Unsupervised Author Identification and Characterization

Stefano Ferilli[1,2(✉)], Domenico Redavid[3], and Floriana Esposito[1,2]

[1] Dipartimento di Informatica, Università di Bari, Bari, Italy
{stefano.ferilli,floriana.esposito}@uniba.it
[2] Centro Interdipartimentale per la Logica e sue Applicazioni, Università di Bari, Bari, Italy
[3] Artificial Brain S.r.l., Bari, Italy
redavid@abrain.it

Abstract. Author identification is a hot topic, especially in the Internet age. Following our previous work in which we proposed a novel approach to this problem, based on relational representations that take into account the structure of sentences, here we present a tool that computes and visualizes a numerical and graphical characterization of the authors/texts based on several linguistic features. This tool, that extends a previous language analysis tool, is the ideal complement to the author identification technique, that is based on a clustering procedure whose outcomes (i.e., the authors' models) are not human-readable. Both approaches are unsupervised, which allows them to tackle problems to which other state-of-the-art systems are not applicable.

1 Introduction

Especially in the last decades, electronic publishing facilities and the spread of the Internet have made the writing activity faster and easier. In this landscape, the huge amount of available documents and writers caused an increase in the number of plagiarism cases, and a much more difficult identification of these cases with traditional (mainly manual) approaches. Although clearly defined, the authorship attribution task is amenable to several variations. We are interested in the following setting: given a small set (no more than 10, possibly just one) of 'known' documents written by a single person, and a 'questioned' document, determine whether the latter was written by the same person who wrote the former. For the sake of clarity, from now on we will refer to the known author (and the corresponding texts) used as a reference as the *base*, and to the unknown author (and the corresponding text) that must be classified as *target*.

Capturing the peculiarities of an author is not trivial, because it requires a deep understanding of many aspects of his behavior. Traditional propositional approaches are not able to seize the whole complex network of relationships between events, objects or a combination thereof that implicitly or explicitly underly a written text. Conversely, relational approaches may provide additional representational power. Adopting this perspective, we express the unstructured

D. Calvanese et al. (Eds.): IRCDL 2015, CCIS 612, pp. 129–141, 2016.
DOI: 10.1007/978-3-319-41938-1_14

texts in natural language by complex patterns on which automatic (relational) techniques can be applied. Exploiting such patterns, the author's style can be modeled, and the model can be in turn used in order to classify a new document and decide whether it was likely written by the same author or not. Since the modeling approach is based on clustering techniques whose outcomes are not human-readable, in this paper we provide a tool that allows to have an insight into the texts under comparison to comfortably check their similarities and differences. This insight may be considered as a kind of (statistical) characterization of the texts themselves.

After describing related works in the next section, the proposed approach to author identification is described and evaluated in Sect. 3. Then, Sect. 4 presents the characterization tool. Lastly, we conclude with some considerations and future works.

2 Related Work

The last decade has witnessed the flourishing of a significant amount of research conducted on Author Identification. Researchers focused on different properties of texts, the so-called *style markers*, to quantify the writing style under different labels and criteria. Generally speaking, the features can be divided into five main groups. Lexical and character features consider a text as a mere sequence of word-tokens (as in [1,12]) or characters (as in [15]), respectively. Syntactic features are based on the idea that authors tend to unconsciously use similar syntactic patterns when writing. Therefore, some approaches (e.g., in [2,13]) exploit information such as PoS-tags, sentence and phrase structures to model the authors. These approaches are affected by two major drawbacks: the former is the need of robust and accurate NLP tools to perform syntactic analysis of texts; the latter is the huge amount of extracted features they require. Semantic approaches, such as the one in [9], rely on semantic dependencies obtained by external resources, such as taxonomies or thesauri. Finally, there are special-purpose approaches, that define application-specific measures to better represent the text style in a given domain. Such measures are based on the use of greetings and farewells in the messages, types of signatures, use of indentation, paragraph length, and so on [7].

All these approaches use a flat (vectorial) representation of the documents. Even syntactic and semantic approaches, such as the one in [14], subsequently create new flat features, losing in this way the relations embedded in the original texts. A different approach that preserves the phrase structure is presented in [10], based on probabilistic context-free grammars (PCFG), but it is practically not applicable in settings in which a small set of documents of only one author is available.

Differently from all of these works, the approach proposed in this paper aims at preserving the informative richness of textual data by extracting and exploiting complex patterns from such complex data.

