A Scalable Framework for Stylometric Analysis Query Processing

Sarana Nutanong Chenyun Yu Raheem Sarwar Peter Xu Dickson Chow
Department of Computer Science, City University of Hong Kong
Kowloon, HKSAR, China
E-mail: snutanon@cityu.edu.hk, {chenyunyu4-c, rsarwar2-c, yaohaixu2-c, tschow9-c}@my.cityu.edu.hk

***Abstract*—Stylometry is the statistical analyses of variations in the author's literary style. The technique has been used in many linguistic analysis applications, such as, author profiling, authorship identification, and authorship verification. Over the past two decades, authorship identification has been extensively studied by researchers in the area of natural language processing. However, these studies are generally limited to (i) a small number of candidate authors, and (ii) documents with similar lengths. In this paper, we propose a novel solution by modeling authorship attribution as a set similarity problem to overcome the two stated limitations. We conducted extensive experimental studies on a real dataset collected from an online book archive, Project Gutenberg. Experimental results show that in comparison to existing stylometry studies, our proposed solution can handle a larger number of documents of different lengths written by a larger pool of candidate authors with a high accuracy.**

I. Introduction

Stylometry is the statistical analyses of variations in the author's literary style. The technique has been used in many linguistic analysis applications, such as, author profiling, authorship identification, and authorship verification, since the 19^{th} century [1], [2]. These applications are based on the observation that there exists such unconscious elements of author literary style that can help detect the original author of a disputed text. In this investigation, we focus on the application domain of plagiarism detection. Plagiarism detection in students text submissions is a major challenge for universities.

Large scale plagiarism detection databases, such as Turnitin (turnitin.com), PlagAware (plagaware.com), PlagScan (plagscan.com) and iThenticate (ithenticate.com) [3] are effective at detecting copy-and-paste plagiarism by comparing the contents of students' essays with a large corpus of documents of known sources. However, they are not designed to detect plagiarized work in which the work itself is *original* but is written by a different author. For example, a student may use an essay writing agency, such as Essay Tigers (essaytigers.com), Grab my Essay (grabmyessay.com) and Essay Thinker (essaythinker.com) as a ghost writer to create an original essay on their behalf [4].

In this investigation, we treat plagiarism detection as the *authorship Identification* problem [5]. Given a documents Q and a set X of verified documents with known authors Y, determine whether the essay was written by the student or find the original author of the essay. Through stylometric analyses, such a problem is generally solved by (i) building a classification model from the features vectors extract from X and their known authors as the associated label set Y; (ii) using the model to predict the author of the document Q of interest. In this way, we can identify whether Q is written by a ghost writer, even though Q is an original document.

Over the past two decades, authorship identification has been extensively studied by researchers in the area of natural language processing. These studies report a high accuracy over 95% using several types of stylometric features including lexical, character, syntactic and structural [6], [7], [8], [2]. However, these studies have the following limitations which makes them inapplicable to our large-scale plagiarism detection problem. 1) *Length Variations:* A slight variation in document lengths may affect the accuracy of authorship identification [9], [2], [8]. 2) *Size of the Candidate Author Set:* Existing studies report a drastic drop in the accuracy as the number of candidate authors increases [6], [7], [8]. In fact, in most of the existing studies [6], [7] reporting a high accuracy, the number of candidate authors do not exceed 20.

In this paper, we propose a novel solution which enables us to apply stylometry to a database-scale plagiarism detection problem. We address the two aforementioned limitations by proposing a new document representation model, as well as, a complete pipeline for authorship identification. Specifically, we represent each document as a set of data points in a high-dimensional space. In this set representation, each data point represents a chunk (a document segment with a fixed size) and each dimension corresponds to a stylometric feature. As a result, a longer document is represented as a set with a larger number of data points.

In order for us to handle a large number of documents, we apply the *probabilistic k-nearest neighbor (PkNN)* classification method to obtain the probabilistic distribution over the candidate authors [10]. Our main motivation for adopting PkNN is that it is an instance-based learning method. That is, classification is done through a comparison with instances stored in memory rather than a generalized model. The main advantages of this learning approach is that (i) little or no training is needed; (ii) the model can represent a complicated target function; (iii) there is no information loss through generalization [11].

Although the described approach helps us overcome the two limitations, it also presents a new computational challenge. As an instance-based learning method, a classification task using PkNN involves comparing the document of interest with multiple similar instances. Furthermore, the set representation also makes indexing and finding similar documents more costly.

