Methods for intrinsic plagiarism detection and author diarization

Notebook for PAN at CLEF 2016

Mikhail Kuznetsov, Anastasia Motrenko, Rita Kuznetsova, and Vadim Strijov

Antiplagiat Research,
Moscow Institute of Physics and Technology,
mikhail.kuznecov@phystech.edu, anastasiya.motrenko@phystech.edu, kuznetsova@ap-team.ru,
strijov@phystech.edu

Abstract The paper investigates methods for intrinsic plagiarism detection and author diarization. We developed a plagiarism detection method based on constructing an author style function from features of text sentences and detecting outliers. We adapted the method for the diarization problem by segmenting author style statistics on text parts, which correspond to different authors. Both methods were tested on the PAN-2011 collection for the intrinsic plagiarism detection and implemented for the PAN-2016 competition on author diarization.

1 Introduction

Traditional *intrinsic* plagiarism detection problem [13,9,10] is formulated as follows. Given a suspicious document, the task is to determine whether the document is written by a single author or contains plagiarized sections. Unlike the *extrinsic* setting, no external collection is given: plagiarism detection should be performed without comparing a suspicious document to the potential sources. The traditional intrinsic plagiarism setting contains an essential condition: there exists one main author who wrote at least 70% of the considered text document.

The «one-main-author» condition designated the following common schema for the intrinsic plagiarism detection [11,14,15,4,8]: 1) split a text document into a set of text segments (e.g. sentences), 2) develop a set of segment features and combine them to an author style function that measures an author-style correspondence for each text segment, and 3) find critical values in the author style function to detect plagiarized segments. The authors in [11] proposed to divide a text document into a set of intersecting segments (a «sliding window» approach) and used character 3-gram frequencies as the main component of an author style function. The other considered style function examples are the n-gram classes (i.e. the inverted frequencies) [4], punctuation, pronouns and part-of-speech tags count [14], normalized word frequency class [15]. Oberreuter et al. [8] proposed to construct a style function which counts a relative deviation of an n-gram frequency from its typical value.

The PAN-2016 competition [12] provided a more general setting for intrinsic plagiarism detection named *author diarization*. Unlike the traditional intrinsic plagiarism formulation, the text document is written by n authors, no single main author is given,

and each author can contribute in arbitrary extent. The task is to distinguish exactly n authors in the given text document, where the number n can be known or unknown.

To deal with the full stack of PAN diarization problems (traditional intrinsic plagiarism detection, diarization with a given number of authors, diarization with an unknown number of authors) we propose a single algorithmic framework with slight modifications for each particular problem. First, we divide a text document into sentences and construct basic stylometric features for each sentence (character and word n-gram frequencies, punctuation and pronouns count). Second, we train a classifier over the constructed feature space using the PAN-2011 evaluation corpus. Third, having the classifier output (that can be also referred to as author style function over text sentences) we make 1) outlier detection for the intrinsic plagiarism problem, 2) classifier statistics segmentation for the diarization problem. If the number of authors is unknown, we compute its estimation by an exhaustive search maximizing a heuristical cluster measure.

2 Intrinsic plagiarism detection

We provide an algorithm description for the traditional intrinsic plagiarism framework and the more general author diarization problem. Three main stages of this method are: composing basic features for a text segment, constructing an author style function, and post-processing with outliers detection. An author style function is constructed as an output of a classifier trained on basic features.

Problem setting. Denote by D a collection of text documents. Each document $d \in D$ has one main author who wrote its main part (at least 70% of a text); the other parts of a document may be written by other authors. The problem is to detect these intrusive fragments in a text document.

We formulate the intrinsic plagiarism problem as text segments classification. A text segment s is a sequence of symbols in a document d such that d splits into a set of segments S, $d = \bigcup\limits_{s \in S} S$. The problem is to find the labels $a(s_i)$ such that $a(s_i) = 0$ if the segment s_i is written by the main author, and $a(s_i) = 1$ if the segment s_i contains plagiarism.

Our method exploits *per-sentence* approach [14] to the segment construction. Unlike the more commonly used sliding-window approach [11,4,8], the sentence method constructs disjoint segments of different length and detects plagiarism on sentence level. To split a document into sentences we use the standard nltk parser (`sent_tokenize` from Natural Language Processing Toolkit, [5]).

