A Language Independent Author Verifier Using Fuzzy C-Means Clustering

Notebook for PAN at CLEF 2014

Pashutan Modaresi[1,2] and Philipp Gross[1]

[1] pressrelations GmbH, Düsseldorf, Germany
{pashutan.modaresi, philipp.gross}@pressrelations.de
[2] Heinrich-Heine-University of Düsseldorf, Institute of Computer Science, Düsseldorf, Germany
modaresi@cs.uni-duesseldorf.de

Abstract In this work we describe our approach to solve the *author verification* problem introduced in the *PAN 2014 Author Identification* task. The *author verification* task presents participants with a set of problems where each problem consists of a set of documents written by the same author and a questioned document with an unknown author. The task is then to decide whether the questioned document has the same author as the other documents or not. Inspired by a psychological personality model, our approach uses basic lexical feature extraction and fuzzy clustering. Using the created fuzzy clusters, the membership values of documents to the clusters can be computed. The distribution of the cluster membership values will be used finally to solve the verification problem.

1 Introduction

Given a set of documents with known authors, *authorship attribution* is the task of identifying the author of an unseen document. Having a small number of candidate authors, this task can be easily solved using the state-of-the-art approaches[1]. A realistic and common scenario for *authorship attribution* is the *author verification* problem. Given a set of documents written by a single author, the task here is to determine whether a questioned document is written by the same author or not.

The *PAN 2014 Author Identification* task focuses on the *author verification* problem. To be more specific, in this task a multi-lingual corpus is provided which consists of several problems. Each problem contains a maximum of 5 documents written by a single author and a questioned document by an unknown author. The task in then to determine whether the questioned document is written by the same author or not.

The fact that an author may consciously or unconsciously vary his or her writing style, makes the task of *author verification* a hard problem[7]. In this paper we introduce a novel approach for solving the task of *author verification*. For this we extract language independent features from our training corpus and use a fuzzy clustering algorithm to construct our models. Finally using the membership distribution of documents over the clusters, we do solve the verification task.

In Section 2 we define the problem of *author verification* formally and introduce some notations. Section 3 addresses the process of feature extraction and normalization.

The process of clustering and model construction is discussed in Section 4. Section 5 covers the process of verification and scoring. An overview of the evaluation results can be seen in Section 6. Finally in Section 7 the work will be concluded.

2 Problem Statement

In this section we formally define the problem of *author verification* in the context of the *PAN 2014 Author Identification* task.

Let $P = \{D, d_u\}$ be a problem consisting of a set of documents $D = \{d_1, \ldots, d_n\}$ with $1 \leq n \leq 5$ written by a single author, and a questioned document d_u with an unknown author. The task in *author verification* is to determine whether the questioned document d_u is written by the same author or not. We denote the author of a document d_i by $\mathcal{A}(d_i)$. In other words an author verifier φ is a binary classification function of the following form:

$$\varphi(d_u, D) = \begin{cases} 1, & \text{if } \mathcal{A}(d_u) = \mathcal{A}(d_i) \ \ \forall d_i \in D \\ 0, & \text{if otherwise} \end{cases} \tag{1}$$

In the *PAN 2014 Author Identification* task, problems are from 4 different languages, namely Dutch, English, Greek and Spanish. The *author verification* algorithm has to be able to deal with documents from the specified languages. The performance of the *author verifier* will be evaluated according to the area under the ROC curve (AUC) of its probability scores and also based on the c@1 measure[8]. The evaluation process will be discussed in more details in Section 6.

In the following section, we start the description of our algorithm by discussing the feature extraction and normalization step.

3 Feature Extraction and Normalization

Feature extraction is considered as one of the important steps in *author verification*[9]. Different kinds of stylometric features like lexical, syntactic or semantic features have been used for solving the *author verification* task. In order to design an efficient *author verification* algorithm, which can deal with huge amounts of documents, we only consider a limited number of lexical features and construct our learning algorithm in a way that would result in an acceptable performance even with a small number of features. Lexical features have the advantage over the syntactic or semantic features, that this kind of features can be computed very efficiently and without the use of any external knowledge or training.

