Detecting a Change of Style Using Text Statistics[⋆]
Notebook for PAN at CLEF 2018

Kamil Safin and Aleksandr Ogaltsov

Antiplagiat Company
Moscow Institute of Physics and Technology
Higher School of Economics
kamil.safin@phystech.edu, avogaltsov@edu.hse.ru

Abstract In this paper we address style change detection problem at PAN'18 author identification task. For this task one should determine whether text is written by the same author or not. We consider supervised problem statement with the whole text as a training object. The roposed approach is based on three types of features: text statistics, hashing and high dimensional text vectors. The final algorithm is the ensemble of classifiers that were independently trained on each feature group.

1 Introduction

Authorship detection is a class of open problems in natural language processing. This class contains a bunch of the tasks that were featured in previous PAN competitions, namely:

1. Author clustering [6,15] – provided with a collection of text documents one should label each document, where label corresponds to one of *n* predefined authors.
2. Author diarization [17,5] – provided with a document written by *n* authors one should link text fragment with its author.
3. Intrinsic plagiarism detection [11,18,8,13] – provided with a document one should determine reused passages without a reference collection [19].
4. Style breach detection [1] – segmentation problem where text should be divided into style consistent passages.

PAN'18 consists of the following tasks: author identification task [3], author profiling task [12], author obfuscation task [10]. This year's author identification task is relaxation of style breach detection, i.e. binary classification task, where positive label corresponds to the case when document has at least one style change. Therefore, we can rely on developed solutions for these task [4,14]. General framework that was applied for previous tasks frequently is following:

1. To obtain text parts using some segmentation scheme. For example, sentence segmentation, *n*-grams with or without overlap.
2. To construct a mapping from text segment into feature space. [2,16,14]

[⋆] This research is supported by RFBR project 18-07-01441

3. Provided with segments features to train an algorithm to classify, cluster, or detect outliers.

However, in this paper we develop a framework that considers the whole text as a training object without any segmentation. On the one hand, such problem statement was inspired by the fact that we deal with binary classification, on the other hand we try to contribute slightly different point of view on the problem.

First, we perform preprocessing procedure that is different for each specific classifier. Next, we extract three types of features: text statistics, hash code of a text, and high-dimensional sparse representation of a text, obtained by simple counting of word *n*-grams appearance in range 1-6. Such *n*-grams counting showed success in different tasks from intrinsic plagiarism detection [16] to author profiling [7]. We train three independent classifiers on each type of features, make linear combination of probabilities given by each classifier and, learn threshold for this linear combination.
All experiments were carried out on TIRA [9].

2 Problem Statement

In this section we state the problem formally. Consider text documents collection D of size m and denote i-th document of collection by D_i, where $i \in 1, \dots m$. Let f be the mapping, such that each document of the collection is mapped to fixed-size vector:

$$f : D \to \mathbb{R}^d.$$

Consider labeling function h, such that:

$$h : \mathbb{R}^d \to y \in \{0, 1\},$$

where class label 1 is for documents written by more than one author and 0 for single-author documents. Let L_D be a empirical risk defined by:

$$L_D(h) = \frac{|\{i : h(D_i) \neq y_i\}|}{m},$$

where y_i is class label for i-th document.
We want to find $\hat{h}$ that minimizes L_D on a given collection D:

$$\hat{h} = \arg\min_{h \in \mathbb{H}} L_D(h),$$

where $\mathbb{H}$ is parametric family of functions.

3 Experiment

3.1 Data

The data corpus consists of user posts from various sites of the StackExchange network. Data is split into training and validation sets that contain 2980 and 1492 texts respectively.

3.2 Quality Criteria

To evaluate the quality of proposed algorithm, the accuracy score was used. Accuracy is the fraction of correct predictions. More formally, for binary classification accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

3.3 Model

Our model consists of three independent classifiers: Statistical, Hashing, and Counting Classifier. Each classifier returns the probability of the fact that text contains style changing. And the final probability is the weighted sum of three probabilities — p_s, p_h, p_c respectively.

Statistical Classifier. Statistical classifier uses 19 statistical features for a text analysis. The most important of them are:

- number of sentences;
- unique words fraction;
- text length;
- punctuation symbols fraction;
- letter symbols fraction.

To produce final probability Random Forest Classifier was used.

Hashing Classifier. This model uses hashing function to build term frequency counts in a text. The hash function employed is the signed 32-bit version of Murmurhash3[1]. As a result, a text is maped into 3000-dimensional vector space. These vectors contains information about occurrences of char n-grams in text. Text representation vector is used to classify whether a text contains style changes or not. Random Forest Classifier was used to produce probability.

Counting Classifier. Counting Classifier uses high-dimensional (3 million) representation of a text. Different dimensions were tried but they showed lower quality. It counts word n-grams form 1 to 6 and turns it to a vector. Logistic Regression is then used to get the probability.

