the total number of characters. The percentage of upper cased characters and the usage of numbers is also computed.

Word and Sentence-based features are composed by the mean number of characters per word, the standard deviation in word length and the difference between the longest and shortest word, the vocabulary richness and the usage of acronyms, stop words and first person pronouns, both singular and plural (as ratios of these values and the total number of words in the text). The mean number of words per sentence is also calculated as well as the standard deviation in sentence length and the difference between the longest and shortest sentence.

Dictionary-based features use 6 different dictionaries: interjections, discourse markers, positive/negative words, abbreviations and curse words. The ratios of found words for each dictionary and the total number of words of the text are used as features. The positive and negative word dictionaries are sentiment analysis lexicons used in [7], which are publicly available. The other ones were compiled for this work.

Syntactic features constitute the largest group of our features. They consist of the frequencies of individual dependency relations in the dependency trees of the sentences in the post as well as the mean width and depth of the dependency trees. The individual dependency relations and their meaning can be found [18] and the frequency of these dependencies gives us valuable information about the structure of the texts. A clear example would be the dependencies that indicate subordinate and coordinate clauses (SUB and COORD); their appearance and their frequency help us understand the complexity of the discourse. The depth of the trees is defined as the longest path between the root and one of the leaves. The width is the maximum number of siblings that there are at some level of the tree. The depth and width of dependency trees can be interpreted as a measure of the complexity of the structure of the corresponding sentences.

4. Experimental Setup

The dataset that was used for our experiments is composed of 4836 journalistic posts of 26 authors from the blogs of the British newspapers *The Guardian'*, *The Independent*,

Table 1. Results with our features, compared with baseline features

Approach	Accuracy
Our Features	**77.65%**
Function word list 1 (FW1)	56.64%
Function word list 2 (FW2)	60.63%
Parts-of-Speech (PoS)	57.27%
Stopwords	59.35%
FW1 + PoS	65.77%
FW2 + PoS	65.92%
Stopwords + PoS	66.29%

and *The Daily Mail*. For each of the 26 authors, between 80 and 250 posts are available. Each post of the dataset is tagged with the name of its author and the features listed in Section 3 are extracted.

For the classification process we use Weka's implementation of Bagging (with Random Forests as base classifier), in a 10-fold cross-validation classification. In other words, we consider author identification as a multi (26)-class classification problem.

To contrast the performance achieved with our features against the performance achieved with some of the features discussed in state-of-the-art literature, we implemented seven different baselines. For all baselines, we used the same classifier used with our model as well as the same dataset. The first two baselines use normalized frequencies of function words, used in [19]. Since the list of function words that was used in the original is not available, we used the lists available at *http://myweb.tiscali.co.uk/wordscape/museum/funcword.html* ("function word list 1" in Table 1 below) and *http://www.sequencepublishing.com/academic.html* ("function word list 2"). The next two baselines use normalized frequencies of parts-of-speech and normalized frequencies of stop words, respectively.[5]. The last three baselines use a combination of the features of the first four baselines.

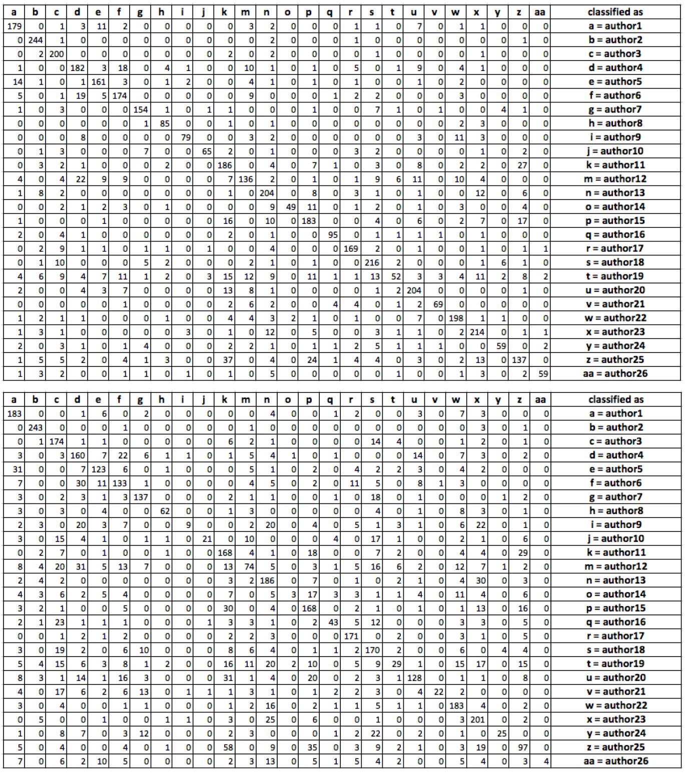
5. Results and Discussion

Table 1 shows the performance of Weka's Bagging algorithm with our features, compared to the performance with the baseline feature sets.

