



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *Workshop on Historical Document Imaging and Processing, 16-17 Sep 2011, Beijing, China*.

Citation for the original published paper:

Wahlberg, F., Dahllöf, M., Mårtensson, L., Brun, A. (2011)

Data Mining Medieval Documents by Word Spotting.

In: *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing* (pp. 75-82).

New York: ACM

<http://dx.doi.org/10.1145/2037342.2037355>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-162428>

Data Mining Medieval Documents by Word Spotting

Fredrik Wahlberg
Centre for Image Analysis
Uppsala University
fredrik.wahlberg@cb.uu.se

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

Lasse Mårtensson
Dept of Scandinavian Languages
Uppsala University
lasse.martensson@nordiska.uu.se

Anders Brun
Centre for Image Analysis
Swedish University of
Agricultural Sciences
anders.brun@cb.uu.se

ABSTRACT

This paper presents novel results for word spotting in Latin and Old Swedish medieval manuscripts. A word or grapheme of some kind is marked by a user, and the method automatically finds similar matches in the document. We present a method with improved accuracy for this kind of documents. The method automatically finds pages and lines. An advantage of the new method is that it finds the query within a text line without solving the difficult problem of segmenting a text line into individual words or graphemes. We evaluate the method on two medieval manuscripts and show how it can help a user navigate a text and present graphs of word statistics as a function of page number.

Categories and Subject Descriptors

I.4.6 [Image Processing and Computer Vision]: Segmentation; I.4.7 [Image Processing and Computer Vision]: Feature Measurement

General Terms

Keywords

OCR, handwriting, medieval, HTR

1. INTRODUCTION

This paper presents a study on applying word spotting based on dynamic time warping as a method for full document matching on two varieties of medieval handwriting. These experiments are motivated in the context of an interdisciplinary effort to establish a large-scale project under the heading “From quill to bytes – a Swedish infrastructure for automatic transcription of pre-modern handwritten texts”. This project sets out to develop automatic image and language analysis techniques for data mining of historical manuscripts. Our goal, in a long-term perspective, is to develop software for transcribing historical manuscripts, such

as religious works, administrative documents, financial accounts, and medical journals. This will necessitate cooperative work involving image analysis in the field of handwritten text recognition (HTR), humanistic research in the form of palaeographical and philological expertise, and language technology. Our mission is to establish a research infrastructure that will push the “digital horizon” back in time, by enabling digital interpretation and data mining of handwritten historical materials for both researchers and the public. This would revolutionize research on pre-modern history, making it possible to tackle questions about the past hitherto unanswerable.

1.1 Data

The present study is concerned with two Medieval manuscripts, both belonging to the Uppsala University Library. They are labelled C61 and C64, see [2] for a catalogue. We are concerned with the pages 539–1104 of C61, which hold books 5–8 in Old Swedish of the Revelations of Saint Bridget of Sweden. The script falls into the category *Cursiva Recentior*, using the criteria of [5]. The 230 folios, with text on each side, of C64 contain a handbook in Latin for priests, attributed to Laurentius of Vaksala, called *Summula de ministris et sacramentis ecclesiasticis*. The manuscript is dated to the 14th century, and is possibly from Uppsala. We classify the script as Northern Textualis, again following [5].

Medieval manuscripts, such as C64, are often illuminated in that textual elements and miniature illustrations are visually integrated, e.g. in decorated initials, borders, etc. Ruling, i.e. lines drawn to guide the scribe, may also be prominent. Textual elements in a stricter sense can be defined as those which are instances of graphemes, i.e. of linguistic sign types given by the linguistic code.

Among graphematic elements – see Figures 1 and 2 – we find letters, punctuation marks, and ligatures, i.e. letters joined into a single graphic form. There is a frequent and sophisticated use of abbreviations; Cappelli’s standard dictionary [4] lists thousands of types. Macron strokes, often indicating nasals, are part of the scribal abbreviation system. In Old Swedish manuscripts, such as C61, abbreviations are used to a much lesser degree than in the Latin ones. Neither spelling nor use of abbreviations follows strict rules in medieval writing; there is a high degree of variability due to

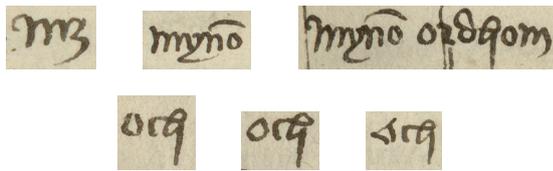


Figure 1: The words *medh* (with), *mynom* (my), and the phrase *mynom ordhom* (my words) and three tokens of *och* (and) (Manuscript C61). (Section 1.1.)



