Authorship Identification and Author Fuzzy "Fingerprints"

Nuno Homem
INESC-ID
TULisbon – Instituto Superior Técnico
Lisbon, Portugal
nuno.homem@hotmail.com

Joao Paulo Carvalho
INESC-ID
TULisbon – Instituto Superior Técnico
Lisbon, Portugal
joao.carvalho@inesc-id.pt

Abstract— **Fingerprint identification is a well-known technique in forensic sciences. The basic idea of identifying a subject based on a set of features left by the subject actions or behavior can be applied to other domains. Identifying text authorship based on an author "fingerprint" is one such application. This paper considers the problem of extracting "fingerprints" from texts and matching them with those obtained from a set of known authors. It presents an innovative fuzzy fingerprint algorithm based on vector valued fuzzy sets. Words and other stylometric features are used to create the fingerprint. The implementation is based on an approximated fast and compact algorithm that allows the method to be used on near real time, even for a large number of authors and texts.**

Keywords: fuzzy fingerprints, vector valued fuzzy sets, similarity, frequent elements, approximate algorithms, data streams.

I. INTRODUCTION[1]

Text authorship attribution is an area with a long research history that has evolved significantly in last years with the introduction of modern machine learning classification techniques. History presents us with many unknown author texts for which several authors have been proposed. In the past, most of these authorship attribution attempts were based on circumstantial evidences or in text style comparisons. Style comparisons were largely dependent on the perception of the authors of the study. The first scientific studies of authorship date from the late nineteenth century, with the works of Mendenhall (1887), who studied the authorship of texts attributed to Bacon, Marlowe and Shakespeare. These first scientific studies tried to relate authorship with word length and relative frequencies.

In the twentieth century, Zipf [16] gave significant steps in the understanding of the distribution of word usage in a languague. Zipf's presented an empirical law that states that in a language, the frequency of any word is inversely proportional to its rank in the frequency table. The most frequent word appears twice the second most frequent word, three times the third most frequent, etc. The Zipf's Law is discrete power law probability distribution that can be used to describe several physical and social sciences' phenomena. An interesting consequence of Zipf's Law is that most of the words occur only once in a set of texts. Those words are designated as hapax legomena (sometimes abbreviated to hapaxes), a Greek term meaning "[something] said [only] once". Hapax legomena have also been used for authorship attribution.

Text authorship encompasses several related albeit distinct problems:

- The one out of many problem – identifying a text author from a pool of possible or suspect authors where the author is always in the pool of suspects.
- None or one out of many problem – similar to the above, but the author may not be in the pool of suspects.
- The single author problem – estimating the probability of a text having been written by the given author.

The difficulty of the text authorship problem is naturally exponentially higher the larger number of possible authors. The availability of author text samples is also a major constraint when approaching this problem.

In this work one considers the problem of extracting a fingerprint from a set of texts and then using that fingerprint to identify the author of a distinct document.

Fingerprint identification is a well-known technique in forensic sciences and widely documented. In computer sciences a "fingerprint" is a procedure that maps an arbitrarily large data item (such as a computer file, or author set of texts) to a much compact information block, its "fingerprint", that uniquely identifies the original data for all practical purposes, just as human fingerprints uniquely identify people for practical purposes.

In computer sciences, fingerprints are typically used to avoid the comparison and transmission of bulky data. For example, in order to efficiently check if a remote file has been modified, a web browser or proxy server can simply fetch its fingerprint and compare it with the fingerprint of the previously fetched copy. Fingerprints are a fast and compact way to identify items.

To serve the author identification purposes, a fingerprint must be able to capture the identity of that author. In other words, the probability of a collision, i.e., two authors yielding the same fingerprint, must be small. The fingerprint has also to be robust; a text should be identified even if the author changes some aspects of the style. The idea of identifying text authorship based on an author fingerprint is a very appealing

This work was in part supported by FCT (INESC-ID multi annual funding) through the PIDDAC Program funds.

978-1-61284-968-3/11/$25.00 ©2011 IEEE

one, because identification can theoretically be made on near real time.

