UJM at CLEF in Author Identification
Notebook for PAN at CLEF 2014

Jordan Fréry[1], Christine Largeron[1], and Mihaela Juganaru-Mathieu[2]

[1] Laboratoire Hubert Curien, Université de Lyon, F-42023, Saint-Etienne, France
[2] Institut H. Fayol, École Nationale Supérieure des Mines, F-42023 Saint-Etienne, France
jordan.frery@gmail.com, christine.largeron@univ-st-etienne.fr, mathieu@emse.fr

Abstract This article describes our proposal for the Author Identification task in the PAN CLEF Challenge 2014. We adopt a machine learning approach based on several representations of the texts and on optimized decision trees which have as entry various attributes and which are learnt for every training corpus separately for this classification task. Our method ranked us at the 2nd place with an overall AUC of 70.7%, and C@1 of 68.4% and, between the 1st and the 6th place on the six corpora .

1 Introduction

The task Author Identification (AI) in the CLEF-PAN Challenge is to solve a large set of problems like : *given a set A of samples texts, all texts in A are written by only one author and a mysterious document u, determine if u was written by the author of A*. The difficulties of this task are various : the lack of data we have per author: sometimes, A has only one text, some languages that we do not know or we are not able to understand.

We decided to represent the documents in different vector spaces and by various type of features :

- length of the sentences,
- variety of vocabulary,
- words, n-characters grams, n-words gram,
- punctuation marks.

For each feature, we considered two numerical values : a mean and a counter. An other global counter was also used. Because we are not able to indicate or to justify the features which are the most important, we used decision trees, an adapted version of CART, to learn a decision model suited for a kind of document. Thus, each corpus defined by a language and a genre, has its own learned tree.

So, our proposal is based on:

- the proposition of vector space models and attributes that represent the documents in a way as optimal as possible.
- the formulation of Author Verification problem as a supervised classification problem.
- the evaluation of this approach on different groups of problems in the challenge context.

Section 2 describes the vector spaces that we choose to represent the documents. Section 3 is dedicated to the methodological approach. The section 4 presents the experiments and the results obtained on the training set and for the challenge. We will finish with some conclusions and future perspectives.

2 Textual representation

A problem inside a corpus consists in a given set A of documents written by the same author and another document u whose author is unknown. The aim is to decide whether u has the same author as all documents d_i in A.

2.1 Vector space models

In order to represent the textual documents as vectors we use different vector space models. The first one is the well known term frequency-inverse document frequency weighting scheme (tf-idf) introduced by Salton [1]. This model is very efficient to isolate terms (words or characters) that are frequent in one document and not in the others. A document d in a corpus A is represented as a vector of weights $\boldsymbol{d} = (w_1, \ldots, w_j, \ldots, w_{|T|})$ where the weight w_j of the term t_j in d corresponds to the product of the term frequency tf_j of the term t_j in d by the inverse document frequency $idf(j)$ defined by:

$$idf(j) = log \frac{|A|}{|\{d \in A : t_j \in d\}|}$$

This representation can be defined for terms corresponding either to words or characters. Moreover, in order to take into account the variety of the style and vocabulary, we consider representations based on the punctuation, length of phrases and diversity of the vocabulary as detailed in Table 1.

	Representation space		Comparison method
	Term	Model	
$R1$	Character 8-grams	tf-idf	cosine similarity
$R2$	Character 3-grams	tf-idf	correlation coefficient
$R3$	Word 2-grams	tf-idf	correlation coefficient
$R4$	Word 1-gram	tf-idf without the 30% most frequent words	correlation coefficient
$R5$	Word 1-gram	tf-idf without stop words	correlation coefficient
$R6$	Phrases	word per sentence mean, word per sentence standard deviation	correlation coefficient
$R7$	Vocabulary diversity	total number of different terms divided by the total number of occurrences of words	euclidean distance
$R8$	Punctuation	average of punctuation marks per sentence characters taken into account: "," ";" ":" "(" ")" "!" "?"	cosine similarity

Table 1. List of representation spaces and comparison measures

2.2 Documents comparison

Our approach requires to compare all documents inside a corpus using the cosine similarity, euclidean distance or the correlation coefficient. These measures are normalized, between 0 and 1 for the euclidean distance and cosine similarity and, between -1 and 1 for the correlation coefficient. For two documents represented as vectors $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$, the cosine similarity $cos(\boldsymbol{d_i}, \boldsymbol{d_j})$ is defined as follows:

$$cos(\boldsymbol{d_i}, \boldsymbol{d_j}) = \frac{\boldsymbol{d_i} \cdot \boldsymbol{d_j}}{||\boldsymbol{d_i}||\boldsymbol{d_j}||}$$

The cosine similarity equals to 1 when the documents have the same representation. Conversely, if two documents are highly different, cosine similarity will tend to be 0.