To address the computational challenge, we apply an approximation technique called *locality sensitive hashing (LSH)* and design pruning methods for three different set similarity measures. LSH is often used as a tool to transform a high-dimensional similarity search problem into a collection of exact match lookup queries. The crux of our efficiency improvement solution lies in the way we use LSH to identify similar documents and the way we prune candidate documents to obtain the top-k results for PkNN classification.

We performed an experimental evaluation using a corpus obtained from Project Gutenberg[1], an online book archive. Our corpus contains 2386 novels from 136 different authors, which is significantly larger than any of those in the existing studies on stylometric analysis.

Our contributions are given as follows: (i) a new stylometric data representation model for handling documents with different lengths and for handling a large number of authors; (ii) an efficient solution for identifying candidate documents using LSH; (iii) an error analysis on the proposed LSH-based solution; (iv) an experimental study using a real dataset. (v) an entropy-based method to help determine whether a prediction should be trusted.

The rest of the paper is organized as follows. Section II provides a literature review. Section III presents problem formulation. Our proposed solution overview is discussed in Section IV. Section V presents our proposed efficiency improvement techniques. In Section VI, we report results from our extensive experimental studies. Section VII presents our conclusioning remarks and a future work discussion.

II. Literature Review

A. Stylometry

Stylometric analysis tasks can be organized into two types, authorship identification and writing style similarity detection [1]. Each task is consist of two steps (i) finding appropriate stylometric features (ii) forming efficient approaches to apply on these features. The objective of authorship identification task is to compare query texts against the writing samples of the candidate authors. A well-known example of authorship identification is the Federalist papers [12] in which 12 disputed/anonymous essays were compared against writing samples of Alexander Hamilton and James Madison. Since, all candidate authors (possible classes) are known a prior, the authorship identification task can use supervised or unsupervised classification techniques.

Stylometric features are writing-style-markers or attributes that are effective discriminators of authorship. A vast array of stylometric features have been proposed so far including lexical, syntactic and structural. Lexical features are character-based and word-based statistical measures of lexical variations. These include word and character lengths, sentence lengths [6]. Syntactic features include function words [12], part-of-speech tag n-grams [2]. Structural features include the style-markers related to the layout and organization of text. For example, number of words in sentence and number of words in paragraph.

Existing studies used the equal size of the text to compare for authorship identification [8], [2]. This is a challenge in real datasets. This is because, a database of student assignments may contain documents with different lengths written by many different students. In this investigation we use author-group level stylometric features to make our system significantly faster than existing techniques with reasonable accuracy.

In this work, we use 56 stylometric features which are categorized into three types: *lexical*, *syntactic* and *structural*. The lexical features can be further subcategories into 14 character-based and 13 word-based features [6]. We use 27 syntactic features, which include function words [12], part-of-speech [2]. There are 2 structural features, the number of words in sentence and the number of words in paragraph.

B. Similarity Search in a High Dimensional Space

Since our work involve identifying similar writing styles with respect to a given documents using the stylometric features, we discuss techniques for similarity search in a high dimensional space in this subsection.

Indexing and Querying in a Real-Valued Vector Space. Locality sensitive hashing (LSH) is a method for approximate similarity search queries in high dimensional spaces [13]. Gan et al. [14] propose a technique called the *collision counting LSH (C2LSH)* method. Using C2LSH, the collision frequency is used to estimate the similarity of two points in a high-dimensional Euclidean space which is more suitable for range search than E2LSH. They also provide an error analysis for the range query.

Set Similarity and Outlier Management. The Hausdorff distance is a well-known measure for comparing two sets of points in a real-valued vector space. Specifically, the standard definition of the Hausdorff distance is given by

$$H(A, B) = \max\{h(A, B), h(B, A)\},$$

where $h(A, B)$ is defined as $\max_{a \in A} \min_{b \in B} d(a, b)$ and $d(a, b)$ is the distance between a and b. Using this measure, the two sets A and B are consider similar *iff* for every element in A, there is at least one element in B in proximity, and vice versa.

When we are required to only match A to a subset of B, we can use the function $h(A, B)$ to perform a directed distance calculation. We call this variant the directed Hausdorff distance. Note that $h(A, B)$ is actually *not* a distance function, since it does not satisfy the *identity of indiscernibles* and *symmetry* properties. However, the term *distance* is used in this paper for this type of function in the interest of brevity.

