8
Writer Identification for Handwritten Words

Forensic handwriting investigation has dependably been subjecting to contentions since scientific specialists have been utilizing their subjective investigation by a visual examination of the authorship. In this chapter we address the task of writer identification of the handwritten words. We provide an extensive study of allographic methods for codebook generation. We also analyze the effects of segmentation on the writer identification task for handwritten words by constructing grapheme and character-based feature clusters.

This chapter is organized as follows. Section 8.1 introduces the challenges in writer identification. In Section 8.2 we present a brief summary of writer identification at allograph level. Section 8.3 states the methodology that we adopted to identify the author of a word. In Section 8.4 we describe the experiments we conducted to investigate the performance of the various allograph level clusters. We conclude this paper in Section 8.5.

8.1 INTRODUCTION

The challenges for writer identification include the variation in writing style depending on, the use of different pens, whether the writer has written in a hurry or not, and the inherent variability in human writing style, so that one word is rarely written, in exactly the same way, twice. The handwriting of a person may change over years and therefore makes further identification a harder task. Even in the presence of these challenges it can be seen that at least for humans, the writing differs clearly for different writers.

In recent years, many different approaches have been proposed for writer identification for different writers. There have been a number of new and effective attempts to identify documents [Bulacu and Schomaker, 2006; Schomaker *et al.*, 2007; Jain and Doermann, 2011; Muriel Visani and Bui, 2011; Fiel and Sablatnig, 2013; Jain and Doermann, 2014; Paraskevas *et al.*, 2014], textlines [Slimane and Margner, 2014], words [Zhang and Srihari, 2003; Tomai *et al.*, 2004; Chaabouni *et al.*, 2011; Slimane and Margner, 2014], and characters [Leedham and Chachra, 2003b; Pervouchine and Leedham, 2007] penned by specific writer.

Writer analysis can be studied under two levels: the Texture level [Said *et al.*, 2000; Zhang and Srihari, 2003; Wang *et al.*, 2003; Bulacu *et al.*, 2003; Tomai *et al.*, 2004; Schomaker and Bulacu, 2004; Chaabouni *et al.*, 2011; Fiel and Sablatnig, 2013; Jain and Doermann, 2014], and Character level [Pervouchine and Leedham, 2007; Schomaker *et al.*, 2007; Siddiqi and Vincent, 2007; Muriel Visani and Bui, 2011; Newell, 2013; Paraskevas *et al.*, 2014]. Features at texture level exhibit a global nature; they are informative for the habitual pen grip and the preferred writing slant [Schomaker and Bulacu, 2004]. Features at allograph level possess local nature and hence reveal the true character shapes [Schomaker and Bulacu, 2004]. For analyzing an allograph, first the handwriting has to be segmented. Thus, the writer identification process involves a segmentation method, which can introduce errors. After segmentation, features for individual writers can be calculated on these allographs. The writer identification methods, which use textural features, skip this segmentation step and calculate features directly on the image. They can be readily applied to any language

Figure 8.1 : The three types of handwritten Word Allographs from CVL Dataset

without any prior knowledge of the letters or allographs [Newell, 2013]. They are fast because they encode a whole document or passage in a single feature vector.

The allographs are the smaller unit of handwritten text; they can be further studied at grapheme, bi-gram and character levels. Graphemes are small strokes of handwriting, which are extracted by applying a robust segmentation algorithm [Paraskevas *et al.*, 2014]. Bi-gram is a combination of a character with another adjacent character, and characters are the single alphabet of a word usually manually segmented. Figure 8.1 gives example of the three types of allographs.

In this work, we exploit individual character shapes to identify the writers of handwritten words from a set of specified writers. So far, there have been few works that address offline writer identification for handwritten words [Zhang and Srihari, 2003; Tomai *et al.*, 2004; Chaabouni *et al.*, 2011; Slimane and Margner, 2014]. However, a proper exploration at allograph level is still lacking. We identify the writer of a particular word at allograph level, when multiple writers have penned a single document. In this paper, we examine the discriminability of writer identification at allograph level.