Statistical, Hashing, and Counting Classifiers were trained on the train set in order to maximize performance measure — accuracy — independently from each other. Resulting performances are shown in the table below.

[1] `http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html`

	Accuracy
Statistical Classifier	0.67
Hashing Classifier	0.65
Counting Classifier	0.74

Model. The final score for text d is the weighted sum of probabilites:

$$\text{score}(d) = \alpha_s p_s + \alpha_h p_h + \alpha_c p_c,$$

where coefficients $\alpha_s, \alpha_h, \alpha_c$ are selected from $(0, 1)$.
If the score for a text exceeds the threshold δ, then this text is marked as text with change of style:

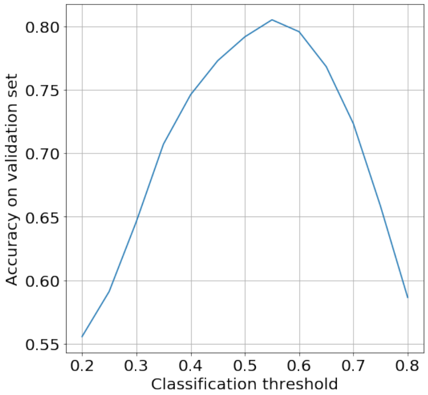
$$\text{score}(d) > \delta \Rightarrow d \text{ has change of style.}$$

3.4 Parameters Tuning

Coefficients $\alpha_s, \alpha_h, \alpha_c$ and threshold δ were tuned on the validation set by grid search in order to maximize accuracy. Each of the coefficient $\alpha_s, \alpha_h, \alpha_c$ shows the importance of corresponding classifier. Optimal parameters for the final model are:

$$\alpha_s = 0.4,\ \alpha_h = 0.2,\ \alpha_c = 0.4.$$

We can see, that Statistical and Counting Classifiers are the most informative.
And the value of threshold is: $\delta = 0.55$.
The relation between accuracy and value of threshold is shown on the figure below.



3.5 Results

The proposed model was tested on PAN'18 data set. The results of its performance are shown below.

	Validation	Test
Accuracy	0.805	0.803

4 Conclusion

We proposed an algorithm for style change detection task. This algorithm uses three independent classifiers: Statistical, Hashing, and Counting. Each classifier gives its own probability that a text may contain a change of style. Final score is computed as weighted sum of three probabilities. And if the score exceeds the threshold, a text will be marked as it containing a change of style.
The method was implemented for the PAN'18 style change detection task. The model has achieved accuracy score 0.803 on the test dataset.

References

1. Overview of the Author Identification Task at PAN 2017: Style Breach Detection and Author Clustering (2017)
2. Bensalem, I., Rosso, P., Chikhi, S.: Intrinsic plagiarism detection using n-gram classes. EMNLP (2014)
3. Kestemont, M., Tschugnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the Author Identification Task at PAN-2018: Cross-domain Authorship Attribution and Style Change Detection. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)
4. Khan, J.A.: Style breach detection: An unsupervised detection model. In: CLEF (2017)
5. Kuznetsov, M., Motrenko, A., Kuznetsova, R., Strijov, V.: Methods for intrinsic plagiarism detection and author diarization. Notebook for PAN at CLEF 2016 (2016)
6. Layton, R., Watters, P., Dazeley, R.: Automated unsupervised authorship analysis using evidence accumulation clustering. Natural Language Engineering 19(1), 95–120 (2013)
7. Ogaltsov, A., Romanov, A.: Language variety and gender classification for author profiling in pan 2017. In: CLEF (2017)
8. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th international competition on plagiarism detection. CLEF (Online Working Notes/Labs/Workshop). Citeseer (2012)
9. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
10. Potthast, M., Hagen, M., Schremmer, F., Stein, B.: Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)
11. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. Proceedings of the 23rd international conference on computational linguistics (2010)
12. Rangel, F., Rosso, P., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)

13. Safin, K., Kuznetsov, M., Kuznetsova, M.: Methods for intrinsic plagiarism detection. Informatics and Applications (2017)
14. Safin, K., Kuznetsova, R.: Style breach detection with neural sentence embeddings. In: CLEF (2017)
15. Samdani, R., Chang, K.W., Roth, D.: A discriminative latent variable model for online clustering. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 1–9. PMLR, Bejing, China (22–24 Jun 2014)
16. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles (2009)
17. Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by authorship within and across documents. CEUR Workshop Proceedings (2016)
18. Stein, B., Barron, Cedeno, L., Eiselt, A., Potthast, M., Rosso, P.: Overview of the 3rd international competition on plagiarism detection. CEUR Workshop Proceedings (2011)
19. Zechner, M., Muhr, M., Kern, R., Granitzer, M.: External and intrinsic plagiarism detection using vector space models. Proc. SEPLN. vol. 32 (2009)