Software Library for Authorship Identification

Ivan Ivanov[1], Cvetina Hantova[1], Maria Nisheva[1,2],
Peter L. Stanchev[2, 3], Phillip Ein-Dor[4]

[1] Faculty of Mathematics and Informatics, Sofia University, Bulgaria
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
[3] Kettering University, Flint, USA
[4] Faculty of Management, Tel Aviv University
ivan.vladimirov.ivanov@gmail.com, cvetinahantova@gmail.com,
marian@fmi.uni-sofia.bg, pstanche@kettering.edu,
eindor@tau.ac.il

Abstract. The aim of this paper is to review some methods for text authorship attribution and to discuss the development of a software library with tools for automatic authorship attribution. The presentation is focused on an analysis of two groups of tools oriented to: (1) methods for extraction of features and (2) methods for computing the distance between character strings based on data compression algorithms.

Keywords: text authorship identification, compression algorithms, normalized compression distance, n-grams, natural frequency zoned word distribution.

1 Introduction

The problem of authorship identification (or authorship attribution) has a long and instructive history. The methods and techniques used to solve it varied down the ages. At the beginning the humanists compared literary works by different authors, but nowadays the need to identify the authorship of online texts, such as emails, blog posts, tweets, is also essential. Tools for authorship attribution have been integrated in e-Learning environments, forensics, etc. but so far with inconsistent results.

In the typical authorship attribution problem, given a set of candidate authors for each of whom text samples are available, a text of unidentified authorship is assigned to one candidate author. Beyond this typical problem, several other authorship analysis tasks are often defined, including the following [Stamatatos, 2009]:

- author verification – deciding whether a given text was written by a particular author or not;
- plagiarism detection – finding similarities between two texts;
- author profiling or characterization – extracting information about the age, education, sex, etc. of the author of a given text;
- detection of stylistic inconsistencies, as may happen in collaborative writing.

Multiple sets of features are extracted from the documents to define the author's style. In the 19th century, stylometric markers that tried to measure text complexity and vocabulary richness were dominant. Later, features were categorized in groups by the linguistic level they represent – character, syntactic, lexical, and semantic. Some of the best results are achieved using lexical features, especially functional word frequencies, even though they are very simple as features. The development of Natural Language Processing (NPL) makes it possible to parse the grammatical structure of the sentences more precisely and leads to more frequent use of syntactic features as author style determiners.

The variety of approaches to differentiating between different authors' styles after the features have been extracted is also large. In the early years of the authorship attribution problem the main techniques applied were statistical analysis and probability distributions. But as the machine learning approaches – Neural Networks, Decision trees, Naïve Bayes classifier and also Support Vector Machines – increase their influence, more and more experiments made use of them to attribute authorship.

In this way a number of potential practical applications of software tools for authorship identification became quite realistic. Among the most important modern application areas of authorship attribution it is worth to note the following:

- detection of plagiarism in e-Learning environments and e-Publishing systems;
- assistance in preparation of expert reports in criminology;
- identification of channels of threats in cybersecurity;
- providing digital libraries with tools for studying authors' style of writing.

This paper is aimed at a discussion of some first results in building a software library with various tools for automatic authorship attribution. The rest of the paper is organized as follows: Section 2 presents a brief overview of modern methods for authorship attribution; Section 3 and Section 4 present in a nutshell the set of tools available in the library at present; the last section (5) contains some concluding remarks.

2 Methods for authorship identification used in the digital age

An authorship identification problem is defined in terms of a given set of candidate authors, a set of text samples of known authorship covering all the candidate authors (known as the training corpus), and a set of text samples of unknown authorship (test corpus). Each element of the test corpus should be attributed to a candidate author. The most widely used authorship identification approaches may be differentiated according to whether they treat each training text individually or cumulatively for the particular author.

The approaches of the first group concatenate all the available training text samples for each author into one file and extract a cumulative representation of this author's style. So they form the author's *profile* from this concatenated text. In this way the differences between texts written by the same author are ignored. This group of approaches is called *profile-based approaches* [Mosteller, 1964].