In this work, a codebook of graphemes is constructed after extracting the graphemes using a sliding window. In the following work, an assessment is made on 10 writers from CVL dataset [Kleber *et al.*, 2013]. The dataset includes handwritten words from a set of 4 documents of each writer. For each writer we use two of his document for training and the remaining two for testing. The grapheme level clusters are trained with 140 different words and further tested on 150 different words for each writer. We conduct two sets of experiments by building the codebook with overlapping and non-overlapping windows. The character level codebook is learned from 5698 characters from a total 10 writers it is tested upon 150 different words for each writer. K-means is used for codebook generation and one vs. rest SVM is used for further classification. In the end, majority voting decides the author of the given handwritten word.

8.2 RELATED WORK

Numerous studies have attempted to study the writer identification problem at texture level, where the methods identify the writer of a document based on the overall look and feel of the writing. The problem has been attempted at the global features of complete document as well as at local level that deals with the various shapes of the handwritten text. We here present the past work of writer identification with a focus at the allograph level and word level writer identification.

8.2.1 Allographs features

8.2.1.1 Grapheme level:

[Bulacu and Schomaker, 2006] worked at grapheme level for free handwriting in only lower case text. They claimed an improvement in writer identification rate by infusing both allographic and texture features. The Probability Distribution Function (PDF) was computed for features such as contour direction, contour hinge, contour co-occurrence, run length, and grapheme emission. The system yields a performance of 87% on 900 writers at document level, which is build by

combining writers from Firemaker, IAM and ImUnipen datasets. For grapheme extraction, a cut at lower contour is made where the distance between the upper and lower contour of the image is comparable to the ink width. [Siddiqi and Vincent, 2007] identified the writer with only grapheme features and reported an identification rate of 94% at document level. To extract graphemes, the handwritten text is divided into a large number of small windows of fixed size. Then a correlation similarity measure is used to cluster similar graphemes. Each document is then modeled as a Gaussian Mixture of grapheme codewords. Bayes decision theory is employed for document classification.

[Muriel Visani and Bui, 2011] claimed that for less than 150 writers in a system, grapheme-based method outperforms the character-based method but only for online handwritten text. They used both on-line and off-line features jointly to identify the writer and then recognize online handwritten text. This is termed as writer adaptive handwriting recognition system. Online graphemes are segmented using MyScript Builder and are classified into four groups depending on the initial writing direction. Further k-means is used to create several prototypes within each group and a similarity function is used to encode each writer's document into a weight matrix. The weight matrices are matched and the writers are identified for each document. In handwriting, the stroke thickness varies from pen to pen, ink to ink and paper to paper. In view of it, [Paraskevas *et al.*, 2014] suggests to reduce the stroke thickness down to one pixel by skeletonizing the text. Using a fixed squared window size the graphemes are extracted and the codebook is build by Kohonen SOFM [Kohonen, 1989]. The main contribution of the system lies in introducing an improvement in edge directional features, which results in achieving an accuracy of 95.6% with Manhattan distance.

8.2.1.2 Character level:

[Leedham and Chachra, 2003b] proposed a writer identification scheme by manually segmenting offline handwritten characters. They performed two experiments on CEDAR database for 30 writers one with local codebook while the other with global codebook. The local codebook is composed of sub codebooks of 52 types of characters. K-means is used as a clustering scheme for creating the codebook. For each writer, the nearest codebook is searched using Euclidean distance. Therefore, for each writer one histogram (in the case of global codebook) or 52 histograms (one per character, in the case of local sub-codebooks) are obtained. A Probability Distribution Function (PDF) is computed from each whether common or sub codebooks, by histogram binning. This PDF is used to characterize each writer. Finally writer identification is done by a distance function. Their work concludes that i) working with local sub codebook results in much better performance than using a unique single codebook, and ii) when some of the sub codebooks are combined only slight difference in performance was seen.