Other variants of SHD include the *modified Hausdorff distance (MHD)* [15] and *partial Hausdorff distance (PHD)* [16]. These variants are proposed to address the outlier sensitivity

[1] https://www.gutenberg.org/

drawback of SHD [17], [18]. They argue that even a single point can drastically change the SHD value and propose two different approaches to handle outliers.

III. Problem Formulation

The problem of authorship identification is decomposed into two parts, namely *candidate identification* and *classification*. The main focus of our investigation lies in the candidate identification part, while the task of identifying the author using the candidate documents is done by applying an existing classification technique.

The candidate identification part of our solution is concerned with identifying similar documents in the corpus. We model each document as a set of stylometric vectors and identification of documents with similar styles is done by comparing set distances with respect to the query document. Specifically, we consider three different set distance measures, the *standard Hausdorff distance (SHD)*, *modified Hausdorff distance (MHD)*, and *partial Hausdorff distance (PHD)*. A set distance function (SHD, MHD, or PHD) is used to identify the top-k documents with the minimum set distances, where the k value is the desired number of candidate documents.

The classification part makes use of the similar documents (with known authors) to predict the most likely author of the query document. We solve this classification problem using the probabilistic k-nearest neighbor (PkNN) classifier [10]. Specifically, the PkNN classifier accepts a set of k objects (documents) where object is accompanied by its class (author) and the distance with respect to the query as input. As output, PkNN produces a *probability mass function (PMF)* of the classes of objects.

IV. Proposed Solution Overview

Figure 1 provides an overview of our proposed system which consists of major components: (i) *data storage and management*; (ii) *runtime query processing*; (iii) *result presentation*. These components are described in the following subsections.

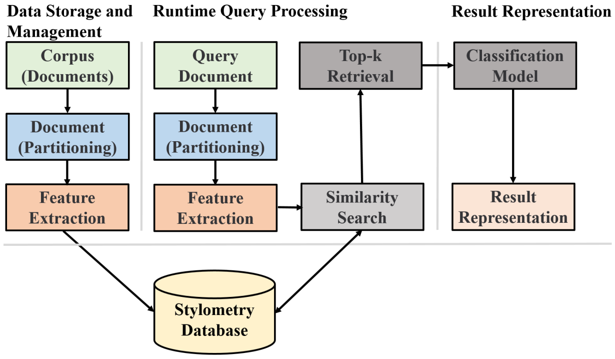


Fig. 1. System overview: similarity detection based on stylometric features

Data Storage and Management We create our corpus by extracting data from an online book archive, Project Gutenberg[2]. Our corpus contains 2386 novels from 136 different authors, which is significantly larger than any of those in the existing studies on stylometric analysis [6], [7], [8]. There is a substantial variation among the *lengths* of the sample documents of the candidate authors, i.e., 19,500 tokens - 1,684,500 tokens.

As mentioned earlier, in order to effectively handle documents of different lengths with large number of authors, we proposed a new stylometric data representation model. We partitioned document into chunks of equal size. We set the size of each chunk to 1,500 tokens [2]. A result, a longer document results with a greater number of data points in its set representation. Note that the last chunk is discarded to the variation in lengths.

Runtime Query Processing: Set Similarity Search. When a query document is submitted, our system performs the data transformation process described in Section IV. After the transformation, the query document is represented as a set Q of points where each point represents a chunk and each dimension corresponds to a stylometric feature. At this point, the query document is in the same data representation format as the documents in the corpus. We compare the query point set Q against the corpus in order to identify top-k similar documents. As described in Section II-B, these set distance measures are considered in this investigation.

Result Presentation and Classification. In this work, we choose the PkNN classifier [10] to provide a probabilistic classification result based on the top-k candidate documents. Specifically, the PkNN classifier provides a PMF over a set of possible authors. In addition, we also use the entropy of the probabilistic result to determine how confident we are with our prediction.

V. Efficiency Improvement

In this section, we show that we can exploit the *MaxMin* nature [19] of the Hausdorff distance to avoid evaluating the distance of each document in the database with respect to the query document Q. Specifically, by specifying the distance threshold r for every query point q in Q, we can separate candidate documents D into two groups, i.e., those with $h(Q, D)$ less than or equal to r and the rest. For the top-k query, as long as the size of the former is greater than k, we can ignore the latter. In addition, we also generalize this pruning concept to the modified Hausdorff and partial Hausdorff distance functions. Furthermore, we show that the process of identifying points in proximity to q can be greatly accelerated by an approximate range query processing method, C2LSH. We also provide an error analysis for using C2LSH to identify documents in proximity to Q.

