Token-level author diarization using clustering of stylistic contexts

Ivan Grubišić, Milan Pavlović, Author3

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{ivan.grubisic, milan.pavlovic, }@fer.hr

Abstract

This document provides the instructions on formatting the TAR system description paper in LaTeX. This is where you write the abstract (i.e., summary) of the work you carried out within the project. The abstract is a paragraph of text ranging between 70 and 150 words. This document provides the instructions on formatting the TAR system description paper in LaTeX. This is where you write the abstract (i.e., summary) of the work you carried out within the project. The abstract is a paragraph of text ranging between 70 and 150 words.

1. Introduction

In this paper we will focus on the author diarization task proposed on PAN 2016 competition[1]. The aim of this task is to decompose a document into its authorial parts, i.e. to split a text into segments and assign an author to every segment (Koppel et al., 2011; Aldebei et al., 2015). This is one of the unsupervised variants of a well known authorship attribution problem since text samples of known authorship are not available (Rosso et al., 2016). As we will describe, in two out of three subtasks of this task only a correct number of authors for a given document is known.

The simplest variant of authorship attribution problem is about finding the most likely author for a given document, from a set of candidate authors whose authentic writing examples are available (Stamatatos, 2009b; Stein et al., 2011; Ding et al., 2016). Such descirbed problem can be tackeled with supervised machine learning techniques as a single-label multiclass text classification problem, where one class represents one author (Stamatatos, 2009b).

Authorship attribution problem is also known as authorship identification and it is a part of authorship analysis (Stamatatos, 2009b; Ding et al., 2016). Authorship analysis is a field of stylometry and studies information about the authorship of a document, based on features derived from that document (Layton et al., 2013). Moreover, stylometry analyzes literary style with statistical methods (Stein et al., 2011).

Rosso et al. (2016) divided PAN 2016 author diarization task into three subtasks. First subtask is traditionally called intrinsic plagiarism detection (IPD). The goal of this task is to find plagiariarized parts of a document in which 70% of text is written by main author and the rest by one or more other authors. The term *intrinsic* means that a decision whether plagiarized parts exist has to be made only by analysing a given document, without any comparisons with external sources. In the rest of the paper we refer to this subtask as a task *a*.

Other two subtasks are more related to the general task of author diarization. In the second subtask we need to segment a given document and group identified segments by author. In the rest of the paper we refer to the second subtask as a task *b*. Third subtask differs from the second one in the fact that exact number of authors is unkown. In the rest of the paper we refer to the third subtask as a task *c*.

Table 1: Basic characteristics of train datasets. * represents that there is a true author and plagiarism segments which don't have to originate from a single author.

Task	Number of documents	Average length (in tokens)	(min, max) authors
Task *a*	71	1679	(2, 2)*
Task *b*	55	3767	(2, 10)
Task *c*	54	3298	(2, 10)

For all three subtasks a different training datasets are publicly available[1]. Rosso et al. (2016) explain that they are collections of various documents which are part of Webis-TRC-12 dataset (Potthast et al., 2013). Every document in that dataset is constructed from texts of various search results (i.e. authors) for one of the 150 topics in total. By varying different parameters such as the number and proportion of the authors, places in a document where an author switch occurs (between words, sentences or paragraphs), three training and test datasets were generated (Rosso et al., 2016). Test datasets are currently not publicly available and we could not use them for evaluation of our approach. Some basic characterisics of training datasets are shown in Table 1.

2. Related work

The basic assumption in authorship analysis is that texts of different authors are mutualy separable because each author has more or less unique writing style (Stamatatos, 2009b; Ding et al., 2016). More precisely, Koppel et al. (2009) explain that methods used in authorship analysis must be able to distinguish writing styles, but also tolerate shallow differences inside the same style because an author's stylistic habbits can conciously or unconciously vary over time. Therefore, most of related work tries to find the better features and methods which writing style will be quantified and measured with.