To be useful, the fingerprint should comply with some basic criteria:

- Include a minimal set of features that describe the author in a compact format.
- Allow for update operations whenever new information (texts) on the author is available.
- Allow for a fast comparison process once a new text needs to be identified.
- Scalability, i.e., performance should not degrade significantly when the number of texts or authors in the pool increases.
- Flexibility, i.e., should allow new authors to be included in the process, whenever information is available.

This paper proposes a new method for identifying text authorship given a set of possible authors, and proposes an authorship test to use in the single author problem. By using the word frequencies as a proxy for the individual behind a specific text, one can gather information on the author and identify other texts. The use of word frequencies is a well-known technique; the bag-of-words model for a text has been used for many years in this area.

The first step in the proposed method is to gather the top-*k* word frequencies in all known texts of each known author. An approximated algorithm is used for this purpose since classical exact top-k algorithms are inefficient and require the full list of distinct elements to be kept (storing 100000 words per author is inefficient if only the top-1000 are needed.) The Filtered Space-Saving algorithm [5, 6] is used for this purpose since it provides a fast and compact answer to the top-*k* problem although it only gives an approximate solution. This paper defends that the algorithm approximation is not an issue, as a degree of change or randomness has to be expected and incorporated into the detection method.

Once the top-k word frequencies are available, the fingerprint is constructed by applying a fuzzifying function to the word frequencies. This paper proposes the innovative method of fuzzifying the set of features based on their order on the top-*k* list instead of their frequency value.

The last part in the process is to perform the same calculations for the text being identified and then to compare this text fuzzy fingerprint with all the available author fuzzy fingerprints. The most similar fingerprint is chosen and the text is assigned to the fingerprint author.

II. Relation With Previous Work

Text authorship identification is a problem with a long history and multiple applications. Juola [8], Stamatatos [15] and Koppel [12] present extensive and comprehensive surveys of the history and state or art in this area. One very interesting experiment in this area was the competition organized by Juola [7] in 2004.

Initial research in authorship identification was focused in finding statistical features for quantifying the writing style. This line of research is known as stylometry. Measures for word length, sentence length, character frequencies, word frequencies or ratio of unique words, attempted to capture the essence and the differences between authors.

Authorship identification has improved significantly since the late 1990s due to the use of new machine learning techniques. Focus has shifted towards identifying a relevant set of features and them applying standard classification algorithms. Features include character features, such as character frequencies, uppercase, lowercase, digit, etc., and n-grams frequencies.

N-grams are obtained by extracting all sequences of n characters from a text. Frequencies are then calculated. N-grams are one of the most widely used features and with remarkably good results [9].

Word frequencies are other widely used features. In most cases, the bag-of-words model is used. In this model, a text is seen as a set of distinct words without any ordering. Texts are simply modeled by distinct word counts. This simple model has also proved very successful. In most cases, only the most frequent words are considered as in Burrows Delta method [2], [3]. This method uses the *k* most frequent words in the training set, computes the standard deviation of each word frequency and computes the sum of the *z*-scores for each word between frequencies of the author and the document to attribute (absolute difference between frequencies, divided by the standard deviation of the training set). The document is attributed to the author with the lowest *z*-score sum. Performance of Burrows' Delta has been used as a reference to measure other methods performance, quoting Juola [8]:

"Performance of Burrows' Delta has generally been considered to be very good among attribution specialists, and it has in many cases come to represent the baseline against which new methods are compared."

Syntactic and semantic features can also be used, but these depend on the use of natural languages tools to parse and classify the text. These features and their extraction process are very dependent on the text language.

Standard classification methods have been applied to this problem. Mosteller and Wallace [14] were the first to apply Naïve Bayes classification to a set of function words (a few dozen words commonly used in the language, such as *the*, *with*, *in*, *by*, *about*). This work and others using word frequencies have demonstrated this as a reliable method for authorship identification. It has been demonstrated that the Burrows Delta method is in fact a maximum likelihood classifier when word frequencies follow a Laplacian distribution [1].

