Author Identification by Automatic Learning

Jordan Frery, Christine Largeron
Laboratoire Hubert Curien, Universitié Jean Monnet, Saint-Etienne, France
Mihaela Juganaru-Mathieu
Institut H. Fayol, École Nationale Supérieure des Mines de St Etienne, France
email: jordan.frery@gmail.com, christine.largeron@univ-st-etienne.fr, mathieu@emse.fr

***Abstract*—The problem of author identification can be defined in the following way: given a set of documents written by an author and a new document, we have to decide if this last one was written or not by the same author as the other documents. For solving this problem we have suggested and implemented various approaches: counting method, vote technique and supervised learning which explore several models of document representation. The experiences carried out using the collections of PAN-CLEF 2014 challenge have confirmed the interest of our approaches as well as their performance in terms of running time.**

I. Introduction

Everyone is able to tell sometimes: "This song seems to be from Supertramp or this music from Chopin". Listening an excerpt allows to identify the singer or the music's style, even if the singer is or seems unknown. For an instrumental piece we are able to recognize the author, the interpreter, or some instruments. For a textual document, the problem of author identification (i.e. profiling or recognition) is quite similar and appears very often, not only in the field of literature. Thus, for example, security or forensics are also confronted with this issue. For example, Abbasi and Chen [1] did a deep analysis on forum messages from Dark Web to detect criminal profiles.

In the context of documents [1], three main problems may arise:

- detect various characteristics of the author like his age, his gender or his education level; this problem is named Author Profiling [2]
- decide who has written a given text among several known authors; this problem is named Author Verification or Author Recognition [3]
- identify if an author has written or not a given document, using a small collection of documents, all written by this author; this problem is named Author Identification [4].

To solve these problems, the documents must be represented in a suited space and then compared. We think that there is not an unique model of documents and that the choice of the representation space should depend on the language and the type of documents (novel, mail, news, etc.). For this reason, in our work we introduce several representation spaces. We suggest also to formalize the Author Identification task as a supervised clustering problem and we propose three methods to solve it: DCM, DCM-voting and DCM-classifier.

[1]A document means any textual production like blog posts, messages, newspaper articles, programs.

The following section II is dedicated to the state of the art. Section III describes the vector spaces that we choose to represent the documents. Section IV presents our methodological approach and the three solving methods. The experiments and the results obtained on the corpora of PAN-CLEF 2013 and 2014 are detailed in Section V. Conclusions and perspectives are given in the last Section.

II. Related work

The Author Identification can be seen as a classification problem of texts: *"Given a set of documents written by a same author, the set can be large or composed of only one element, we have to decide if a new document has been written by the same author as the others"*. We have to solve a problem of classification having as response a binary value ("yes" or "not") or a probability to belong to the set of known documents. However, one of the specificities of this problem is that only elements belonging to one of the two possible classes are given: the documents having the same author, but the second class is not explicitly described. Moreover, sometimes the number of positive examples is reduced to only one document and, the task becomes much more difficult.

To mitigate the absence of negative examples, one can try to produce some of them. This way is explored by different authors among which Seidman [5] who builds a class of impostors randomly chosen on the Web on the basis of ten more frequent words in the available documents. Other authors, like Zhang et al. [6] and Halvani et al. [7] transform this problem of classification with two classes into a problem with several classes, either by adding external classes or by dividing the initial classes into several. These same authors [7] increase the size of the class containing the known documents when this last one is reduced to only one. Thus, these approaches allow to transform the problem into a classical form of classification, but during the construction of the set of negative examples there is a risk to take some documents very different from the known documents.