3 Relational Author Identification

Text documents in Natural Language are a complex kind of data, because they have several (often hidden) connections to the culture, feelings and objectives of the authors, which are partly expressed by their writing style. The approach to Author Identification proposed in [3], that will be briefly summarized in the following, aims at exploiting as much as possible of this rich deal of information. It translates textual data into a structural description that tries to explicitly capture (part of) the complex patterns representing the author's style. These descriptions can be clustered, provided that a similarity measure for relational representations and a stopping criterion for the grouping procedure are available. Applying this procedure to both the base and the target text we build two corresponding models, and obtain a final classification as a result of the comparison between these two models. The underlying idea is that, assuming that each model describes a set of ways in which the author composes the sentences, if the writing habits expressed by the target model can be brought back to the base model, then one can conclude that the author is the same.

We borrowed the following pre-processing techniques, that transform the text into a more standard and machinable form, from ConNeKTion [6,11], a system for conceptual graph learning and exploitation that also provides a structured representation of the processed texts:

Collocation Extraction. Collocations are linguistic expressions consisting of two or more words that denote a compound concept whose meaning results from the specific composition of the constituent words.

Anaphora Resolution. Anaphora are references to concepts already cited in previous portions of a text, usually expressed by pronouns. So, to make a sentence autonomous, it is necessary to identify the referred concept and replace the pronoun by the explicit concept.

Parsing. A significant improvement in text understanding can be obtained by stepping up from the purely lexical level to the syntactic level. The syntactic relationships between subjects, verbs and (direct or indirect) objects in a sentence can be represented in a tree that reproduces its phrase structure.

Dependencies Extraction. Based on the parse tree of a sentence, a set of grammatical (typed) dependencies among the sentence components can be identified. These dependencies can be expressed as binary relations between pairs of words, the former of which represents the governor of the grammatical relation, and the latter its dependent. ConNeKTion can currently deal only with English, although the proposed strategy is general and applicable, in principle, to any language for which such dependencies can be extracted.

Term Normalization. Since word inflection is nearly irrelevant for identifying the underlying concepts expressed by terms, it is useful to select as a reference a normalized version of each word in the text. ConNeKTion uses lemmatization instead of stemming, which may allow to distinguish the grammatical role of the word and is more comfortable to read by humans.

These pre-processing steps allow to translate each sentence into a relational pattern, that is expressed as a Horn clause [8].

After obtaining a relational description for each sentence in the available documents (both base and target ones), the author identification procedure evaluates the similarity between all pairs of sentences using the similarity framework presented in [4]. In the resulting upper triangular matrix of similarities, the top-left part reports the similarity scores between pairs of sentences both belonging to known documents (base); the bottom-right part includes the similarities between pairs of sentences both belonging to the unknown document (target); and the top-right part reports the similarity scores across known and unknown documents.

Then, two separate agglomerative clustering procedures are carried out on the base and on the target sub-matrices, respectively. Initially, each description yields a different singleton cluster. Then, the dendrogram is built according to a *complete link* strategy, by which two clusters are merged if "the similarity of the farthest items of the two clusters under consideration must be greater than a given threshold". Note that more than one pair might satisfy such a requirement, and that the ordering in which the pairs of clusters are merged affects the final model. To deal with these issues, the procedure ranks the pairs of clusters that might be merged according to the average similarity among all pairs of elements (i.e., sentences) taken from each of the two clusters. Then, for each iteration, only the pair of clusters yielding the highest average similarity is merged. In the end, the resulting set of clusters is considered as a *model* of the writing style of the clustered documents. Since the outcome of the clustering procedure is determined by threshold T, and since it is unlikely that a single fixed threshold works for all possible cases, a flexible approach to determining such a threshold for each specific set of input data is proposed in [3].

As soon as the appropriate thresholds are chosen, the base and target models have been defined, each having its own threshold, and the classification phase may take place. This phase considers only clusters that are not singletons. The ratio of such clusters in the target model that can be merged with at least one such cluster in the base model under the complete link assumption is computed, and taken as a *Score* expressing how much the two models overlap. Such merging check exploits the similarities in the top-right submatrix and the maximum threshold obtained in the previous step for the base and target models. If *Score* passes a pre-defined value τ, then the response is that author is the same, otherwise it is not. The lower such value, the less overlapping is required between the models to classify the target as being written by the same author as the base, and hence the less reliable the classification. E.g., using $\tau = 1.0$ encourages accurate classifications: indeed, it denotes a cautious behavior and makes harder a full alignment between the models. The approach also provides for an option by which it may understand, using suitable heuristics, that the available data are too poor to obtain a reliable outcome, and abstains from returning a decision. We call this setting *smoothed evaluation*, in contrast to the normal setting that is called *boolean evaluation*.