To train a classification model we use the labeled collection from the PAN-2011 contest [13]. We use the similar notation for the ground truth information: y_i is an indicator variable showing whether the sentence s_i is written by the main author of document d. The sentence s_i is classified correctly if the label $a(s_i)$ equals to the ground truth label, $a(s_i) = y_i$, for document d.

The initial label information y_i is given in the form of character labeling. Sentences are labeled by the rule: if more than a half of characters in s_i are plagiarized, then assign $y_i = 1$, otherwise $y_i = 0$.

2.1 Features construction

To vectorize text sentences and construct feature description, mapping

$$s_i^d \mapsto \mathbf{x}_i^d \in \mathbb{R}^n,$$

the common methods from [11,14,15] were implemented with slight modifications. The list of methods is provided below.

Word frequencies. A word frequency feature is based on analyzing occurrences of text words w, the lowercased sequences of letter characters excepting the stopwords. Let $n_d(w)$ be a number of occurrences of word w in document d, $n_s(w)$ is a number of occurrences of word w in sentence s, and w_d^* is the most frequent word in document d. By $\nu_d^s(w)$ denote a *relational frequency* of word w in document d *without* sentence s:

$$\nu_d^s(w) = \log_2 \frac{n_d(w_d^*)}{n_d(w) - n_s(w) + 1}. \tag{1}$$

A relational frequency (1) characterizes specificity of word w in sentence s. Similarly, a set of word frequencies $\nu_d(s) = \{\nu_d^s(w) : w \in s\}$ characterizes specificity of sentence s: the more specific words has the sentence, the more it deviates from the main author style. The mean, 5% and the 95% percentiles of a set $\nu_d(s)$ compose feature description for sentence s. That is, for each sentence the algorithm constructs three word-based features that can be interpreted as 1) mean frequency of words in a sentence, 2) frequency of the most rare word in a sentence, 3) frequency of the most frequent word in a sentence.

n-gram frequencies. Together with word frequencies the algorithm computes the n-gram character frequencies using the same technique as above. The only difference is text parsing: the document and each sentence are splitted into character n-grams instead of words. Finally, the algorithm computes three statistics (mean, 5% and 95% percentiles) for each sentence and for each n. The experiments show that the best practice is to use 1-grams, 3-grams and 4-grams jointly. That is, the resulting n-gram feature returns nine statistics, three for each of the considered n-grams.

Count and length. For each sentence the algorithm computes the number of occurrences of the most common punctuation symbols (`!,.?-;`) and the universal part-of-speech tags (`VERB`, `NOUN`, `PRON`, `ADJ`, `ADV`, `ADP`, `CONJ`, `DET`, `NUM`, `PRT`) using the nltk parser [5]. Since the sentence lengths differ, the counts are additionally normalized by the sentence words number.

Finally for each sentence the algorithm computes its length in characters, and the mean length of the sentence words.

2.2 Classification and author style function

In this section we describe classification and outlier detection stages of plagiarism detection. The method constructs classifier function over vectorized sentences $\mathbf{x}_1, ..., \mathbf{x}_m$

and trains it according to the following schema. Each sentence s_i has a ground truth plagiarism label y_i. To consider the impact of nearby sentences, the classifier predicts label $a(s_i)$ using two sentences from left and right of s_i. That is, to predict label $a(s_i)$ classifier function f uses an extended description $[\mathbf{x}_{i-2}; \mathbf{x}_{i-1}; \mathbf{x}_i; \mathbf{x}_{i+1}; \mathbf{x}_{i+2}]^T$,

$$a(s_i) = f([\mathbf{x}_{i-2}; \mathbf{x}_{i-1}; \mathbf{x}_i; \mathbf{x}_{i+1}; \mathbf{x}_{i+2}]), \tag{2}$$

and maximizes accuracy of prediction of the ground truth labels y_i.

