A Relational Unsupervised Approach to Author Identification

F. Leuzzi[1], S. Ferilli[1,2], and F. Rotella[1]

[1] Dipartimento di Informatica – Università di Bari
{fabio.leuzzi, stefano.ferilli, fulvio.rotella}@uniba.it
[2] Centro Interdipartimentale per la Logica e sue Applicazioni – Università di Bari

Abstract. In the last decades speaking and writing habits have changed. Many works faced the author identification task by exploiting frequentist approaches, numeric techniques or writing style analysis. Following the last approach we propose a technique for author identification based on First-Order Logic. Specifically, we translate the complex data represented by natural language text to complex (relational) patterns that represent the writing style of an author. Then, we model an author as the result of clustering the relational descriptions associated to the sentences. The underlying idea is that such a model can express the typical way in which an author composes the sentences in his writings. So, if we can map such writing habits from the unknown-author model to the known-author model, we can conclude that the author is the same. Preliminary results are promising and the approach seems practicable in real contexts since it does not need a training phase and performs well also with short texts.

1 Introduction

Speaking and writing habits have changed in the last decades, and many works have investigated the author identification task by exploiting frequentist approaches, numeric techniques and writing style analysis. The spreading of documents across the Internet made the writing activity faster and easier compared to past years. Thus, author identification became a primary issue, due to the increasing number of plagiarism cases. In order to face such problems, several approaches have been attempted in the Machine Learning field [1, 4, 12, 18].

The authorship attribution task is well-understood (given a document, determine who wrote it) although amenable to many variations (given a document, determine a profile of the author; given a pair of documents, determine whether they were written by the same author; given a document, determine which parts of it were written by a specific person), and its motivation is clear. In applied areas such as law and journalism knowing the author's identity may save lives.

The most common approach for testing candidate algorithms is to cast the problem as a text classification task: given known sample documents from a small, finite set of candidate authors, assess if any of those authors wrote a questioned document of unknown authorship. A more lifelike approach is: given a set of documents by a single author and a questioned document, determine whether the questioned document was written by that particular author or not. This is more interesting for professional forensic linguistics because it is a primary need in that environment.

This setting motivated us to face the following task: 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.

Performing a deep understanding of the author to seize his style is not trivial, due to the intrinsic ambiguity of natural language and to the huge amount of common sense and linguistic/conceptual background knowledge needed to switch from a purely syntactic representation to the underlying semantics. Traditional approaches are not able to seize the whole complex network of relationships, often hidden, between events, objects or a combination of them. Conversely relational approaches treat natural language texts as complex data from which mining complex patterns.

So after extracting and making explicit the typed syntactical dependencies of each sentence, we formally express them in a First-Order Logic representation. In this way the unstructured texts in natural language are expressed by complex (relational) patterns on which automatic techniques can be applied. Exploiting such patterns, the author's style can be modelled in order to classify a new document as written by the same author or not.

For the sake of clarity, from now on we refer to the known author used for training as the *base*, and to the unknown author that must be classified as *target*.

This work is organized as follows: the next section describes related works; Section 3 outlines the proposed approach, that is evaluated subsequently. Lastly, we conclude with some considerations and future works.

2 Related Work

There is a huge amount of research conducted on Author Identification in the last 10 years. With the spread of anonymous documents in Internet, authorship attribution becomes important. Researches focus on different properties of texts, the so-called *style markers*, to quantify the writing style under different labels and criteria. Five main types of features can be found: lexical, character, syntactic, semantic and application specific. The lexical and character features consider a text as a mere sequence of word-tokens or characters, respectively. An example of the first category is [2] in which new lexical features are defined for use in stylistic text classification, based on taxonomies of various semantic functions of certain choice words or phrases. While this work reaches interesting results, it is based on the definition of arbitrary criteria (such as the 675 lexical features and taxonomies) and requires language-dependent expertise.

In [20] the authors build a suffix tree representing all possible character n-grams of variable length and then extract groups of character n-grams as features. An important issue of such approaches based on character feature is the choice of n, because a larger n captures more information but increases the dimensionality of the representation. On the other hand, a small n might not be adequate to learn an appropriate model.

Syntactic features are based on the idea that authors tend to unconsciously use similar syntactic patterns. Therefore they exploit information such as PoS-tags, sentence and phrase structures. In addition to the need of robust and accurate NLP tools to perform syntactic analysis of texts, a major drawback of these approaches is the huge amount of feature extracted they require (e.g., in [19] there are about 900k features).