Figure 2: The words *debet*, *hac*, *enim*, *non* in abbreviated form; *orationem* in two versions; three instances of the $\chi\rho\iota$ abbreviation of *Christi* (genitive of *Christus*), with two Greek letters (Manuscript C64). (Section 1.1.)

scribers' preferences and ad hoc concerns.

2. PREVIOUS WORK

For image data where complete OCR is intractable, words can still be compared and clustered. This technique is called word spotting [12]. The basic idea is to select template words and let the machine find matches. The matched word occurrences can be used to refine the classification after human correction. The result is several clusters of matched words that can be used for research and indexing. This process can be quite hard to manage using traditional techniques for pixel-by-pixel template matching. Handwritten text is often highly irregular and pixel-based methods are slow.

To be able to do matching with irregular and degraded handwritten text, the concept of using dynamic time warping has been proposed [14]. The dynamic time warping approach treats the matching problem as the matching between one-dimensional signals. Using dynamic programming, the optimal match, with respect to some distance measure, is found. This concept has been shown to work in many different applications. In speech recognition the natural signal representation is one-dimensional, and this approach has been successfully used in [15] to match spoken words. It is also used for computed tomography volume registration of colons [13]. The colon is virtually sliced into sections perpendicular to its centre direction and thickness is used as a feature. This reduces the original three-dimensional problem to one dimension problem, allowing dynamic time warping to be applied.

The approaches of [14] and [7] have been extended by [9]. They use more information of word shape and size to cor-

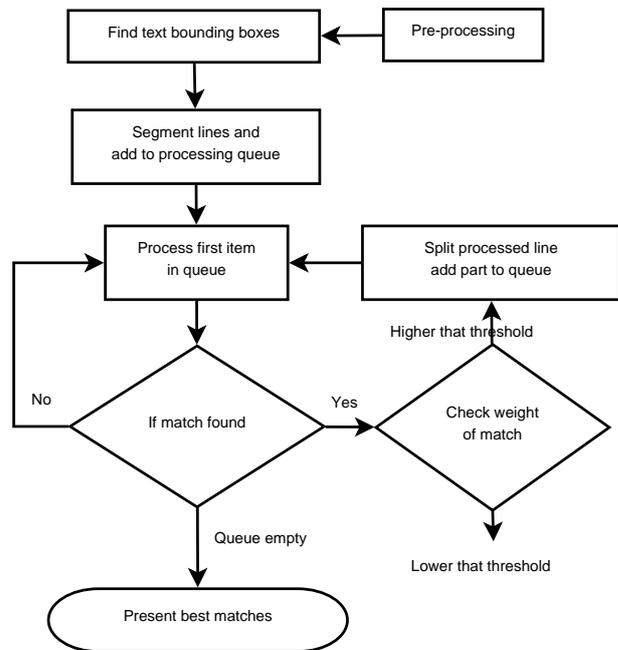


Figure 3: A flowchart showing complete process of the word spotting software. The steps are described in detail in Section 3.

rectly identify matches yielding higher accuracy. These studies use handwritten texts from the George Washington collection as source material. In Washington's letters, a good word segmentation is possible. The segmentation of single words is also a prerequisite for their matching techniques.

In [8], an adaptive approach to spotting words is described. This approach tries to find clusters of word images. These images are used to refine the search. This approach is more robust and less sensitive to noise than ordinary template matching. However, the problem with the speed of the pixel-by-pixel comparison remains.

The approaches of [6] and [10] use local features to find clusters of words. Features like local colour gradient are used that does not demand word or line segmentation. These approaches work well on source material that is difficult to segment and significantly degraded. However, extracting feature data is hard and the matching problem large.