Table 1. Author identification outcomes (NC = not classified)

Type	boolean evaluation		smoothed evaluation				
Set	acc	err	acc	err	NC	Δ_{err}	Δ_{acc}
Training	0.7	0.3	0.7	0.1	0.2	0.2	0
Test 1	0.7	0.3	0.65	0.15	0.2	0.15	0.05
Test 2	0.45	0.55	0.41	0.28	0.31	0.27	0.04
Total	**0.58**	**0.42**	**0.55**	**0.20**	**0.25**	**0.22**	**0.03**

Type	boolean evaluation			smoothed evaluation					
Set	P	R	F	P	R	F	Δ_P	Δ_R	Δ_F
Training	0.7	0.7	0.7	0.87	0.7	0.77	0.17	0	0.07
Test 1	0.7	0.7	0.7	0.81	0.65	0.72	0.11	-0.05	0.02
Test 2	0.45	0.45	0.45	0.6	0.41	0.49	0.15	-0.04	0.04
Total	**0.58**	**0.58**	**0.58**	**0.73**	**0.55**	**0.62**	**0.15**	**-0.03**	**0.04**

The author identification procedure was evaluated using the English dataset provided in the '9th Evaluation Lab on Uncovering Plagiarism, Authorship, and Social Software Misuse' (PAN), held as part of the CLEF 2013 conference. The dataset is split into 3 sub-datasets: a training set 'Training' (involving 10 problems for the English dataset), an early-bird evaluation dataset 'Test 1' (involving 20 problems for the English dataset) and the complete evaluation dataset 'Test 2' (involving 30 problems for the English dataset), which is a superset of 'Test 1'. Table 1 reports the results of an experiment aimed at investigating how good the approach is in the boolean and smoothed evaluation setting. As regards the difference in error rate (*err*) and accuracy (*acc*) between the two settings, a positive difference in the former (Δ_{err}) can be interpreted as a *gain* in performance, while a positive difference in the latter can be interpreted as a *loss*, in using the smoothed evaluation with respect to the boolean one. Table 1 shows that, for each sub-dataset (and hence for the entire dataset as well), the gain is much more than the loss. E.g., in Test 1 the gain (i.e., reduction in error rate) is $0.3 - 0.15 = 0.15$, whereas the loss (in accuracy) is just $0.7 - 0.65 = 0.05$. Concerning to Precision (P), Recall (R) and F1-measure (F). In this case, a positive difference Δ_P between the Precision scores in the smoothed and boolean evaluation settings can be seen as a *gain* (since, reducing the number of cases in which a classification is given, we keep only the most reliable cases, and thus we expect the undecided cases to contain more errors than correct outcomes). Conversely, the difference Δ_R between the Recall scores can be seen as a *loss* (since, reducing the number of cases in which a classification is attempted, we expect that also some correct outcomes having a borderline classification are lost). Unlike Accuracy and Error Rate, here both gain and loss are referred to correct classifications. In particular, the gain represents the decrease in misclassifications with respect to the cases in which the approach gives a response, whereas the loss represents the correct classifications over the entire dataset. As for Accuracy and Error Rate, the gain is always much more than the loss. Such

a good performance of the smoothed evaluation setting, and the positive balance between gain and loss, is confirmed by the F-measure.

4 Author Characterization

When people write texts, they (often unconsciously) make choices, based on their preferences or on the type of document under development. These choices involve the selection and composition of terms and other linguistic elements and patterns. Based on this consideration, it might be sensibly assumed that the linguistic features of the text may somehow characterize the author, reflecting his style and acting as a kind of fingerprint for him. For these reasons, we developed a tool for document comparison that can extract and manage statistics on several kinds of document features: frequency of words, letters, part-of-speech tags, punctuation, words bi-grams, part-of-speech tags bi-grams and letters n-grams ($n = 2, \ldots, 5$). Also, we used the linguistic features extracted by the unsupervised methods provided by the tool in [5]: word suffixes, prefixes, stems and stopwords.

