Authorship Verification via k-Nearest Neighbor Estimation

Notebook for PAN at CLEF 2013

Oren Halvani, Martin Steinebach, and Ralf Zimmermann

Fraunhofer Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
{FirstName.LastName}@SIT.Fraunhofer.de

Abstract In this paper we describe our k-Nearest Neighbor (k-NN) based Authorship Verification method for the Author Identification (AI) task of the PAN 2013 challenge. The method follows an ensemble classification technique based on the combination of suitable feature categories. For each chosen feature category we apply a k-NN classifier to calculate a style deviation score between the training documents of the true author $\mathcal{A}$ and the document from an author, who claims to be $\mathcal{A}$. Depending on the score and a given threshold, a decision for or against the alleged author is generated and stored into a list. Afterwards, the final decision regarding the alleged authorship is determined through a majority vote among all decisions within this list. The method provides a number of benefits as for instance the independence of linguistic resources like ontologies, thesauruses or even language models. A further benefit is the language-independency among different Indo-European languages as the approach is applicable on languages like Spanish, English, Greek or German. Another benefit is the low runtime of the method, since there is no need for deep linguistic processing like POS-tagging, chunking or parsing. Moreover, the method can be extended or modified for instance by replacing the classification function, the threshold or the underlying features including their parameters (e.g. n-Gram sizes or the amount of feature frequencies). In addition to the PAN 2013 AI-training-corpus, where we gained an overall accuracy score of 80%, we also evaluated the algorithm on our own dataset with an accuracy of 77.50%.

1 Introduction

Authorship Verification (AV) is a subdiscipline of Authorship Analysis [8, Page: 3], where the task is to verify if a given document $\mathcal{D}_{\mathcal{A}?}$ of an alleged author $\mathcal{A}$ has been really written by $\mathcal{A}$ or not. In order to verify the authorship of $\mathcal{D}_{\mathcal{A}?}$ some components are mandatorily required. Generally, this includes a set of sample documents from $\mathcal{A}$, a set of features (concrete "style markers") and at least one classification method that makes the decision regarding the alleged authorship. From the perspective of Machine Learning, AV appears to be an instance of binary classification problems, since the expected output of AV system regarding the alleged authorship is either "yes" or "no" (formally $\mathcal{A}$ or $\neg\mathcal{A}$). On a second glance, however, it turns out that AV must describe an one-class-classification problem, [3] since $\mathcal{A}$ is the only target class to be distinguished among all other classes (authors), which can be theoretically infinite.

In contrast to the related discipline Authorship Attribution (AA), which is also a subfield of Authorship Analysis, AV has not gained the same popularity in research. This statement can be observed from the number of publications within scientific gateways/portals as for instance ACM, CiteCeer or Springer in Table 1.

Portal	**Number of publications according to both disciplines**	**Source**
ACM Digital Library	50 for AV vs. 391 for AA	[4]
CiteCeerX	27 for AV vs. 377 for AA	[1]
SpringerLink	25 for AV vs. 267 for AA	[7]

Table 1: Publishing statistics of scientific portals.

One possible reason for this is that AV can be more complicated to handle as AA, since it represents an open-class problem by default, while AA can be both an open or a closed-set problem. Open-set means that the true author of a given document is not included in a set, while in a closed-set the opposite is the case. Upendra et al. mentioned in [6] that most of the research focusses on the closed-set case, which is easier to solve as the open-set case problem. The latter one, however, is closely related to AV, since a decision must be formulated that descibes if an alleged author is accepted to be the true author or not.

In this paper we propose an intuitive scheme for AV, based on our previous work in [2, Page: 99]. The core of the method relies on the Nearest Neighbor one-class-classification technique described in [9, Page: 69]. The main idea of our approach is to compute and compare style deviation scores depending on so-called feature category combinations between $\mathcal{D}_{\mathcal{A}?}$ and a set of training documents of $\mathcal{A}$. Afterwards the overall decision regarding the alleged authorship is determined, based on a majority vote among all single decisions (according to the chosen feature category combinations).

2 Notation

To increase the readability of this paper we decided to follow the notation in Table 2, which we have also defined in our previous work [2]. The notation will be used both in tables and in continuous text.