To predict plagiarism labels the algorithm uses Scikit-learn implementation of the Gradient Boosting Regression Trees (GBRT) [6]. The optimal parameters (`n_estimators=200`, `max_depth=4`) were set by maximization of the Area-Under-Curve classification measure. The output of the GBRT model is also called an author style function: the model combines features and returns sentence scores $a(s_1), ..., a(s_m)$, which indicate degree of mismatch with main author style.

In a final step the algorithm detects outliers in an author style statistics. The outlier sentences are finally labeled as plagiarized. The outlier detection method is threshold-based: all sentences with a classifier label more than a certain threshold marked as outliers. A threshold is determined by a grid of quantiles of style function values. Among the 0.71,0.72,...,0.99 quantiles the algorithm chooses the one which optimizes the F1-measure for final plagiarism detection. For the PAN-2011 dataset an optimal quantile is 0.94. That means approximately 6% of the dataset sentences are plagiarized.

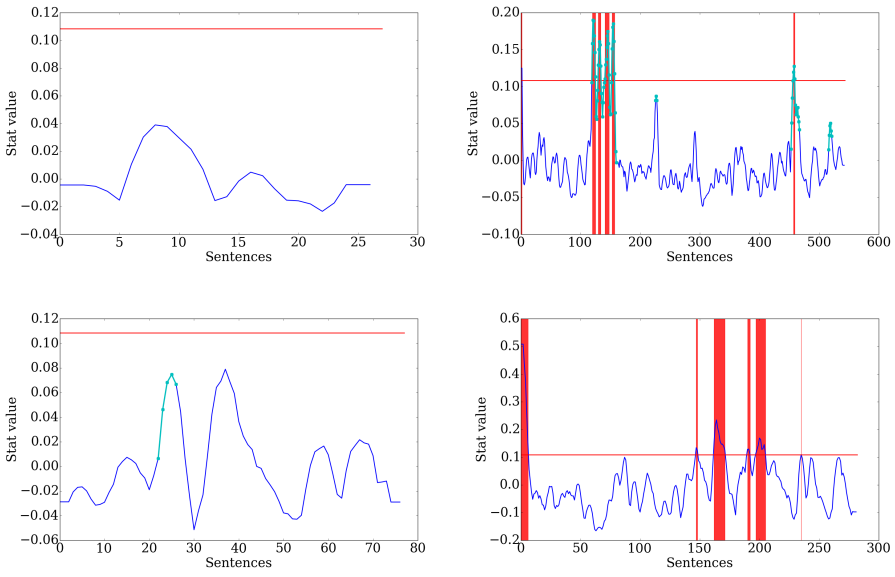

Figure 1: Plagiarism detection examples

Figure 1 shows the examples of the style function and detected outliers. Blue lines illustrate a classifier output ("Stat value" at y-axis), red lines — outlier thresholds. Red segments indicate detected plagiarism. In turn, cyanic coloured parts of the blue lines correspond to the ground truth information about plagiarized sentences. The upper figures show the cases when the detector works correctly, the bottom right and left illustrate the first- and second type errors, correspondingly.

3 Author diarization

The author diarization problem with given number of authors is to segment a document into parts corresponding to the different authors. No main author is given, each of writers can contribute in arbitrary extent.

3.1 Known number of authors

The intrinsic plagiarism method was adapted to solve the diarization problem. The algorithm splits a document into sentences and vectorizes sentences as it is described in section 2.1. The algorithm also uses a trained model (2) and computes the series statistics $a(s_1), ..., a(s_m)$ for the sentences $s_1, ..., s_m$.

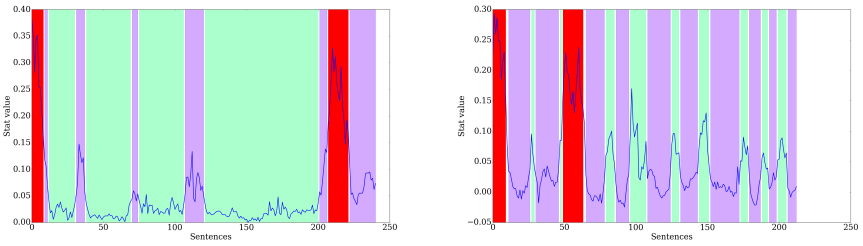

Figure 2: Segmentation examples

Instead of the outlier detection stage, the diarization method provides segmentation of series $a(s_1), ..., a(s_m)$ using the Hidden Markov Model approach with Gaussian emissions [7]. The algorithm uses hmmlearn [1] implementation of the Viterbi algorithm with fixed number of hidden components equalling to number of authors n. The examples of segmented time series are shown on figure 2 (number of authors n equals three on both figures).

