Clustering writing components from medieval manuscripts

Mats Dahllöf
Department of Linguistics and Philology
Uppsala University
Sweden
mats.dahllof@lingfil.uu.se

Abstract

This article explores a minimally supervised method for extracting components, mostly letters, from historical manuscripts, and clustering them into classes capturing linguistic equivalence. The clustering uses the DBSCAN algorithm and an additional classification step. This pipeline gives us cheap, but partial, manuscript transcription in combination with human annotation. Experiments with different parameter settings suggest that a system like this should be tuned separately for different categories, rather than rely on one-pass application of algorithms partitioning the same components into non-overlapping clusters. The method could also be used to extract features for manuscript classification, e.g. dating and scribe attribution, as well as to extract data for further palaeographic analysis.

1 Introduction

1.1 Purpose

The present work explores a minimally supervised method for extraction and clustering of writing components from historical manuscripts. The primary purpose is to locate letters and to group them into classes capturing graphemic equivalence. The method will to some extent also find ligatures, scribal abbreviations, parts of letters, and letter sequences.

1.2 Motivation and applications

Clustering of writing components can be used as a first step in the transcription of manuscripts. By annotating clusters which group elements correctly – and manually correcting those that almost do – we will get cheaply acquired transcriptions, admittedly incomplete, associated with regions of the images. Components labelled in this way can be used as data for training of systems for handwritten text recognition. Another possible application is to use these data to compare different manuscripts, for instance, in the kind of digital palaeography proposed by Ciula (2005), or in letter-based scribe attribution in the style of Dahllöf (2014). Clustering can also be used as a tool for presenting manuscript data for qualitative palaeographic analysis.

1.3 Related work

The dominant approach in the field of handwritten text recognition is building systems based on supervised machine learning. This kind of work relies on labelled data, i.e. transcriptions. Less attention has been paid to methods for automatic analysis of handwriting which do not rely on transcriptions. When no linguistic labelling is available, clustering is a way of finding structure in writing. Rath and Manmatha (2007) use clustering of words from historical manuscripts (18th century) as a means for obtaining labelled data for word spotting. Vuurpijl and Schomaker (1997) use clustering to find allography in on-line handwriting data. Another application in the field of digital palaeography is proposed by Stutzmann (2016), who is interested in the use of clustering for the categorization of medieval script types.

2 Design and experimental procedure

The two main components of the current system are responsible for component extraction and clustering, respectively. As these components have to be based on certain a priori assumptions, they can be designed in a variety of ways, and their operation has to be guided by a number of parameters. We will present and evaluate a baseline system, which has proved useful as a point of departure during the development phase. Its parameter setting is to

a large extent comprehensible as reflecting properties of writing. After having looked at the baseline system, we will proceed to look at a few modified set-ups.

We will evaluate the clustering results using the measures *precision* and *recall* (see e.g. Manning et al. (2008)). In this context, these scores will be based on labels assigned to the clusters by a manual analysis. We will consider a cluster as capturing a category if at least 60% of its members belong to the category. So, the precision of a cluster is the ratio of elements belonging to the category associated with it. This also makes it possible to characterize a clustering outcome (the set of clusters) in terms of which categories it has managed to capture. Given a labelled cluster, we can also estimate its recall from the number of actual instances of the category in a sample of manuscript pages and the number of these which are included in the cluster.

3 Extraction of components

The first steps in the processing of the image files (JPEG, TIFF) are scaling and binarization by a version of the algorithm of Otsu (1979). The system ignores the outer margins of the images. To be more specific, images are cropped in such a way that, if l is the smallest value of the image width, w, and height, h, the further processing is concerned with the image in the centred rectangle of size $(w-0.05l)\times(h-0.05l)$.

In order to adapt the component extraction to the actual size and scale of the writing, the processing is guided by the typical stroke width, w_s (for the manuscript images being analysed). The system estimates w_s by determining the most common width of sequences of continuous horizontal foreground (ink) pixels separated by at least two pixels of background. After that, the system rescales the images to make sure all manuscripts are processed at roughly the same writing-relative resolution. In the experiments we discuss here, $w_s = 7$ pixels.

The component extraction process is guided by five parameters, $(t_i, w_{mn}, w_{mx}, h_{mn}, h_{mx})$. First, the system performs connected component labelling to find connected stretches of writing. Components whose width and height are in the intervals $[w_{mn}, w_{mx}]$ and $[h_{mn}, h_{mx}]$, respectively, are extracted, while those wider than w_{mx} are fed to a segmentation module. Loosely speaking, t_i is the thickest amount of ink that allows a vertical cut to be made.