Symbol	**Decription**
$\mathcal{A}$	Denotes the true author.
$\mathcal{D}_{\mathcal{A}?}$	Denotes a document of an alleged author, who claims to be $\mathcal{A}$.
$\mathbb{D}_{\mathcal{A}}$	Denotes a set $\{ \mathcal{D}_{1\mathcal{A}}, \mathcal{D}_{2\mathcal{A}}, \dots \}$ containing sample documents of $\mathcal{A}$.
F	Denotes a feature category consisting of $\{ f_1, f_2, \dots \}$ features.
$\mathcal{F}$	Denotes a feature vector (e.g. $\mathcal{F}_{\mathcal{A}?}$ as the feature vector represenatation of $\mathcal{D}_{\mathcal{A}?}$).

Table 2: Notation used in this paper.

3 Features

In this section we describe which features we used in our AV system for the PAN task. Typically features are taken from linguistic layers, which can be understood as structured units of an unstructured text. Stamatatos [8] and Loose [5] for instance mention the following linguistic layers, which can be looked-up for features:

1. **Phoneme layer:** This layer includes phoneme-based features that can be won out of texts by using (pronouncing) dictionaries, e.g. IPA.
2. **Character layer:** This layer includes character-based features as for instance prefixes, suffixes or letter n-Grams.
3. **Lexical layer:** This layer includes token-based features as for instance function words or Pos-Tags (Part-Of-Speech Tags).
4. **Syntactic layer:** This layer includes syntax-based features as for instance constituents (e.g. nominal phrases) or collocations.
5. **Semantic Layer:** This layer includes semantic-based features especially semantic relations (homonyms, hyponymous, synonymys, meronyms, etc.).

Instead of utilizing features according to these layers, we decided to define the term "feature categories" that allows us to distinguish more precisely among single features. For example, considering the features `a`,`o` and `u`, the appropriate feature category would be *letters*. Hence, a feature category can be seen as concrete concept of features belonging to a specific (more abstract) linguistic layer. Table 3 lists all feature categories that have been used by our AV system.

F_i	**Feature category**	**Examples**	**Remarks**
F_1	*Punctuation marks*	`-`, `_`, `,`, `.`, `:`, `;`, `()`, `[]`, `{}`	–
F_2	*Letters*	`a`, `b`, `c`, ..., `x`, `y`, `z`, `A`, `B`, `C`, ..., `X`, `Y`, `Z`	–
F_3	*Letter n-Grams*	`en`, `er`, `th`, `ted`, `ough`	$n \in \{2,3,4\}$
F_4	*Token k-prefixes*	`[removed]` $\rightsquigarrow$ `[re]`, `[confirmed]` $\rightsquigarrow$ `[con]`	$k \in \{2,3\}$
F_5	*Token k-suffixes*	`[extended]` $\rightsquigarrow$ `[ed]`, `[available]` $\rightsquigarrow$ `[able]`	$k \in \{2,4\}$
F_6	*Function words*	`and`, `or`, `the`, `on`, `in`, `while`	–
F_7	*Function word n-Grams*	`(which, is, or)`, `(that, on, the, above)`	$k \in \{3,4\}$
F_8	*Sentence k-beginning function words*	`(The ...)`, `(Since the ...)`	$k \in \{1,2\}$
F_9	*Token n-Grams*	`(such that)`, `(it could not)`	$k \in \{2,3\}$
F_{10}	*Token n-Gram lengths*	`(of the)` $\rightsquigarrow$ `(2,3)`, `(are known as)` $\rightsquigarrow$ `(3,5,2)`	$k \in \{2,4\}$
F_{11}	*Token n-Gram k-prefixes*	`(has been more)` $\rightsquigarrow$ `(ha, be, mo)`	$n = 3, k = 2$
F_{12}	*Token n-Gram k-suffixes*	`(has been more)` $\rightsquigarrow$ `(as, en, re)`	$n = 3, k = 2$

Table 3: Feature categories

3.1 Parameters

The majority of the feature categories in Table 3 can be parameterized in various ways, e.g. n-Gram sizes, k-prefix/suffix lengths or the number of entries within dictionary-based feature categories (for instance F_6). Moreover, the frequencies of the extracted features for each F_i can be adjusted, such that one can decide to use all or just the top t

extracted features. As a consequence, there is a dependency of these parameters when constructing feature vectors, which also affects our AV system regarding accuracy and runtime. For the PAN competition, however, we could not find the best parameters, due to the fact that the parameter-space is extremely huge and can only be searched through by special techniques as for instance evolutionary algorithms. Such techniques, however, are beyond the scope of our approach and are subject of future work.