3.2 Unknown number of authors

To deal with unknown number of authors we make its estimation computing an averaged t-statistic for all pairs of author segments. Iterating through probable number

of authors n from 2 to 20, the diarization method computes the time series segmentation $c_1, ..., c_n$ for each n. For each segmentation it computes $Q(n)$, the measure of clusters discrepancy:

$$Q(n) = \sum_{i,j=1}^{n} \frac{|m(c_i) - m(c_j)|}{\sqrt{\frac{\sigma(c_i)^2}{l(c_i)} + \frac{\sigma(c_j)^2}{l(c_j)}}},$$

where $m(c_i)$ is the mean of elements in cluster c_i, $\sigma(c_i)$ is the mean deviation, and $l(c_i)$ is the cluster size.

The final estimation $\hat{n}$ maximizes clusters discrepancy $Q(n)$. Having obtained the estimation, the algorithm performs a diarization method with known number of authors $\hat{n}$.

4 Experiment

We conducted several computational experiments on the PAN-2011 collection for intrinsic plagiarism detection [2]. The test collection consists of 4753 documents and is splitted into 10 folds. Each folds contains 500 documents except for the smaller fold 10.

Quality criteria. The criteria from [10] and [3] were used to measure quality of the methods. By $\mathbf{y}$ denote a ground truth character plagiarism segment, a sequence of labeled characters in a document. By $\mathbf{a}$ denote a detected character plagiarism segment. By Y and A denote the sets of given and detected plagiarism segments, respectively.

Micro- and macro precision and recall criteria are defined as follows:

$$\text{Prec}_{\text{micro}}(Y, A) = \frac{\left|\cup_{(\mathbf{y},\mathbf{a})\in(Y\times A)}(\mathbf{y}\cap\mathbf{a})\right|}{|\cup_{\mathbf{a}\in A}\mathbf{a}|}, \quad \text{Rec}_{\text{micro}}(Y, A) = \frac{\left|\cup_{(\mathbf{y},\mathbf{a})\in(Y\times A)}(\mathbf{y}\cap\mathbf{a})\right|}{|\cup_{\mathbf{y}\in Y}\mathbf{y}|},$$

$$\text{Prec}_{\text{macro}}(Y, A) = \frac{1}{|A|}\sum_{\mathbf{a}\in A}\frac{\cup_{\mathbf{y}\in Y}(\mathbf{y}\cap\mathbf{a})}{|\mathbf{a}|}, \quad \text{Rec}_{\text{macro}}(Y, A) = \frac{1}{|Y|}\sum_{\mathbf{y}\in Y}\frac{\cup_{\mathbf{a}\in A}(\mathbf{a}\cap\mathbf{y})}{|Y|}.$$

F1 measure is a combination of precision and recall for both micro and macro cases:

$$\text{F1} = 2\frac{\text{Prec}\cdot\text{Rec}}{\text{Prec}+\text{Rec}}$$

An overall score pladget is an F1-measure normalized by granularity:

$$\text{pladget}(Y, A) = \frac{\text{F1}}{\log_2(1+\text{gran}(Y, A))}.$$

Results. The first series of experiment uses a cross-validation schema to estimate and compare different models on test folds. Take first five folds of the collection and construct five models. Each time use four of five folds for training, use the remaining fold as the test sample. Quality results for different models are shown on table 1. The best result (F1-measure 0.32, pladget 0.24) was achieved by the model tested on the fold 4.

The remaining folds of the collection were used to validate the model quality. Table 2 shows the results for a single best model, separately by folds 6-10 and on average. The finally achieved quality is 0.29 for macro F1-measure and 0.21 for macro-pladget.