In fact, most classification methods such as Neural Networks, Decision Trees, SVM, etc, have already been applied to this problem. References to these studies can be found in the mentioned surveys. Unfortunately, not much work has been done in the authorship identification area using fuzzy techniques.

A recent work by Cormode et al. [4] presents the concept of signature algorithms applied to iterations between individuals and provides some signature schemes to network traffic and telecommunication calls. A generic approach and a theoretical framework for signatures communication graphs analysis are provided. The "signatures" concept has some common points to the proposed author fingerprint. However, one prefers to use the "fingerprint" designation, as the algorithm aims at extracting information about the author that he has not provided knowingly, while a "signature" usually refers to information that was created specifically to identify someone or something. In [4] the feature extraction is exact; the use of approximated algorithms is suggested and several distinct distances are proposed.

The fuzzy fingerprint concept is a generalization of the Vector Valued Fuzzy Sets (VVFS) concept introduced by Kóczy [10]. The qualitative meaning of an object is represented by the quantities of the VVFS.

The vector valued fuzzy sets concept has also been used in [11] to introduce the fuzzy signature concept. Fuzzy signatures can model sparse and hierarchically correlated data with the help of hierarchically structured VVFS and a set of not-necessarily homogenous and hierarchically organized aggregation functions.

III. THE FILTERED SPACE-SAVING ALGORITHM

To allow the use of authorship identification techniques in near real time and for a large number of potential authors and documents, a key issue is to be able to extract the relevant features using an efficient algorithm with reduced memory usage. In this case, features are the most frequent words in the author's texts. The choice was to use an approximate top-*k* algorithm capable of generating good quality estimates using a reduced memory footprint. The Filtered Space-Saving algorithm [6], modified to handle weighted counting, was chosen. Filtered Space-Saving, originally presented in [5], is an evolution from Space-Saving algorithm presented by Metwally and al. [13].

The Filtered Space-Saving (FSS) algorithm uses a bitmap counter with h cells, each containing two values, α_i and c_i, standing for the error and the number of monitored elements in cell i. An hash function that transforms the input values (words) into an uniformly distributed integer range is used to obtain $h(x)$. The hashed value $h(x)$ is then used to increment the corresponding cell on the bitmap counter. Initially all values of α_i and c_i are set to 0.

The second storage element is a list of monitored elements A with size m. The list is initially empty. Each element contains three parts; the value itself v_j, the estimate count f_j and the associated error e_j.

The minimum required value to be included in the monitored list is always the minimum of the estimate counts, $\mu = min\ \{f_j\}$. While the list has free elements, the minimum is set to 0.

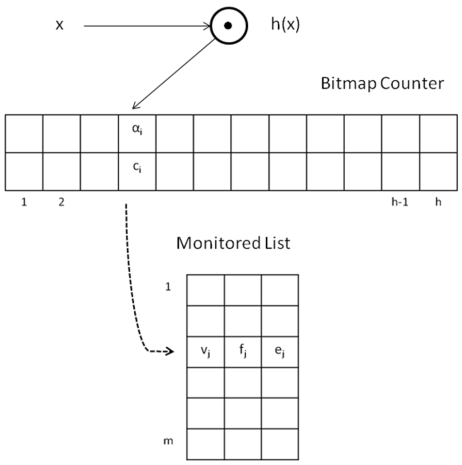

Figure 1 – FSS Algorithm Diagram.

The algorithm is quite simple. When a new word is received, its hash is calculated and the bitmap counter is checked. If there are already monitored elements with that same hash $(c_i > 0)$ the list is searched to see if this particular element is already there. If the element is in the list then the estimate count f_j is incremented. If the element is not in the list then it is checked to see if it should be added.