Zu Eissen and Stein (2006) manually created a labeled

[1] https://tinyurl.com/y9m4zntm

corpus of plagiarized documents and used it for intrinsic plagiarism detection task. They used average sentence length, part of speech tags, average stopword number and the averaged word frequency class as input features for their linear discriminant analysis and suport vector machine (SVM) models. They approached that task in a supervised fashion.

Stamatatos (2009a) created a feature vector of normalized occurence of character tri-grams in the whole document. That vector represented a document's profile. Using a sliding window of fixed length he created same profiles for every window and compared them with the profile of the whole document. Result of comparison was an output from a style change function whose peaks were indicators of place in a document where style change occurs. All values above the predefined passage criterion were considered a result of plagiarism. That approach was unsupervised.

Rahman (2015) classified sections of documents from PAN 2011 dataset with the help of SVM. Those sections were again obtanined by sliding a window of fixed length over the document. He also proposed new kind of information theoretical features - entropy, relative entropy, correlation coefficient and n-gram frequency class calculated from character tri-gram frequency profiles of each window and the whole document. He also used function word bigram and tri-gram frequency profiles with 1, 2, 3 and 4 skips. The value of style change function introduced by Stamatatos (2009a) was also incorporated in feature vectors.

Stein et al. (2011) defined IPD as the one class classification problem where the text of main author belongs to a one target class and the rest are outliers. To find them, they estimated probability distributions of various stylistic features for the target class and outliers. Then a naive Bayes' algorithm was applied to feature vectors whose values lie outside the predefined uncertainity intervals. An additional outlier post-processing methods were also tested. The most successful was the unmasking technique described by Koppel et al. (2009). The main sense of unmasking is to iteratively remove the best features that distinguish two classes and observe the speed with which cross-validation accuracy of again trained classifier drops. If the drop is slow and smooth, the outliers are indeed outliers because after *n* iterations of removing discriminative features they are still separable from the main auhor's work.

Koppel et al. (2011) used two staged approach in clustering of pre-segmented mixed biblical text written by two authors. First they used normalized cuts algorithm with cosine similarity to obtain initial clusters of segments which were represented only by normalized counts of synonymes from Hebrew synsets. Samples from initial clusers were separated in core and non-core samples via an iterative procedure, and core ones were labeled. SVM classifier was used to classify non-core samples, but now a bag-of-words feature vectors were used. The whole approach resulted with very good clusters. They also tried this method on an unsegmented case. Text was first splitted in a way that minimizes doubly-represented synonyms in segments and the same procedure was repeated. The clustering performance was lower than in pre-segmented case.

Brooke et al. (2013) concluded that a very good initial segmentation of text, at least in poems written by T. S. Elliot, is needed for a good performance of their modified k-means algorithm in clustering of voices. Except often character, lexical and syntactic features, they used features such as average frequency in a large external corpus (Brants and Franz, 2006). One of the most promising feature they considered is the centroid of 20 dimensional distributional vectors obtained by applying latent semantic analysis on a large web corpus (Landauer and Dumais, 1997).

The works by Kuznetsov et al. (2016) and Sittar et al. (2016) were submitted on the PAN 2016 competition for three aforementioned tasks. For the task *a*, Kuznetsov et al. (2016) trained a Gradient Boosting Regression Trees (GBRT) model on PAN 2011 dataset as a style change function used for threshold based outlier detection. Every sentence was vectorized using word frequencies, n-gram frequencies, punctuaiton symbols and the universal POS tags count, sentence length and mean length of sentence words. The final input to the model was concatenation of center sentence vector and ones from context of size ± 2. In task *b* they used a Hidden Markov Model with Gaussian Emissions for document segmentation over the same sentence scores from task *a*. To estimate the unknown number of authors *n* in task *c*, they chose *n* from 2 to 20 which maximizes their cluster discrepancy measure $Q(n)$.