To be more specific, the segmentation process

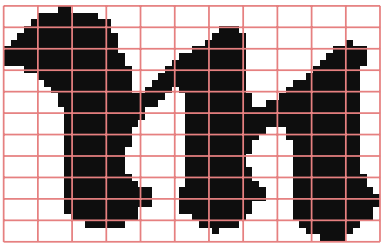

Figure 1: The grid of $11 \times 11 = 121$ rectangles corresponding to the features used to capture the distribution of foreground (component ink). So, the first five and the last five values for the example would be something like $(0.1, 0.8, 0.5, 0.1, 0.0, \ldots, 0.3, 0.0, 0.2, 0.9, 0.2)$ if we read top to bottom and left to right.

operates on the column sum of foreground (i.e. ink) $I(x)$, as computed with reference to the bounding box surrounding the component. It scans the component pixel by pixel, x_L being the current position. When $x_L = 0$ or $I(x_L) \leq t_i$, the system looks for a $x_R \in [x_L + w_{mn}, x_L + w_{mx}]$ where $I(x_R) \leq t_i$ and $I(x_R)$ is the smallest value. If it is not unique, the leftmost (smallest) x_R is preferred. A component spanning x_L to x_R is then proposed, and scanning resumes with $x_L = x_R$. If no x_R meets the condition, scanning resumes with $x_L = x_L + 1$. If the height of a proposed component spanning x_L to x_R is in the interval $[h_{mn}, h_{mx}]$, it is added to the set of extracted components.

In order to normalize the segmentation with respect to the size and scale of the writing, each of the segmentation parameters is defined as the product of a constant and w_s (the typical stroke width). We used $(t_i, w_{mn}, w_{mx}, h_{mn}, h_{mx}) = (1.0w_s, 3.0w_s, 8.0w_s, 3.0w_s, 15.0w_s)$ in the set-ups discussed here. These parameters represent a heuristic assumption about the appearance of the handwriting. We have tailored them to medieval book hands, aiming for a "wide-spectrum" retrieval of letter-size elements, but more or less excluding the letter **i** (bold case will be used for linguistic categories, rather than something like ⟨i⟩) and other "minim" components.

4 Clustering

Clustering is performed in three steps. The first one uses the density-based DBSCAN algorithm for obtaining a "core clustering". Secondly, small (< 40 elements here) clusters are removed. The third step is classification with the purpose of as-

signing additional not yet clustered components to the remaining (larger) core clusters.

4.1 Feature model and distance metric

Each manuscript component is characterized by a feature vector, which quantifies the distribution of foreground pixels as captured by a grid of 11×11 equal subrectangles over the bounding box. This consequently gives us 121 features, as illustrated by Figure 1. Each value is the ratio of the number of foreground pixels to the size of the subrectangle region (i.e. $f_i \in [0.0, 1.0]$).

The clustering and classification relies on Euclidean distance operating on these vectors: $distance(f,g) = \sqrt{\sum_{i=1}^{n} (f_i - g_i)^2}$.

4.2 Core clustering

The system uses the density-based DBSCAN algorithm (Ester et al., 1996) to obtain a "core clustering". DBSCAN was proposed for applications, like the present one, where a fair amount of noise data points are present. The clustering process is guided by two parameters: *Eps* (epsilon) and *minPts*. *Eps* is the largest distance between two points which are to be counted as neighbours. A smaller *Eps* makes the algorithm more reluctant to cluster data points by requiring a higher degree of similarity for clustering to take place. The *minPts* parameter is the minimal number of neighbouring points required for the formation of a same-cluster dense region. ($minPts = 11$ for the baseline set-up discussed below.) We can understand *minPts* as a constraint on the amount of evidence required for the stipulation of a cluster. Clusters smaller than *minPts* may however be proposed, if another cluster "steals" data points from a previously established cluster in the clustering process (which is sensitive to the order in which data points are visited). The DBSCAN algorithm typically leaves a subset of the data points (referred to as "noise") outside of the clusters, viz those points which are not part of the neighbourhood of any other point as defined by *Eps* and *minPts*.

The absolute distance values, as defined by the feature model and the distance metric (and thereby *Eps*), are hard to work with in a direct and intuitive way. For this reason, we suggest a data-oriented criterion, sensitive to the properties of the set of components to be clustered, for selecting the *Eps* value. Thus, we estimate *Eps* from a given threshold, p_{Eps}, in such a way that p_{Eps} is the probability that two randomly selected image components be at most *Eps* distant from each other. The intuition is that this probability should correspond to that of two random components being evident instances of the same graphemic type. In the baseline set-up, we estimated *Eps* from $p_{Eps} = 0.0007$. If we compare this number with English electronic text, we can note that the corresponding character-related probability is around 0.07, i.e. roughly 100 times higher than the number we assume here.

After the DBSCAN step, which typically gives us clusters of highly varying sizes, we remove the clusters which we think are too small (size < 40 in the experiments here) to merit attention. This will make the output easier to inspect and use.

4.3 Extending clusters by classification

We add more elements to the clusters formed by the core clustering, using a "nearest neighbour" classification procedure on the DBSCAN noise and the components from the removed small clusters. If we have retrieved n core clusters, with sets of elements $E_1, \ldots, E_n$, respectively, the system computes the centroids, $c_1, \ldots, c_n$, for each cluster. It also determines the distance d_i, such that a certain fraction, f, of the elements of E_i is within that distance of the centroid. So, d_i is the smallest distance such that $\left|\{e \in E_i \,|\, distance(e,c_i) < d_i\}\right| > f * \left|E_i\right|$. In the experiments reported here, $f = 0.9$. This will, loosely speaking, exclude peripheral components.