Table 1 shows that the classifier trained on our features outperforms the baselines by a wide margin. The obvious explanation for this is that the accuracy of an approach that uses function words depends heavily on the choice of the words in the precompiled list. Adding Part-of-speech improves the baseline's accuracy, but structural features still outperform it. To further compare the performance of our features with the performance of FW2, see Figure 1. The confusion matrixes are very illustrative in that they show where the classifier erred and what the cause for this was.

The function word list approach works reasonably well in cases in which 200 or more texts from one author are available for training. With authors that have less than 100 texts, the results are much worse. This behavior can be observed in several cases. For example, in the case of class "i", our model predicts correctly 79 instances of the

[5] The list of stop words that was used is available in Python's Natural Language Toolkit (NLTK).



a	b	c	d	e	f	g	h	i	j	k	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	classified as
179	0	1	3	11	2	0	0	0	0	0	3	2	0	0	0	1	1	0	7	0	1	1	0	0	0	a = author1
0	244	1	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	1	0	b = author2
0	2	200	0	0	0	0	0	0	0	2	0	2	0	0	0	0	1	0	0	0	0	1	0	0	0	c = author3
1	0	0	182	3	18	0	4	1	0	0	10	1	0	1	0	5	0	1	9	0	4	1	0	0	0	d = author4
14	1	0	1	161	3	0	1	2	0	0	4	1	0	1	0	1	0	0	1	0	2	0	0	0	0	e = author5
5	0	1	19	5	174	0	0	0	0	0	9	0	0	0	1	2	2	0	0	0	3	0	0	0	0	f = author6
1	0	3	0	0	0	154	1	0	1	1	0	0	0	1	0	0	7	1	0	1	0	0	4	1	0	g = author7
0	0	0	0	0	0	1	85	0	0	1	0	1	0	0	0	0	0	0	0	0	2	3	0	0	0	h = author8
0	0	0	8	0	0	0	0	79	0	0	3	2	0	0	0	0	0	0	3	0	11	3	0	0	0	i = author9
0	1	3	0	0	0	7	0	0	65	2	0	1	0	0	0	3	2	0	0	0	0	1	0	2	0	j = author10
0	3	2	1	0	0	0	2	0	0	186	0	4	0	7	1	0	3	0	8	0	2	2	0	27	0	k = author11
4	0	4	22	9	9	0	0	0	0	7	136	2	0	1	0	1	9	6	11	0	10	4	0	0	0	m = author12
1	8	2	0	0	0	0	0	0	0	1	0	204	0	8	0	3	1	0	1	0	0	12	0	6	0	n = author13
0	0	2	1	2	3	0	1	0	0	0	0	9	49	11	0	1	2	0	1	0	3	0	0	4	0	o = author14
1	0	0	0	1	0	0	0	0	0	16	0	10	0	183	0	0	4	0	6	0	2	7	0	17	0	p = author15
2	0	4	1	0	0	0	0	0	0	2	0	1	0	0	95	0	1	1	1	1	0	1	0	0	0	q = author16
0	2	9	1	1	0	1	1	0	1	0	0	4	0	0	0	169	2	0	1	0	0	1	0	1	1	r = author17
0	1	10	0	0	0	5	2	0	0	0	2	1	0	1	0	0	216	2	0	0	0	1	6	1	0	s = author18
4	6	9	4	7	11	1	2	0	3	15	12	9	0	11	1	1	13	52	3	3	4	11	2	8	2	t = author19
2	0	0	4	3	7	0	0	0	0	13	8	1	0	0	0	0	1	2	204	0	0	0	0	1	0	u = author20
0	0	0	0	0	1	0	0	0	0	2	6	2	0	0	4	4	0	1	2	69	0	0	0	0	0	v = author21
1	2	1	1	0	0	0	1	0	0	4	4	3	2	1	0	1	0	0	7	0	198	1	1	0	0	w = author22
1	3	1	0	0	0	0	0	3	0	1	0	12	0	5	0	0	3	1	1	0	2	214	0	1	1	x = author23
2	0	3	1	0	1	4	0	0	0	2	2	1	0	1	1	2	5	1	1	1	0	0	59	0	2	y = author24
1	5	5	2	0	4	1	3	0	0	37	0	4	0	24	1	4	4	0	3	0	2	13	0	137	0	z = author25
1	3	2	0	0	1	1	0	1	0	1	0	5	0	0	0	0	0	1	0	0	1	3	0	2	59	aa = author26