We represent documents as vectors in $\mathbb{R}^4$. Each component of these 4-dimensional vectors can be computed using the feature extraction functions. Independent of the document language we use the following functions to compute the feature vector components of documents:

Average Sentence Length *(f_{sl})* : Using a sentence detector, sentence boundaries of the document will be detected (In our case we use a regular expression based sentence detector for optimizing the performance). For each sentence s in the document, its length $l(s)$ will be computed. We denote the set of all sentences inside a document with S. Finally the average sentence length of the document can be computed as follows:

$$f_{sl}(d) = \frac{\sum_{s \in S} l(s)}{|S|} \tag{2}$$

Punctuation Marks Usage *(f_{pm})* : Using a predefined set of punctuation marks $T =$ { ( ) , : ; ! ? } the frequency of the elements of the set T inside the document will be computed and finally normalized by the length of the document. With $f(t, d)$ we denote the frequency of the punctuation mark t in document d.

$$f_{pm}(d) = \frac{\sum_{t \in T} f(t, d)}{|d|} \tag{3}$$

Space After Comma *(f_{sac})* : Our experimental results show that whether a space is used after a comma or not, can be a good discriminating feature in the *author verification* task. Let α denote the number of times a comma is followed by a space and β be the number of times a comma is not followed by a space. In his way f_{sac} can be defined as follows:

$$f_{sac}(d) = \frac{\alpha - \beta}{|d|} \tag{4}$$

Analogue to f_{sac} we define f_{sbc} which is the *Space Before Comma* feature. Through this feature, authors that use a space before comma can be discriminated from the ones who do not use a space before comma.

As the extracted features may exhibit significant differences in their range and distribution, out learning algorithm could be more sensitive to features that are in a wider range (e.g. Average Sentence Length). In order to avoid this behavior we use feature normalization through which we can modify the mean and variance of the features using a transformation function. The transformation function that we use in this work is the *min-max* function. Given a feature f, the *min-max* transformation function which is defined as follows:

$$\boldsymbol{f'} = \frac{\boldsymbol{f} - min(\boldsymbol{f})}{max(\boldsymbol{f}) - min(\boldsymbol{f})} \tag{5}$$

In the above formula $\boldsymbol{f}$ denotes the feature vector and $\boldsymbol{f'}$ is the transformed feature vector.

4 Fuzzy Clustering and Model Construction

In this section we illustrate the main idea behind our learning algorithm. We believe that different personality dimensions have a close relationship with the writing style of authors. In psychology, the *Big Five Personality Traits* are 5 dimensions of personality that are used to describe the personality of humans[3]. Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism are the personality dimensions which are described as the factors of the *Big Five* model. Based on these dimensions, each persons personality can be described using a combination of the above dimensions. Inspired by the *Big Five* model, we construct c clusters, where each cluster represents a personality dimension. An author's personality can then be determined by computing his or her membership to these clusters. Finally two authors that have the same (or similar) membership distribution over the clusters would be considered as the same.

For this we collect all the documents in our training set from which we know that they are written by the same author and extract their features (See Section 3). This will result in a matrix $Z = [z_1^{tr}, z_2^{tr}, \ldots, z_N^{tr}] \in \mathbb{R}^{4 \times N}$ where N is the number of collected documents and z_i^{tr} denotes the transpose of the vector z_i. As already mentioned the personality of an author can be determined using his or her membership values to the available clusters. Due to this consideration, we use the *Fuzzy C-Means*[2] clustering algorithm to construct fuzzy clusters. For constructing c clusters, we assign initial cluster membership values for each document in the collection (The collection of these values constructs the partition matrix $U = [\mu_{ik}] \in \mathbb{R}^{c \times N}$). The partition matrix will be updated after each iteration of the algorithm until no significant changes are observable. After initializing the partition matrix randomly, the *Fuzzy C-Means* algorithm can be summarized as follows:

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster centers with $m \in [1, \infty)$

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c \tag{6}$$

Step 2: Compute the distances

$$D_{ik}^2 = \left\| z_k - v_i^{(l)} \right\|^2 = (z_k - v_i^{(l)})^T (z_k - v_i^{(l)}), \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \tag{7}$$

Step 2: Update the partition matrix:

for $1 \leq k \leq N$

if $D_{ik} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ik}/D_{jk})^{2/(m-1)}} \tag{8}$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ik} > 0, \text{ and } \mu_{ik}^{(l)} \in [0,1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1 \tag{9}$$

Until $\left| U^{(l)} - U^{(l-1)} \right| < \epsilon$

We use the cluster information produced by the cluster algorithm, to verify whether two documents are written by the same author or not. The process of *author verification* will be discussed in the following section.