The classification procedure assigns each component, e (which does not yet belong to a cluster), to cluster i if and only if c_i is the nearest centroid among $c_1, \cdots, c_n$, and $distance(e,c_i) < d_i$.

4.4 Implementation and output

The system exists in the form of a Java implementation. For the purpose of the experiments reported here, it presents the set of clusters by means of images and HTML files, as in Figure 2. The members of each cluster are exhibited in a separate document as in Figure 3, page by page and ordered by their closeness to the centroid.

5 Experiments

In the experiments we conducted in order to evaluate the system, we used eight 7th–15th century book manuscripts as data. We tried a number of different clustering parameter settings. Complete outputs for all combinations of data and set-ups discussed here are available as Supplementary Materials at `http:`

//stp.lingfil.uu.se/~matsd/ch2018/.

5.1 Data

In our experiments, we applied the system to eight different manuscripts, representing four different important medieval periods and styles, each in two clearly different instances, spanning roughly seven centuries, see Table 1. The styles are Irish and Carolingian minuscule, textualis, and cursiva, The first four manuscripts are in Latin; the rest in Old Swedish. The pages of B 59 (the oldest book in Swedish) are worn and stained. B 10 suffers from bleed-through. Otherwise, the data are from fairly well-preserved books. The digitizations are of high quality – we used the highest resolutions available – in JPEG or TIFF format. Each file is between 2MB and 90MB in size. The terms of use allow the images to be used freely for research purposes. (The URLs for images and metadata are found in Table 1).

5.2 Baseline set-up

We applied the system with the baseline parameter settings (specified above) on the eight manuscripts. From each sequence of pages, the system extracted exactly 20 000 components. Between 25 and 44 images had to be read. The clustering assigned between 5000 and 14 800 of them to clusters. We manually inspected the clusters and performed an analysis of which linguistic categories they represent. We also estimated the precision scores. In the case of letter categories, we accepted also a few similar majuscule instances of the same letter, e.g. **O** among **o**, as true positives. Table 2 summarizes the results.

Clusters for between 5 and 15 letters (not counting allographs) were established. Letters like **a**, **d**, **m**, and **q** (**q** only for Latin) had a strong tendency to appear. As expected, no **i** cluster was found. In a few cases, two different clusters for the same letter had been established. The number of clusters for ligatures and bigrams (ordinary two-letter sequences) were strikingly higher for the textualis and cursiva manuscripts, reflecting the fact that connected letters are typical of these styles.

We can distinguish two kinds of outcome: For some manuscripts, components were only assigned to meaningful clusters (Gen. 1, CS 557, CS 564, C 61(b)). In the other cases, also large "useless" clusters were established (CS 60, B 59, B 10, C 61(a)). This suggests that more generous clustering settings could be worth exploring in the former cases, while the latter situation invites using more reluctant DBSCAN parameter values.

5.3 More generous settings

We tried to cluster the manuscripts Gen. 1, CS 557, CS 564, and C 61(b), whose components were potentially "underclustered" by the baseline set-up, with $p_{Eps} = 0.0014$ (i.e. doubled), otherwise using the baseline settings. Table 3 summarizes the results. A general tendency is, as can be expected, that the clusters become larger and less pure. The change also gives us new useful clusters, e.g. **h** for Gen. 1 and **b**, **h**, **d̄** (ligature d with macron), &, and **𝒫** (*pro* abbreviation) for CS 564. The CS 557 output covers 18 minuscule letter categories. However, this setting also to some extent leads to merging of categories discerned in the baseline set-up. So, for instance, in the Gen. 1 output, **f** joins the **p**∪**s**$_2$ blend, and instances of **n** and **r** – just like instances of **a** and **o** in C 61(b) – get mixed up in the same cluster.

5.4 More restrictive settings

We also tried two ways of making the clustering of components more restrictive. The first one was tightening the DBSCAN neighbourhoods by halving the probability p_{Eps} (i.e. $p_{Eps} = 0.00035$) from which *Eps* is estimated. Secondly, we doubled the neighbourhood size requirement, i.e. used $minPts = 22$. Applying the two settings to two of the "overclustered" manuscripts produced the outputs characterized in Table 3. Figure 4 aligns two cluster set overviews for B 59, one for the baseline set-up and one for $p_{Eps} = 0.00035$.

We can see that the more restrictive settings, as expected, make the clusters fewer and smaller across the board. For CS 60 the outcome is otherwise similar to that for the baseline set-up. A new cluster is formed by the detachment of a **p**∪**r** category. For B 59 the response to the more restrictive settings is more pronounced and fruitful. Fairly pure clusters appear for three or four additional letters, **d** (only with the lower p_{Eps}), **f**, **k**, and **þ** (thorn). Some letter sequence categories disappear, but we also see a few new ones. For both manuscripts, the two set-ups ($p_{Eps} = 0.00035$ and $minPts = 22$) have fairly similar consequences.