Given two documents, the tool allows to compare them according to the normalized frequency of occurrence of items for each of the above features, and to

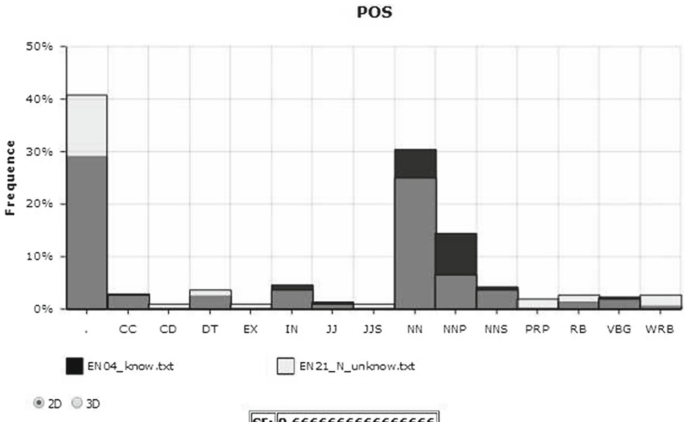

Fig. 1. Document comparison tool (pairwise histogram comparison)

display a graphical and numerical report of the outcome, as shown in Figs. 1 and 2 for the 'PoS' statistic comparison. At the bottom it shows the list of items that are present in both documents, and the lists of the items that occur in either of the two. For the common items, a histogram is plot at the top, that visually summarizes the occurrences of each item in the two documents (see Fig. 1). For each item, the bars representing the percentage of occurrence in the two documents are overlapped. In gray is the overlapping part, in light gray the exceeding percentage when it is due to the former document, and in black the exceeding percentage when it is due to the latter document. Alternatively, the user may display a histogram that shows, for each item, a single bar representing the difference in occurrences between the two documents, i.e., it corresponds to the exceeding part of the overlapping bars for that item in the previous histogram (see Fig. 2). The color (black or light gray) indicates which document this exceeding part comes from. Global statistics are also provided to the user: in addition to the average, minimum and maximum values for the frequencies, two similarity measures are computed. The former (CO) is the sum of the overlapping parts of the bars in the first histogram (i.e., the gray parts in Fig. 1), expressed in percentage of occurrences: the larger this value, the more similar the documents. The latter (SF) is based on the formula in [4], using $\alpha = 0.5$, l as the number of common items in the two documents, and n, m as the number of

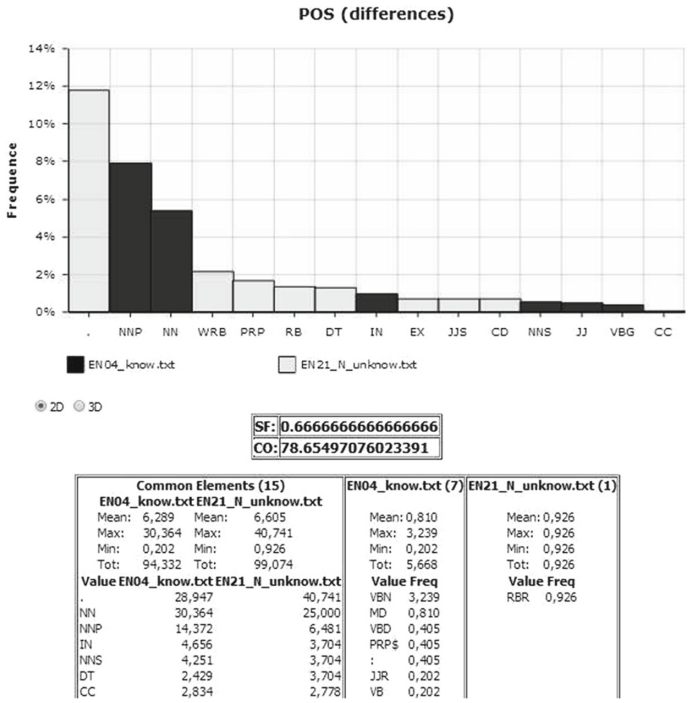

Fig. 2. Document comparison tool (difference histogram)

items that are present in only either of the two, respectively. CO only considers the common information; SF smooths it with the different information, providing a different perspective to the user. While the author identification procedure described in the previous section did not provide any intuitive explanation for its classification, this tools allows the user to have a clear (both numerical and graphical) insight into the documents under comparison, which may be very valuable in order to understand how much the two documents/authors differ, and in what exactly.