3. METHOD

3.1 Image acquisition

The source material was scanned by the reproduction department at the Uppsala University library, Carolina Rediviva. It was scanned using a Digibook Suprascan A0 from I2S. The colour space was RGB and the resolution 15 Mpix and 19 Mpix for C61 and C64, respectively.

3.2 Pre-processing

The source images are pre-processed before the segmentation step. In preparation for the segmentation, the images needed to be converted into binary format using Otsu's thresholding method. An advantageous feature of the

manuscripts we worked with is that the same type of pen is used for all pages. Also, the angle of the pen tip does not vary much over time. This feature was exploited using a morphological opening with a diagonal structure element, which keeps binary pixels written by a specific pen type, removing lines and most noise. A small 5×5 median filter was also used to remove noise.

3.3 Segmentation

Earlier work in word spotting used data where each word could be segmented. The method proposed here is intended for source material where single word segmentation is hard or impossible. Our method does not need segmented words and works with one complete line at a time when spotting words. This simplifies the segmentation significantly when dealing with cluttered or damaged source material.

We performed segmentation in four steps. In the first step, the bounding boxes of the text on each page was determined. Each image depicts one spread of the manuscript and hence holds two pages. In the second step, the method identifies the positions of text lines. The third step refines the estimates of line positions by splitting the page vertically. This refines the estimate to account for a line not being completely straight. The texts were cluttered, and straight cuts between lines could not be made without cutting through letters. Hence, the fourth step refined the estimation of the line positions by using information from earlier steps and cutting the lines apart.

To find the bounding boxes of the text on each page and a rough estimate of the lines, segmentation methods based on projection have been shown to be effective on this type of source material [11]. The pixels of the binary images were projected on the x and y -axes followed by a Gaussian filtering. As size for the filter, roughly the width of the sought for features was used (i.e. the line height and the desired horizontal margin of the bounding box). As a result, the bounding box could be grown from the centre of the text and out. The results of this method were adequate for our purposes and identified all bounding boxes successfully.

When the bounding boxes had been identified, a rough approximation of the position of each line was made, by a projection of the binary image of each page on the vertical axis. This projection, combined with a Gaussian filter, gave a smooth curve. The width of the Gaussian filter was approximately the height of a text line. Local minima and maxima then corresponded to lines and spaces between lines, respectively. To refine this approximation, we split each page vertically to smaller pieces (columns) and used the same procedure as above to refine the estimate. This approach gave a good estimate of where the lines were placed in the document, even for bent text lines. However, it did not allow for a good horizontal cut through the text without cutting through letters. If, for instance, a letter g on some line would be close to a letter f on the line below, a cut would have to go around the letter regions. Also, if letters would be connected, the cut through them should be minimal.

A graph-based method was used to solve the problem of finding an optimal cut through the page between the text lines. The weight of the graph was generated from the image

after applying distance transforms and normalization. The first distance transform was performed on the text regions. This would penalize making the cut through a letter, and if unavoidable to make a cut as short as possible. Then a distance transform on the background was performed. This would make the cut keep away from letter regions to allow for errors in the thresholding. These two maps were added and the first normalized to a value above 1, the second to a value below 1.

The path through the graph with the lowest accumulative weight corresponded to a cut through the page with the properties described above. We used Dijkstra's algorithm to find the optimal path through the graph.

3.4 Searching

A variety of techniques have been applied to the problem of word spotting. Many of them use some kind template matching, trying to find matches between images. Others use local points of interest, trying to match point clusters together to find similar areas in an image. We will try to solve the problem of word spotting using a technique where one dimensional features are extracted from the source's text lines. The sequences generated will then be matched with a template that is a cut out of the sequence generated from the position of the sought for word.

3.4.1 Dynamic Time Warping

Dynamic time warping is a technique for matching one-dimensional signals. It finds the optimal way of matching one signal to another, with respect to some distance measure. The mapping is non-linear with the possibility of skipping and replicating samples in the matching. While the corresponding problem in two dimensions and more is NP-complete, dynamic time warping (1-D) can be solved using dynamic programming. Hence, the found match is the optimal match between the signals, with respect to the cumulative weight of the match and distance measure. The signals are represented as samples in a time series.