4 Verification via k-NN Estimation

In this section we give a detailed description of our AV system. For overview purposes we divided the entire process into three subsections, where we first explain what kind of preprocessing we perform to clean and normalize the data, then we describe the core verification algorithm and finally we show how our system determines the decision regarding the alleged authorship.

4.1 Preprocessing

Given $\mathcal{D}_{\mathcal{A}?}$ as an input document which is going to be verified regarding the alleged authorship and $\mathbb{D}_{\mathcal{A}}$ as the set of sample documents of the true author, the first step is to perform preprocessing on all documents in terms of normalization and noise reduction. Normalization is essential to treat all documents uniquely, while noise reduction is important to increase the quality of extracted features. We normalize the documents by several operations as for instance substituting successive blanks by only one or by replacing diacritics by their appropriate representatives, e.g. "ñ" $\rightsquigarrow$ "n". Moreover, we equalize the lengths of the training documents. For this we concatenate all $\mathcal{D}_{i\mathcal{A}} \in \mathbb{D}_{\mathcal{A}}$ into a single document, which is then splitted up into $\ell + 1$ (near) equal-sized training documents $\mathcal{D}_{1\mathcal{A}}$, $\mathcal{D}_{2\mathcal{A}}$, $\ldots$, $\mathcal{D}_{\ell+1\mathcal{A}}$. Regarding noise reduction we remove citations, markup tags, formulas and non-words as for instance "DFT-Eq.(6.1)".

4.2 Verification

Our verification algorithm is based on a k-NN classifier which, by its nature, is only able to work on numeric values. Therefore, we first must construct the feature vector representations $\mathcal{F}_{\mathcal{A}?}$ from $\mathcal{D}_{\mathcal{A}?}$ and $\mathcal{F}_{1\mathcal{A}}$, $\mathcal{F}_{2\mathcal{A}}$, $\ldots$, $\mathcal{F}_{\ell+1\mathcal{A}}$ from all the training documents. Beforehand, we have to choose a set of feature categories that should be taken into account. This set is denoted by $\mathbb{F}$ and is always odd-numbered, otherwise the later majority-vote determination would be inapplicable. Each feature vector consists of exactly $n = |F_i|$ values that represent relative frequencies within its underlying document, according to $F_i \in \mathbb{F}$. It should be noted that in our scheme it can never be the case that a feature vector is mixed (includes features among more than one category). After all feature vectors have been constructed, we calculate style deviation scores between $\mathcal{F}_{\mathcal{A}?}$ and each $\mathcal{F}_{j\mathcal{A}}$ for all $F_i \in \mathbb{F}$. A style deviation score describes a value $s_j \in \left(\mathbb{R}_+ \cup \{0\}\right)$ and is calculated through a distance function $dist(\cdot, \cdot)$ as for example the euclidean distance. Hence, the deviation score can be written formally as $s_j = dist(\mathcal{F}_{\mathcal{A}?}, \mathcal{F}_{j\mathcal{A}})$. The lower s_j is, the lower is the style deviation between $\mathcal{F}_{\mathcal{A}?}$ and $\mathcal{F}_{j\mathcal{A}}$. Therefore

$s_j = 0$ is interpretated as absolutely identical, while $s_j \to \infty$ is interpretated as totally different in terms of style. The resulting deviation scores are stored together with their corresponding feature vectors into a list, which is sorted (according to the scores) by ascending order:

$$Outer\_Distances = \big((s_1, \mathcal{F}_{1\mathcal{A}}), (s_2, \mathcal{F}_{2\mathcal{A}}), \ldots, (s_{\ell+1}, \mathcal{F}_{\ell+1\mathcal{A}}) \big)$$