A new element will be inserted into the list if $\alpha_i + 1 \geq \mu$. If the element is not monitored then α_i is incremented. In fact this α_i stands for the number of elements with hash value i that have not been counted in the monitored list; it is the maximum number of times an element that is not in the list and that has this hash value could have been observed.

```
Algorithm: FSS(h cells, m counters, S stream)
begin
for each element, x, with value w, in S {
    set min to min {fj}
    let i be the hash(x) mod h
    if ci is not 0 {
        if x is monitored {
            let j be the index of x in the list
            increment fj
            continue for next x
        }
    } // this will only be executed if x is not
monitored
    if  αi + 1 >= min {
        if list size equals m {
            let m be the index with lower fj
            and for same fj with higher ej
            let k be the hash(x) mod h
            decrement ck
            set αk = fi
            remove vm
        }
        include x in the list in index j
        set vj to x
        set ej to αi and fj to αi+1
        increment counter ci
    } else {
        increment αi
    }
}// end for
end
```

Figure 2 – The FSS Algorithm

978-1-61284-968-3/11/$25.00 ©2011 IEEE

If the element is included in the monitored list, then c_i is incremented and set $f_j = \alpha_i + 1$ and $e_j = \alpha_i$.

If the list has exceeded its maximum allowed size, then the element with the lower f_j is selected. If there are several with the same value, the one of those with the larger value of e_j is selected. The selected element is removed from the list, the corresponding bitmap counter cell is updated, c_j is decresead and α_i is set with the maximum error incurred for that position in a single element, which is the estimate for the removed element, $\alpha_i = f_j$. When $h=1$, FSS is exactly the Space-Saving algorithm.

IV. The Fuzzy Fingerprint Algorithm

The main concept behind this algorithm is that authors have a stable enough behavior that allows a set of features to be extracted, fuzzified and then compared. The most frequent words in the texts of a single author present the required stability.

The bag-of-words model is used. This is a simplified model used in natural language processing and information retrieval. In this model, a text is represented as an unordered collection of words, disregarding grammar and even word order. The simplifying assumption that those variables are independent is considered.

One additional decision to consider is how punctuation and other stylometric features are handled. The way punctuation is used and features like the length of sentences and paragraphs can distinguish authors, and are therefore relevant for a fingerprint. In fact, the proposed method handles not only words but also any token as long as tokens are consistently handled, so this does not constitute any problem.

Author texts can then be analyzed as a set of word counts generated by a stable distribution that depends only on the author. Only word counts are in fact relevant and used as variables.

The set of words to consider in the fingerprint should be large enough to allow a comprehensive sample of the author style and vocabulary. The FSS algorithm requires two parameters, the size of the monitored list m, and the size of the bitmap counter h. In all the tests the parameters were set proportional to the number of words used in the fingerprint: $m = 3k$, $h = 9k$.

A. Fuzzy Fingerprint Creation

The full set of known texts are processed through the modified FSS algorithm to compute the approximated top-k list and frequencies for each author. Consider T_j is the set of texts by the author j. The result consists of a list of k tuplets $\{v_i, n_i\}$ where v_i is the i-th most frequent word and n_i the corresponding count estimate.

To create the actual fingerprint, the top-k list has to be fuzzified. The choice of the fuzzifying function is critical and the chosen approach is to assign a membership value to each word in the set based only on the order in the list. In fact, experiments have shown that the order of the frequency seems more relevant than its actual value. The more frequent words will have a higher membership value.

Several alternative membership attribution functions for each element i of the top-k list are tested in this paper. The simplest one is:

$$\mu\_(i) = \frac{k-i}{k}. \tag{1}$$

Function μ_{ab}, a function that weights more the initial words in the list is also tested:

$$\mu_{ab}(i) = \begin{cases} 1-(1-b)\dfrac{i}{k} & \text{if } i < a \\ a\left(1-\dfrac{i-a}{k-a}\right) & \text{if } i \geq a \end{cases} \tag{2}$$

The third function is μ_{erfc}, based on the complementary error function:

$$\mu_{erfc}(i) = 1 - erf\left(\frac{2i}{k}\right), \tag{3}$$

where *erf()* is the Gauss error function. Figure 3 presents the used functions.