The shape and position of the warped template matching a signal sequence is given by the minimal cost path found in the cost map. An example of a cost map is given in figure 4. On one axis of the cost map correspond to samples of the template, the other axis the samples of the longer signal. Each point in the map is then calculated from a distance measure between the template sample and the sample from the longer signal. Distance measures are discussed below. From the cost map the cumulate weight for paths through the cost map is calculated as in equation 1, where D is the cumulative weight map.

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{array} \right\} + d(x_i, y_j) \quad (1)$$

When a problem with more dimensions than one is to be solved, a feature of the high dimensional data can be extracted. On this feature, the matching can be performed and the result brought back to the higher dimension. There can even be many features used in the same matching. This

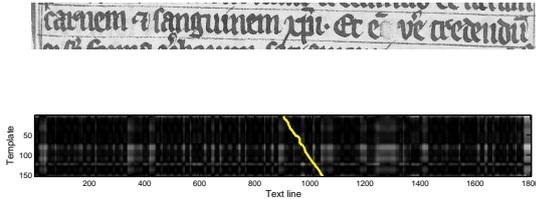


Figure 4: An example of the cost map used to generate the weighted graph used in searching with dynamic time warping. The optimal path through this map is shown as a line. Note that the weights in this map come from the distance measure used to find the dissimilarity between the features of the template and the text line (Section 3.4)

assumes that there is a distance measure that can be used for measuring similarity at a given point of the data.

Dynamic time warping has been proven successful on well segmented sources. Here the source is harder to segment and the risk of noise from erroneous segmentation is much higher. We have used the features proposed by [14] and extended it with new distance measures.

When searching for matches on a whole line the template is matched to a much longer sequence than itself. To make the template able to fit anywhere in the longer sequence, the path through the cost matrix can go from anywhere on the top row to anywhere on the bottom row. The problem of finding a match is a path-finding problem. Modifying the time warping procedure to allow different paths have been done successfully. Instead of matching single words with other single words, we can match a word the whole text line. Another advantage to this kind of matching is that we can spot longer sequences of words. The need for segmentation is minimal, reducing risk of errors from having to solve a hard segmentation problem to find single words. The coordinates of the path through the cost matrix will correspond to the matchings between the samples of the template and the longer sequence.

In many dynamic time warping applications, the *Sakoe-Chiba* band constraint is used to keep the warping from finding a match which differs much in size from the template. We can not use this kind of technique and have relied on a post match length comparison.

3.4.2 Feature selection

We used three features proposed by [14] as features for the word spotting. The first is the vertical projection of the segmented text. This is simply the sum of pixel values from each column of the image. The second and third features are the upper and lower contour of the text. The contours are interpolated between letters.

$$f_1(I, j) = \sum_{n=1}^{height} I(n, j) \quad (2)$$

Because of issues with the image capturing and text quality, the segmented lines sometimes had a non-horizontal slope that could vary on one single line. Hence, the contour features needed to be compensated for this irregularity. Otherwise, the matching would suffer because the extracted features would include this error as feature data. A baseline was estimated from the lower contour feature using a wide median filter (≈ 200 samples). This estimated baseline was then subtracted from both contour features.

In earlier works on word spotting using dynamic time warping, the square Euclidean distance measure was used to find the distance between a template and the text features. In equation 3, this measure is shown using the image I and the template T , the function f_n representing the n :th feature.

$$d(x_i, y_j) = \sum_{n=1}^3 (f_n(I, i) - f_n(T, j))^2 \quad (3)$$

Note that the distance measure is the square distance. This penalizes outliers very hard. The distance measure should be viewed as a measure of the dissimilarity between two samples.

The feature vectors have to be normalized to account for scale differences between text lines. Normalization used in earlier papers stretch the signal between 0 and 1. This way of normalizing was difficult to use on our sources. Our segmentation did not always find a good cut and at many places in the source material a good cut was impossible. Stretching the signal would then lead to an unstable normalization, since noise risked being used as the outliers in the normalization.