a	b	c	d	e	f	g	h	i	j	k	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	classified as
183	0	0	1	6	0	2	0	0	0	0	0	4	0	0	1	2	0	0	3	0	7	3	0	0	0	a = author1
0	243	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	1	0	b = author2
0	1	174	1	1	0	0	0	0	0	6	2	1	0	0	0	0	14	4	0	0	1	2	0	1	0	c = author3
3	0	3	160	7	22	6	1	1	0	1	5	4	1	0	1	0	0	0	14	0	7	3	0	2	0	d = author4
31	0	0	7	123	6	0	1	0	0	0	5	1	0	2	0	4	2	2	3	0	4	2	0	0	0	e = author5
7	0	0	30	11	133	1	0	0	0	0	4	5	0	2	0	11	5	0	8	1	3	0	0	0	0	f = author6
3	0	2	3	1	3	137	0	0	0	2	1	1	0	0	1	0	18	0	1	0	0	0	1	2	0	g = author7
3	0	3	0	4	0	0	62	0	0	1	3	0	0	0	0	0	4	0	1	0	8	3	0	1	0	h = author8
2	3	0	20	3	7	0	0	9	0	0	2	20	0	4	0	5	1	3	1	0	6	22	0	1	0	i = author9
3	0	15	4	1	0	1	1	0	21	0	10	0	0	0	4	0	17	1	0	0	2	1	0	6	0	j = author10
0	2	7	0	1	0	0	1	0	0	168	4	1	0	18	0	0	7	2	0	0	4	4	0	29	0	k = author11
8	4	20	31	5	13	7	0	0	0	13	74	5	0	3	1	5	16	6	2	0	12	7	1	2	0	m = author12
2	4	2	0	0	0	0	0	0	0	3	2	186	0	7	0	1	0	2	1	0	4	30	0	3	0	n = author13
4	3	6	2	5	4	0	0	0	0	7	0	5	3	17	3	3	1	1	4	0	11	4	0	6	0	o = author14
3	2	1	0	0	5	0	0	0	0	30	0	4	0	168	0	2	1	0	1	0	1	13	0	16	0	p = author15
2	1	23	1	1	1	0	0	0	1	3	3	1	0	2	43	5	12	0	0	0	3	3	0	5	0	q = author16
0	0	1	2	1	2	0	0	0	0	2	2	3	0	0	0	171	0	2	0	0	3	1	0	5	0	r = author17
3	0	19	2	0	6	10	0	0	0	8	6	4	0	1	1	2	170	2	0	0	6	0	4	4	0	s = author18
5	4	15	6	3	8	1	2	0	0	16	11	20	2	10	0	5	9	29	1	0	15	17	0	15	0	t = author19
8	3	1	14	1	16	3	0	0	0	31	1	4	0	20	0	2	3	1	128	0	1	1	0	8	0	u = author20
4	0	17	6	2	6	13	0	1	1	1	3	1	0	1	2	2	3	0	4	22	2	0	0	0	0	v = author21
3	0	4	0	0	1	1	0	0	0	1	2	16	0	2	1	1	5	1	1	0	183	4	0	2	0	w = author22
0	5	0	0	1	0	0	1	1	0	3	0	25	0	6	0	0	1	0	0	0	3	201	0	2	0	x = author23
1	0	8	7	0	3	12	0	0	0	2	3	0	0	0	1	2	22	0	2	0	1	0	25	0	0	y = author24
5	0	4	0	0	4	0	1	0	0	58	0	9	0	35	0	3	9	2	1	0	3	19	0	97	0	z = author25
7	0	6	2	10	5	0	0	0	0	2	3	13	0	5	1	5	4	2	0	0	5	4	0	3	4	aa = author26

Figure 1. Confusion matrixes of our model (top matrix) with the FW2 baseline (bottom matrix)

class, while the baseline approach predicts only 9 of them correctly. The situation is similar with the classes “j”, “o”, “y”, “aa”, and “v”. That is, features that incorporate syntactic phenomena lead to a more accurate author identification. The use of function words is partially also stylistically motivated. But partially their use is purely grammatical (as, e.g., in the case of governed prepositions). Therefore, a larger training dataset is necessary to adequately cover the stylistic use of function words.