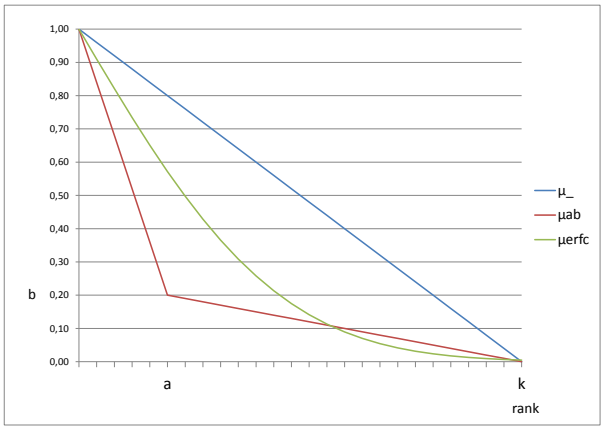

Figure 3 – Fuzzyfying functions

The fingerprint based on the top-k list (size-k fingerprint, Φ), consists on a size-k fuzzy vector where each position i contains a word v_i and a value μ_{vi} representing the fuzzified value of v_i's rank (the membership of the rank).

An author j will be represented by its fingerprint $\Phi(j) = \Phi(T_j)$. The set of all author fingerprints will constitute the fingerprint library.

B. Fuzzy Fingerprint Detection

In order to find the author of an unknown text D, one starts by computing the size-k fingerprint of D, $\Phi(D)$. Then one compares the fingerprint of D with the fingerprints $\Phi(j)$ of all authors present in the fingerprint library. Authorship is attributed to the author j that has the most similar fingerprint to $\Phi(D)$. Fingerprint similarity, $sim\Phi_{D,j}$, is calculated using (4):

$$sim\Phi_{D,j} = \sum_{v} \frac{\min\left(\mu_v(\Phi(D)), \mu_v(\Phi(j))\right)}{k}, \tag{4}$$

where $\mu_v(\Phi(x))$ is the membership value associated with the rank of word v in fingerprint x.

Note that all the definitions remain valid if n-grams are used instead of words.

V. Experimental Results

The experiments used a large set of newspaper articles from 87 distinct authors. This set comprises 5177 articles with a total of 1489947 words. The articles are written in Portuguese and were published during a period of 180 days in daily newspaper Público (considered a reference in Portuguese daily newspapers).

Articles were processed in chronological order and divided into two blocks, each with half of articles of each author. The first block was used to create the fingerprints for each author and the second was used to test authorship identification. The training set comprises 2569 articles and the test set 2608. Each of the articles in the test set was tested against 87 possible authors. Note that this is a much higher number of possible authors than in most works on this field, where only 5 to 10 authors are usually considered.

The set of tests compares distinct tokenization and techniques. Three distinct tokenization methods were used:

- 4-gram character splits;
- Words and punctuation;
- Words, punctuation and additional stylometrics features.

Punctuation improves significantly the identification rate, so it was always included, even in n-gram splits. The additional stylometrics features used were:

- Number of words per clause, truncated to the values 0, 3, 6…27, 30, 40, 50, 60, 80, 100;
- Number of clauses per sentence.

Distinct algorithms were used for each tokenization:

- Burrow's Delta *z-score*;
- Fuzzy fingerprints using basic membership function $\mu\_$;
- Fuzzy fingerprints using membership function μ_{ab} with $a = 0.2k$ and $b = 0.2$;
- Fuzzy fingerprints using membership function μ_{erfc}.

Figure 4 shows the correct author identification rate (first candidate author correctly identified) for word and punctuation tokenization approach, as a function of the number of tokens used in the identification process. Tests with Burrow's Delta *z-score* use the k highest global frequency tokens; tests fuzzy fingerprints use the top-k author and test article frequency tokens. The value of k was changed from 100 to 2500 in steps of 100 tokens.