Sittar et al. (2016) used k-means algorthm to cluster *ClustDist* scores of each sentence, where the number of groups was equal to the known number of authors in tasks *a* and *b*. In task *c*, a number of groups was generated randomly. Although they defined a *ClustDist* score for a single sentence as an average distance between current and every other sentence vector, which is again a similar concept like a style change function, in provided example they used only a sum of unknown distance measures. Fifteen features in total were used for sentence vectorization, including average word and sentence lengths, count and ratios of characters, digits upperpase letters, spaces and tabs.

The most of described approaches combine supervised and usupervised methods and operate on the level of longer text segments or sentences. Since the style change in our tasks can occur even between two tokens in a same sentence, we wanted our model should be able to work on the token level. We were also inspired by Brooke et al. (2013) who said that a more radical approach would not separate described tasks in segmentation and clustering steps, but rather build an authorial segments that would also form good clusters. Instead of clustering tokens directly, we decided to cluster their vectorized stylistic contexts because they obviously contain more valuable sylistic information than tokens alone.

3. Author diarization and intrinsic plagiarism detection

Let Δ be the domain of documents. We define a document $D \in \Delta$ as a finite sequence of tokens $(t_i)_{i=1}^{n}$, where n can differ among documents. Given a document, each of its tokens is unique and defined by its character sequence and position in the document. Therefore, a document can be equivalently represented by its set of tokens $T_D = \{t_i\}_{i=1}^{n}$.

For each document, there is a corresponding mapping to a sequence of labels $(a_i)_{i=1}^n$ that are representing groupings of tokens by authors. The labels a_i are indices of authors of the document. Each token $t_i \in T_D$ is assigned a document-level label $a_i \in \{1..c\}$ associating it to one of c authors. The exact value of the label is not important. It is only required that all tokens corresponding to the same author have the same label. Therefore, there are $m!$ equivalent such mappings given a document. In the case of intrinsic plagiarism detection, there are only 2 labels: 0 representing the main author, and 1 representing plagiarized text.

Equivalently, the codomain of the mapping can also be defined as a set of segmentations Σ. A segmentation $S \in \Sigma$ is a minimal set of segments, where each segment s represents a set of consecutive tokens $\{t_i\}_{i=i_1}^{i_2}$ where each is assigned the same label. For a segmentation to be valid, the segments must cover all terms in the document and not overlap:

$$\bigcup_{s \in S} s = T_D \wedge \bigcap_{s \in S} s = \{\}. \tag{1}$$

The correct mapping of a documents to the corresponding segmentations will be denoted with $\sigma : \Delta \to \Sigma$.

Let $\mathcal{D} \subset \Delta \times \Sigma$ be a dataset consisting of a finite set of pairs of documents and corresponding segmentations., i.e. $\mathcal{D} = \{(D_i, \sigma(D_i))\}_{i=1}^N$. The goal is to find the model $\hat{\sigma}$ that best approximates the correct mapping σ, i.e. makes good predictions given unseen documents.

3.1. Evaluation measures

For evaluation of intrinsic plagiarism detection, Potthast et al. (2010) define multiple measures for different aspects of a system's perfromance. The main measures are binary macro-averaged nad micro-averaged precision (P), recall (R) and F_1-score. For evaluating author diarization, we use *BCubed* precision, recall and F_1 measures described by Amigó et al. (2009), which are specialized for evaluation of clustering results. The same measures were used for evaluation on the PAN 2016 competition (Rosso et al., 2016).