Next, we pick s_1 (which represents the smallest style deviation) and its corresponding feature vector $\mathcal{F}_{1\mathcal{A}}$ from the first tuple within $Outer\_Distances$ and discard the remaining list. After that, we calculate again style deviation scores but this time between $\mathcal{F}_{1\mathcal{A}}$ and each $\mathcal{F}_{i\mathcal{A}} \in \{ \mathcal{F}_{2\mathcal{A}}, \mathcal{F}_{3\mathcal{A}}, \ldots, \mathcal{F}_{\ell+1\mathcal{A}} \}$. The scores $s_j^{'} = dist(\mathcal{F}_{1\mathcal{A}}, \mathcal{F}_{i\mathcal{A}})$ are then stored (without their feature vectors) into the following list:

$$Inner\_Distances = \big(s_2^{'}, s_3^{'}, \ldots, s_{\ell+1}^{'} \big)$$

4.3 Decision

To obtain a single decision ($\mathcal{A}$ or $\neg\mathcal{A}$) regarding a chosen feature category F_i we first calculate the average of the k scores within $Inner\_Distances$ and denote it as avg_dist. Here, k refers to the k nearest neighbours of $\mathcal{F}_{1\mathcal{A}}$. Then, we define the acceptance decision as follows:

$$\frac{s_1}{avg\_dist} \leq \theta$$

If the acceptance decision holds such that the threshold θ is not exceeded, $\mathcal{A}$ is accepted as the true author of $\mathcal{D}_{\mathcal{A}?}$, otherwise not. It should be noted that in all our experiments we chose $\theta = 1$, since we have observed quite stable results with it in comparison with other thresholds. After all the decisions concerning $\mathbb{F}$ have been determined, we store them into a list denoted by $Decisions$, which can look like this, for example:

$$Decisions = \big(\mathcal{A}, \mathcal{A}, \mathcal{A}, \neg\mathcal{A}, \neg\mathcal{A} \big)$$

Finally we apply a majority-vote over $Decisions$ and treat the resulting outcome as the overall decision. In the case of the above example the AV system will judge that $\mathcal{D}_{\mathcal{A}?}$ has been written by $\mathcal{A}$.

5 Experiments

In our experiments we consider the PAN corpus ("PAN 2013 AI-training-corpus"), which has been provided for the Author Identification task. It consists of 189 training documents across the languages Greek (GR), English (EN) and Spanish (SP) and is divided into the datasets $|\mathcal{C}_{GR}| = 20$, $|\mathcal{C}_{EN}| = 10$ and $|\mathcal{C}_{SP}| = 5$.

For our AV system we tested ten different combinations of feature categories and the parameters listed in Table 4. The results are calculated in simple and weighted accuracies, where a simple accuracy is calculated as:

$$\varnothing = \frac{\varnothing_{\mathcal{C}_{GR}} + \varnothing_{\mathcal{C}_{EN}} + \dots}{|\mathcal{C}_{GR} \cup \mathcal{C}_{EN} \cup \dots|} \text{ , with } \varnothing_{\mathcal{C}_i} = \frac{\text{Number of correct answers per dataset } \mathcal{C}_i}{\text{Total number of documents per dataset } \mathcal{C}_i}$$

while in contrast, a weighted accuracy is calculated as:

$$(\text{weighted})\varnothing = \frac{|\mathcal{C}_{GR}| \cdot \varnothing_{\mathcal{C}_{GR}} + |\mathcal{C}_{EN}| \cdot \varnothing_{\mathcal{C}_{EN}} + \dots}{|\mathcal{C}_{GR} \cup \mathcal{C}_{EN} \cup \dots|}$$

F_i	n-Gram	k-prefix/suffix	Top-t (features)	Dictionary entries
F_1	–	–	all	18 per language
F_2	–	–	all	$\approx$ 50 per language
F_3	7	–	100	–
F_4	–	2	all	–
F_5	–	3	all	–
F_6	–	–	all	$\approx$ 200 per language
F_7	–	–	all	–
F_8	–	–	all	–
F_9	2	–	all	–
F_{10}	3	2	160	–
F_{11}	3	2	200	–
F_{12}	3	3	200	–

Table 4: Parameter settings for each feature category according to the PAN corpus evaluation.

It should be highlighted that, due to the low accuracies of some feature categories, not all F_i appear within the feature category combinations in Table 5.