Semantic approaches rely on semantic dependencies obtained by external resources, such as taxonomies or thesauri. In [13] the authors exploits WordNet[5] to detect "semantic" information between words. Although the use of an external taxonomic or ontological resource can be very useful for these purposes, such resources are not always available and often do not exist at all for very specific domains.

Finally, there are non-general-purpose approaches, that define application-specific measures to better represent the style in a given text 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 [11].

While the various approaches faced the problem from different perspectives, a common feature to all of them is their using a flat (vectorial) representation of the document/phrases. Even the two before the last approach, although starting from syntactic trees or word/concept graphs, 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 [15]. In this work a probabilistic context-free grammar (PCFG) is built for each author and then each test document is assigned to the author whose PCFG produced the highest likelihood for such a document. While this approach takes into account the syntactic tree of the sentences, it needs of many documents per author to learn the right probabilities. Thus it is not applicable in settings in which a small set of documents of only one author is available. Moreover we believe that the exploitation of only parse trees is not enough to characterize the author's style, conversely it should be better to enrich the syntactical relationships with grammatical ones.

Differently from all of these, our approach aims at preserving the informative richness of textual data by extracting and exploiting complex patterns from such complex data.

3 Proposed Approach

Natural Language Text is a complex kind of data encoding implicitly the author's style. We propose to translate textual data into a relational description in order to make explicit the complex patterns representing the author's style. The relational descriptions are clustered using the similarity measure presented in [6], where the threshold to be used as a stopping criterion is automatically recognized. We apply this technique to build both base and target models. Then, the classification results from the comparison of these two models. The underlying idea is that the target model describes a set of ways in which the author composes the sentences. If we can bring back such writing habits to the base model, we can conclude that the author is the same.

3.1 The representation formalism

Natural language texts are processed by ConNeKTion [9] (acronym for 'CONcept NEtwork for Knowledge representaTION'), a framework for conceptual graph learning and exploitation. This framework aims at partially simulating some human abilities in the text understanding and concept formation activity, such as: extracting the concepts expressed in given texts and assessing their relevance [7]; obtaining a practical description of the concepts underlying the terms, which in turn would allow to generalize concepts having similar description [16]; applying some kind of reasoning 'by association', that looks for possible indirect connections between two identified concepts [10]; identifying relevant keywords that are present in the text and helping the user in retrieving useful information [17].

In this work we exploit ConNeKTion in order to obtain a relational representation of the syntactic features of the sentences. In particular exploiting the *Stanford Parser* and *Stanford Dependencies* tools [8, 3] we obtain phrase structure trees and a set of grammatical relations (typed dependencies) for each sentence. These dependencies are expressed as binary relations between pairs of words, the former of which represents the governor of the grammatical relation, and the latter its dependent. Words in the input text are normalized using lemmatization instead of stemming, which allows to distinguish their grammatical role and is more comfortable to read by humans. ConNeKTion also embeds JavaRAP, an implementation of the classic Anaphora Resolution Procedure [14]. Indeed, the subject/objects of the sentences are often expressed as pronouns, referred to the first occurrence of the actual subject/object. After applying all these pre-processing steps, we translate each sentence into a relational pattern. In particular, each sentence is translated into a Horn Clause of the form:

$$sentence(IdSentence) \text{ :- } description(IdSentence).$$

where $description(Idsentence)$ is a combination of the following atoms which reflect the relations between the words in the sentence:

- $phrase(Tag, IdSentence, Pos)$ represents a constituent whose Tag is the type of phrase (e.g. NP, VP, S,...) and Pos is the term position in the phrase;
- $term(IdSentence, Pos, Lemma, PosTag)$ defines a single term whose position in the sentence is Pos, its lemma is $Lemma$ and its part-of-speech (e.g. N,V,P,...) is $PosTag$;
- $sd(IdSentence, Type, PosGov, PosDep)$ represents the grammatical relation $Type$ (e.g. dobj, subj,...) between the governor word in position $PosGov$ and the dependent word in position $PosDep$.

This allows us to represent all the relationships between the terms, their grammatical relations and the phrases to which they belong.

3.2 The similarity measure

The similarity strategy exploited here was presented in [6]. It takes values in $]0,4[$ and is computed by repeated applications of the following formula to different parameters extracted from the relational descriptions:

$$sf(i', i'') = sf(n, l, m) = \alpha \frac{l+1}{l+n+2} + (1-\alpha) \frac{l+1}{l+m+2}$$

where:

- i' and i'' are the two items under comparison;
- n represents the information carried by i' but not by i'';
- l is the common information between i' and i'';
- m is the information carried by i'' but not by i';
- α is a weight that determines the importance of i' with respect to i'' (0.5 means equal importance).