A. Threshold-based Pruning for Top-k Processing

The main objective of the proposed pruning mechanism is reducing the number of documents that we need to compute the set distance. As stated earlier, the intuition behind our pruning mechanism is exploiting the MinMax nature of the Hausdorff distance using a range threshold. Figure 2 shows how to avoid computing the set distance for every document

[2] https://www.gutenberg.org/

by identifying data points close to every query point q in Q. Assume that the number of points in each point sets is 4. We can see that the distance $h(Q, D_1)$ is guaranteed to be smaller than the threshold r. This is because, for every q in Q, there is at least one data point from D_1. On the other hand, for D_2, D_3, and D_4, there is at least one data point missing from one of the query ranges. As a result, we can conclude that the distances $h(Q, D_2)$ $h(Q, D_3)$, and $h(Q, D_4)$ must be greater than r. Assume that we wish to find the top-1 document with respect to Q. We can directly return D_1 as the query result and safely ignore D_2, D_3, and D_4 without evaluating the actual distances.

Let us consider how to generalize this concept to MHD and PHD. Similar to SHD, MHD and PHD from Q to D can be computed by ranking the minimum distances $\min_{p \in D} d(q, p)$ for each q in Q. Unlike SHD, however, MHD involves computing the average of maximum distances. Hence, computing a lower bound of $h_m(Q, D)$ involves considering the lower bound of multiple query points rather than just one.

Based on the range query example in Figure 2, an example of this MHD lower bound calculation process is given in Table I. The table shows that the MHD $h_{m,50}$ is given as the average of the 50% of the distances. For example, $h_{m,50}(Q, D_1)$ is $\frac{3+2}{2}$ which is equal to 2.5 units. The same principle can also be applied to PHD. Specifically, for $h^{75}_{p,50}()$, we identify the distances that are in between the percentiles of 50% and 75%, which is the 2^{nd} one in the example (Table I). In order to identify the top-k most similar documents, we apply the best first search principle to compute the result set in an incremental fashion [20].

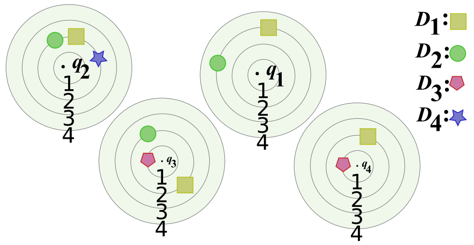


Fig. 2. Four instances of the range query retrieving data points in proximity of q_1, q_2, q_3, and q_4

TABLE I
DISTANCE AND DISTANCE LOWER BOUND CALCULATIONS OF THE EXAMPLES IN FIGURE 2

Doc.	Ranked Distances				SHD	MHD	PHD
	1_{st}	2_{nd}	3_{rd}	4_{th}	1^{st}	$[1^{st}, 2^{nd}]$	$[2^{nd}]$
D_1	3	2	2	2	3	2.5	2
D_2	> 4	3	2	2	> 4	> 3.5	3
D_3	> 4	> 4	1	1	> 4	> 4	> 4
D_4	> 4	> 4	> 4	2	> 4	> 4	> 4

B. Locality Sensitive Hashing

Locality sensitive hashing (LSH) is an approximation method for similarity search queries in high dimensional spaces [13]. In this work, we use C2LSH [14] as the method to identify similar points in a high-dimensional space. Specifically, we make use of the C2LSH range query support to accelerate the candidate pruning process described in the previous subsection.

In this work, we extend the C2LSH error analysis [14] for individual query points to support the Hausdorff distance and its variants. Specifically, by exploiting the aggregate nature of the Hausdorff distance, we show that the error rate of the Hausdorff distance (as well as its variants) is negligible. The following analysis is applicable to SHD, MHD, and PHD. Hence, we use the term "Hausdorff distance" to refer to all of them.

Our analysis is focused on the false negative rate (FNR). Specifically, a false negative is a document D with a Hausdorff distance less than the range r but *not* identified as a near neighbor of the query document Q. Such a set-level false negative is caused by false negatives and false positives at the point level.