In addition to the question of the available data to solve the problem, Author Identification task is then confronted with two classical questions in text mining: how to represent the documents and, once the space of representation chosen, which methods apply to solve the problem of ranking? As we already noticed, Author Identification can be realized using very different types of documents: emails [8], [9], programs, parts of a literary document, unstructured texts [6], even parts of chat [10] or sequences of Unix commands [11]. We think that the choice of the representation spaces and of the features used to describe the documents, we speak about the *stylometric*

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

features, must depend on the type of documents, sometimes the language and also on the quality of the initial text.
The features known as "specific to the applications" relate rather to the number of the tabulations and other separators like the position of the brackets and hooks closing for the programs or the blank lines for the emails. The semantic features are taken into account mainly for web documents (forum and chat). They correspond to abbreviations, frequent conclusive words, like *well*, or of the concentrated transcriptions of the oral expressions like *sse u*. Finally, syntactic features, like spelling mistakes or abbreviations, are also considered. However, if we have to work within a generic framework with documents in various languages and various genres, we rather use features based on characters or words, or successive characters or words (n-grams) [9], [11]. We can also have recourse to part-of-speech tagging, but its use increases considerably the processing time [4] and the results will depend on the tagger's quality [12].

When the choice of these features is made, the documents can be transformed into vectors, using the $tf-idf$ model or the frequency vector. Then, according to the representation of the text adopted, one can compare the documents using classical similarity functions such as the cosine, the correlation, more rarely data compression measures such as Fast Compression Distance like in [13] or Common N-Gram dissimilarity like in [14].

Concerning the resolution of the problem of classification itself, one can apply classical methods such as k-NN [6], [15], [7] or SVM [12]. Some authors, like [16], [14], [17], propose methods based on the choice of a threshold or a vote and on formulas to computate the distance between the unknown document and the others. The differences between these approaches lie in the phase of preprocessing, the extraction of the characteristics, the choice of the threshold and the function of dissimilarity.

Compared to former works, our contribution is within a broader framework: our objective is to propose a generic methodology, applicable to collections very different as well by the kind of the documents as by the language. This requires an approach allowing an automatically choice of the textual representation the most adapted for a given collection.

III. Problem statement and representation of documents

The author identification problem can be formulated as follows. Given a corpus composed of documents of the same type (mail, blog post, novel, instructions, *etc.*) written in the same language (English, French, Java, *etc.*), we have for each problem p one or several documents A_p from the corpus which have been written by a same author and one document u_p whose author is unknown. The task consists in deciding if u_p has been written or not by the same author as the documents of A_p.

If a training set is available, in other words, if we have a set of problems P such that for each problem p in P we know if u_p has been written or not by the same author as the associated documents A_p, then the task can be defined as a supervised clustering problem which can be solved with machine learning methods. The difficulty lies in the choice of a suited representation space for the documents and in the definition of descriptive features which allow to predict effectively if the unknown documents $u_p, p \in P$ have the same author as their associated documents A_p.

	Representation spaces	
	Term	Model
$R1$	Character 8-grams	tf-idf
$R2$	Character 3-grams	tf-idf
$R3$	Word 2-grams	tf-idf
$R4$	Word 1-gram	tf-idf without the 30% most frequent words
$R5$	Word 1-gram	tf-idf without stop words
$R6$	Sentences	Average and standard deviation of words per sentence
$R7$	Vocabulary diversity	Words number divided by occurrences number
$R8$	Punctuation	average of marks per sentence for "," ";" ":" "(" ")" "!" "?"
$R678$	Concatenation	R6 + R7 + R8

TABLE I: The proposed representation spaces

Among the well-known models usually used to represent the documents, there is the tf-idf introduced by [18] according which a document d is represented by a vector $(w_1, \ldots, w_j, \ldots, w_{|T|})$ where the weight w_j of the term t_j in d corresponds to the product of the term frequency tf_j in d and the inverse document frequency $idf(j)$ of t_j. This model is very efficient to identify terms (characters, words, sequences of words or characters corresponding to n-grams) frequent in the given document and rare in the others but, as mentioned in introduction, other characteristics can be taken into account to describe the documents. Indeed, we think that there is not an universal model suited to all the documents and that the choice of the representation space must be done in function of the type and the language of the corpus.

This leads us to propose different representation spaces given in Table I. In addition to the tf-idf model defined for the words, after elimination of the stop words with a dictionary (R5) or by a frequency based filtering (R4), sequences of words or characters (R1, R2, R3), we introduce three other models (R6, R7 and R8) to evaluate the author's writing style. In R6, a document is described by the average and standard deviation of the number of words per sentence. In R7, a measure of the diversity of the vocabulary is associated to the document. This measure corresponds to the number of distinct words used divided by the total number of words occurrences (*i.e.* the document length). The model R8 corresponds to the Salton's model where the punctuation marks are considered instead of the terms (words or characters). Finally, R678 is a mixture of the three previous models: each document is represented by a vector giving the average per sentence of the punctuation marks ":" , ";" , ",", the average and standard deviation of the number of words per sentence and the diversity of the vocabulary.