More precisely, the overall similarity measure carries out a layered evaluation that, starting from simpler components, proceeds towards higher-level ones repeatedly applying the above similarity formula. At each level, it exploits the information coming from lower levels and extends it with new features. At the basic level terms (i.e., constants or variables in a Datalog setting) are considered, that represent objects in the world and whose similarity is based on their properties (expressed by unary predicates) and roles (expressed by their position as arguments in n-ary predicates). The next level involves atoms built on n-ary predicates: the similarity of two atoms is based on their "star" (the multiset of predicates corresponding to atoms directly linked to them in the clause body, that expresses their similarity 'in breadth') and on the average similarity of their arguments. Since each of the four components ranges into $]0,1[$, their sum ranges into $]0,4[$. Then, the similarity of sequences of atoms is based on the length of their compatible initial subsequence and on the average similarity of the atoms appearing in such a subsequence. Finally, the similarity of clauses is computed according to their least general generalization, considering how many literals and terms they have in common and on their corresponding lower-level similarities.

3.3 Building models

Obtained a relational description for each sentence as described in Section 3.1, we applied the similarity measure described in Section 3.2 to pair of sentences. In particular, for each training-test couple we computed an upper triangular similarity matrix between each pair sentences. As can be seen in Figure 1 the global matrix can be partitioned into three parts, the top-left submatrix (filled with diagonal lines) contains the similarity scores between each pair of sentences of known documents (base). The bottom-right one (filled with solid grey) includes the similarities between pairs of sentences belonging to the unknown document (target). The top right submatrix reports the similarity scores across known and unknown documents.

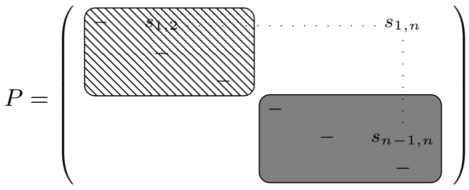

Fig. 1. Global similarity matrix. Each $s_{i,j}$ represents the similarity between the sentence i and j calculated as explained in Section 3.2

Then, we performed an agglomerative clustering to both base and target submatrices according to Algorithm 1. Initially each description makes up a different singleton cluster; then the procedure works by iteratively finding the next pair of clusters to be merged according to a *complete link* strategy. Complete link states that *the distance of the farthest items of the involved clusters must be less than a given threshold.* In this work we refer to a *model* as a possible grouping of similar descriptions, as obtained by running the clustering algorithm with a given threshold.

In this perspective, there is the need of establishing the threshold by which pairwise clustering is carried out for each *model.* Our approach is based on the idea that as long as the threshold increases, also the number of clusters grows, and thus the merging becomes more and more difficult. So, we consider as cut point the greatest gap between the number of clusters obtained with a threshold and the next one obtained by performing clustering with a greater threshold. It is easy to note that a given difference value obtained with many clusters is less significant than the one obtained with a smaller number of clusters.

Taking into account such considerations we have defined the following function that encodes such intuitive assumptions. Given a sequence of models $< m_1, ..., m_n >$ obtained by repeating the clustering procedure with the increment of the cut-threshold by step 0.05, and $c(m_i)$ that computes the number of clusters in the i-th model, we can define:

$$g(i) = \frac{c(m_{i+1})}{c(m_i)} \quad \text{and} \quad th_i = \arg\max_i g(i)$$

where $0 \leq i < n$ and th_i is the desired threshold associated to the model m_i yielding the greatest distance from the model m_{i+1} (see Algorithm 2). Since our similarity measure ranges in $]0, 4[$, the thresholds varies within such range.

Chosen the appropriate thresholds, we defined base and target models and thus we performed the classification. In particular, as can be seen in Algorithm 3, for each cluster in the target model having more than one item, if it can be merged with at least one cluster in the base model (under the complete link assumption), the author is the same, otherwise it is not. Such merging check exploits the top right submatrix. Moreover it uses the maximum threshold between base and target model, which making harder a full alignment between target and base clusters and ensures more precision in the classifications.

Algorithm 1 Relational pairwise clustering.
Interface: *pairwiseClustering(M,T)*.