The approaches of the other group – the so-called *instance-based approaches* [Luyckx, 2011] – use multiple training text samples for each particular author in order

to develop an accurate attribution model. Each training text is individually examined as an instance of the author's style.

2.1 Profile-based approaches

The main characteristic of these approaches is that they are based on concatenation of the available training texts per author in a single text file. This possibly very big file is used to extract the properties of the author's style. Then an unidentified text is compared with each author file using a specific distance measure and the most likely author is identified. As a side effect, the stylometric measures extracted from the concatenated text file may be different from those of each original training text. The profile-based approaches use a very simple training process. Their training phase contains only the extraction of profiles for the candidate authors. Then, the attribution model is usually based on a distance function that computes the differences between the profile of an unidentified text and the profile of each particular author.

2.1.1 Probabilistic methods.

The probabilistic methods are the earliest approaches to authorship identification. They try to maximize the probability of a given text belonging to a candidate author. Standard naïve Bayesian classifiers are often used. Peng et al. [Peng, 2004] achieved very good results for authorship attribution by the use of probabilistic word-level models for a specific corpus.

2.1.2 Compression methods

This group of methods does not produce a concrete vector representation of the author's profile. All the available texts for each author are first concatenated to form a single file and then a compression algorithm is called to produce a compressed file. The difference in bit size between the compressed file for a particular "new" text with unidentified author and the compressed file for each candidate author indicates the similarity of the "new" text and the writing style of the author.

Several compression algorithms have been used, for example: RAR, LZW, GZIP, BZIP2, 7ZIP, etc. RAR was found to be most precise.

These methods are applicable to both character and word sequences.

2.2 Instance-based approaches

Most of the modern authorship identification approaches consider each training text sample as a separate unit that contributes independently to the attribution model. More precisely, each training text sample is represented as a vector of attributes and the classification algorithm is trained using this set of vectors as a training set in order to develop an adequate attribution model. It is important to note that this type of classification algorithm strongly require multiple training instances for each class. Therefore, in case we have only one but possibly quite long training text for a particular candidate author (e.g., an entire book), it should be segmented into multiple parts of approximately equal length. On the other hand, if there are multiple training text sam-

ples of variable length per author, the training text instance length should be normalized.

2.2.1 Vector space methods

A text can be considered as a vector in a multivariate space. Statistical and machine learning algorithms can be used to build a classification model such as: discriminant analysis, Support Vector Machines, decision trees, neural networks, genetic algorithms, memory-based learners, etc. Some of these algorithms can handle high dimensional, noisy, and sparse data. This type of algorithms can be used when very short texts of some authors are available as a training set.

2.2.2 Similarity-based methods

These methods are based on the nearest-neighbor algorithm. The word frequency can be calculated using z-score. The Delta measure can be used for calculating the difference between the training texts set and an unknown text. Similarity-based methods are very effective for texts of at least 1,500 words.

3 Tools for feature extraction

Currently our software library contains two groups of implementation methods [Hantova, 2015]:

- methods for extraction of features and
- methods for computing the distance between character strings based on data compression algorithms.

The first group of tools implement two feature extraction methods that are not so frequently used. The methods were tested on several corpora with different characteristics so that the effect of the author set size and the effect of the data size was evaluated [Hantova, 2015].

The first method is based on syntactic n-grams [Cavnar, 1994]. As opposed to traditional n-grams, syntactic n-grams are obtained based on the order in which the elements are presented in syntactic trees. Their main advantage is the fact that they make use of the syntactic relations of the words, ignoring the arbitrariness that is introduced by the surface structure. To achieve a higher level of abstraction and independence from the topic, the elements used to construct the syntactic n-grams are not concrete words but syntactic relation tags.