As a first test, we used the comparison tool to study the author identification dataset, and specifically the Training subset of its English portion. It allowed us to discover that the 10 problems actually described just 5 cases. More specifically, each set of base documents for a known author was repeated twice, once associated to an unknown document by the same author, and once associated to an unknown document by a different author. This was discovered since the tool reported a substantial similarity in all statistics for those groups of documents. Encouraged by this result, we ran the same analysis on the test sets, and discovered other correspondences, as summarized in Table 2 (where each G_i denotes a group of problems sharing the same base documents).

Table 2. Correspondences in the English portion of the author identification dataset

G_1	{EN01, EN12}
G_2	{EN02, EN08}
G_3	{EN03, EN28, EN37}
G_4	{EN04, EN07, EN36}
G_5	{EN05, EN09}
G_6	{EN06, EN22, EN34}
G_7	{EN10, EN17, EN39}
G_8	{EN11, EN30, EN38}
G_9	{EN13, EN19}
G_{10}	{EN14, EN29}
G_{11}	{EN15, EN25}
G_{12}	{EN16, EN26, EN35}
G_{13}	{EN18, EN24}
G_{14}	{EN20, EN32}
G_{15}	{EN21, EN23, EN40}
G_{16}	{EN27, EN31, EN33}

This allowed us to better focus on single authors, considering for each of them the subset of problems that exploit the same set of known documents against different positive and negative unknown documents. First of all, for each such subset of problems, we built a single document consisting of the concatenation

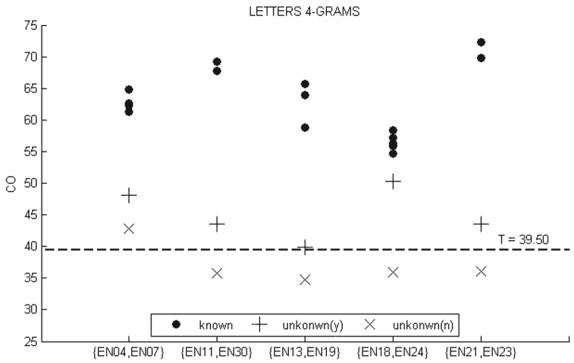

Fig. 3. Similarity of known and unknown documents, and corresponding threshold

of all known documents. Then, we compared it with the corresponding unknown documents and with the single known documents. Finally, we computed all the above statistics for each group of problems and used CO and SF to compare the documents for each feature. A graphic representation of the comparison on the letters 4-grams for the groups of problems in the Training set is shown in Fig. 3, where comparisons with single known documents are represented as filled circles, comparisons with positive test documents are represented as '+', and comparisons with negative test documents are represented as '×'. Interestingly, we observed that not only positive examples usually yield higher similarity values than negative ones, as expected. In fact, there is a clear separation among positive and negative examples throughout the different problems. This suggested the possibility that a similarity threshold could be found that separates positive and negative examples, to be used as an additional criterion for author identification: documents whose similarity value with the compound base document is greater than the threshold are classified as written by the same author. Conversely, document whose similarity value with the compound base document is less than the threshold are classified as written by a different author. To assess such a threshold, for each feature, we considered in turn each example in the training set, partitioned all the remaining examples in two groups (those having similarity above and below the similarity of the selected example, respectively) and counted the number of misplaced examples (i.e., the negative ones above the example under consideration, and the positive ones below it). Then, we fixed the threshold as the similarity value of the example that minimizes the number of misclassifications. E.g., in Fig. 3, the best value is 39.05, because it allows to separate positive and negative examples making a single misclassification.

After determining the thresholds for the different features on the Training set, we applied them on the test set to classify unknown documents, and check whether those thresholds could be used for prediction purposes as well. Indeed, we noted that, overall, a correlation emerges between accuracy on the training and on the test set, which can be used to predict how reliable a learned

threshold will be on unknown documents. So, the similarity value between the base and target document can be used as a support or as a complement to the outcome of the relational author identification technique proposed in the previous sections. More precisely, CO reaches higher average accuracy than SF, both in the training ($CO = 80.56\,\%, SF = 68.89\,\%$) and in the test set ($CO = 66.85\,\%, SF = 53.70\,\%$). For this reason, in the following we will focus on the predictive performance of the former.