[Muriel Visani and Bui, 2011] in their work found that character based method is relatively robust to character segmentation errors. They also claimed that grapheme level identification is computationally more expensive and less effective than character extraction based method, for an increasing number of writers. But these findings were restricted to only online handwritten text and their contribution lies in combining online features with offline features. PCA was applied to reduce the dimensions of the online and off-line features. Their work reported an accuracy of 93% on PSI database for 88 writers for online writer identification on documents.

8.2.1.3 Bi-gram level:

Very few studies have investigated the impact of working with bigram level features on writer identification. The paper [Pervouchine and Leedham, 2007] presents a study of structural features of handwriting extracted from three characters ``d'', ``y'', and ``f'' and a bigram ``th''. They concentrated on the extraction of the micro level features like height, width, height to width ratio, relative height of ascender, slant of ascender, final stroke angle, fissure angle, relative height of

descender, descender loop completeness, descender slant, final stroke angle, slant at point, slant of t-stem, slant of h-stem, position of t-bar. Neural network is used as a classifier and genetic algorithm is applied for searching an optimal feature set for writer identification. The paper claims that the bigram possessed significantly higher discriminating power than any of the three single characters studied, which supports the opinion that a character form is affected by its adjacent characters. They reported an accuracy of 58% for 200 writers from 600 samples of the CEDAR letter dataset.

A recent study by [Newell, 2013] states that identification performance can be improved when bigrams are used for dictionary building. His work focused on the van der Maarten dataset comprising 251 writers with each writer writing a text passage. Each passage in the dataset contains only 32 types of characters in all. The test set contains only 20 characters. These characters are segmented from the text passage and histograms are formed either for each character for each writer or for group of characters. For the test passage these histograms are matched using NN classifier and on the majority voting the decision is built on the authorship of a particular writer. His work claimed that in case when characters of a writer are bi-paired with each other, the one paired with the same character gives the best identification rate.

8.2.2 Word-level writer identification

Recognition of handwritten words is more challenging. [Zhang and Srihari, 2003] used GSC (Gradient, Structural and Concavity) features on words and performed writer discrimination and verification. Their study majorly focused only on four characteristic words, ``been'', ``Cohen'', ``Medical'', and``referred''. Their dataset comprises 1000 writers who wrote only these four words on three documents and reported an accuracy of 83%. Features are drawn only from these words by dividing their images in n x n blocks and are classified by applying k-NN based on Correlation measure as similarity function. Following this, [Tomai *et al.*, 2004] used segmentation and segmentation-free feature extraction approaches to identify writer of a word image. Their dataset includes 75000 words images, representing 25 different words, written by more than 1000 writers. Their choice of features included GSC (Gradient, Structural and Concavity) features, WMR (Word Model Recognizer) features, SC (Shape Curvature) and SCON (Shape Context) features and reported an accuracy of 62% for identification with GSC feature.

Further [Chaabouni *et al.*, 2011] performed word level writer identification by combining the on-line and off-line features. For the same word the writer will form two images one offline and the other online. Then from both the word images high density of information points are identified and then they are surrounded by box. The points identified are called fractal and the features extracted from them are termed multi fractal features. The experiments are performed on 100 writers of ADAB database on 25 words and following conclusions were drawn:

1. The on-line fractal features (84.6%) outperform the off-line fractal features (80.9%).

2. The combination of online and offline fractal produces higher accuracies (93.2%).

3. To characterize the styles of writings online are better because they are more informative.

In a recent study, [Slimane and Margner, 2014] claimed to achieve an accuracy of 23.03% at word level and 69.48% at line level for writer identification. They used sliding window approach to extract graphemes and avoided manual segmentation. A GMM is built for each writer of the AHTID dataset with 53 writers. Their work also states the comparison of using GMMs instead of HMMs for writer identification and also states that writer identification by a single word is more complex than by a single text line.

8.2.3 Our Contribution

Much of the work on handwritten word recognition has focused on a set of few words written by multiple writers. In this work we develop a framework that does not need to be trained on specific words. The handwritten words used for training need not be the same as the handwritten words used for testing. Our system is applicable to documents where multiple authors have annotated the same page. Instead of working with codewords, we work with the clusters of features. Writer-specific classifiers are then trained on each cluster. In this work, we examine grapheme and character level using sliding windows, and obtain a feature set on them. We also present an analysis of feature clustering applied to graphemes and characters for word level writer identification.