The other method is Natural Frequency Zoned Word Distribution Analysis [Chen, 2012]. The words in the documents are divided into zones based on their natural frequency. Each zone is represented by two features that imply the frequency of the zone in the text and also the distribution of the words in the zone throughout the text.

The machine learning approach used to train a model that will determine the author of an unknown text after the features are extracted is Support Vector Machines [Diederich, 2003]. They are the learning method of choice in contemporary computational research.

The selected algorithms provide good results especially when the corpus is constructed of a small set of authors and there is enough training data for each of them.

The developed software tool is a web application. It builds the infrastructure to execute the experiments with the algorithms using different corpora and also visualizes various graphics to illustrate the results and make their interpretation easier.

4 Tools for automatic authorship identification using compression methods

The second group of tools currently included in our library implement a non-traditional method [Ivanov, 2013]. Its main idea is that standard data compression algorithms and techniques can be used to determine the distance between character strings. More specifically, the distance is computed with the so-called Normalized Compression Distance (NCD) metric [Cilibrasi, 2005]. A number of experiments were conducted to assess the accuracy of the approach on different types of text. These include pamphlets from the American Revolutionary era (the famous Federalist papers [Adair, 1944] which have become a classic instance of the authorship attribution problem) and texts from Bulgarian literature (short prose and poetry). The method was also applied for automatic authorship attribution of a particular type of text – source code of computer programs.

As a baseline for comparison in all the experiments a more traditional method with a reputation for accuracy was used – the method based on frequencies of symbol n-grams. All experiments have consistently shown the superiority of the method based on compression.

In all cases, where comparison was possible, our results in automatic authorship attribution using compression based methods are at least as good as the best results published in the available literature.

The developed software tool is web-based and provides infrastructure for the implementation of experiments on a selected text corpus and displays a variety of graphs showing the results and enabling their better interpretation. The results of the implemented methods are good especially in cases where the shell is composed of a small set of authors and there exists a wide variety of texts for each one of them.

Three types of experiments were carried out [Ivanov, 2013].

Measuring the precision of identification of the authorship of Federalist papers

This experiment aimed to determine how well different approaches to identification of authorship work in the context of the well-known Federalist papers (http://www.gutenberg.org/files/1404/1404-8.txt).

Two groups of algorithms were experimentally compared: on one hand, the ZLIB implementation of DEFLATE and BZIP2 as examples of compression algorithms, and on the other hand, particular algorithms based on frequencies of symbol n-grams. The cross-validation technique was applied with three different values of the size of the validation set (1, 3, and 5). The training set consisted of all works that were not included in the validation set. The sizes of the validation set and the training set are given in the columns |BM| and |OM| respectively.

Table 1. Precision of different algorithms in authorship identification of Federalist papers

|BM|	|OM|	DEFLATE (ZLIB)	BZIP2	n-grams			
				1-grams	2-grams	3-grams	4-grams
1	65	93.00% ± 7.7%	96.00% ± 5.9%	62.00% ± 14.6%	73.00% ± 13.4%	85.00% ± 10.8%	81.00% ± 11.8%
3	63	90.33% ± 4.8%	94.33% ± 3.8%	59.33% ± 8.9%	75.00% ± 6.9%	83.33% ± 6.3%	81.00% ± 6.2%
5	61	91.80% ± 3.4%	96.20% ± 2.4%	55.40% ± 6.9%	77.00% ± 5.1%	83.40% ± 4.5%	79.20% ± 5.6%

Table 1 illustrates some results of this experiment. It is evident that BZIP2 is superior to the other methods and especially to those based on frequencies of symbol n-grams.

Experiment: Automatic authorship identification of source code.

Source code written in different programming languages (C/C++, Java, Python, and Perl) as solutions of problems within the [Google Code Jam] contest of 2012 was used for the purpose.

A set of results of this experiment are shown in table 2.