Table 3 reports, for each feature, the selected threshold and its performance, both in discriminating the training documents (as regards Accuracy and F1-measure) and in predicting the test ones (for all statistics: Accuracy, Precision, Recall and F1-measure). Values above 70 % are highlighted in bold. As regards the training set, the values express how neat were the thresholds, revealing that the most reliable features are single words, word 2-grams and non-stopwords (100 % for both Accuracy and F1-measure), followed by letter 4-grams, letter 5-grams, stopwords and word length. Of these, only the non-stopwords feature does not have a corresponding good performance on the test set. Except for this, bold values on the training set always correspond to at least

Table 3. Classification performance of CO similarity for the different statistics: Threshold (T), Accuracy (A), Precision(P), Recall (R) and F-measure (F)

	T	Training		Test			
		A (%)	F (%)	A (%)	P (%)	R (%)	F (%)
words	43.25	**100.00**	**100.00**	**73.33**	66.67	**85.71**	**75.00**
words 2-grams	10.63	**100.00**	**100.00**	**73.33**	**80.00**	57.14	66.67
pos 2-grams	53.93	**80.00**	**75.00**	66.67	**100.00**	28.57	44.44
pos	84.43	**80.00**	**75.00**	66.67	**83.33**	35.71	50.00
punctuation	86.91	**70.00**	66.67	66.67	**83.33**	35.71	50.00
sentence letters	56.28	60.00	33.33	60.00	**100.00**	14.29	25.00
sentence words	70.83	**80.00**	**80.00**	63.33	61.54	57.14	59.26
word length	90.72	**90.00**	**88.89**	**83.33**	**76.47**	**92.86**	**83.87**
prefixes	20.67	60.00	33.33	53.33	50.00	14.29	22.22
suffixes	5.09	**70.00**	**76.92**	50.00	47.83	**78.57**	59.46
stems	20.67	60.00	33.33	53.33	50.00	14.29	22.22
stopwords	50.92	**90.00**	**90.91**	**76.67**	**70.59**	**85.71**	**77.42**
non-stopwords	13.33	**100.00**	**100.00**	66.67	64.29	64.29	64.29
letters	90.14	**70.00**	**76.92**	60.00	53.85	**100.00**	**70.00**
letters 2-grams	76.95	**80.00**	**83.33**	63.33	56.52	**92.86**	**70.27**
letters 3-grams	56.95	**80.00**	**83.33**	66.67	59.09	**92.86**	**72.22**
letters 4-grams	39.05	**90.00**	**90.91**	**80.00**	**72.22**	**92.86**	**81.25**
letters 5-grams	26.90	**90.00**	**90.91**	**80.00**	**78.57**	**78.57**	**78.57**

one bold value on the test set. Concerning the test set, the values express the predictive performance of the thresholds. The highest performance in Accuracy and F1-measure is obtained using words length (83.33 % accuracy), followed by letter 4-grams and 5-grams. The lowest overall performance corresponds to suffixes (50 % accuracy), prefixes and stems (22.22 % F1-measure). Prefixes and stems, together with non-stopwords and sentence words, do not pass the 70 % performance for any parameter. Conversely, word length, stopwords, letter 4-grams and letter 5-grams have predictive performances above 70 % for all parameters. This is somehow surprising, since one would expect more content-based features to be more significant. A possible explanation is that the unknown documents are so short that they do not allow a correct extraction of language resources, which in turn leads to ineffective comparison. Also, the language is likely to affect the results: indeed, English has a much poorer inflection than other languages (e.g., Italian, French), which clearly penalizes suffixes and skews the letter n-grams frequency toward the stems rather than the suffixes. For this reason, in future work, we plan to evaluate the result of these statistics with a multi-language dataset. However, a deeper analysis reveals that content-based features (e.g., PoS, punctuation, word 2-grams) tend to yield better precision, while low-level ones (e.g., letter n-grams with $n \leq 3$) tend to improve recall, which is intuitive. Interestingly, except for word 2-grams, F1-measure is greater than Accuracy in all highlighted cases. Cases in which Accuracy and F1-measure are not both above 70 % are usually associated to very high values in only one between Precision and Recall, and in low values on the other.

5 Conclusions

This work proposed an approach to author identification and characterization based on both relational and statistical representations. The former are exploited for identification purposes, and are motivated by the assumption that the syntactic structure of the sentences written by an author somehow capture his writing style. Experimental results have shown that this technique reaches results that are comparable with the state-of-the-art, while not requiring any training and being effective even for short texts. The technique can be applied to any natural language for which suitable linguistic resources are available. Moreover, it is able to autonomously identify cases in which the classification is less reliable.