The correlation coefficient $corrcoef(d_i, d_j)$ between two documents is given by:

$$corrcoef(d_i, d_j) = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}}$$

where C_{ij} denotes the covariance between the documents $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$.

Table 1 presents the different representation spaces and the measures we used to compare the documents belonging to a corpus. In our methodological approach, we extract two attributes for each representation space of Table 1 in order to represent the unknown documents.

3 Methodological approach

Given a corpus $\mathcal{P}$ containing all the documents having the same language and the same type, we have $p \in \mathcal{P}$ problems to solve and, for each problem there are one or several documents written by the same author and one document (u) whose author is unknown. Thus, the dataset of the supervised learning problem contains all the unknown documents of one corpus, described by 17 attributes but also by the class which has two modalities: SameAuthor or DifferentAuthor. In supervised learning, models are learnt by splitting the dataset into two subsets. The first one, called learning set, is used to learn the model, in our case, decision tree. The second subset, called test set, is used to evaluate the model. The decision tree learnt during the learning step is use to define the class of each unknown document corresponding to a problem. The evaluation of the quality of the decision rules is done by computing the well classification rate or the area under the ROC curve (AUC) obtained by comparing the predicted class and the true class for the unknown documents belonging to the test set. The accuracy of the models depends largely of the attributes predictive power. That leads us to define two attributes per representation space and a global attribute.

3.1 Attributes definition

We use a dissimilarity counter method that we designed during our experimenting on the PAN 2013 corpora in Author identification and which yielded very good results [2].

We chose to use it back for PAN 2014 in a modified version. It is wise to note that this method only works for problems with at least two known texts ($|A| >= 2$).
Given $\mathcal{P}$, the set of problems provided for one corpus defined by A_p the set of documents written by one author and u_p the unknown document for a problem $p, p = 1, ..., |\mathcal{P}|$, such as:

$$\mathcal{P} = \{(A_p, u_p), p \in 1, ..., |\mathcal{P}|\}$$

For each document u_p, corresponding to a given problem, and for each representation space R_v, $v \in \{1, .., 8\}$, we calculate two attributes $count_v(u_p)$ and $mean_v(u_p)$ as follows:

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

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

These two attributes are computed for each representation space. Consequently, since $v \in \{1, .., 8\}$ we have 16 attributes. A last attribute, $TOT_{count}(u_p)$ is built to have a more global representation:

$$TOT_{count}(u_p) = \sum_{v=1}^{8} count_v(u_p)$$

Finally we have 17 attributes describing each unknown document belonging to a problem provided for one corpus composed of the documents having the same language and the same genre.

3.2 Decision tree classifier

For the task of Author Verification, we used the Classification and Regression Trees (CART) algorithm which constructs binary trees using the features and thresholds that yield the largest information gain at each node [3]. The trees are built by using each corpus of the training set separately in such a way to obtain a tree per corpus. We train the classifier with the attributes detailed previously plus the true label for the given unknown documents. At each step, the attribute that best splits the set of unknown documents into the two classes is chosen using the giny impurity. To avoiding overfitting, we apply post-pruning that consists to build the tree which classify the training set perfectly and then prune the tree [4].

For each problem of the corpus, the decision tree has the following information for the unknown document:

- $count_v(u_p), \forall v \in \{1, .., 8\}$
- $mean_v(u_p), \forall v \in \{1, .., 8\}$
- $TOT_{count}(u_p)$
- $class(u_p)$, the true label of a problem

The previous data allow us to build rules in such a way that we classify 100% of problems correctly. In order to handle overfitting we remove all leaves with less than 5% of the total number of problems so we could keep more general rules. Moreover, we decided to not answer problems that were not significant enough. The rule we set is that when the probability for a text to be written by the same author is between 0.4 and 0.6, we change the probability to 0.5 so that we choose to not answer this problem. So finally there are 3 modalities for the class: sameAuthor, differentAuthor or undefined.

4 Experimentation and results

For the learning step, the implementation has been done in Python. We used scikit-learn library [3] for the n-grams representation and for CART.

