SU@PAN'2015: Experiments in Author Verification

Stanimir Nikolov, Dobrinka Tabakova, Stefan Savov, Yasen Kiprov[1], and Preslav Nakov[2]

[1] Sofia University "St. Kliment Ohridski", Bulgaria,
yasen.kiprov@gmail.com
[2] Qatar Computing Research Institute, HBKU

Abstract. We describe the submission of the Sofia University team for the Author Identification Task, part of the PAN 2015 Challenge. Given a small set of documents by a single person and a "questioned" document, possibly of a different genre and/or topic, the task is to determine whether the questioned document was written by the same person who wrote the known document set. This is a hard but realistic formulation of the task, also known as *author verification*. We experimented with an SVM classifier using variety of features extracted from publicly available resources. Our solution was among the fastest, and running time was an official evaluation metric; however, our results were not so strong on AUC and C1.

Keywords: author identification, forensic linguistics, text mining, machine learning.

1 Introduction

We describe the submission of the Sofia University team, registered as *kiprov15*, to the *Author Identification* task of the PAN 2015, the 13th evaluation lab on uncovering plagiarism, authorship, and social software misuse.

Given a small set of documents (no more than five, possibly as few as one) by a single person and a "questioned" document, the task is to determine whether the questioned document was written by the same person who wrote the known document set. In practical terms, the output of the participating systems was expected to be a real number between 0 and 1, which corresponds to the probability of a positive answer. This is a hard but realistic formulation, known as *author verification*, of the more general author identification task. Note that unlike previous editions of the task, this year the questioned document differed from the documents in the known set in terms of genre and/or topic.

The organizers provided training data in four languages (English, Greek, Spanish, and Dutch), and we submitted a system for all four. We experimented with an SVM classifier using variety of features extracted from publicly available resources. Our system was among the fastest-running ones: each testset was fully analysed and scored under two minutes, and time was an official evaluation metric; however, we were not so strong on AUC and C1.

2 Method

We used GATE [3] to annotate the documents; then, we extracted features from these annotations, and we used them to train a classifier using LibSVM [1]. We used SVM as it has proven its strength in a number of natural language processing tasks, e.g., spam detection [8] and information extraction [7]. The main reason for us to choose GATE was that it could be initialized programmatically within a Java project. This makes our solution faster and easily modifiable.

Our GATE processing pipeline includes the following components:

1. Reset PR
2. ANNIE Tokenizer [2]
3. ANNIE Sentence Splitter
4. Paragraph Transfer
5. Groovy script for adding features
6. Groovy script for adding n-grams

2.1 Features

Once the annotations were done, we extracted some features, which we then used in the SVM classifier. While the challenge contained documents in different languages (as well as genres and topics), we aimed to use as few language-specific markers as possible; thus, most of our features are token-based. Here is the list of the features we used:

1. Average sentence length to character count ratio
2. Average sentence length to word count ratio
3. Average word length
4. Average paragraph length to word count ratio [4]
5. Average paragraph length to sentence count ratio
6. Punctuation to word count ratio
7. Sentence count to word count ratio
8. Word based n-grams of sizes 1,2,3

2.2 Classification

Previous research has shown that machine learning can be used successfully to tackle the task of author identification; see [6, 9] for an overview. In our experiments, we used an SVM classifier. SVM has shown that it performs well in high-dimensional spaces, and this was applicable to our task.

We extracted the above-described features, and as a result, for each document we obtained a feature vector. Given a problem, i.e., a set of known documents and a questioned document, we aggregated these feature vectors for all known documents. Similarly, we built a feature vector for the questioned document (but this time there was nothing to aggregate as it is only one). Finally, we produced a 10-dimensional vector for the (known set, questioned document) pair as follows:

for the first seven features (i.e., excluding the n-grams), we just subtracted them, and for the n-gram features, we calculated separately the cosine similarity for the 1-grams, the 2-grams and the 3-grams, and we used the values as eighth, ninth and tenth features. Then, we scaled the real values to the [0;1] range, and we saved the scaling factors. We further added a class label: *same* or *different* (author). On testing, we produced the 10-dimensional vectors in the same way, except that we reused the scaling factors from training.