IV. DCM, DCM-voting and DCM-classifier

The representation spaces being chosen, the documents can be compared using similarity measures like the Cosine, the correlation coefficient or the Euclidean distance and then

one of the proposed methods (DCM, DCM-voting, DCM-Classifier) can be applied to solve the identification problem. The first method DCM solves a problem p of P using only the similarities between the vectorial representations of the documents in one space. The two other methods, based on DCM, allow to combine different representation spaces, with a voting technique in the case of DCM-voting, with a supervised learning method which requires the definition of predictive features for DCM-classifier. These methods are detailed in the following subsections.

A. Similarities counting methods: DCM and DCM-voting

Given a problem $p \in P$ defined by a set A_p of documents written by a same author, an unknown document u_p whose author is undetermined, represented in the same space, and a threshold δ, the method *DCM for Dissimilarity Counter Method*, detailed in the procedure *Dissimilarity Counter Method*, provides the value *True* if u_p has the same author as the documents of A_p or the value *False* otherwise. This method exploits the similarities (or distances) between the available documents. The document u_p is assigned to the class *same author* if the most of the documents in A_p are nearest from u_p than at least one document of A_p. More precisely, the smallest similarity of each document d_x of A_p with the other elements of A_p is computed and then compared to the similarity between d_x and u_p. If the first one is lower, an indicator is incremented. When all the documents A_p have been studied (end of the first **for**), it gives the proportion of documents of A_p which are nearer from u_p than from other documents of A_p and if this proportion is higher than the threshold δ, DCM decides that u_p and the documents of A_p are from the same author.

procedure DISSIMILARITY COUNTER METHOD
  Input data: A_p, u_p, δ
  Result: *True* if A_p and u_p have the same author, *False* otherwise
  A_p : set of known documents from a same author
  u_p : unknown document
  δ : threshold
  smin : minimum of similarity for the decision

  count $\leftarrow 0$
  for $d_x \in A_p$ **do**
    smin $\leftarrow 1$
    for $d_y \in A_p - \{d_x\}$ **do**
      compute $s(d_x, d_y)$ // similarity betwwen d_x and d_y
      if *smin* $> s(d_x, d_y)$ **then**
        smin $\leftarrow s(d_x, d_y)$
      end if
    end for
    if $s(u, d_x) >$ *smin* **then**
      count $\leftarrow$ *count* $+ 1$.
    end if
  end for
  if *count* $> \delta$ **then return** *True*
  else return *False*
  end if
end procedure

This method is called DCM for Dissimilarity Counter Method because the decision is based on the smallest similarity among the known documents or, equivalently the highest dissimilarity. It consists to assignn the document to the class known if it is not the most dissimilar from a majority of known documents. DCM is easy to implement and it allows to solve a problem p of P independently of the others but it presents the drawback to exploit only one representation space.

To overcome this limit, one can use the *DCM-voting* method which consists in applying *DCM* with several representation spaces, in odd number if possible. Then the unknown document is assigned to the majority class returned (known or unknown). However, as outlined above, the different representation spaces are not equivalent and they should not have the same importance in the final decision but it is very difficult, even for an expert, to set their weights as well as the value of the threshold δ. For these reasons, a more general learning method, called DCM-classifier, has been introduced: it allows to exploit simultaneously several representation spaces and to automatically learn their weights in the final decision.

B. DCM-classifier

In supervised learning, we consider a subset of problems $P_A \subset P$ for which we know if the unknown documents u_p are or not from the same author as the associated documents *i.e.* the class (*same author* or *different author*) of the unknown documents belonging to P_A is available. This subset P_A is split into a training set P_a used to build the decision model and a test set used to evaluate it. During the learning step, the training set is used to learn a model which maps the descriptive features with their class. Then, this model can be used to identify the author of a new document whose class is unknown. The accuracy of the prediction depends greatly on the predictive power of these descriptive attributes that we propose to define as follows.