Table 1: Results for test folds, selecting the best model

			Macro				Micro			
Test	F1-raw	Gran	Rec	Prec	**F1**	**Pladget**	Rec	Prec	F1	Pladget
fold 1	0.43	1.58	0.36	0.23	0.28	0.207	0.48	0.43	0.45	0.329
fold 2	0.41	1.57	0.35	0.23	0.28	0.205	0.45	0.40	0.42	0.311
fold 3	0.36	1.70	0.30	0.20	0.24	0.168	0.41	0.39	0.40	0.278
fold 4	0.45	1.53	0.38	0.28	0.32	**0.242**	0.45	0.46	0.46	0.341
fold 5	0.43	1.62	0.34	0.30	0.32	0.228	0.44	0.51	0.47	0.338

Table 2: Results for validation

			Macro				Micro			
Valid	F1-raw	Gran	Rec	Prec	**F1**	**Pladget**	Rec	Prec	F1	Pladget
fold 6	0.43	1.62	0.39	0.22	0.28	0.203	0.50	0.40	0.45	0.320
fold 7	0.45	1.73	0.39	0.25	0.31	0.213	0.48	0.46	0.47	0.323
fold 8	0.41	1.56	0.37	0.22	0.28	0.203	0.48	0.41	0.44	0.326
fold 9	0.43	1.69	0.37	0.26	0.31	0.216	0.44	0.43	0.43	0.303
fold 10	0.36	1.48	0.33	0.19	0.24	0.186	0.43	0.34	0.38	0.290
mean	**0.42**	**1.62**	**0.37**	**0.23**	**0.29**	**0.206**	**0.47**	**0.41**	**0.44**	**0.315**

5 Conclusion

The proposed intrinsic plagiarism detection method splits a text document into sentences, vectorizes the sentences, trains a classification model and finds outliers in the classifier output. To adapt the framework for the author diarization problem, it additionally segments an output statistics into a set of clusters corresponding to the different authors. If the number of authors is unknown, the method estimates it by maximization of cluster discrepancy measure.

The method was implemented to the PAN-2016 competition in author diarization [12]. The model achieved f1-measure 0.2 for the intrinsic plagiarism problem, bcubed-f measure 0.54 for author diarization with known number of authors, and bcubed-f measure 0.5 for unknown number of authors.

References

1. Implementation of the hidden markov models in python. http://hmmlearn.readthedocs.io/en/latest/, accessed: 2016-05-10
2. The pan plagiarism corpus 2011. http://www.uni-weimar.de/en/media/chairs/webis/corpora/corpus-pan-pc-11/, accessed: 2016-05-24
3. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information retrieval 12(4), 461–486 (2009)
4. Bensalem, I., Rosso, P., Chikhi, S.: Intrinsic plagiarism detection using n-gram classes. In: EMNLP. pp. 1459–1464 (2014)

5. Bird, S.: Nltk: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive presentation sessions. pp. 69–72. Association for Computational Linguistics (2006)
6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
7. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. Data mining in time series databases 57, 1–22 (2004)
8. Oberreuter, G., L'Huillier, G., Rıos, S.A., Velásquez, J.D.: Approaches for intrinsic and external plagiarism detection. Proceedings of the PAN (2011)
9. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeno, A., Gupta, P., Rosso, P., et al.: Overview of the 4th international competition on plagiarism detection. In: CLEF (Online Working Notes/Labs/Workshop). Citeseer (2012)
10. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Proceedings of the 23rd international conference on computational linguistics: Posters. pp. 997–1005. Association for Computational Linguistics (2010)
11. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles (2009)
12. Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by Authorship Within and Across Documents. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
13. Stein, B., Barrón Cedeño, L.A., Eiselt, A., Potthast, M., Rosso, P.: Overview of the 3rd international competition on plagiarism detection. In: CEUR Workshop Proceedings. CEUR Workshop Proceedings (2011)
14. Zechner, M., Muhr, M., Kern, R., Granitzer, M.: External and intrinsic plagiarism detection using vector space models. In: Proc. SEPLN. vol. 32, pp. 47–55 (2009)
15. Zu Eissen, S.M., Stein, B.: Intrinsic plagiarism detection. In: Advances in Information Retrieval, pp. 565–569. Springer (2006)