Let l be a function that associates lengths in characters to segments. Specially, $l(\{\}) = 0$. For notational convenience, we also use l to denote the sum of lengths of all segments in a set of segments: $l(S) = \sum_{s \in S} l(s)$, where S is as set of segments. Given a document D, let $S_\mathrm{p} \subseteq \sigma(D)$ be a set of all true plagiarism segments of the document and $\hat{S}_\mathrm{p} \subseteq \hat{\sigma}(D)$ the segments predicted as plagiarism by the model. With $S_\mathrm{tp} = \bigcup_{(s,\hat{s}) \in S_\mathrm{p} \times \hat{S}_\mathrm{p}} l(s \cap \hat{s})$, the micro-averaged evaluation measures for intrinsic plagiarism detection are defined as follows:

$$P_\mu = \frac{l(\hat{S}_\mathrm{tp})}{l(\hat{S}_\mathrm{p})}, \tag{2}$$

$$R_\mu = \frac{l(\hat{S}_\mathrm{tp})}{l(S_\mathrm{p})}, \tag{3}$$

$$F_\mu = \frac{2}{P_\mu^{-1} + R_\mu^{-1}}. \tag{4}$$

The macro-average evaluation measures treat all plagiarism segments as equally important and are not affected by their lengths:

$$P_\mathrm{M} = \frac{1}{|\hat{S}_\mathrm{p}|} \sum_{\hat{s} \in \hat{S}_\mathrm{p}} \frac{\sum_{s \in S_\mathrm{p}} l(s \cap \hat{s})}{l(\hat{s})}, \tag{5}$$

$$R_\mathrm{M} = \frac{1}{|S_\mathrm{p}|} \sum_{\hat{s} \in S_\mathrm{p}} \frac{\sum_{s \in \hat{S}_\mathrm{p}} l(s \cap \hat{s})}{l(s)}, \tag{6}$$

$$F_\mathrm{M} = \frac{2}{P_\mathrm{M}^{-1} + R_\mathrm{M}^{-1}}. \tag{7}$$

In author diarization document segments have to be clustered into c clusters, where c is the number of authors that may or may not be known to the system. We divide the segments from the true segmentation S and the predicted segmentation $\hat{S}$ each into sets of segments $S_i, i = 1..c$ and $\hat{S}_j, j = 1..\hat{c}$, where c is the true number of authors, and $\hat{c}$ the predicted number of authors. We use the following *BCubed* measures for evaluation:

$$P_{\mathrm{B}^3} = \sum_{i=1}^{c} \frac{1}{l(S_i)} \sum_{j=1}^{\hat{c}} \sum_{(s,\hat{s}) \in S_i \times \hat{S}_j} l(s \cap \hat{s})^2 \tag{8}$$

$$R_{\mathrm{B}^3} = \sum_{j=1}^{\hat{c}} \frac{1}{l(\hat{S}_j)} \sum_{i=1}^{c} \sum_{(s,\hat{s}) \in S_i \times \hat{S}_j} l(s \cap \hat{s})^2 \tag{9}$$

$$F_{\mathrm{B}^3} = \frac{2}{P_{\mathrm{B}^3}^{-1} + R_{\mathrm{B}^3}^{-1}}. \tag{10}$$

4. The proposed approach

Our approach can generally be described as a pipeline $P = (f_\mathrm{b} : \Delta \to \Phi_{n_b}, f_\mathrm{t} : \Phi_{n_b} \to \Phi_{n_t}, f_\mathrm{c} : \Phi_{n_t} \to C)$. Here Δ is the tokenized document domain as defined in section (((((((○))))))). Φ_{n_b} and Φ_{n_t} are sets of variable-length sequences of feature vectors with dimension n_b and n_t respectively. n_b and n_t can either be fixed or depend on the document being processed. C is the set of all sequences of length n (the number of tokens in the document) with elements being indices of authors/clusters.

The basic feature extractor denoted with f_b is used to extract stylistic features from the contexts of all tokens. If D is a document with n tokens, the basic feature extractor outputs a sequence of n feature vectors, each vector representing the context of one token. n_b denotes the dimension of those vectors. The next step in the pipeline is the feature transformation f_t that maps the basic features to a space that they can be better clustered in. The dimension of the new feature space n_t generally doesn't equal n_b. The final step in the pipeline is clustering denoted with f_c. The clustering algorithm implicitly clusters tokens because it actually clusters their stylistic contexts, each cluster representing an author. Depending on the task, the clustering algorithm can either be given a known number of authors, or try to predict it.