For each representation space R_v, $v \in \{1, .., V\}$, each document u_p is represented by two features $count_v(u_p)$ and $mean_v(u_p)$ respectively defined with the similarity measure s by :

$$count_v(u_p) = \frac{1}{|A_p|} |\{di \in A_p / min\{s(d_i, d_j), d_j \in A_p - d_i\} < s(d_i, u_p)\}| \quad (1)$$

$$mean_v(u_p) = \frac{1}{|A_p|} \times \sum\nolimits_{d_i \in A_p} s(d_i, u_p) \quad (2)$$

$count_v(u_p)$ gives the number of associated documents which are more similar to u_p than to other documents while $mean_v(u_p)$ is the average of the similarities between u_p and the associated documents of A_p.

A last feature $TOT_{count}(u_p)$ is also computed to obtain a more synthetic representation. It is equal to the mean of the values obtained for count on the different representation spaces and, it is defined by:

$$TOT_{count}(u_p) = \frac{1}{V} \sum_{v=1}^{V} count_v(u_p) \quad (3)$$

Thus, during the learning step, we consider the unknown documents of the problems P_a described by the numerical predictive features and by their true class ($count_v(u_p)$ and $mean_v(u_p)$, $\forall v \in \{1, .., V\}$, $TOT_{count}(u_p)$ and $class(u_p)$). Several learning methods (SVM, *etc*) can then be used to

	EN	EE	SP	GR	DR	DE	Total
training2013	10	-	4	20	-	-	34
eval2013	20	-	10	20	-	-	50
training2014	100	200	100	100	100	96	696
eval2014	200	200	100	100	100	96	796

TABLE II: Number of problems per corpus

solve the problem. In our experiments, we choose decision trees because the features selection is included into the process and the parameters of the model are automatically determined.

V. Experimentation and results

A. Datasets and evaluation measures

We have evaluated the three methods: DCM, DCM-Voting and DCM-classifier on the corpora created for the challenges PAN CLEF 2013 and PAN CLEF 2014[2]. For each year, we have a training set (training), used to learn the model, and a testing set (eval) used for the evaluation. Each set of PAN is composed of several corpora and each corpus contains different problems of the same language (European) and the same genre (novel, essay, articles...). In 2013 the two sets, training2013 and eval2013, were composed of documents in three languages: English, Spanish and Greek when in 2014, they were composed of six corpora (EN: English Novel, EE: English Essay, SP: Spanish Article, GR: Greek Articles, DR: Dutch Review, DE: Dutch Essay). Each corpus has a certain number of problems as shown in Table II. Each problem is composed of "known" documents (written by the same author), at least one, and one unknown document. The task consists in determining if this last one has been written by the same author as the "known" documents.

The results obtained on each corpus from the sets eval2013, training2014 and eval2014 have been evaluated thanks to the usual indicators: precision, recall and F1 measure. The error rate indicated by the $F1$ measure being syntactical, the methods have also been compared with the area under the curve (AUC), the indicator $c@1$, the product of these two indicators and the execution time. The indicator $c@1$ allows to give more importance to a correct answer than the absence of decision (corresponding to a probability to belong to a class equals to 0.5). This indicator is defined by:

$$c@1 = \frac{1}{n}(n_c + \frac{n_c}{n} n_u)$$

where n is the number of problems in the corpus, n_c is the number of correct answers, n_u the number of problems with no decision.

B. Results on the collection 2013

For the DCM method, we used the representation $R1$ (character 8-grams) that gave the best results on the training set of 2013, with a threshold δ equals to $\frac{|A_p|}{2}$ when for DCM-voting, we choose the representation spaces R1, R2, R3, R4 and R478 (cf. table I). For the representation spaces based on n-grams words or n-grams characters, documents are represented with the $tf - idf$ model. The table III shows the results yielded by the three methods on the collection eval13 and the best results obtained during the competition per corpus and then on all the corpora. The low precision of DCM-classifier on the Spanish corpus (SP) can be explained by the lack of problems for this language (only four) in the training corpus which makes harder the design of an effective model. The three methods yield satisfying results. However, DCM and DCM-voting are limited to problems that are composed by at least two known documents. If we compare the results obtained by our methods with those of the winner of the competition per corpus, we observe that DCM-classifier is the best only on the Greek corpus with 85%, but on the other hand, on all the corpora, DCM-voting as well as DCM-classifier yield the best results or equivalent to the winner of the competition for all evaluation measures (F1, precision and recall).