5.5 Recall and the classification step

Tables 2–4 give an idea about how the recall scores for different categories compare to each other. (The recall is 0 for categories for which the system does

Table 1: The eight medieval manuscript page sequences used as data. CS: St. Gallen, Stiftsbibliothek. UUB: Uppsala University Library. The UUB images cover spreads.

Abbr.	Source citation, URL	Script, date
Gen. 1	Schaffhausen Stadtbibliothek, pp. 6 ff. http://dx.doi.org/10.5076/e-codices-sbs-0001	"Irische Halbunziale" 7th/8th century
CS 60	Cod. Sang. 60, pp. 6 ff. http://dx.doi.org/10.5076/e-codices-csg-0060	"irischer Schrift" 8th century
CS 557	Cod. Sang. 557, pp. 13 ff. http://dx.doi.org/10.5076/e-codices-csg-0557	"Qualifizierte St. Galler Carolina" late 9th century
CS 564	Cod. Sang. 564, pp. 16 ff. http://dx.doi.org/10.5076/e-codices-csg-0564	"Grosse, sorgfältige Spätcarolina" late 12th century
B 59	National Library of Sweden, pp. 3 recto ff. (image 9 ff.) https://data.kb.se/datasets/2015/01/fornsvenska/B_59_002611384	Textualis late 13th century
B 10	UUB, pp. 24 verso ff. http://urn.kb.se/resolve?urn=urn:nbn:se:alvin:portal:record-90664	Textualis 1350–1399
C 61(a)	UUB, C 61, pp. 138 ff. (spread image 74 ff.) http://urn.kb.se/resolve?urn=urn:nbn:se:alvin:portal:record-55762	Cursiva recentior late 15th century
C 61(b)	The same codex, pp. 540 ff. (spread image 275 ff.) http://urn.kb.se/resolve?urn=urn:nbn:se:alvin:portal:record-55762	Cursiva recentior (a different hand) late 15th century.

not establish a corresponding cluster.) Letters like **a**, **d**, **m**, and **q** seem to be "easy" ones, while the more infrequent letters are more difficult to retrieve or have been filtered out by the size constraint. The component extraction set-up is probably more or less responsible for the scarcity of **i**, **f**, and **l** clusters.

We will not provide a fully-fledged analysis of absolute recall, but we have estimated recall for a few cases and categories. For this purpose, we annotated the three first pages (or four, i.e. two spreads for C 61(b)) of the sequences for Gen. 1, CS 557, CS 564, and C 61(b) as regards three letter categories, **e**, **m**, and **o**. We counted black-ink minuscule non-ligature letter instances in the main text columns as the relevant ones, excluding e.g. majuscule and red ink writing and later additions. (The ligature versus non-ligature distinction was sometimes difficult to apply, e.g. for Gen. 1.) We manually counted the true positives using the overviews of the clustered components (arranged page by page, as in Figure 3). Table 5 gives the precision and recall scores for the two set-ups described earlier for the manuscripts. Scores for the baseline core clustering (i.e. excluding the contribution by the classification step) are also reported. We see that the absolute recall scores vary considerably. The low values for **e** and CS 564 pertain to the "clean-cut" **e** category; many **e**:s are assigned to sequence categories. The **m** and **o** categories seem to reach high recall scores in several cases. Instances of these are easy to extract because they are internally well connected and isolated from other letters. We may also expect the feature model to "agree" with the shapes of **m** and **o** in well capturing the differences between them and other letters.

The $p_{Eps} = 0.0014$ set-up leads to a clear increase in recall in most cases, without loss in precision. For **m** and **o** in CS 564, and C 61(b) precision is somewhat compromised. For C 61(b) it even leads to the merging of the **a** and **o** clusters (separated in the baseline output). The **o**-precision for the **a**∪**o** cluster is just 29%.

The classification step is responsible for 13–35% (comparing manuscript outputs) of the components being assigned to clusters in the baseline set-up. (These components carry a tag indicating their classification-based assignment.) In the small-scale study of Table 5, classification generally leads to a pronounced improvement in recall compared to the core DBSCAN output of the baseline system. In only two cases, a fall in precision can be ob-

Table 2: Outcome of the baseline experiments. Manuscripts, with c: the number of clusters established, n: the number of components assigned to some cluster (k: in thousands) (the rest is "noise"), and p: the number of images that the component extraction module read. Clusters for letters, ligatures and bigrams, and mixtures (∪: union). Three dots "···" indicate preceding or succeeding material in classes of components with one "dominating" letter. The precision scores are $> 99.5\%$, unless the cluster size is marked by a symbol indicating a lower level of precision, as follows: ⋆: $> 98\%$, †: $> 80\%$, or ‡: $> 60\%$. Plus signs in the size numbers indicate that several clusters were established. Allographs: $\mathbf{d}_1$: ordinary, $\mathbf{d}_2$: uncial. Compare Figure 2.