8.3 METHODOLOGY

Features are extracted from segments or windows defined on the word image. The features are then clustered using *k*-means clustering. We envisage that allographs that are similar will get grouped into the same cluster. The set of allographs that belong to a cluster may be coming from samples by several writers. For each cluster we train a suite of one-vs-rest SVM classifiers using samples that belong to that cluster. The feature vector extracted for each word segment (window) is associated with the closest cluster and is classified by the suite of SVMs trained for that cluster. A majority voting among the classification decisions for all the windows gives the final writer label assigned to the word.

8.3.1 Feature Extraction

We extract features from each segment of the word. A segment could either encompass a character or a grapheme. When we consider segments to be characters we do a manual segmentation of words into characters. For other allographs we use a sliding window, which encompasses a portion of the word – that could be a partial or complete character or even portions of multiple characters, depending on the position and width of the window. Allographic features extracted from a segment are considered to be at character level or grapheme level, depending on how wide is the segment. In this work we adopt features from the work reported in [Slimane and Margner, 2014].

1. Pixel Density: It comprises total foreground pixels of the given image.

2. Center deviation from lower baseline: It is the vertical position of the gravity center in the whole character or glyph (W) with respect to the lower baseline normalized by the height h of the word. This feature is described by the following equation: $f = \frac{G_y - L}{h}$ here G_y is the vertical gravity center of the character or glyph, L is the lower baseline inside the concerned window of the word. The baseline here is detected by the method described in Chapter 7 in this thesis.

3. Zernike moments: For each image Zernike moment [Khotanzad and Hong, 1990] is calculated for an order five with no negative repetitions. A vector of 12 moments is generated whose mean will be the final feature value.

4. Mean of vertical projection normalized by the image width.

5. Mean of horizontal projection of the image.

6. Standard deviation of vertical projection normalized image width.

7. Standard deviation of horizontal projection of the image.

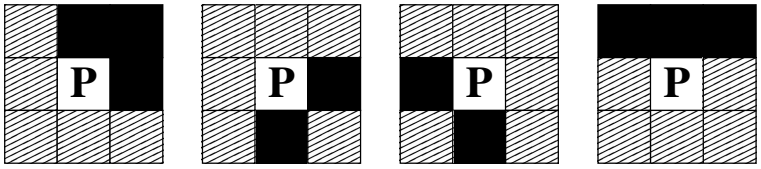

Figure 8.2 : The four topological masks with P as the central pixel, the dark region as the foreground pixel and the filled pattern as *do not care* care region.

8. Mean of Derivate of vertical projection vector profile of the image.
9. Mean of Derivate of horizontal projection vector profile of the image.
10. Mean of vertical runs of the image.
11. Mean of horizontal runs of the image.
12. Standard deviation of vertical runs image.
13. Standard deviation of horizontal runs of the image.
14. Typological Masks Matching: Handwriting can be quite similar in structure. However it varies over formation of structures like loops in characters 'l','e','h' and curves in characters 'B','P','D'. Therefore, it acts as a good feature to capture the handwriting style for different writers. There are four types of typological masks used in our system. Each of these masks is matched against the image. Each mask surrounds the central pixel with foreground pixels triggered as ON at some place and *don't care* for the rest of the places. This is illustrated in Figure 8.2. For each match the corresponding mask counter is incremented and finally, the mean of the feature vector is used as a feature.