$\mathbb{F}$	$\varnothing_{\mathcal{C}_{SP}}$	$\varnothing_{\mathcal{C}_{EN}}$	$\varnothing_{\mathcal{C}_{GR}}$	$\varnothing$	(weighted) $\varnothing$
$\{ F_1, F_3, F_9 \}$	**80** %	**90** %	**70** %	**80** %	**77.14** %
$\{ F_1, F_3, F_7, F_8, F_{12} \}$	**80** %	80 %	65 %	75 %	71.42 %
$\{ F_1, F_2, F_3 \}$	**80** %	80 %	55 %	71.67 %	65.71 %
$\{ F_1, F_4, F_9 \}$	**80** %	80 %	60 %	73.33 %	68.57 %
$\{ F_1, F_3, F_9, F_{11}, F_{12} \}$	**80** %	80 %	55 %	71.67 %	65.71 %
$\{ F_7, F_9, F_{11} \}$	60 %	60 %	50 %	56.67 %	54.28 %
$\{ F_3, F_6, F_7, F_{11}, F_{12} \}$	60 %	50 %	55 %	55 %	54.28 %
$\{ F_2, F_5, F_6 \}$	**80** %	40 %	40 %	53.33 %	45.71 %
$\{ F_3, F_7, F_9 \}$	20 %	70 %	50 %	46.67 %	51.43 %
$\{ F_4, F_6, F_7 \}$	40 %	40 %	60 %	46.67 %	51.43 %

Table 5: Evaluation results according to the PAN corpus.

In order to test the language independence of our method, we decided to extend the PAN corpus with an additional german dataset denoted by $\mathcal{C}_{DE}$. The $\mathcal{C}_{DE}$ dataset consists of 400 documents, extracted from various German-speaking e-Books from the publicly available "Online Book Catalog" of the Gutenberg Project website:

`http://www.gutenberg.org/browse/languages/de`

$\mathcal{C}_{DE}$ includes 40 problem-cases, where each case includes nine sample documents of $\mathcal{A}$ and one document from an author, who claims to be $\mathcal{A}$. The average size of one document is 18 KByte, where each document comprises $\approx$ 4KByte of noisy data. Table 6 summarizes the results according to the extended PAN corpus.

$\mathbb{F}$	$\varnothing_{\mathcal{C}_{SP}}$	$\varnothing_{\mathcal{C}_{EN}}$	$\varnothing_{\mathcal{C}_{GR}}$	$\varnothing_{\mathcal{C}_{DE}}$	$\varnothing$	(weighted) $\varnothing$
$\{ F_1, F_3, F_9 \}$	**80** %	**90** %	**70** %	67.5 %	**76.86** %	72 %
$\{ F_1, F_3, F_7, F_8, F_{12} \}$	**80** %	80 %	65 %	**77.5** %	75.63 %	**74.67** %
$\{ F_1, F_2, F_3 \}$	**80** %	80 %	55 %	75 %	72.5 %	70.67 %
$\{ F_1, F_4, F_9 \}$	**80** %	80 %	60 %	62.5 %	70.63 %	65.33 %
$\{ F_1, F_3, F_9, F_{11}, F_{12} \}$	**80** %	80 %	55 %	62.5 %	69.38 %	64 %
$\{ F_7, F_9, F_{11} \}$	60 %	60 %	50 %	60 %	57.5 %	57.33 %
$\{ F_3, F_6, F_7, F_{11}, F_{12} \}$	60 %	50 %	55 %	62.5 %	56.88 %	58.67 %
$\{ F_2, F_5, F_6 \}$	**80** %	40 %	40 %	65 %	56.26 %	56 %
$\{ F_3, F_7, F_9 \}$	20 %	70 %	50 %	67.5 %	51.86 %	60 %
$\{ F_4, F_6, F_7 \}$	40 %	40 %	60 %	60 %	50 %	55 %

Table 6: Evaluation results according to all corpora.

In our experiments we observed several phenomenons, where the most important one is that the parameter settings seems to have a very strong influence on the results. In Table 7 we list alternative parameter settings for the best feature category combination $\mathbb{F} = \{ F_1, F_3, F_9 \}$ regarding the PAN corpus (without $\mathcal{C}_{DE}$).