Input: M is the similarity matrix; T is the threshold for similarity function.
Output: set of clusters.

```
$pairs \leftarrow empty$
$averages \leftarrow empty$
for all $item : M.rows$ do
   $newCluster \leftarrow item$
   $clusters.add(newCluster)$
end for
for all $pair(C_k, C_z) \mid C_k, C_z \in clusters$ do
   if $completeLink(M, C_k, C_z, T)$ then
      $pairs.add(C_k, C_z)$
      $averages.add(getScoreAverage(C_k, C_z))$
   end if
end for
$pair \leftarrow getBestPair(pairs, averages)$
$merge(pair)$
return $clusters$
```

completeLink(matrix,cluster$_1$*,cluster*$_2$*,threshold)* → TRUE if complete link assumption for the passed clusters holds, FALSE otherwise.
getBestPair(pairs,averages) → returns the pair having the maximum average.

4 Evaluation

We evaluated our procedure using the training set provided in the 9th evaluation lab on uncovering plagiarism, authorship, and social software misuse (PAN) held as part of the CLEF 2013 conference. Although this challenge has already taken place, we could not compare our outcomes with the official challenge results because the test set is not yet publicly available. However, since our approach does not require a training phase, we were able to exploit the training set for this purpose. Such a dataset is composed by 10 problems for the English language, 20 problems for Greek and 5 problems for Spanish. In this evaluation we will consider the English problems only, since the current version of ConNeKTion is based on the Stanford NLP tools, that cannot deal with the other two languages. However, our approach can be easily extended to the other languages, as long as suitable NLP tools for them are available.

Table 1 reports, for each problem (whose ID, as indicated in the original data set, is reported in the first column), information both on the documents written by the base author and on the unknown document written by the target author. As to the former, it specifies the number of documents, the number of clauses generated by the sentences they contain and their average length. As to the latter, it shows the number of corresponding clauses/sentences and their average length. It also reports the experimental outcomes including the expected

Algorithm 2 Best model identification.
Interface: *getBestModel(M,T_{lower},T_{higher})*

Input: M is the similarity matrix; T_{lower} is the starting threshold, T_{higher} is the maximum threshold that can be attempted.
Output: $bestModel$ that is the model having the best threshold.

```
$t \leftarrow T_{lower}$
$models \leftarrow \emptyset$
$thresholds \leftarrow \emptyset$
for all $t < T_{higher}$ do
    $clusters \leftarrow pairwiseClustering(M)$
    $models.add(clusters)$
    $thresholds.add(t)$
    $t \leftarrow t + 0.5$
end for
$maxHop \leftarrow 0$
$bestModel \leftarrow null$
for all $m_i \mid m_i \in models, i > 0$ do
    $hop \leftarrow (m_i.size * 100)/m_{i-1}.size$
    if $maxHop < hop$ then
        $maxHop \leftarrow hop$
        $bestModel \leftarrow m_{i-1}$
    end if
end for
return $bestModel$
```

class, the class predicted by our approach and a score computed as the number of aligned cluster across models over the total number of target-model clusters. The average number of sentences for the known and unknown documents is respectively 164.9 and 56.2, and the average lengths of the descriptions are respectively of 176.01 and 203.97. It can be noted that on average the known documents consist of many short sentences while the unknown documents are composed of few long sentences. A singular situation happens in problem ‘EN23’ where summing all the sentences of the known documents we obtain half of the sentences belonging to the corresponding unknown one.

In our experimentation we reported the same measures required in the original challenge. Precision and Recall (and, consequently, F_1 measure) are equal to 0.7 in the case of a ‘hard’ classification obtained considering as Yes only the scores equal to 1.0. Softening the classification threshold to 0.9, these statistics become 0.8 for all metrics. In fact, it is worth noting that the correct classification of the problem ‘EN23’ was not reached for just 0.08, but this is the problem with the smallest number of clauses to be clustered, hence we could hypothesize that a sufficiently refined model has not been reached for a while.

The complete quantitative results of the PAN 2013 challenge (English section) are reported in Table 2. In this table are shown the nick names of the challenge participants with the performances of their systems. Considering that

Algorithm 3 Complete classification procedure.