The Mahalanobis distance measure is commonly used in pattern recognition. It is defined in equation 4 where \vec{x} and \vec{y} are data vectors and S the covariance matrix of the space where the vectors live. This measure takes correlation into account and is scale invariant. Also, the equivalent of a normalization would then be done using the standard deviation. The standard deviation is very robust against a sudden extreme value in a feature signal.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (4)$$

At first we assumed that the correlation between features were irrelevant. And have tried to use a diagonal covariance matrix in the Mahalanobis distance measure. Then the measure collapses to normalized Euclidean distance, where the feature vectors could be normalized beforehand as in equation 5 (μ_f is the mean of the feature vector and σ_f its standard deviation). Then the Euclidean distance formula (equation 3) could be used.

$$f_i^{(normalized)} = \frac{f_i - \mu_f}{\sigma_f} \quad (5)$$

A sample of the covariance matrix from on line in C61 is

given in equation 6. It shows a large correlation between the projection feature and the upper contour feature. Also, it shows an inverse correlation between the projection and the lower contour feature. There is also some correlation between the contour features, but not as large.

$$S = \begin{pmatrix} 1.0000 & 0.4161 & -0.5169 \\ 0.4161 & 1.0000 & 0.1527 \\ -0.5169 & 0.1527 & 1.0000 \end{pmatrix} \quad (6)$$

The above covariance matrix led us to believe that the correlation between variables could not be ignored. Hence, we tested to used whitening on the features vectors instead of normalization. Using Euclidean distance on the resulting feature vectors is equivalent to using Mahalanobis distance. The local covariance matrices were estimated from the feature vector of each text line.

The template that was used in each search was selected from the image itself. The corresponding feature data was fetched from the feature database before running a search for matching words.

3.4.3 Search procedure

The first step in the search procedure was adding all segmented lines to a processing queue. From this queue, a post was drawn and processed using dynamic time warping. The weight of the warping has compared to a threshold. The threshold was adaptive with the aim of finding the 500 best matches.

When a match was found, the line was split into the part containing the match and the preceding and following parts of the line. The matched part was put in a list of matches while the remaining part of the line were appended to the processing queue. Dynamic time warping always finds the best match. Hence, if the first match on a line had a weight that was under the threshold, on more interesting matches could be found on that line.

3.5 Presentation

The results of the search using dynamic time warping are sorted by normalized weight. The normalized weight is defined as the matching cost, i.e. the cumulate cost of each visited node in the weight map, divided by the path length.

To generate performance statistics, reference statistics for each word used in the search was needed. These word occurrences was picked by hand and automatically compared to the results from the word spotting. We did not have access to a full transcription of the text sources used and segmentation of words was not done. This posed a problem when trying to compare our results with previous work. Below, performance metrics will be presented.

4. EXPERIMENTS

To evaluate and demonstrate the proposed method, we have performed a series of experiments on both script types. All experiments have been performed on an Intel Xeon 3.60GHz, Dual Core and the running times have been on the order of 1

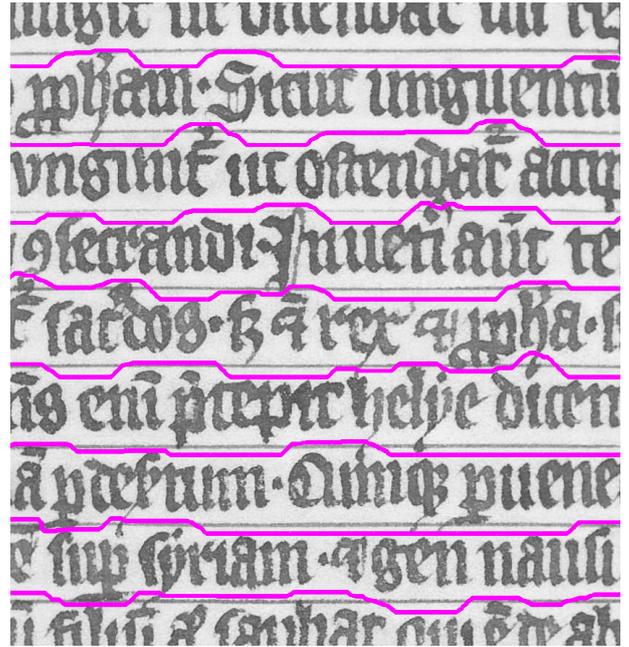


Figure 5: An example of the line segmentation from Manuscript C64. The cuts are shown as lines in magenta. Note the importance of having a method that can cut around protruding letters. (Section 3.3)

– 2 minutes for 20 – 30 spreads. The current implementation is in MATLAB and could be further optimized.