8.3.2 Classification of segment features

When using sliding windows we consider the two settings of overlapping or non-overlapping windows to identify the handwritten word segments. We use k-means to group the features to form clusters. Each cluster will have segments that may be coming from different writers. That is, similar allographs of various writers can be clustered together. The training phase involves analyzing each cluster. If in a given cluster there are allographs of n different writers then we train a set of n one-vs-rest SVM classifiers for that cluster. These n writers are said to be owning that cluster. Thus we learn a suite of classifiers: a set of one-vs-rest classifiers for each cluster. The decision making can be explained as follows. Consider that the features extracted for a given allographic segment α are closest to the cluster C_1in terms of the Euclidean distance. Say the cluster C_1 has samples from k writers. On the execution of all one vs. rest SVM, a set of scores $W_{\alpha c1} = \{w_1, w_4, ..., w_k\}$ are computed which are the SVM decision scores for the writer classes or the ``rest" class (unknown writer). From this set the writer with the highest decision score is selected as the label for the allograph $\alpha$. There can be a possibility that an ``unknown writer" is assigned to an allograph. This will be the case when all the one-vs-rest SVMs classify the allograph to the unknown (rest) class. This happens when the allogprah does not carry enough discriminative features to enable it to get classified to one of the writers owning that cluster.

After the assignment of writer labels to the allographs of segments in a handwritten word, the writer of the complete word is identified using the majority label. If the majority label is the unknown writer, the system refuses to classify the word.

Window Width	20	25	30
Code Book Size	Accuracies		
100	**66.40**	60.28	62.94
200	64.63	60.20	**63.30**
300	64.18	60.55	61.70
400	63.74	**61.44**	62.59
500	62.85	61.26	62.32
600	63.92	60.99	63.39
700	64.10	60.37	63.39
800	63.92	59.75	63.30
900	63.83	59.04	60.99
1000	63.56	60.20	62.15

Table 8.1 : Writer identification Rate for an overlapping window at grapheme level

Window Width	20	25	30
Code Book Size	Accuracies		
100	18.17	16.13	12.21
200	20.48	16.84	12.87
300	21.90	19.06	13.84
400	21.81	19.59	14.90
500	24.20	18.79	14.98
600	20.74	21.63	17.83
700	23.23	21.90	17.43
800	**24.47**	21.81	18.40
900	23.58	20.48	**19.54**
1000	22.52	**22.16**	17.59

Table 8.2 : Writer identification rate for non-overlapping window at grapheme level

Window Width	20	25	30
Code Book Size	Accuracies		
100	17.43	15.96	16.37
200	13.03	16.37	18.00
300	**18.81**	17.67	19.22
400	15.72	18.00	19.71
500	14.74	**19.06**	21.66
600	14.41	17.02	18.00
700	14.33	17.59	19.63
800	17.35	18.89	19.95
900	16.53	16.12	19.54
1000	18.49	18.40	**21.82**

Table 8.3 : Writer Identification Rate for an overlapping window at Character Level

Window Width	20	25	30
Code Book Size	Accuracies		
100	3.66	4.40	3.42
200	7.49	8.31	5.94
300	5.70	8.55	4.89
400	10.10	10.34	8.88
500	10.34	9.93	8.55
600	11.89	11.07	10.50
700	11.56	11.97	9.85
800	13.52	**14.33**	12.54
900	**14.09**	13.27	11.48
1000	13.84	13.36	**12.62**

Table 8.4 : Writer Identification Rate for non-overlapping window at Character Level

8.4 EXPERIMENTS

In order to assess the effectiveness of the proposed approach, we performed a series of experiment on the CVL dataset [Kleber *et al.*, 2013]. In this work, we experiment with 10 writers with four documents from each of them.

8.4.1 Experiments at Grapheme Level:

Graphemes are extracted by means of a sliding window. Hence experiments are carried for both overlapping and non overlapping windows. Table 8.1 presents the accuracies obtained by sliding overlapping windows over a word. We varied the size of the codebook (number of clusters) from 100 to 1000. Table 8.2 provides the results obtained by sliding non-overlapping windows.

8.4.2 Experiments at Character Level:

From manually segmented characters, the similar features are clustered together for training. The testing is carried for both overlapping and non overlapping sliding windows features. Table 8.3 presents the accuracies obtained by sliding overlapping windows over a word. We varied the size of the codebook (number of clusters) from 100 to 1000. Table 8.4 provides the results obtained by sliding non-overlapping windows.