Table 2. Precision of different algorithms in identification of source code authorship

|BM|	|OM|	DEFLATE (ZLIB)	BZIP2	n-grams			
				1-grams	2-grams	3-grams	4-grams
1	236	95.00% ± 6.6%	93.00% ± 7.7%	40.00% ± 14.8%	91.00% ± 8.6%	93.00% ± 7.7%	94.00% ± 7.2%
2	235	96.50% ± 3.8%	93.00% ± 5.2%	33.00% ± 10.3%	88.00% ± 6.8%	92.00% ± 5.9%	91.50% ± 6.4%
3	234	97.67% ± 2.6%	96.33% ± 3.1%	32.33% ± 8.2%	88.33% ± 6.3%	92.33% ± 5.3%	92.67% ± 5.0%

It can be seen that almost all algorithms (except for 1-grams) achieve very good precision in automatic authorship attribution of source code on our exemplary dataset.

Experiment: Automatic authorship identification of Bulgarian literature.

For the purposes of the experiment a corpus of 10 individual works for each of the following five Bulgarian authors: Chudomir, Elin Pelin, Ivan Vazov, Yordan Yovkov, Jordan Raditchkov, was created. Some results of the experiment are shown in table 3.

Table 3. Precision of different algorithms in authorship identification of Bulgarian literature

|BM|	|OM|	DEFLATE (ZLIB)	BZIP2	n-grams			
				1-grams	2-grams	3-grams	4-grams
1	49	70.00% ± 13.8%	75.00% ± 13.1%	19.00% ± 11.8%	49.00% ± 15.1%	44.00% ± 15.0%	54.00% ± 15.0%
3	47	66.67% ± 8.0%	73.33% ± 7.4%	21.00% ± 6.1%	51.00% ± 8.3%	49.00% ± 8.6%	53.33% ± 8.9%
5	45	63.32% ± 6.4%	70.60% ± 6.2%	18.40% ± 5.1%	48.40% ± 6.2%	49.00% ± 5.7%	56.00% ± 6.0%

These results demonstrate a considerable superiority of compression-based algorithms to those based on n-grams. The difference in precision reaches and exceeds 20%.

5 Conclusion

The results of our research and experiments confirm the persuasion that there is no universal approach to authorship attribution that will give reliable results irrespective of the characteristics of the corpus of training texts. Because of that our future plans include some research and development activities directed to a step by step extension of our library with software tools implementing a variety of methods for authorship identification. The efforts will be concentrated on a selection of approaches most suitable in particular contexts. The first step will be to build a set of modules for analysis of the style of writing and identification of the authorship of short texts. An essential extension of the available set of training corpora is considered as well.

References

Adair D. (1944). The Authorship of the Disputed Federalist Papers. The William and Mary Quarterly ser. 3, vol. 1, no. 2: 97-122.

Cavnar W., Trenkle J. (1994). N-gram-based text categorization. In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval SDAIR-94, 161–175.

Chen Z., Huang L., Yang W., Meng P., Haibo Miao H. (2012). More than Word Frequencies: Authorship Attribution via Natural Frequency Zoned Word Distribution Analysis. Cornell University Library.

Cilibrasi R., Vitanyi P. M. B. (2005). Clustering by compression. IEEE Transactions on Information Theory, 51(4), 1523-1545.

Diederich J. (2003). Authorship Attribution with Support Vector Machines. Applied Intelligence 19, 109-123.

Google Code Jam: https://code.google.com/codejam

Hantova C. (2015). Authorship attribution. MSc Thesis, Sofia University, Facuty of Mathematics and Informatics.

Ivanov I. (2013). Automatic authorship attribution using compression methods. MSc Thesis, Sofia University, Facuty of Mathematics and Informatics.

Luyckx K. (2011). Scalability Issues in Authorship Attribution. Vubpress.

Mosteller F., Wallace D. (1964). Inference and disputed authorship: The Federalist. Addison-Wesley.

Peng F., Shuurmans D., Wang S. (2004). Augmenting naive Bayes classifiers with statistical language models. Information Retrieval Journal, 7(1), 317-345.

Stamatatos E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology, 60(3), 538-556.