Manuscript	Letters	Ligatures, bigrams	Mixtures, etc.
Gen. 1 c=20, n=8.9k p=25	**a**:1171, **b**:306, **c**:406, $\mathbf{d}_1$:251, $\mathbf{d}_2$:289, **e**:273, **f**:112, **l**:271, **m**:460, **n**:667, **o**:675, **q**:204+65, **r**:755, $\mathbf{s}_1$:101, **t**:252, **u**:827 [s_1: like modern s]	**en**:54, **&**:90	$\mathbf{p}\cup\mathbf{s}_2$:1715 [$s_2$: tall r-like allograph]
CS 60 c=16, n=14.6k p=27	**a**:1138†, **c**:601⋆, **d**:593⋆, $\mathbf{e}_1$:317, $\mathbf{e}_2$:181, **m**:734⋆, **o**:851, **q**:371, **s**:755, **t**:361 [e_2: part of ligatures]	**er**:62†, **&**:248	**b**∪**h**∪**li**:718⋆, **n**∪**u**:4300‡ **c**∪**e arc**:52, *useless:* 3274
CS 557 c=17, n=5.0k p=44	**b**:241+93, $\mathbf{d}_1$:273, **g**:64, **h**:84, **l**:85, **m**:585, **n**:144, **o**:676, **p**:597, **q**:195, **r**:409+86, **s**:1141, **v**:191	**is**:104, **ss**:56	–
CS 564 c=17, n=6.7k p=42	**a**:389, **d**:563, **e**:115, **m**:998‡, **o**:248, **p**:487, **q**:157, **r**:195+45, **s**:1324, **u**:456⋆	**es**:104, **os**:45, **ss**:92, ···**s**:78, **it**:84	**n**∪**u**:1289‡
B 59 c=23, n=14.8k p=44	**a**:2938†, **m**:361†, **n**:1562‡, **o**:488, **s**:1220‡	**al**:57, **bo**:47, **fa**:105⋆, **fi**:157, **gh**:63, **gi**:94†, **ll**:71†, **sk**:256†, **sti**:53†	**a**···:367, ···**a**:290†, **e**···:110, **sk**∪**st**:52†, *useless:* 3846+2172+223+190+109
B 10 c=26, n=10.1k p=35	**a**:2218⋆, **ä**:178, **b**:216†, **d**:168, **e**:375⋆, **k**:96, **m**:487‡, **o**:933, **s**:951‡, **t**:197⋆, **þ**:314, **v**:487⋆, **w**:54, **y**:112 [**a**: some with lost diacritic]	**an**:130⋆, **fi**:66, **ff**:69, **ffi**:68, **gi**:109⋆	**a**···:116, ···**a**:53, **g**···:189, **t**···:83, *useless:* 2261+103+97
C 61(a) c=24, n=10.6k p=31	**a**:446, **d**:173, **g**:383, **h**:946⋆, **o**:116, **t**:443†	**dh**:69, **eß**:274, **ot**:123, **sk**:50⋆, **ß**:98, **th**:934⋆	···**a**:88, **d**···:498, ···**d**:223, **g**···:128, ···**g**:142+107, **h**···:576†, ···**l**:47, ···**p**:46, **t**···:297⋆, **macron**:863, *useless:* 3535
C 61(b) c=27, n=11.2k p=25	**a**:1594, **ä**:456, **d**:508, **e**:436, **g**:158+60, **k**:117, **m**:905†, **n**:1180†, **o**:511, **r**:148⋆, **s**:237‡, **v**:88, **w**:261, **y**:130	**en**:134, **ff**:63†, **fö**:114†, **hy**:94, **sk**:205, **ta**:52, **ti**:152 **tin**:41	···**h**···:2889+111†, **s**···:461†, **m/n-minim**:109