5 Verification and Scoring

In order to find an answer to an *author verification* problem P, we compute the cluster membership values for documents with known authors and documents with unknown authors. Then using the membership values we will decide if the documents have the same author or not.

Given a problem $P = \{D = \{d_1, \ldots, d_n\}, d_u\}$ and c cluster prototypes (centroids) $V = \{v_1, \ldots, v_c\}$ we compute the membership values of the documents with known authors to the constructed clusters. In this way, for each document $d_i,\ 1 \leq i \leq n$ a cluster membership vector $\mu_i = \{\mu_{i1}, \ldots, \mu_{ic}\}$ will be computed where the j-th element in the vectors represents the membership value of the document d_i to the cluster j.

In the same way we compute the cluster membership values of the document with unknown author d_u. This would result in the membership vector μ_u. At this step the cluster membership values for all documents in the problem P are known. Notice that the documents $d_1, \ldots, d_n$ are assumed to be written by the same author. Theoretically we would expect that the cluster membership vectors of these documents look very similar to each other. Experimental results show that this is usually not the case, which relies on the fact the authors write in different psychological states.

In order to solve the above problem, for the documents with known authors, we compute a mean cluster membership vector. Through this vector a more stable estimation of membership to available personality dimensions can be made. The mean cluster membership vector of a set of documents $d_1, \ldots, d_n$ with known authors can be computed as follows:

$$\tilde{\mu} = \frac{\sum_{i=1}^{n} \mu_i}{n} \tag{10}$$

Now using the cosine similarity between the average cluster membership vector of documents with known authors and the questioned document, the similarity between these two vectors can be computed. The cosine similarity between these two vectors is defined as follows[6]:

$$S_{\tilde{\mu},\mu_u} = \frac{\tilde{\mu} \cdot \mu_u}{\|\tilde{\mu}\| \, \|\mu_u\|} \tag{11}$$

Through the cosine similarity measure we compute the angle between the vectors. A cosine values of 0 means that the vectors are orthogonal to each other and a cosine value of 1 means that the vectors are identical. Through the cosine similarity measure we assigned a score to each problem. Additionally we need a transformation function which can return binary values for author verification problem. In Section 2 we defined the function $\varphi(d_u, D)$. Here we modify this definition and redefine the function:

$$\varphi(d_u, D) = \begin{cases} 1, & \text{if } S_{\tilde{\mu},\mu_u} \geq 0.5 \\ 0, & \text{if otherwise} \end{cases} \tag{12}$$

Using the above function definition, for each problem P it can be decided if the documents inside P belong to the same author or not. A value of 1 means that the documents inside P have the same author and a value of 0 means that the questioned document has a different author than the documents with a known author.

6 Evaluation Results

In order to evaluate our approach we used the training set provided by the *PAN 2014 Author Identification* task. The training set consists of documents belonging to 4 different languages, namely Dutch, English, Greek and Spanish. Dutch documents are divided into essays and reviews, and English documents into essays and novels. Greek and Spanish documents belong only to the genre Articles. In total we constructed 6 models, where each model corresponds to a specific language and a specific genre.

For constructing the clusters of language L and genre G, we randomly selected 20% of the available training data to create the clusters. The experiments have been repeated 1000 times and average $c@1$ measure of the iterations has been computed. The $c@1$ measure of a single iteration can be computed as follows[8]:

$$c@1 = (\frac{1}{n})(n_c + (n_u \frac{n_c}{n})) \tag{13}$$

where, $n =$ number of problems, $n_c =$ number of correct answers and $n_u =$ number of unanswered problems. The results are summarized in Table 1.