Input: O_{known} is the set of descriptions (represented as in Section 3.1) obtained from the known-author documents; $O_{unknown}$ is the set of descriptions obtained from the unknown-author document; T_{lower} is the starting threshold, T_{higher} is the maximum threshold that can be attempted.
Output: Classification outcome.

```
$M_{known} \leftarrow getSimilarities(O_{known})$
$model_{known} \leftarrow getBestModel(M_{known}, T_{lower}, T_{higher})$
$M_{unknown} \leftarrow getSimilarities(O_{unknown})$
$model_{unknown} \leftarrow getBestModel(M_{unknown}, T_{lower}, T_{higher})$
$t \leftarrow max(t_{known}, t_{unknown})$
$O \leftarrow O_{known}$
$O.add(O_{unknown})$
$M \leftarrow getSimilarities(O)$
$class \leftarrow true$
for all $(C_k, C_u) \mid C_k \in model_{known} \wedge C_u \in model_{unknown}$ do
  if $!completeLink(M, C_k, C_u, t)$ then
    $class \leftarrow false$
  end if
end for
return $class$
```

completeLink(matrix,cluster$_1$,cluster$_2$,threshold) → TRUE if complete link assumption for the passed clusters holds, FALSE otherwise.
getSimilarities(list) → returns the similarity matrix between all pairs of objects in 'list'.

the performance of the winner approach in PAN 2013 has reached an F_1 measure equal to 0.8, these preliminary results can be considered very promising and motivate us to further proceed in this research direction.

5 Conclusions

This work proposes a technique for author identification based on First-Order Logic. It is motivated by the assumption that making explicit the typed syntactical dependencies in the text one may obtain significant features on which basing the predictions. Thus, this approach translates the complex data represented by natural language text to complex (relational) patterns that allow to model the writing style of an author. Then, these models can be exploited to classify a novel document as written by the author or not. Our approach consists in translating the sentences into relational descriptions, then clustering these descriptions (using an automatically computed threshold to stop the clustering procedure). The resulting clusters represent our model of an author. So, after building the models of the base (known) author and the target (unknown) one, the comparison of these models suggests a classification (i.e., whether the target author is the same

Table 1. Dataset details and outcomes

ID	Known docs			Unknown doc		Outcomes		
	$\#_{docs}$	$\#_{clauses}$	μ_{length}	$\#_{clauses}$	μ_{length}	Expected	Class	Score
EN04	4	261	121.06	62	136.60	Y	Y	1.0
EN07	4	260	121.48	44	195.47	N	Y	1.0
EN11	2	109	185.87	39	160.41	Y	Y	1.0
EN13	3	109	156.99	65	134.65	N	N	0.6
EN18	5	274	154.25	53	165.49	Y	Y	1.0
EN19	3	139	164.35	56	210.05	Y	N	0.37
EN21	2	109	210.89	24	269.21	N	N	0.67
EN23	2	51	217.29	97	277.29	Y	N	0.92
EN24	5	242	147.06	89	169.08	N	N	0.69
EN30	2	95	189.87	33	322.09	N	N	0.8

Table 2. Performances of PAN 2013 Challenge for English Dataset

Submission		**English**	
	F_1	Precision	Recall
zhenshi13	0.800	0.800	0.800
seidman13	0.800	0.800	0.800
layton13	0.767	0.767	0.767
moreau13	0.767	0.767	0.767
jankowska13	0.733	0.733	0.733
ayala13	0.733	0.733	0.733
halvani13	0.700	0.700	0.700
feng13	0.700	0.700	0.700
ghaeini13	0.691	0.760	0.633
petmanson13	0.667	0.667	0.667
bobicev13	0.644	0.655	0.633
sorin13	0.633	0.633	0.633
vandam13	0.600	0.600	0.600
jayapal13	0.600	0.600	0.600
kern13	0.533	0.533	0.533
baseline	0.500	0.500	0.500
gillam13	0.500	0.500	0.500
vladimir13	0.467	0.467	0.467
grozea13	0.400	0.400	0.400

as the base one or not). The underlying idea is that the model describes a set of ways in which an author composes the sentences in its writings. If we can bring back such writing habits from the target model to the base model, we can conclude that the author is the same.

It must be underlined a small number of documents is sufficient, using this approach, to build an author's model. This is important because, in real life, only a few documents are available for the base author, on which basing a classification. We wanted to stress specifically this aspect in our experiments, using the training set released for the PAN 2013 challenge. Preliminary results are promising. Our approach seems practicable in real contexts since it does not need of a training phase and performs well also with short texts.

The current work in progress concerns the evaluation of our system using the original test set used in PAN 2013 challenge. As a future work, we plan to study the quality of the clusters, pursuing an intensional understanding thereof. In particular, we want to study whether generalizing the clustered clauses we can obtain a theory expressing the typical sentence construction that the author exploits in his texts. Such theory would be the intensional model of the author, which would allow to carry on the investigation in the learning field.