[2]http://clef2014.clef-initiative.eu/

	DCM	DCM-vot.	DCM-classi.	Best result
English	73.3%	76.7%	75%	80% [5]
Spanish	77.3%	81.8%	60%	84% [7]
Greek	73.3%	82%	85%	83% [5]
Score F1	73.1%	76.7%	76%	75.3% [5]
Precision	74.4%	78.1%	76%	75.3% [5]
Recall	71.8%	75.3%	76%	75.3% [5]

TABLE III: $F1$ measure for the three methods on each corpus and $F1$, $precision$ and $recall$ on all the corpora eval2013.

Corpus	EN	EE	DR	DE	SP	GR
Problems#	100	200	100	96	100	100
Problems# with $\|A\| = 1$	100	57	99	62	0	20
AUC	89%	70%	68%	91%	77%	76%

TABLE IV: Results from the 10-cross validation with DCM-classifier on training2014

C. Results on the collection 2014

The set 2014 has a higher number of problems, with a greater number of documents type, than the collection 2013 and it is harder to process since more than half of the problems are composed of only one known document ($|A| < 2$); which makes the methods DCM and DCM-voting ineffective and irrelevant. For this reason, only DCM-Classifier has been evaluated, firstly on the training set using the 10-cross validation which consists in splitting the corpus in two subsets, one to build a model and the other one to test it, secondly on the evaluation set for the challenge. The results obtained by cross validation on all the training set are shown in the table IV. They confirm the performances of the DCM-classifier method. The table V contains the official results obtained in the PAN14 competition in Author Identification [19] which compare our method with the ones from the other participants. DCM-classifier allowed us to be ranked at the second place in this competition, with the best result on the corpus EE and the second one on the corpus DE. Moreover, it yielded good results in a very short time. We can note that the execution times from the winner of the competition are in average around 3 hours when the execution time of DCM-Classifier is around few seconds.

Corpus	EN	EE	DR	DE	SP	GR
AUC	61 %	72%	60%	90%	77%	68%
c@1	59 %	71%	58%	90%	75%	64%
Final rank	7/13	**1/13**	6/13	**2/13**	4/13	7/12
Time (minutes)	3:10	0:54	0:08	0:29	1:00	0:57
Execution time rank	3/13	3/13	3/13	4/13	3/13	3/12

TABLE V: Results with DCM-classifier on eval2014

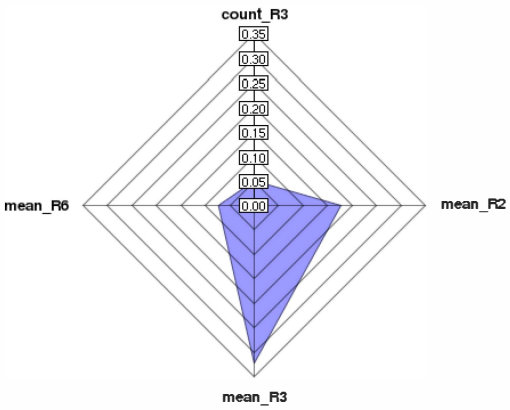

Fig. 1: Attributes importance for GR corpous

One of the advantages of DCM-classifier, based on decision trees, is to highlight the characteristics that allow the most to identify the author of one document regarding the type of the corpus. Indeed, the documents are described by attributes computed on different representation spaces but the learning of our model includes a selection phase where the most discriminant attributes are selected. Thus, with our approach we can identify the best predictive features for each corpus. For example, the Figure 1 shows the different representation spaces used for the GR corpus. In our experiments, we have seen that these spaces and the attribute weights are very different from one corpus to another. This result confirms the interest of combining several representation spaces.