$\mathcal{F}_3$, n-Gram	$\mathcal{F}_3$, Top-t	$\mathcal{F}_9$, n-Gram	$\mathcal{F}_9$, Top-t	$\varnothing_{\mathcal{C}_{SP}}$	$\varnothing_{\mathcal{C}_{EN}}$	$\varnothing_{\mathcal{C}_{GR}}$	$\varnothing$	(weighted) $\varnothing$
7	100	2	all	**80** %	90 %	**70** %	**80** %	**77.14** %
6	100	2	all	**80** %	**100** %	65.50 %	82.67 %	**77.14** %
7	100	2	all	**80** %	80 %	**70** %	76.67 %	74.28 %
6	200	2	all	**80** %	**100** %	55 %	78.33 %	71.42 %
7	100	2	160	**80** %	80 %	60 %	73.33 %	68.57 %
7	100	2	160	**80** %	80 %	55 %	71.67 %	65.71 %
2	100	2	all	**80** %	**100** %	40 %	73.33 %	62.86 %
3	all	2	all	60 %	80 %	55 %	65 %	62.86 %
2	all	2	all	**80** %	80 %	45 %	68.33 %	60 %
6	all	2	all	40 %	80 %	50 %	56.67 %	57.14 %

Table 7: Evaluation results for $\mathbb{F} = \{ F_1, F_3, F_9 \}$ and various settings according to the PAN corpus.

It can be seen easily that the resulting (simple) accuracies, which range between $56.67\,\%$ and $80\,\%$, are highly depending on the various settings.

6 Conclusion & future work

In this paper we presented an intuitive scheme to automatically verify alleged authorships of text documents. Our method provides several benefits as for instance language independence (at least for Indo-European languages) and also independence of linguistic resources (e.g. ontologies, thesauruses, language models, etc.). A further benefit is the low runtime of the method, since there is no need for deep linguistic processing like POS-tagging, chunking or parsing. Another benefit is that the involved components within the method can be replaced easily as for example the style deviation method, the acceptance-threshold or the feature categories including their parameters. Moreover, the components can be extended or combined e.g. through ensemble-techniques (combination of several style deviation methods).

Unfortunately, besides benefits our approach includes several pitfalls, too. One of the biggest challenges, for example, is the inscrutability of the methods parameter-space, due to the fact that the number of configuration settings is near infinite. Such settings include for instance the number of extracted features from a specific category or feature-specific parameters (e.g. n-Gram sizes, k-prefix/suffix lengths, etc.). Therefore, it must be investigated how to find the very best corpus-independent parameter settings. This could be accomplished, for instance, with evolutionary algorithms.

Another challenge that remains unsolved is the influence of the text topic, which can distort the verification results. For some feature categories, e.g. F_3 (*Letter n-Grams*), we encountered the presence of content words which are mostly topic related. Yet it is not clear if the verification results remains the same if training documents derive from different sources (e.g. e-Mails, scientific papers, social networks messages, movie reviews, etc.), which may differ heavily in register, genre and style. Hence, a more detailed investigation is required in the near future, which involves special corpora construction.

7 Acknowledgements

This work was supported by the CASED Center for Advanced Security Research Darmstadt, Germany funded by the German state government of Hesse under the LOEWE programme (http://www.CASED.de).

References

1. CiteSeer: The College of Information Sciences and Technology, The Pennsylvania State University, USA (2013), http://citeseerx.ist.psu.edu/index
2. Halvani, O.: Autorschaftsanalyse im Kontext der Attribution, Verifikation und intrinsischer Exploration. Master thesis (2012)
3. Koppel, M., Schler, J.: Authorship Verification as a One-Class Classification Problem. In: Proceedings of the twenty-first international conference on Machine learning. pp. 62–. ICML '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1015330.1015448

4. Library, T.A.D.: Association for Computing Machinery, New York, NY, USA (2013), http://dl.acm.org
5. Loose, F.: Paarweise Autorenschaftsverifikation von kurzen Texten. Master thesis, Bauhaus-Universität Weimar, Fakultät Medien, Medieninformatik (May 2011)
6. Sapkota, U., Solorio, T.: Sub-Profiling by Linguistic Dimensions to Solve the Authorship Attribution Task. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop) (2012), http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#SapkotaS12
7. SpringerLink: Springer-Verlag GmbH, Heidelberg, Germany (2013), http://www.springerlink.com
8. Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. J. Am. Soc. Inf. Sci. Technol. 60(3), 538–556 (Mar 2009), http://dx.doi.org/10.1002/asi.v60:3
9. Tax, D.M.J.: One-Class Classification. Concept Learning In the Absence of Counter-Examples. Ph.D. thesis, Delft University of Technology (2001), http://www-ict.ewi.tudelft.nl/˜davidt/thesis.pdf