References

[1] Shlomo Argamon, Marin Saric, and Sterling Stuart Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480. ACM, 2003.

[2] Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification using functional lexical features: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):802–822, April 2007.

[3] Marie catherine De Marneffe, Bill Maccartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *In Proc. Intl Conf. on Language Resources and Evaluation (LREC)*, pages 449–454, 2006.

[4] Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2):109–123, May 2003.

[5] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[6] Stefano Ferilli, Teresa M.A. Basile, Nicola Di Mauro, and Floriana Esposito. Plugging numeric similarity in first-order logic horn clauses comparison. In Roberto Pirrone and Filippo Sorbello, editors, *XIIth International Conference of the Italian Association for Artificial Intelligence*, volume 6934 of *LNCS*, pages 33–44. Springer, 2011.

[7] Stefano Ferilli, Fabio Leuzzi, and Fulvio Rotella. Cooperating techniques for extracting conceptual taxonomies from text. In *Proceedings of The Workshop on Mining Complex Patterns at AI*IA XIIth Conference*, 2011.

[8] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.

[9] Fabio Leuzzi, Stefano Ferilli, and Fulvio Rotella. ConNeKTion: A tool for handling conceptual graphs automatically extracted from text. In Tiziana Catarci, Nicola Ferro, and Antonella Poggi, editors, *Bridging between Cultural Heritage Institutions – Proceedings of the 9th Italian Research Conference on Digital Libraries (IRCDL 2013)*, volume 385 of *CCIS*. Springer-Verlag Berlin Heidelberg, 2013.

[10] Fabio Leuzzi, Stefano Ferilli, and Fulvio Rotella. Improving robustness and flexibility of concept taxonomy learning from text. In Annalisa Appice, Michelangelo Ceci, Corrado Loglisci, Giuseppe Manco, Elio Masciari, and Zbigniew W. Ras, editors, *New Frontiers in Mining Complex Patterns - First International Workshop, NFMCP 2012, Held in Conjunction with ECML/PKDD 2012, Bristol, UK,*

September 24, 2012, Revised Selected Papers, volume 7765 of *CCIS*, pages 232–244. Springer-Verlag Berlin Heidelberg, April 2013.

[11] Jiexun Li, Rong Zheng, and Hsinchun Chen. From fingerprint to writeprint. *Commun. ACM*, 49(4):76–82, April 2006.

[12] David Lowe and Robert Matthews. Shakespeare vs. fletcher: A stylometric analysis by radial basis functions. *Computers and the Humanities*, 29(6):449–461, 1995.

[13] Philip M. Mccarthy, Gwyneth A. Lewis, David F. Dufty, and Danielle S. Mcnamara. Analyzing writing styles with coh-metrix. In Geoff Sutcliffe and Randy Goebel, editors, *In Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS*, pages 764–769. AAAI Press, 2006.

[14] Long Qiu, Min-Yen Kan, and Tat-Seng Chua. A public reference implementation of the RAP anaphora resolution algorithm. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*, pages 291–294. European Language Resources Association, 2004.

[15] 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, pages 38–42, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[16] Fulvio Rotella, Stefano Ferilli, and Fabio Leuzzi. An approach to automated learning of conceptual graphs from text. In Moonis Ali, Tibor Bosse, Koen V. Hindriks, Mark Hoogendoorn, Catholijn M. Jonker, and Jan Treur, editors, *Recent Trends in Applied Artificial Intelligence, 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013, Amsterdam, The Netherlands, June 17-21, 2013. Proceedings*, volume 7906 of *Lecture Notes in Computer Science*, pages 341–350. Springer, 2013.

[17] Fulvio Rotella, Stefano Ferilli, and Fabio Leuzzi. A domain based approach to information retrieval in digital libraries. In Maristella Agosti, Floriana Esposito, Stefano Ferilli, and Nicola Ferro, editors, *Digital Libraries and Archives - 8th Italian Research Conference, IRCDL 2012, Bari, Italy, February 9-10, 2012, Revised Selected Papers*, volume 354 of *CCIS*, pages 129–140. Springer-Verlag Berlin Heidelberg, January 2013.

[18] Fiona J. Tweedie, S. Singh, and David I. Holmes. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1):1–10, 1996.

[19] Hans van Halteren. Linguistic profiling for author recognition and verification. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[20] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.*, 57(3):378–393, February 2006.