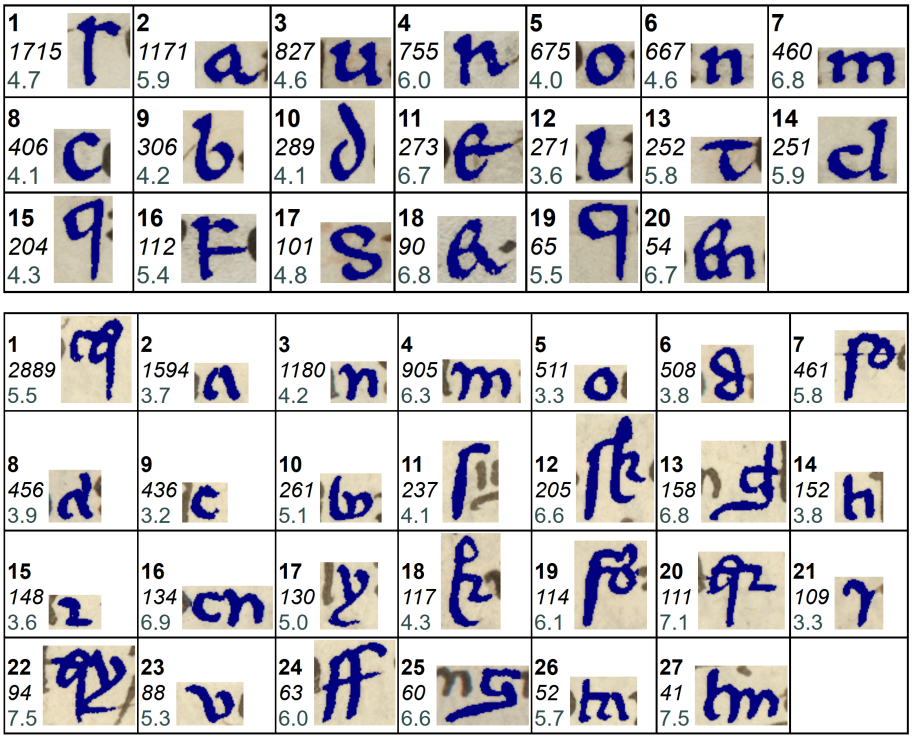

Figure 2: Cluster sets for Gen. 1 (top) and C 61(b) (bottom) from the baseline set-up. In cells: cluster number, size, average width – estimated stroke width (w_s) being the unit –, and the instance closest to the centroid. Component foreground (binarized) rendered in dark blue. Compare Table 2.

Table 3: Clusters produced with $p_{Eps} = 0.0014$, otherwise as baseline set-up, see Table 2 (also for explanations). New clusters (compared to baseline) are underlined.

Manuscript	**letters**	**ligatures, bigrams**	**mixtures, etc.**
Gen. 1 $p_{Eps}=0.0014$ c=23, n=13.2k	**a**:1460*, **b**:341+51, **c**:555, **d**$_1$:284, **d**$_2$:405, **e**:863*, **h**:113, **l**:349, **m**:630, **o**:994, **q**:303, **s**$_1$:213, **t**:615†, **u**:1231 (absent: **f**, **n**, **r**)	**en**:129†, **er**:63, **&**:128, **e**···:55, **li**:79	**n**∪**r**:2306*, **f**∪**p**∪**s**$_2$:1906*, **r-frag.**:156
CS 557 $p_{Eps}=0.0014$ c=27, n=9.5k	**a**:594, **b**:404, **c**:111, **d**$_1$:309, **d**$_2$:167, **e**:423, **E**:54, **g**:112, **h**:108, **l**:109*, **m**:960, **n**:685, **o**:841, **p**:628, **q**:245, **r**:609+138†, **s**:1149, **t**:64, **u**:969, **v**:229	**co**:104, **er**:184, **&**:104, **is**:113, **ri**:64, **ss**:66	–
CS 564 $p_{Eps}=0.0014$ c=28, n=10.9k	**a**:592, **b**:282*, **d**:660, **e**:202, **h**:121, **m**:1369‡, **o**:568, **p**:542*, **q**:219, **r**:276+79, **s**:1331 (absent: **u**)	**ɑ**:54, **er**:248, **&**:85, **it**:245*, **on**:82, **ꝑ**:61, **ri**:119*, **ss**:103	**d**···:64, **n**∪**u**:2992‡, **o**···:54, ···**s**:277, **st**···:55, *useless:* 151+77+54
C 61(b) $p_{Eps}=0.0014$ c=31, n=13.4k	**ä**:488, **d**:625, **e**:483, **g**:214+66, **k**:169, **m**:945†, **n**:1277†, **r**:213*, **s**:296‡, **v**:105, **w**:330*, **y**:163 (absent: **a**, **o**)	**en**:202†, **ff**:84†, **gh**:59†, **hy**:337†, **sk**:217, **ß**:137, **ti**:174†, **tin**:43	**a**∪**o**:2471*, ···**h**···:2860†, **k**···:69, **r**···:78, ···**p**:50, **s**···:788†, **t**···:81†, **th**···:90†, *useless:* 200+40