The following steps are done in predicting the segmentation: (1) raw text is tokenized giving a sequence of tokens $D \in \Delta$, (2) features are extracted for all tokens and their contexts, giving a sequence of feature vectors $\phi_\mathrm{t} = (\mathbf{t}_i)_{i=1}^n = (f_\mathrm{t} \circ f_\mathrm{b})(D)$, (3) the tokens are clustered based on ϕ_t, giving a sequence of author labels $(a_i)_{i=1}^n$,

where $n = |T_D|$, and (4) a segmentation $\hat{S}$ is generated based on the document D and the sequence of predicted author labels.

Tokenization. As a preprocessing step, we tokenized each document D with `nltk`[2] toolkit, to obtain a set of tokens T_D. We also performed Part of Speech tagging with the same tool, to speed up a basic feature extraction which we describe below. We didn't use other preprocessing techniques such as lemmatization, stemming and stop word removal because they would take away a lot of stylometric data from text (Stamatatos, 2009b). The final output from tokenization step was a finite sequence $\{(t_i, o_i, l(t_i), POS_i)\}_{i=1}^{n}$ where o_i is the offset of token t_i, $l(t_i)$ its length in characters and POS_i its POS tag.

Basic feature extraction. We defined the stylistic context of a token t_i as a set of tokens $\{t_k\}_{k=i-c}^{i-1} \cup \{t_k\}_{k=i+1}^{i+c}$ where c is a context size. From each context we extracted the most used stylometric features which we found in previous work.

- *Character tri-grams*. Frequencies of *n*-grams on character level have been very useful in quantifying the writing style (Stamatatos, 2009a). They are able to capture lexical and contextual information, use of punctuation and errors which can be an author's "fingerprint". This feature is also tolerant to noise. Based on work by Stamatatos (2009b) and Rahman (2015), we choose $n = 3$. Maximal dimension of this feature vector was set to 200.

- *Stop words*. According to Stamatatos (2009b), sometimes also called *function words*, these are the most common used topic-independent words in text, such as articles, prepositions, pronouns and others. They are used unconciously and found to be one of the most discriminative features in authorship attribution since they represent pure stylistic choices of authors' burrows-1987,argamon-2005. We used frequencies of 156 english stop words available in `nltk`.

- *Special characters*. We used counts of all character sequences which satisfied a regular expression defined as [REGEX]. Although character n-grams can catch the use of those character sequences, we wanted to have a distinct feature for that purpose. Koppel et al. (2009) mentioned that authors can have different punctuation habbits.

- *POS tag counts*. This is syntactic feature which Koppel et al. (2009) and Stamatatos (2009b) also identified as a discriminative one in authorship analysis and it was used by Kuznetsov et al. (2016). We used all 12 tags from the universal tagset.

- *Average token length*. Used by Kuznetsov et al. (2016), Sittar et al. (2016), **?**) and Stein et al. (2011). Koppel et al. (2009) characterized this feature as a complexity measure.

- *Bag of Words*. Bag of words text representation more captures content, rather than style (Stamatatos, 2009b). We included this feature because it boosted performance of our initial testing, even without words been previously stemmed or lemmatized (stop words were excluded). Vocabulary had a maximum of 100 words.

- *Type-token ratio*. We wanted to use that feature to measure the vocabulary richness in a token's context, but after we finished with performance evaluations we realized that there was a bug in implementation and that feature did not contribute to overal results. The feature should be calculated as the ratio of vocabulary size and total number of tokens of the text (Stamatatos, 2009b).