Window Width	20	25	30
Code Book Size	Accuracies		
100	15.31	13.19	**24.02**
200	14.58	**20.28**	17.43
300	10.67	13.27	23.86
400	16.04	14.82	14.41
500	15.15	14.98	19.46
600	17.26	18.16	20.11
700	**19.71**	13.84	16.94
800	10.67	14.25	22.56
900	13.60	19.63	13.27
1000	12.79	10.59	12.87

Table 8.5 : Writer Identification rate for an overlapping window based on [Slimane and Margner, 2014]

Window Width	20	25	30
Code Book Size	Accuracies		
100	8.79	11.56	3.42
200	11.64	10.83	5.94
300	18.00	13.11	4.89
400	10.42	18.97	8.88
500	16.45	22.72	8.55
600	18.24	**24.10**	10.50
700	13.68	20.68	9.85
800	**20.03**	18.49	12.54
900	18.08	18.81	11.48
1000	14.50	18.73	**12.62**

Table 8.6 : Writer Identification rate for non-overlapping window based on [Slimane and Margner, 2014]

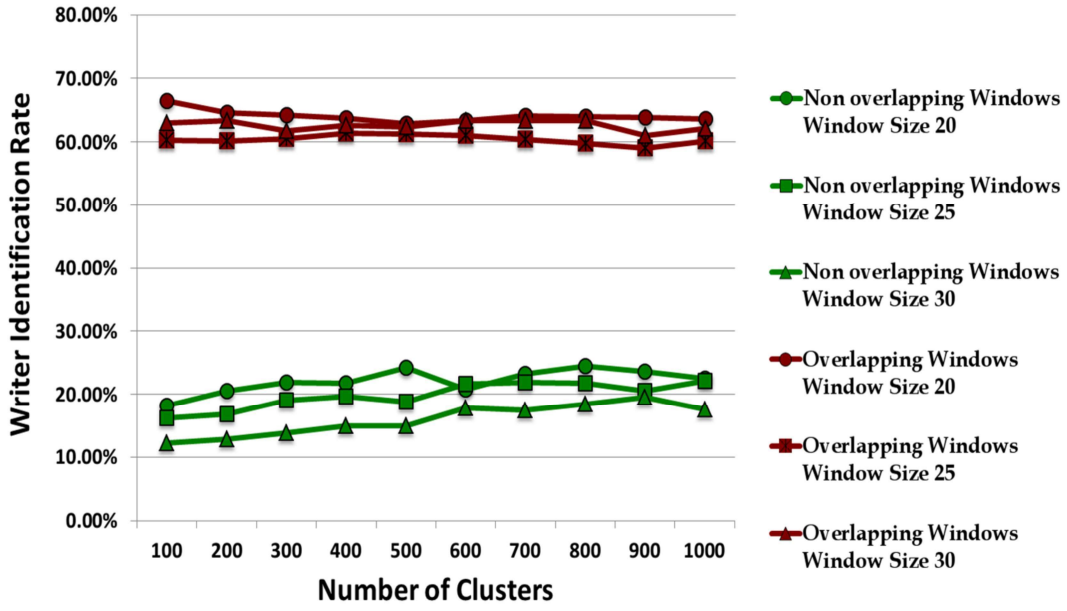

Figure 8.3 : Results for grapheme based features with non overlapping and overlapping window.

It is apparent from Tables 8.1, 8.2, 8.3, and 8.4 that the overlapping window setting outperforms the non-overlapping approach for both the character and grapheme based feature clustering. Possibly the non-overlapping windows result in clusters that are compact and hence are unable to separate the writers properly in the feature space. The results show that the optimal size of the window for achieving the best classification accuracy is 20. From Figure 8.3 and 8.4 it is clear that there is a major improvement in accuracy when using overlapping windows. It is because of the context that adds due to overlapping windows which makes the resulting clusters well populated and non-compact, thereby leading to better results.