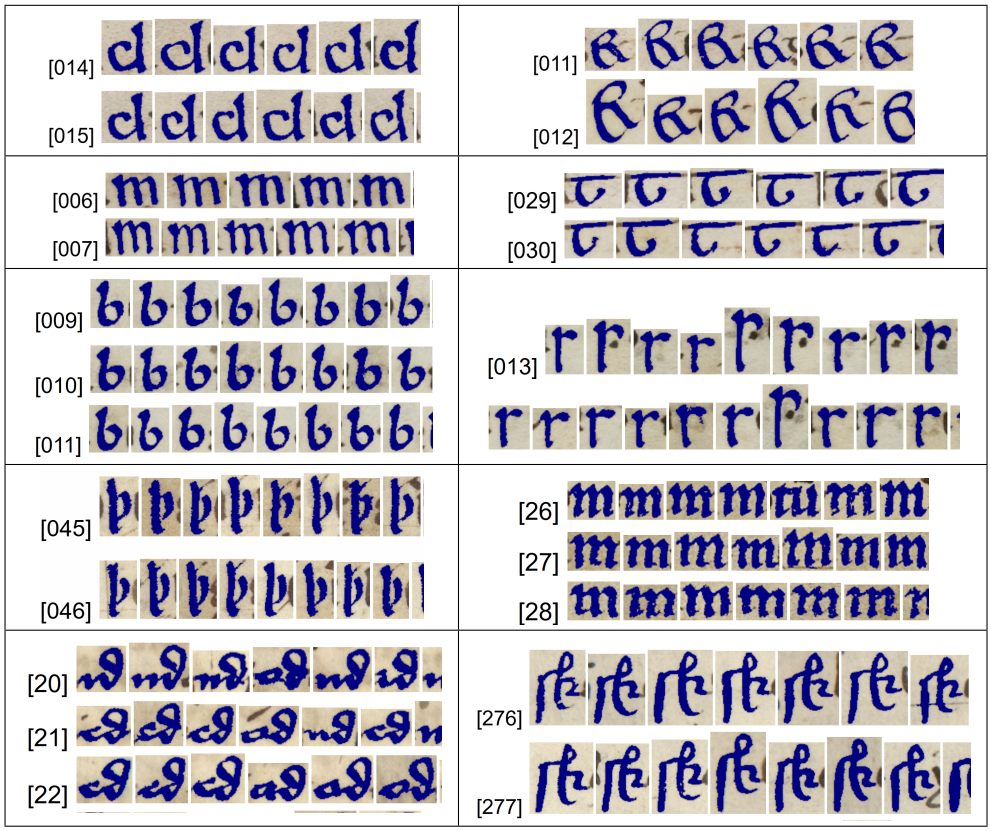

Figure 3: A few cluster instances (excerpts): **d**, **&** (⟨et⟩ ligature), **m**, and **t** from from CS 60, **b** and **p**∪**s**$_2$, both from Gen. 1, **þ** (thorn) from B 10, **m** from B 59 (two errors), ···**d** from C 61(a), and **sk** from C 61(b). Some component images are cut to the right.

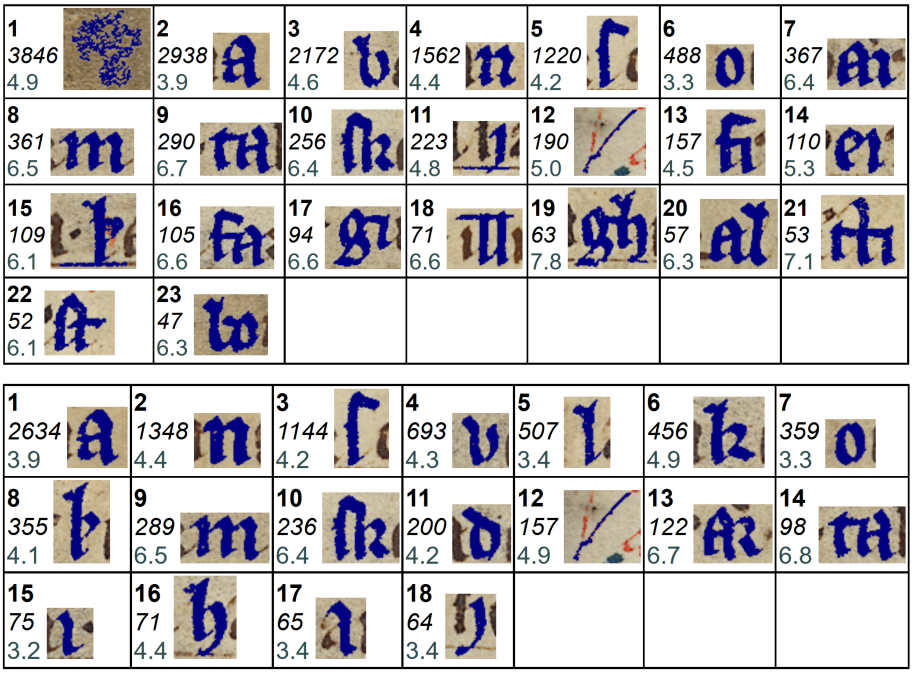

Figure 4: Cluster sets for B 59: Baseline set-up (top) (compare Table 2) and the more reluctant set-up with $p_{Eps} = 0.00035$ (bottom) (compare Table 4). Explanations at Figure 2.

Table 4: Clusters produced by more reluctant settings of p_{Eps} and *minPts*, otherwise as baseline set-up (see Table 2). New clusters (compared to baseline) are underlined.

Data & set-up	**letters**	**ligatures, bigrams**	**mixtures, etc.**
CS 60 $p_{Eps}=0.00035$ c=15, n=12.4k	**a**:985†, **c**:520, **d**:541, **e**$_1$:254, **e**$_2$:97, **m**:698*, **o**:726, **q**:314, **s**$_1$:178, **t**:318	**&**:203	**b**∪**h**∪**li**:635, **n**∪**u**:2894†, **p**∪**r**:806†, *useless:* 3202
CS 60 *minPts*=22 c=15, n=12.9k	**a**:1037†, **c**:549*, **d**:575*, **e**$_1$:271, **e**$_2$:100, **m**:719*, **o**:813, **q**:318, **s**$_1$:468, **t**:309	**&**:209	**b**∪**h**∪**li**:666, **n**∪**u**:2981†, **p**∪**r**:813†, *useless:* 3033
B 59 $p_{Eps}=0.00035$ c=18, n=8.9k	**a**:2634*, **d**:200, **h**:71, **k**:456*, **m**:289†, **n**:1348‡, **o**:359, **s**:1144‡, **þ**:355†	**ar**:122*, **sk**:236†, **ta**:98†	**v**∪**b**:693‡, *useless:* 507+157+75+65+64
B 59 *minPts*=22 c=16, n=6.6k	**a**:2852*, **h**:93, **k**:477*, **m**:335†, **n**:1433‡, **o**:455, **s**:1170‡, **þ**:386†,	**ar**:229‡, **fi**:63, **sk**:223†, **ta**:143†,	**v**∪**b**:943‡, *useless:* 599+155+66

Table 5: Precision (p.) and recall (r.), both in %, for the baseline and $p_{Eps}=0.0014$ set-ups, estimated for three categories (**e**, **m**, and **o**) and four manuscripts. Last cell (36%) corresponding to cluster **a**∪**o** in Table 3. We also report scores for the baseline set-up excluding the classification step (Core).