Differences between the 2 model variants....
basic features

Feature transformation. Differences between the 2 model variants. Training.
id
scaling
concatenation with squared features
transformation trained via SGD

Let $(\mathbf{b}_i)_{i=1}^{n} = f_{\mathrm{b}}(D)$ be the sequence of basic feature vectors with elements from $\mathbb{R}^{n_b}$. Let $(a_i)_{i=1}^{n}$ be the sequence of true author labels with elements from $\{1..c\}$. We want to maximize the *clusterability* of the feature vectors obtained by the feature transformation T. Let $(\mathbf{t}_i)_{i=1}^{n} = (T(\mathbf{b}_i))_{i=1}^{n} = f_{\mathrm{t}}((\mathbf{b}_i)_{i=1}^{n})$ be the sequence of transformed feature vectors with elements from $\mathbb{R}^{n_t}$.

Clustering. clustering algorithms
k-means HAC DBSCAN auto-k-means

Segmentation generation. something short
features
differences: fixed features + transformation vs document-dependent features

5. Experimental results

1 baselines repo link Mention cofidences.

5.1. Intrinsic plagiarism detection

setup results don't forget to describe tolerance

5.2. Author diarization with known numbers of authors

setup

5.3. Author diarization with unknown numbers of authors

setup

6. Conclusion

Conclusion is the last enumerated section of the paper. It should not exceed half of a column and is typically split into 2–3 paragraphs. No new information should be presented in the conclusion; this section only summarizes and concludes the paper.

[2] `http://www.nltk.org`

Table 2: This is the caption of the table. Table captions should be placed *above* the table.

Model	R_μ	P_μ	F_μ	R_M	P_M	F_M
Dummy	0	1	2			
One	0	1	2			
One	0	1	2			
One	0	1	2			

7. Further work

Acknowledgements

If suitable, you can include the *Acknowledgements* section before inserting the literature references in order to thank those who helped you in any way to deliver the paper, but are not co-authors of the paper.

References

Khaled Aldebei, Xiangjian He, and Jie Yang. 2015. Unsupervised Decomposition of a Multi-Author Document Based on Naive-Bayesian Model. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 501–505.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram corpus version 1.1. *Google Inc*.

Julian Brooke, Graeme Hirst, and Adam Hammond. 2013. Clustering voices in the Waste Land. In *Proceedings of the 2nd Workshop on Computational Literature for Literature (CLFL'13), Atlanta*.

Steven HH Ding, Benjamin Fung, Farkhund Iqbal, and William K Cheung. 2016. Learning stylometric representations for authorship analysis. *arXiv preprint arXiv:1606.01219*.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26.

Moshe Koppel, Navot Akiva, Idan Dershowitz, and Nachum Dershowitz. 2011. Unsupervised decomposition of a document into authorial components. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1356–1364. Association for Computational Linguistics.

Mikhail Kuznetsov, Anastasia Motrenko, Rita Kuznetsova, and Vadim Strijov. 2016. Methods for intrinsic plagiarism detection and author diarization. *Working Notes Papers of the CLEF*.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Robert Layton, Paul Watters, and Richard Dazeley. 2013. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering*, 19(01):95–120.

Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. 2010. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 997–1005. Association for Computational Linguistics.

Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. 2013. Crowdsourcing interaction logs to understand text reuse from the web. In *ACL (1)*, pages 1212–1221.

Rashedur Rahman. 2015. Information Theoretical and Statistical Features for Intrinsic Plagiarism Detection. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 144.

Paolo Rosso, Francisco Rangel, Martin Potthast, Efstathios Stamatatos, Michael Tschuggnall, and Benno Stein. 2016. Overview of pan'16. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 332–350. Springer.

Abdul Sittar, Hafiz Rizwan Iqbal, and A. Nawab. 2016. Author Diarization Using Cluster-Distance Approach. *Working Notes Papers of the CLEF*.

Efstathios Stamatatos. 2009a. Intrinsic plagiarism detection using character n-gram profiles. *threshold*, 2(1,500).

Efstathios Stamatatos. 2009b. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Benno Stein, Nedim Lipka, and Peter Prettenhofer. 2011. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45(1):63–82.

Sven Meyer Zu Eissen and Benno Stein. 2006. Intrinsic plagiarism detection. In *European Conference on Information Retrieval*, pages 565–569. Springer.