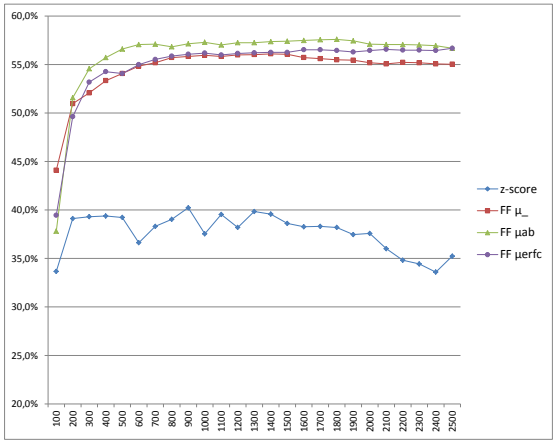

Figure 4 – Accuracy as a function of k when using words and punctuation

Figure 4 shows very clearly the differences in performance of each algorithm. Use of *z-score* with words and punctuation leads to unsatisfactory results. The fuzzy fingerprints algorithms give much better and more stable results. The best results are obtained with fuzzy fingerprints using improved membership attribution function μ_{ab} with $a = 0.2k$ and $b = 0.2$

Figure 5 shows the accuracy results for words, punctuation and additional stylometrics features.

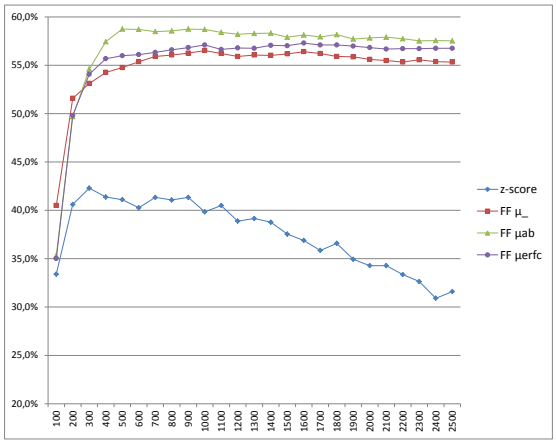

Figure 5 – Accuracy as a function of k when using words, punctuation and additional stylometrics features

Results are better than those observed with words and punctuation, the additional features introduced improve the author discrimination. Once again fuzzy fingerprints using improved membership function μ_{ab} *with* $a = 0.2k$ and $b = 0.2$ give the best results.

Figure 6 shows the accuracy results when 4-grams are used. Performance increases slowly with k, reaching very similar levels of those achieved with words, punctuaction and additional stylometrics features but with much higher k values. The basic membership function $\mu\_$ generates better results, probably due to the fact that 4-grams distribution is less skewed than that of words. Burrows's Delta applied to 4-grams leads to much better results than those achieved with words but stil worse than those obtained with fuzzy fingerprints. Fuzzy fingerprints using membership function μ_{erfc} achieve reasonable results either with words or n-grams.

978-1-61284-968-3/11/$25.00 ©2011 IEEE

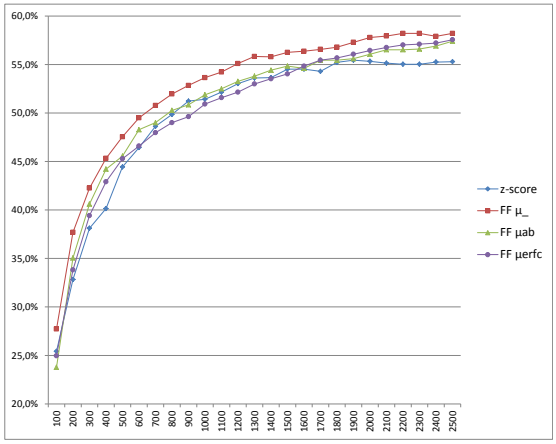

Figure 6 – Accuracy as a function of *k* when using 4-grams

VI. Conclusions and Future Work

This work shows how fuzzy methods may be used to identify authors of new texts. The use of simple fuzzy techniques based on approximate algorithms leads to very interesting results.