3 Experiments and Evaluation

3.1 Experimental Setup

We used the following datasets for training (we used them one at a time, as we trained and tested separate models for each of the four languages):

- pan15-authorship-verification-training-dataset-dutch-2015-04-19
- pan15-authorship-verification-training-dataset-english-2015-04-19
- pan15-authorship-verification-training-dataset-greek-2015-04-19
- pan15-authorship-verification-training-dataset-spanish-2015-04-19

3.2 SVM Parameters

As we mentioned above, we used LibSVM. We chose a C-SVM type of classifier with radial basis function (RBF) kernel. We further set the following parameter values: $C = 1$, $\gamma = 0.5$, $\varepsilon = 0.001$. We used different cache sizes depending on the purpose of the executions: testing on our machines or testing on TIRA.

We selected the above SVM parameter values experimentally, following the recipe in the *Practical Guide to Support Vector Classification*[3], which can be found on the LibSVM website. After setting the described kernel and SVM type, we manually tried different values for the C and the γ parameters as prescribed, and we ended up with the above values. Given that the search process was not automated, we might have missed some better parameter values.

3.3 Official Results

Our official results are summarized in Table 1. More detailed results can be found at the corresponding TIRA page:[4] look for *kiprov15*, which is our team's name.

We can see in Table 1 that our highest score is for Greek, where our team was ranked 9th out of 15. A notable characteristic of our solution is its runtime. In all instances, our system was among the fastest-running ones: each testset was fully analysed and scored under two minutes.

However, our results for English, Spanish and Dutch are suprisingly low, which could indicate a bug in the execution of the above-described pipeline. We are planning a detailed investigation in future work.

[3] http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf

[4] http://www.tira.io/task/authorship-verification/

Language	AUC	C1	Final Score	Runtime	Placement
Greek	0.7086	0.6400	0.4535	00:01:01	9/15
English	0.4926	0.5243	0.2582	00:01:35	15/18
Spanish	0.2802	0.3400	0.0953	00:01:09	17/17
Dutch	0.2560	0.3476	0.0890	00:00:47	17/17

Table 1. Official results for the Author Identification task for our team *kiprov15.*

A possible explanation for our relatively good results for Greek could be the number of the known documents for each problem. It turns out that for Greek, there are generally less known documents per problem in the training data. This could help as having more known documents might be confusing for the classifier. When these documents are from different genres and have different text structure, the classifier might pick the wrong characteristics to focus on, i.e., try to model genre/structure instead of author.

4 Future Work

We have described the submission of the Sofia University team for the PAN'2015 Author Identification Task. We experimented with an SVM classifier using variety of features extracted from publicly available resources. Our solution was among the fastest, but it did not perform very well in terms of AUC and C1.

Note that our solution is configurable and can be easily expanded and tweaked, which we plan to explore in future work. For example, it is very easy to generate new language-specific features, e.g., by adding new processing resources to the GATE pipeline. A wide range of these are readily available, but a careful selection and evaluation might be further required. Such features can include lists of stopwords, language-specific resources, character n-grams [5], part-of-speech n-grams, etc.

Another idea is to try to artificially expand the training data by using some of the examples in the known set as questioned examples; with such a tweak, we will provide more training examples to the SVM, which is likely to improve its predictions at testing time.

5 Source Code

The project source code can be found on BitBucket:
https://bitbucket.org/StanimirNikolov/pan-author-identification

6 Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments, which have helped us improve the paper.

References

1. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
2. Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, ACL '02, pages 168–175, Philadelphia, Pennsylvania, USA, 2002.
3. Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. *Text Processing with GATE (Version 6)*. 2011.
4. G. Kokkinakis E. Stamatatos, N. Fakotakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35:193–214, 2011.
5. Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, volume 3 of *PACLING '03*, pages 255–264, Harifax, Canada, 2003.
6. Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26, 2009.
7. Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. SVM based learning system for information extraction. In *Deterministic and statistical methods in machine learning*, pages 319–339. Springer, 2005.
8. Ruslan Sharapov and Ekaterina Sharapova. Using of support vector machines for link spam detection. In *Proceedings of the 2011 International Conference on Graphic and Image Processing*, ICGIP '11, page 828503, Cairo, Egypt, 2011.
9. Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.