Next, we estimate the probability that a document D with $h(Q, D) \leq r$ being a false negative. Consider a single query point q in the query set Q. Let N_i denote the number of points in D that fall inside the range r with respect to q and N_o denote the number of points that fall outside. In order for q to misjudge D as a point set whose minimum distance is greater than r the two conditions must be met. First, all of the N_i points inside the range must be false negative. Second, all of the N_o points outside the range must be true negative. Let P_1 denote the point-level FNR and P_2 denote the point-level FPR. The probability that the stated two conditions are satisfied is given by $P_1^{N_i}(1 - P_2)^{N_o}$.

Now let us consider the worst case, which is the case of SHD. Recall the pruning rule for the similarity search method based on SHD, the documents D is removed if there is no data point in D appearing in the result set of any query point q in Q. In order for a document D to be a set-level false negative with respect to Q, there has to be at least one query point q in Q that satisfies the stated conditions. As a result the, set-level FNR is given by

$$\text{Set-level FNR} = 1 - (1 - P_1^{N_i}(1 - P_2)^{N_o})^{|Q|} \qquad (1)$$

Let σ denote our desired set-level FNR bound. The values of P_1 and P_2 can be determined by

$$\ln(1 - P_1^{N_i}(1 - P_2)^{N_o}) \leq \frac{1}{|Q|} \ln(1 - \sigma) \qquad (2)$$

After we have determined the values of P_1 and P_2, we can use the analysis provided by Gan et al. [14]. Since SHD is the worst case, this analysis is also applicable to MHD and PHD.

VI. PERFORMANCE EVALUATION

A. Experimental Setup

Dataset: We create our corpus by extracting data from an online book archive, Project Gutenberg[3]. Our corpus contains 2386 novels from 136 different authors with a substantial

[3] www.gutenberg.org

variation among the *lengths* of the sample documents. Each document is partitioned into *chunks* and a feature vector is extracted from each chunk according to the process described in Section IV with a chunk size of 1,500 tokens. As a result, each document is represented as a set of 56-dimensional vectors. Each dimensions is also normalized to the range of [0, 1].

For the query documents, we choose documents written by authors who have written 30 to 60 documents. In order to demonstrate that our method can handle a substantial variation in lengths we sample the documents according to the number of chunks. That is, the query documents Q are organized into three groups according to the size $|Q|$: $(20, 40)$, $(40, 60)$, and $(60, 80)$. The total number of query documents is 60, where each group contains 20 documents.

Evaluation Measures. Our experimental studies include assessments on both efficiency and accuracy of our solution. We evaluate the efficiency of our solution based on two measures: *execution time (exec. time)* and *the number of set distance calculations (Set Dist. Cals.)*.

For the accuracy assessments, we measure the *strong* and *weak* accuracy. For the strong accuracy, a prediction is considered correct if the correct author is the most likely author. For the weak accuracy, we assume that our method is used for candidate author generation. As a result, a prediction is considered correct if the correct author is included as a candidate author based on the top-k document result set. In addition to the accuracy measures, we also assess the entropy of each prediction to quantify its uncertainty level.

Parameters. We also compare the efficiency and accuracy of different solutions by varying the values of $|Q|$, while the number k of candidate documents is fixed to 20. Specifically, the query set sizes $|Q|$ are organized into three groups as described in the previous paragraph.

Environment and parameter settings. Experiments were conducted on an `Intel(R) Xeon(R) CPUE5-2620 v2 @ 2.10GHz` dual-processor server with 96GB main memory. All algorithms and experimental study were implemented in Python. The LSH parameters are given as follows. (i) The number L of compound hash functions is set to 200. (ii) The number K of projections per compound hash function is set to 3.[4] (iii) The bucket width w of each projection is set to 1.5. (iv) The collision threshold is set to 20 which corresponds to the similarity retrieval range r of $0.15\sqrt{D}$, where D represents the number of dimensions, i.e., 56, in this case.

Let us now consider the Hausdorff distance variants, MHD and PHD. We choose the MHD percentage value of 50% and the PHD percentage range of [50%,75%].

B. *Efficiency Studies*

In this subsection, we evaluate the efficiency of our proposed solution using the cost measures described earlier. We compare our proposed LSH-based pruning method described in Section V-B with the baseline method which makes use of the pruning technique described in Section V-A.