Table 1: Evaluation results for the training set

Language	Genre	#Clusters	m	c@1	AUC	c@1 · AUC
Dutch	Essays	4	4	0.731	0.752	0.549
Dutch	Reviews	3	4	0.680	0.763	0.518
English	Essays	3	3	0.664	0.651	0.432
English	Novels	4	3	0.852	0.852	0.725
Greek	Articles	4	3	0.671	0.697	0.467
Spanish	Articles	3	5	0.684	0.712	0.487

In Table 1 the number of created clusters and the parameter m are also specified. These parameters are the ones that returned the best results during our experiments. As we can see the algorithm returns the best results for the English novels with an $c@1$ value of 0.852. The worst results are also for the English documents but the ones in the genre essays. Even though the $c@1$ values for all languages and genres are greater than 0.66.

Beside the above approach, we evaluate the performance of our algorithm according to the area under the ROC curve (AUC)[4] of its returned probability scores. Table 1 summarizes the results. As it can be seen in the table, the AUC values are consistent and comparable with c@1 values. The reason for this is that the verification algorithm outputs very high probability scores for the positive cases, and very low probability scores for the negative cases.

For ranking the performance of participants in the competition a test corpus has been provided. We have evaluated our algorithm using *Tira*[5] which is a service for running experiments in computer science. Table 2 represents the performance results and also the run-time of our algorithm on the test corpus.

From the performance results based on the test corpus it can be seen that our algorithm performs very well for English *Novels*, and *Essays* reaching a final score of 0.508 and 0.349 respectively. But for the other languages the results are not as satisfactory as expected. This difference between the results indicates that for languages other than English, a deeper feature engineering is needed.

Table 2: Evaluation results for the test set

Language	Genre	c@1	AUC	c@1 · AUC	Runtime (in seconds)
Dutch	Essays	0.635	0.594	0.377	4
Dutch	Reviews	0.500	0.493	0.246	6
English	Essays	0.580	0.602	0.349	6
English	Novels	0.715	0.711	0.508	7
Greek	Articles	0.540	0.543	0.293	4
Spanish	Articles	0.650	0.640	0.416	7

The run-time of our algorithm on different data sets also shows that the introduced algorithm can be efficiently used for large collections of *author verification* problems. This is due to the small number of features that we extract from documents. This has from one side the advantage that the *author verification* problems can be solved very efficiently, but from the other side, it will result in a lower performance for specific languages.

7 Conclusion

In this work we have described our approach to solve the author verification problem introduced in the *PAN 2014 Author Identification task*. Using the *fuzzy c-means* clustering algorithm, we partitioned the provided training set (Section 4) into several clusters. Given an *author verification* problem, we used the membership values of the documents inside the problem to verify whether two documents have the same author or not.

In order to design an efficient algorithm we only considered a limited number of features for each language. This resulted in very low run-times for our algorithm. Accordingly we acquired the 1st place among the participants regarding the run-time of algorithms.

Our introduced approach also revealed sound results for the English language achieving the 1st place for English Novels and the 5th place for English Essays among the 13 participating teams. For other languages we did not get the expected satisfactory results.

The reason for this lies in the small amount of training set that we use for constructing our fuzzy clusters. We also use the same set of features for all available languages which is probably the main reason for insufficient results for languages other than English.

References

1. Argamon, S.: Scalability issues in authorship attribution.kim luyckx. LLC 27(1), 95–97 (2012)
2. Bezdek, J., Ehrlich, R., Full, W.: FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10(2-3), 191–203 (1984)
3. Digman, J.M.: Personality Structure: Emergence of the Five-Factor Model. Annual Review of Psychology 41(1), 417–440 (1990)
4. Fawcett, T.: Roc graphs: Notes and practical considerations for researchers. ReCALL 31(HPL-2003-4), 1–38 (2004)
5. Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Recent trends in digital text forensics and its evaluation. In: Forner, P., MÃijller, H., Paredes, R., Rosso, P., Stein, B. (eds.) Information Access Evaluation. Multilinguality, Multimodality, and Visualization, Lecture Notes in Computer Science, vol. 8138, pp. 282–302. Springer Berlin Heidelberg (2013)
6. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)
7. 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)
8. Peñas, A., Rodrigo, A.: A simple measure to assess non-response. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 1415–1424. HLT ’11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
9. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. 60(3), 538–556 (Mar 2009)