Together these results provide an important insight into the fact that very small or very large window sizes lead to deterioration of classification results. The classification accuracy is observed to be somewhat invariant to the number of clusters used for grapheme and character level allographic features. In the overlapping setting more windows are generated which increases the

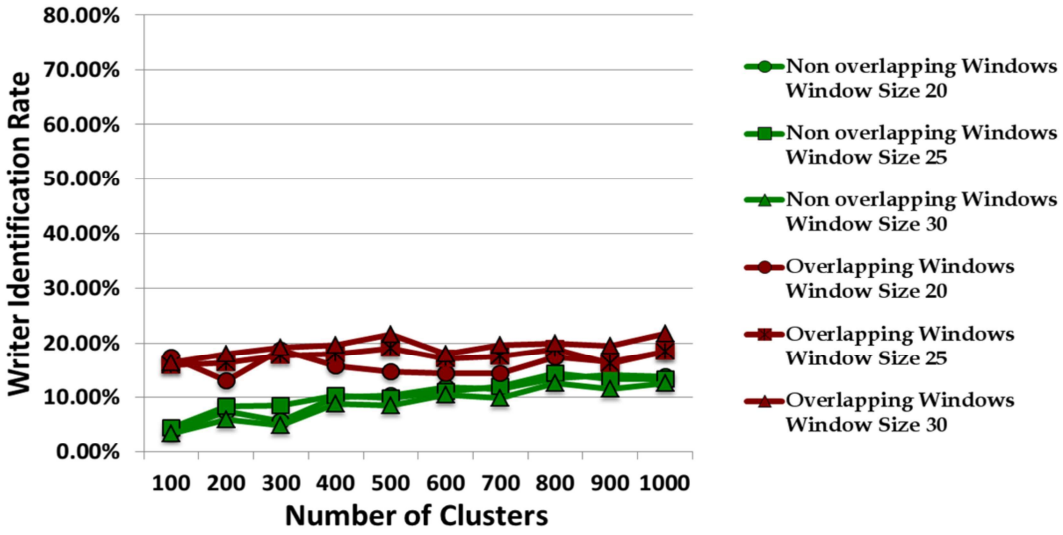

Figure 8.4 : Results for Character based features with non overlapping and overlapping window.

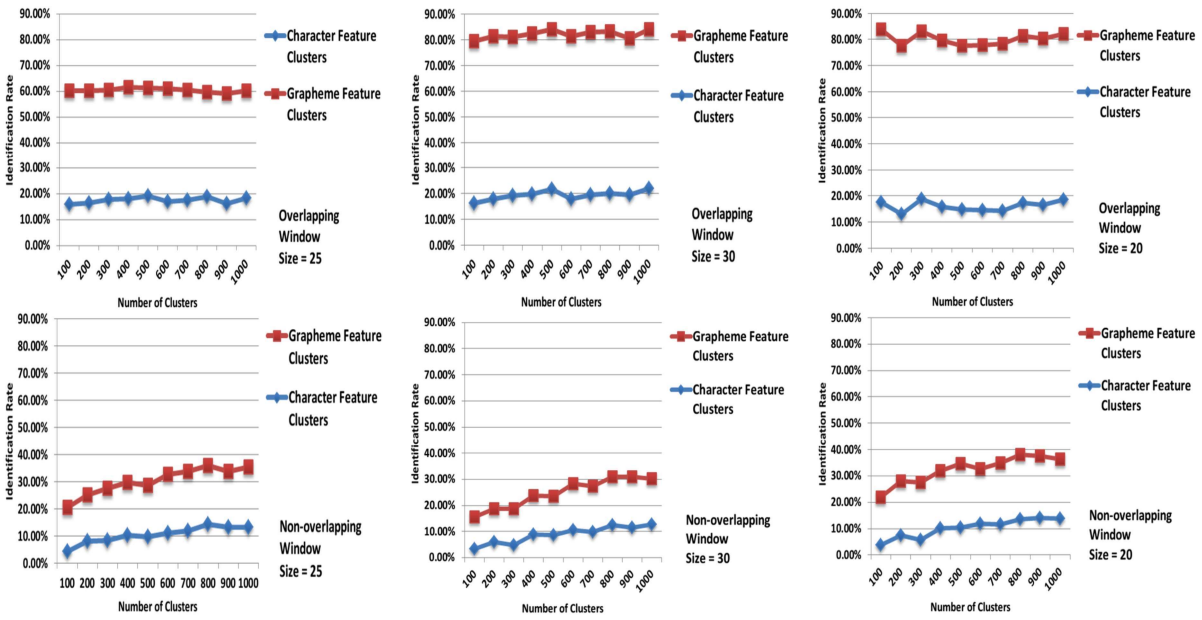

Figure 8.5 : Performance comparison between grapheme and character feature clusters.

allographs extracted from the sample words. It is also found that the grapheme level outperforms the character level analysis irrespective of window size (see Figure 8.5).