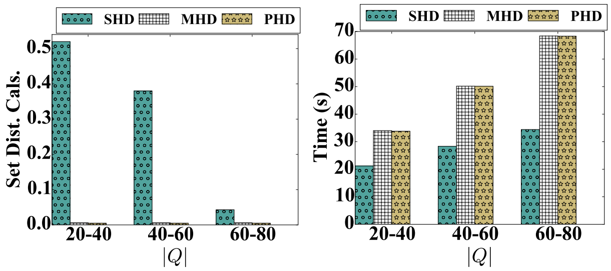
Table II provides a summary of efficiency studies' results. Each measurement is the average computed from 60 queries and the number k of candidate document is 20. As can be seen, A significant degree of candidate pruning were obtained from both methods.

TABLE II
SUMMARY OF EFFICIENCY STUDIES' RESULTS

Dist.	Method	Set. Dist. Cals.	Exec. Time
SHD	Baseline	10.22%	452.88
	LSH-based pruning	10.83%	27.65
MHD	Baseline	0.60%	463.00
	LSH-based pruning	0.62%	50.21
PHD	Baseline	0.59%	461.90
	LSH-based pruning	0.59%	50.79

In terms of the execution time, the table show that our LSH-based pruning method provides approximately an order of magnitude speedup in comparison to the baseline. We can also see that the execution times of MHD and PHD are consistently higher than that of SHD for both baseline and LSH-based pruning method. This is because the top-k processing for MHD and PHD involves a best-first search on candidate documents and lower bound calculations. For SHD on the other hand, we can safely disregard out-of-range candidate documents without lower bound calculations or a search algorithm. Since the baseline method is significantly outperformed by our proposed LSH-based pruning, we omit their results from the remaining of the section.

Let us now examine the effect of $|Q|$ on the distance calculation cost and the query execution time. The results are illustrated in Figure 3. As can be seen, the set distance calculation costs decreases as $|Q|$ increases. This is because, the Hausdorff distance, by nature, is a MaxMin distance. As the number of query points in Q increases the number of documents that fall in to the range r also reduces.



(a) Set Dist. Cal. Cost (b) Execution Time

Fig. 3. Effect of the query set size Q

Figure 3(b) shows that the query execution time increases as $|Q|$ increases. This is due to the increased number of points that we need to process for each query set.

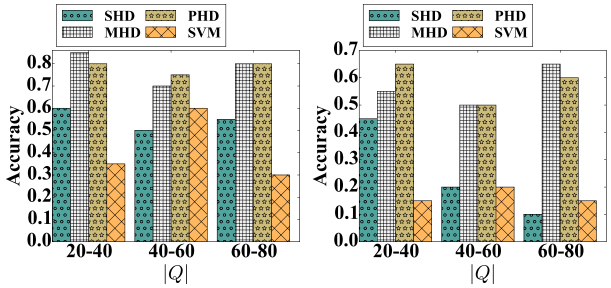
C. *Accuracy Studies*

For the accuracy studies, we introduce a comparative classification method based on the *support vector machine*

[4] Note that in the original definition of C2LSH, the K value is set to 1. In order for us to reduce the number of false positive, we increase the value of K to 3 and set the rest of the parameters according to the analysis given in the original C2LSH paper [14].

(SVM) [21]. Specifically, we create a probabilistic SVM classifier using the data points in the corpus. The label of each data point is given by the author ID of the corresponding document. The training set is the corpus subtracted by the test set, which is the set of query documents described in Section VI-A. The probabilistic SVM classifier provides a prediction result as a probability distribution over a set of classes (i.e., authors).

Accuracy: Effect of the query set size $|Q|$. Figure 4(a) illustrates the effect of $|Q|$ on the weak accuracy. We can see that MHD and PHD outperform SHD and SVM. This shows that MHD and PHD provide a more effective outlier management mechanism than SHD. We can also see that the weak accuracy of SVM is significantly lower than that of MHD and PHD.



(a) Weak Accuracy (b) Strong Accuracy

Fig. 4. Accuracy: Effect of the query set size $|Q|$

Figure 4(b) shows a significant gap between PHD and SVM in term of the strong accuracy. This result demonstrates the effectiveness of the set distance-based solution and our outlier management mechanism.

Accuracy: Entropy Analysis. One benefit of using a probabilistic classifier like PkNN is that we can determine the prediction uncertainty using the entropy. Consequently, one can decide whether to use the prediction result based on the entropy.