4.1 Learning

The experimentation has been made on the training corpus which contains 696 problems labelled as DE, DR, GR, EN, EE or SP where D stands for Dutch (DE,DR), GR for Greek, SP for Spanish and E for English (EE,EN). We have essays and review for Dutch (DE,DR) and essays and novels for English (EE,EN). For experimentation, we have made a 10 cross validation for each group of problems in order to evaluate the quality of the decision trees on the training set.

The table 2 shows for each corpus: the number of problems and the result calculated with the area under the ROC curve (AUC) on the training dataset.

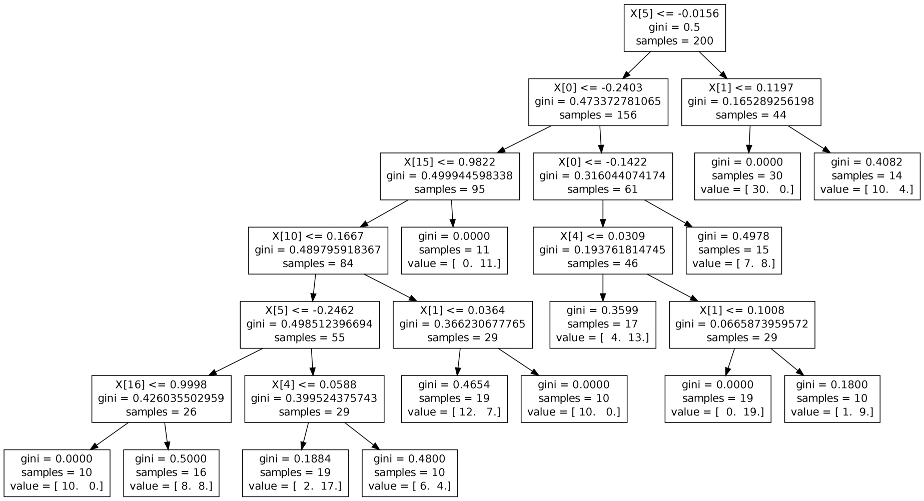
Corpus	EN	EE	DR	DE	SP	GR
Problems#	100	200	100	96	100	100
AUC	89%	70%	68%	91%	77%	76%

Table 2. 10 cross validation on the training corpus

The following tree is the one used over the English essays corpus where "samples" is the number of problem remaining at a node. There are 200 problems to treat.

X[5] = $mean_{R5}(u_p)$
X[0] = $mean_{R3}(u_p)$
X[1] = $mean_{R2}(u_p)$
X[15] = $mean_{R6}(u_p)$
X[4] = $mean_{R1}(u_p)$
X[10] = $count_{R7}(u_p)$
X[16] = $mean_{R8}(u_p)$

[3] http://scikit-learn.org



4.2 Evaluation

The evaluation of the decision trees built during the learning step was done during the competition. The table 3 contains the official results of PAN14 in Author Identification for our team computed by the organizers of the challenge.

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

Table 3. Challenge evaluation results

5 Conclusion

With a overall scores of 0.707 for AUC, 0.684 for C@1 we obtain a final score of 0.484 which is the second best submission. As shown in Table 3, we obtain the 1st rank for the English essays corpus, 2nd for the Dutch essays corpus and 4th for the Spanish corpus. For the previous corpora, the results we obtained were consistent with the ones we had while training our decision tree. However we lost a lot of accuracy on the English novels corpus (near 30% of loss). We would need to study the evaluation corpus to understand why we had such a loss of accuracy. Moreover our approach is not time-consuming as shown in Table 3.

During this challenge we saw that the most difficult part was to gather features that are complemented each other. The use of CART allows to identify good predictive features. However, we have not tried all possibilities for text representations. Moreover, building efficient features, like with the counter method, highly improves the accuracy of CART for some corpora.

References

1. G. Salton, M.M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY, USA (1983)
2. Juola, P., Stamatatos, E.: Overview of the Author Identification Task at PAN 2013. Pamela Forner, Roberto Navigli and Dan Tufis edn., Working Notes Papers of the CLEF 2013 Evaluation Labs
3. L. Breiman, J. Friedman, R.O., Stone, C.: Classification and Regression Trees (1984)
4. Quinlan, J.R.: Simplifying decision trees. Int. J. Man-Mach. Stud. 27(3), 221–234 (Sep 1987)