Our methodology (using graphemes) which is a discriminative approach presents an improvement in identification accuracy over the generative approach described in [Slimane and Margner, 2014] that uses window based features and models a GMM for each writer. Table 8.1 and Table 8.5 illustrates an increase in the identification accuracy from 19% to 63% at grapheme level for overlapping windows. Table 8.2 and Table 8.6 illustrates an increase in the identification accuracy from 18% to 22% at grapheme level for non-overlapping windows. It is observed that discriminative approach increases the offline writer identification rate at word level rather than

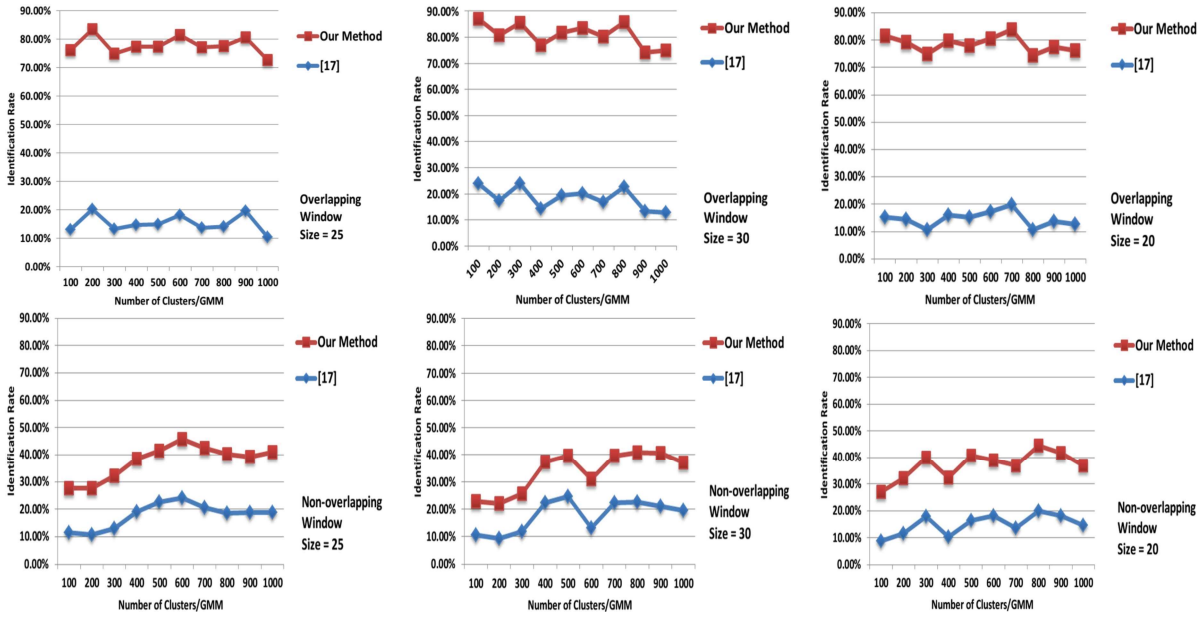

Figure 8.6 : Identification rate improvement between [Slimane and Margner, 2014] and our method using graphemes.

generative approach, irrespective of window size (see Figure 8.6).

8.5 CONCLUSIONS

In this chapter we described a method which exploited the allographic variations in the writing style at word level for writer identification. Allographic features at grapheme level exhibit discriminative properties when using overlapping windows. Our framework naturally allowed to assign a test word to the unknown writer class in case the allographs in the word did not exhibit clear resemblance to any particular writer in the test set.

...