The obtained results also show that a commonly used and text specific method of comparable complexity, Burrow's Delta, used in this work as a reference, performed badly in this set of tests. The issue seems to be the large number of possible authors. Most other authorship identification studies based on Burrow's Delta use a much lower number of candidates. Burrow's Delta seems not to be a good choice in situations with large number of authors, at least with the size of text samples available in this case. However, the use of n-grams as an alternative to words in Burrow's Delta provided much better results. This approach hasn't been detailed in other studies and might justify further analysis.

The proposed method achieves always better results (either with words of n-grams) than Burrow's Delta. The use of words in the proposed method allows a much lower number of features than the use of n-grams, resulting in reduced memory use. The fact that the results remained stable for a wide range of values of *k* and for several fuzzification functions shows that the proposed method is robust. However, further analysis should be done on other languages and sets of texts to establish a set of parameterization rules.

The use of simple stylometric features such as the number of words or number of clauses in a sentence further improves the results. Further research can identify other interesting features.

The present study uses a very large set of candidate authors, while most other studies use less than 10 authors. Future research should extend the comparison between this method and others.

The large number of authors it supports and the ability to include new authors or update existing author fuzzy fingerprints is critical to the use of the method in long-term detection processes. New fuzzy fingerprints can be created and added to the author's library at any time, and new texts from known authors can be added to the fuzzy fingerprint. Update to one fuzzy fingerprint does not influence all the others.

The proposed method should not be seen as a text specific method; it can be used in other domains to identify individual behavior based on events. The proposed method will identify individuals as long as they present a stable event distribution.

References

[1] S. Argamon, Interpreting Burrows's Delta: Geometric and Probabilistic Foundations , Lit Linguist Computing , Vol. 23, pages 131-147, 2008.

[2] J. Burrows, Delta: A measure of stylistic difference and a guide to likely authorship, Literary and Linguistic Computing, Vol. 17, pages 267–287, 2002.

[3] J. Burrows, Questions of authorship: Attribution and beyond, Computers and the Humanities, Vol. 37, No. 1, pages 5–32, 2003.

[4] G. Cormode, F. Korn, S. Muthukrishnan, Yihua Wu, On signatures for communication graphs, IEEE 24th International Conference on Data Engineering (ICDE), 2008.

[5] N. Homem and J. Carvalho, Estimating Top-k Destinations in Data Streams, Computational Intelligence for Knowledge-Based Systems Design, Springer Berlin / Heidelberg, pages 290-299, 2010.

[6] Nuno Homem and Joao P. Carvalho, Finding top-k elements in data streams, Information Sciences, 180(24), pp. 4958-4974, Dec. 2010, Elsevier.

[7] P. Juola, Ad-hoc authorship attribution competition. In Proc. 2004 Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities (ALLC/ACH 2004), 2004.

[8] P. Juola, Authorship Attribution, Foundations and Trends in Information Retrieval: Vol. 1: No 3, pages 233-334, 2006.

[9] V. Keselj, F. Peng, N. Cercone, and C. Thomas, N-gram-based author profiles for authorship attribution, in Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING03, pp. 255–264, Dalhousie University, Halifax, NS, August 2003.

[10] L. Kóczy, Vector valued fuzzy sets, BUSEFAL- BULL STUD EXCH FUZZIN APPL, pages 41–57, 1980.

[11] L. Kóczy, T. Vámos, G. Biró, Fuzzy signatures, in: EUROFUSE-SIC99, 1999.

[12] M. Koppel, J. Schler and S. Argamon, Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology, 60: 9–26. doi: 10.1002/asi.20961, 2009.

[13] A. Metwally, D. Agrawal and A. Abbadi, Efficient Computation of Frequent and Top- k Elements in Data Streams, Technical Report 2005-23, University of California, Santa Barbara, September 2005.

[14] F. Mosteller and D. L. Wallace, Inference and Disputed Authorship: The Federalist. Reading, MA: Addison-Wesley, 1964.

[15] E. Stamatatos, A survey of modern authorship attribution methods, Journal of the American Society for Information Science and Technology, Volume 60, Issue 3, pages 538-556, 2009.

[16] G. Zipf, Selective studies and the principle of relative frequency in language, 1932.