6. Conclusions and Future Work

We have shown that a relatively small set of features composed mainly by syntactic dependency features is very competitive in the author attribution task. The accuracy achieved in a 26-class supervised machine learning experiment outperformed the baselines by a large margin. This is quite promising and could have great impact in the world of plagiarism detection.

In the future, we plan to implement semi-supervised and unsupervised models. We expect them to be useful for forensics, where clean, labeled data is scarce. We also want to study the influence of sexual orientation in the writing style of authors.

References

[1] H Baayen, H van Halteren, and F Tweedie, 'Outside the cave of shadows: using syntactic annotation to enhance authorship attribution', *Literary and Linguistic Computing*, **11**(3), 121–132, (1996).

[2] Bernd Bohnet, 'Very high accuracy and fast dependency parsing is not a contradiction', in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, COLING '10, pp. 89–97, Stroudsburg, PA, USA, (2010). Association for Computational Linguistics.

[3] David Crystal and Derek Davy, *Investigating English Style*, Longman Group Ltd., London, 1969.

[4] C. DiMarco and G. Hirst, 'A computational theory of goal-directed style in syntax', *Computational Linguistics*, **19**(3), 451–499, (1993).

[5] Michael Gamon, 'Linguistic correlates of style: authorship classification with deep linguistic analysis features'. International Conference on Computational Linguistics, (August 2004).

[6] D. I. Holmes and R. S. Forsynth, 'The federalist revisited: New directions in authorship attribution', *Literary and Linguistic Computing*, **10**(2), 111–127, (1995).

[7] Minqing Hu and Bing Liu, 'Mining and summarizing customer reviews', in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 168–177, New York, NY, USA, (2004). ACM.

[8] Robert AJ Matthews and Thomas VN Merriam, 'Neural computation in stylometry i: An application to the works of shakespeare and fletcher', *Literary and Linguistic Computing*, **8**(4), 203–209, (1993).

[9] Frederick Mosteller and David L. Wallace, 'Inference in an authorship problem', *Journal of the American Statistical Association*, **58**(302), 275–309, (1963).

[10] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song, 'On the feasibility of internet-scale author identification', in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pp. 300–314, Washington, DC, USA, (2012). IEEE Computer Society.

[11] Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj, 'Language independent authorship attribution using character level language models', in *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pp. 267–274, Stroudsburg, PA, USA, (2003). Association for Computational Linguistics.

[12] Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney, 'Authorship attribution using probabilistic context-free grammars', in *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pp. 38–42, Stroudsburg, PA, USA, (2010). Association for Computational Linguistics.

[13] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernndez, 'Syntactic n-grams as machine learning features for natural language processing', *Expert Systems with Applications*, **41**(3), 853 – 860, (2014). Methods and Applications of Artificial and Computational Intelligence.

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

[15] Efstathios Stamatatos, 'A survey of modern authorship attribution methods', *Journal of the American Society for Information Science and Technology*, **60**(3), 538–556, (2009).

[16] Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A Sanchez-Perez, and Alberto Barrón-Cedeño, 'Overview of the author identification task at pan 2014', *analysis*, **13**, 31, (2014).

[17] Efstathios Stamatatos, George Kokkinakis, and Nikos Fakotakis, 'Automatic text categorization in terms of genre and author', *Comput. Linguist.*, **26**(4), 471–495, (December 2000).

[18] Mihai Surdeanu, Richard Johansson, Adam Meyers, Llu'ıs Màrquez, and Joakim Nivre, 'The conll-2008 shared task on joing parsing of syntactic and semantic dependencies', in *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pp. 159–177, Stroudsburg, PA, USA, (2008). Association for Computational Linguistics.

[19] Ying Zhao and Justin Zobel, 'Effective and scalable authorship attribution using function words', in *Information Retrieval Technology*, eds., GaryGeunbae Lee, Akio Yamada, Helen Meng, and SungHyon Myaeng, volume 3689 of *Lecture Notes in Computer Science*, 174–189, Springer Berlin Heidelberg, (2005).