		Core			**Baseline**			$p_{Eps}=0.0014$		
Manuscript		**e**	**m**	**o**	**e**	**m**	**o**	**e**	**m**	**o**
Gen. 1	p.	100	100	100	100	100	100	100	100	100
	r.	10	46	54	14	54	58	46	68	83
CS 557	p.	–	100	100	–	100	100	100	100	100
	r.	0	13	46	0	44	61	8	71	76
CS 564	p.	–	97	100	100	65	100	100	57	100
	r.	0	46	1	4	58	17	7	61	36
C 61(b)	p.	100	100	100	100	97	100	100	96	29
	r.	28	58	25	28	63	28	31	65	36

served. The effect of the classification step varies even more for individual categories. Just to take two extreme examples from the full baseline output for CS 557 (see Table 2): for **m**, 322 of 585 elements were retrieved by classification, but only 1 of 1141 for **s**. (100% precision in both cases.)

6 Conclusions

In this study, we have shown that simple component extraction and clustering in combination with limited human intervention can be used to produce partial transcriptions of medieval manuscripts in a range of styles. This kind of pipeline will provide a low-cost method for initial annotation, which is potentially useful in many contexts of handwriting analysis and digital philology. However, the basic modules of the system are simple and invite improvement.

The component extraction module relies on very basic methods for binarization, connected component labelling, and component segmentation. We have not considered more sophisticated designs in this paper, but that is certainly one of the more obvious points for future work. Just modifying the segmentation parameters of the current implementation would give us other components to cluster and consequently very different outputs. The system works without making use of modules for layout analysis or line segmentation, other than what is achieved by the connected component labelling. This makes the system more robust with regard

to some kinds of manuscript, but using such modules would potentially be useful for guiding the component extraction in other cases.

The feature model seems to work quite well for the styles studied here, because letter distinctions generally correspond to marked contrasts in how ink is distributed in the bounding box. Admittedly, we have only studied fairly regular book styles. The **b**∪**h**∪**li** mixture produced for C 60 is an example where the model, so to speak, fails to separate the categories. The 11×11 "resolution" is reasonable for letters, but will blur larger components, e.g. letter sequences. The model does not capture absolute or relative size of the components. This is an advantage when there is linguistically insignificant size variation, but could have helped to separate **p** and **s** in Gen. 1, see Figure 3. This letter pair also illustrates another problem: A significant part of the **p**'s is a typically unconnected dot. Another conclusion that should be drawn is that a system like this could benefit from also looking at the contexts in which the components occur. The current system only "sees" the foreground components as framed by the bounding box.

When we looked at the results of the clustering we saw that settings that are fruitful for the retrieval of one letter category might lead to merging of other categories. And, contrariwise, a set-up that will to keep the two categories separated might prevent the system from establishing clusters for other categories. This suggests that a system like the present one should be tuned separately for different categories, rather than rely on a one-pass application of algorithms partitioning the components into non-overlapping clusters. This kind of approach would also benefit from the use of statistical criteria which would help us find good parameter settings automatically.

Acknowledgments

The author is grateful for the support of two projects, funded by the Swedish Research Council (Vetenskapsrådet, Dnr 2012-5743) and Riksbankens Jubileumsfond (Dnr NHS14-2068:1), and led by Anders Brun and Lasse Mårtensson, respectively. Thanks also to the participants of the Workshop on Computational Methods in the Humanities (COMHUM 2018) for discussion.

References

Ciula, Arianna (2005). Digital palaeography: using the digital representation of medieval script to support palaeographic analysis. *Digital Medievalist*, 1.

Dahllöf, Mats (2014). Scribe attribution for early medieval handwriting by means of letter extraction and classification and a voting procedure for larger pieces. In *2nd International Conference on Pattern Recognition (ICPR)*, pages 1910–1915.

Ester, Martin, Hans-Peter Kriegel, Jiirg Sander, and Xiaowei Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Otsu, Nobuyuki (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66.

Rath, Tony M. and R. Manmatha (2007). Word spotting for historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 9:139–152.

Stutzmann, Dominique (2016). Clustering of medieval scripts through computer image analysis: Towards an evaluation protocol. *Digital Medievalist*, 10.

Vuurpijl, Louis and Lambert Schomaker (1997). Finding structure in diversity: a hierarchical clustering method for the categorization of allographs in handwriting. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 387–393. IEEE.