The relational approach provides a classification that can be hardly traced back to the original texts, in order to provide the user with a better understanding and insight on what makes the two texts/authors alike or different. For this reason, the author identification mechanism was complemented by a tool that computes statistics on several different linguistic features of a given text. Applying these statistics to the two texts under comparison, it obtains indicators that are shown to the user in the form of lists of similar/different items, frequencies, aggregate similarity values and histograms. In addition to providing the user with an intuitive description of what makes the two texts similar or different, it turned out that these statistics are related to the classification in a way that

can be mathematically captured. This provides matter for future work, that, in addition to improving the relational approach, may check whether and how these statistics may be used to improve the author identification performance in the smoothed evaluation setting.

Acknowledgments. The authors would like to thank Fabio Leuzzi, Fulvio Rotella and Domenico Grieco for their work in setting up the system and running the experiments. This work was partially funded by the Italian PON 2007–2013 project PON02_00563_3489339 "Puglia@Service".

References

1. Argamon, S., Whitelaw, C., Chase, P., Hota, S.R., Garg, N., Levitan, S.: Stylistic text classification using functional lexical features: research articles. J. Am. Soc. Inf. Sci. Technol. **58**(6), 802–822 (2007)
2. Feng, V.W., Hirst, G.: Authorship verication with entity coherence and other rich linguistic features notebook for PAN at CLEF 2013. In: CLEF 2013 Labs and Workshops - Online Working Notes, Padua, Italy, PROMISE, September 2013
3. Ferilli, S.: A sentence structure-based approach to unsupervised author identification. J. Intell. Inf. Syst. 1–19. Published on-line: 19 December 2014
4. Ferilli, S., Basile, T.M.A., Biba, M., Di Mauro, N., Esposito, F.: A general similarity framework for horn clause logic. Fundamenta Informaticæ **90**(1–2), 43–46 (2009)
5. Ferilli, S., Esposito, F., Grieco, D.: Automatic learning of linguistic resources for stopword removal and stemming from text. Procedia Comput. Sci. **38**, 116–123 (2014)
6. Leuzzi, F., Ferilli, S., Rotella, F.: ConNeKTion: a tool for handling conceptual graphs automatically extracted from text. In: Catarci, T., Ferro, N., Poggi, A. (eds.) IRCDL 2013. CCIS, vol. 385, pp. 93–104. Springer, Heidelberg (2014)
7. Li, J., Zheng, R., Chen, H.: From fingerprint to writeprint. Commun. ACM **49**(4), 76–82 (2006)
8. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, Heidelberg (1987)
9. Mccarthy, P.M., Lewis, G.A., Dufty, D.F., Mcnamara, D.S.: Analyzing writing styles with coh-metrix. In: Florida Artificial Intelligence Research Society International Conference (FLAIRS), pp. 764–769. AAAI Press (2006)
10. Raghavan, S., Kovashka, A., Mooney, R.: Authorship attribution using probabilistic context-free grammars. In: ACL 2010 Conference Short Papers, ACLShort 2010, pp. 38–42. Association for Computational Linguistics (2010)
11. Rotella, F., Ferilli, S., Leuzzi, F.: A domain based approach to information retrieval in digital libraries. In: Agosti, M., Esposito, F., Ferilli, S., Ferro, N. (eds.) IRCDL 2012. CCIS, vol. 354, pp. 129–140. Springer, Heidelberg (2013)
12. Seidman, S.: Authorship verification using the impostors method - notebook for PAN at CLEF 2013. In: CLEF 2013 Labs and Workshops - Online Working Notes, Padua, Italy, PROMISE, September 2013
13. van Halteren, H.: Linguistic profiling for author recognition and verification. In: 42nd Annual Meeting on Association for Computational Linguistics, ACL 2004. Association for Computational Linguistics (2004)

14. Vilariño, D., Pinto, D., Gómez, H., León, S., Castillo, E.: Lexical-syntactic and graph-based features for authorship verification - notebook for PAN at CLEF 2013. In: CLEF 2013 Labs and Workshops - Online Working Notes, Padua, Italy, PROMISE, September 2013
15. Zheng, R., Li, J., Chen, H., Huang, Z.: A framework for authorship identification of online messages: writing-style features and classification techniques. J. Am. Soc. Inf. Sci. Technol. **57**(3), 378–393 (2006)