VI. Conclusion

In order to solve the author identification problem, we proposed three methods. The first one, DCM, is a counting method that exploits only one representation space. It yields good results but its effectiveness will highly depend on the number of known documents since it can only handle problems that contain at least two known documents from the same author. The extension of this method, DCM-voting, allows to overcome one of the limits of DCM since it uses several representation spaces but it can not handle problems with only one known document. A second extension, DCM classifier, based on decision trees which exploits the results of DCM to build the input attributes, overcomes all the limits of DCM. In the evaluation of the challenge PAN-CLEF 2014, the method DCM-classifier, with a general score of 70.7 %, got the second best performance, very near from the winner, but with execution times very low and the results obtained on the evaluation are coherent with those obtained on the training phase. Finally, our work confirms the interest of combining several representation spaces and an advantage of our approach is that it allows, with a training phase, the selection of those that seem the most suited for the chosen corpus.

References

[1] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group web forum messages," *Intelligent Systems, IEEE*, vol. 20, no. 5, pp. 67–75, 2005.

[2] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, and G. Inches, "Overview of the Author Profiling Task at PAN 2013," in *Proceedings of PAN at CLEF 2013*, 2013.

[3] H. V. Halteren, "Author Verification by Linguistic Profiling: An Exploration of the Parameter Space," *ACM Trans. Speech Lang. Process.*, vol. 4, pp. 1:1–1:17, Feb. 2007.

[4] P. Juola and E. Stamatos, "Overview of the Author Identification Task at PAN 2013," in *Information Access Evaluation. Multilinguality, Multimodality, and Visualization. 4th International Conference of the CLEF Initiative, CLEF* (P. Forner, R. Navigli, and D. Tufis, eds.), pp. 23–26, 2013.

[5] S. Seidman, "Authorship Verification Using the Impostors Method," in *Notebook for PAN at CLEF 2013*, pp. 13–16, 2013.

[6] C. Zhang, X. Wu, Z. Niu, and W. Ding, "Authorship identification from unstructured texts," *Knowledge-Based Systems*, vol. 66, no. 0, pp. 99 – 111, 2014.

[7] O. Halvani, M. Steinebach, and R. Zimmermann, "Authorship Verification via k-Nearest Neighbor Estimation Notebook for PAN at CLEF 2013," in *Notebook for PAN at CLEF 2013*, 2013.

[8] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-Mail Content for Author Identification Forensics," *SIGMOD Rec.*, vol. 30, pp. 55–64, Dec. 2001.

[9] M. Chaurasia and D. S. Kumar, "Natural Language Processing Based Information Retrieval for the Purpose of Author Identification," *International Journal of Information Technology and Management Information Systems (IJITMIS)*, vol. 1, no. 1, pp. 45–54, 2010.

[10] G. Inches and F. Crestani, "Overview of the International Sexual Predator Identification Competition at PAN-2012," in *Proceedings of PAN at CLEF 2012*, 2013.

[11] B. Szymanski and Y. Zhang, "Recursive data mining for masquerade detection and author identification," in *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC*, pp. 424–431, June 2004.

[12] D. Vilariño, D. Pinto, H. Gómez, S. León, and E. Castillo, "Lexical-Syntactic and Graph-Based Features for Authorship Verification," in *Notebook for PAN at CLEF 2013*, 2013.

[13] D. Cerra, M. Datcu, and P. Reinartz, "Authorship analysis based on data compression," *Pattern Recognition Letters*, vol. 42, no. 0, pp. 79 – 84, 2014.

[14] R. Layton, P. Watters, and R. Dazeley, "Local n-grams for Author Identification Notebook for PAN at CLEF 2013," in *Notebook for PAN at CLEF 2013*, 2013.

[15] M. R. Ghaeini, "Intrinsic Author Identification Using Modified Weighted KNN," in *Notebook for PAN at CLEF 2013*, 2013.

[16] M. V. Dam, "A basic character n-gram approach to authorship verification," in *Notebook for PAN at CLEF 2013*, 2013.

[17] M. Jankowska and E. Milios, "Proximity based one-class classification with Common N-Gram dissimilarity for authorship verification task Notebook for PAN at CLEF 2013," in *Notebook for PAN at CLEF 2013*, 2013.

[18] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[19] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. A. Sanchez-Perez, and A. Barrón-Cedeño, "Overview of the author identification task at pan 2014," in *Proceedings of CLEF PAN 2014*, 2014.