Figure 5 shows the attempted prediction ratio and the two types of accuracy measures with different *entropy cutoff* values. Specifically, we only use the prediction result when the entropy value is below the cutoff. As can be seen, as the cutoff value increases the number of used predictions reduces, while the strong accuracy and weak accuracy increase. For example, with the cutoff value of 0.8, the used prediction ratio is 82% and the weak accuracy is 96%, while the strong accuracy is 71%. However, if we use all predictions, the weak accuracy is 90%, while the strong accuracy is 67%. As can be seen, by using the entropy to rule out 18% of the predictions, we can significantly increase both weak accuracy and strong accuracy.

VII. Conclusion

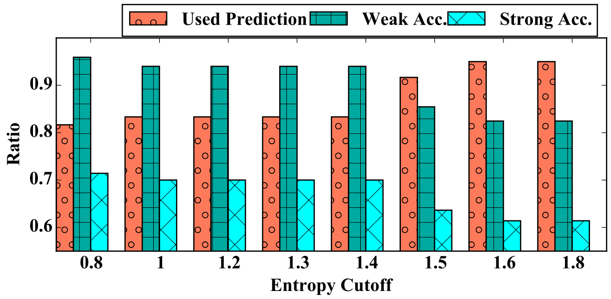


Fig. 5. Used Prediction Ratio, Weak and Strong Accuracy

We have presented a scalable method for authorship attribution on a real world dataset collected from an online book archive, Project Gutenberg. Experimental results show that in comparison to existing stylometry studies, our proposed solution can handle a larger number of documents of different lengths written by a larger pool of candidate authors with a high accuracy. We show that our method outperforms existing method in terms of query processing costs and accuracy.

Acknowledgment

This research was partially supported by two CityU research grants (CityU Project No.7200387 and No.6000511).

References

[1] T. C. Mendenhall, "The characteristic curves of composition," *Science*, pp. 237–249, 1887.

[2] E. Stamatatos, "A survey of modern authorship attribution methods," *JASIST*, vol. 60, no. 3, pp. 538–556, 2009.

[3] A. Ali, H. Abdulla, and V. Snasel, "Overview and comparison of plagiarism detection tools," pp. 161–172, 2011.

[4] R. Clarke and T. Lancaster, "Eliminating the successor to plagiarism? identifying the usage of contract cheating sites," in *PICAI*, 2006.

[5] G. Ledger and T. Merriam, "Shakespeare, fletcher, and the two noble kinsmen," *LLC*, vol. 9, no. 3, pp. 235–248, 1994.

[6] J. Grieve, "Quantitative authorship attribution: An evaluation of techniques," *LLC*, vol. 22, no. 3, pp. 251–270, 2007.

[7] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *TOIS*, vol. 26, no. 2, p. 7, 2008.

[8] K. Luyckx and W. Daelemans, "The effect of author set size and data size in authorship attribution," *LLC*, pp. 35–55, 2010.

[9] M. Eder, "Does size matter? authorship attribution, small samples, big problem," *PDH*, pp. 132–135, 2010.

[10] C. Holmes and N. Adams, "A probabilistic nearest neighbour method for statistical pattern recognition," *J R Stat Soc Series B Stat Methodol*, vol. 64, no. 2, pp. 295–306, 2002.

[11] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[12] F. Mosteller and D. Wallace, "Inference and disputed authorship: The federalist," 1964.

[13] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *STOC*, 1998, pp. 604–613.

[14] J. Gan, J. Feng, Q. Fang, and W. Ng, "Locality-sensitive hashing scheme based on dynamic collision counting," in *SIGMOD*, 2012, pp. 541–552.

[15] R. Lipikorn, A. Shimizu, and H. Kobatake, "A modified hausdorff distance for object matching," in *Pattern Recognition*, vol. 1, 1994, pp. 566–568.

[16] D. P. Huttenlocher, G. A. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[17] B. Achermann and H. Bunke, "Classifying range images of human faces with hausdorff distance," in *ICPR*, vol. 2, 2000, pp. 809–813.

[18] W. J. Rucklidge, "Locating objects using the hausdorff distance," in *ICCV*, 1995, pp. 457–464.

[19] S. Nutanong, E. H. Jacox, and H. Samet, "An incremental hausdorff distance calculation algorithm," *PVLDB*, vol. 4, no. 8, pp. 506–517, 2011.

[20] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," *ACM Trans. Database Syst.*, vol. 24, no. 2, pp. 265–318, 1999.

[21] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Appl. Intell.*, vol. 19, no. 1-2, pp. 109–123, 2003.