4.1 Segmentation

An important sub-goal of the paper is accurate segmentation of lines. Figures 5 and 6 show some examples from the proposed method.

4.2 Searching

During interactive search, the user selects a word using the mouse and get back a series of hits that are ordered according to similarity. The results of the search using dynamic time warping are sorted by normalized weight. The normalized weight is defined as the matching cost, i.e. the cumulate cost of each visited node in the weight map, divided by the path length. Overall, this is working fine and some results can be seen in figures 7, 8 and 9. However, this mode of operation is not suitable for an objective validation of the method.

4.3 ROC curve

To test the accuracy of the method, as a tool for finding all occurrences of a word, we have performed a ROC analysis. In this experiment, we have manually marked all occurrences of some words in the two datasets. We have then analyzed the percentage of true positive hits vs false positive hits, for the

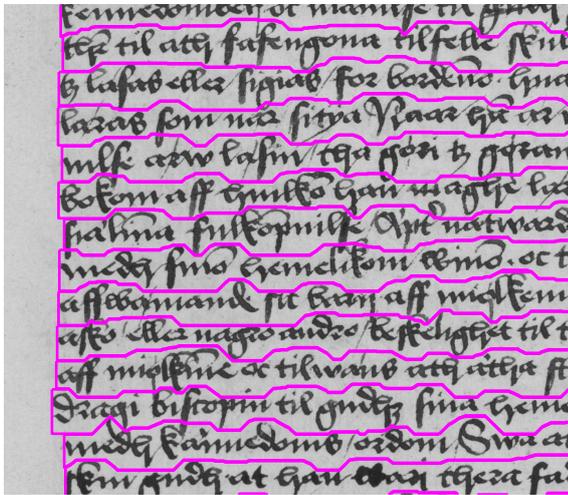


Figure 6: An example of the line segmentation from Manuscript C61. The cuts are shown as lines in magenta. Note the importance of having a method that can cut around protruding letters. (Section 3.3)

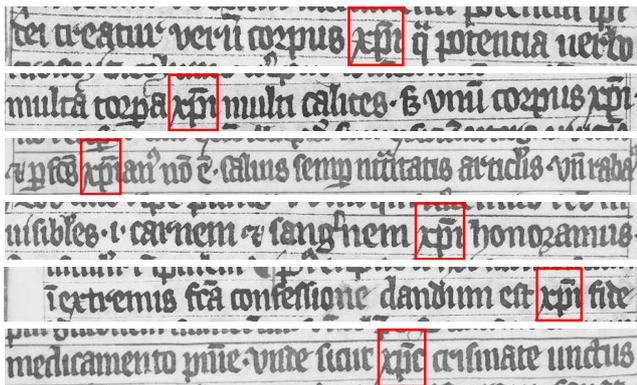


Figure 7: Example of occurrences of the word xpi. The template has been fitted to the text line by dynamic time warping. (Section 3.4.3)

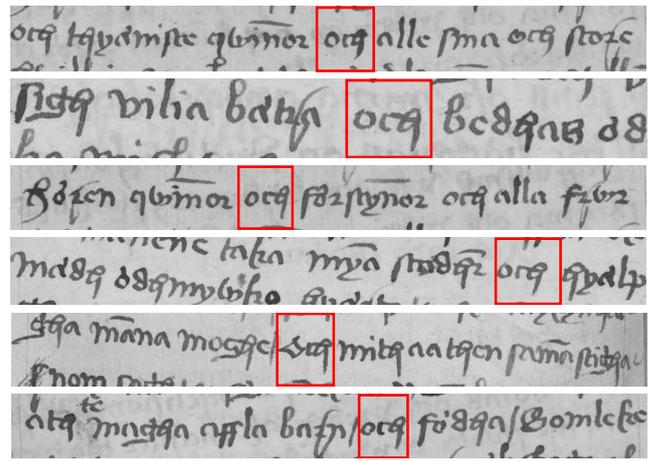


Figure 8: Example of occurrences of the word och. The template has been fitted to the text line by dynamic time warping. (Section 3.4.3)

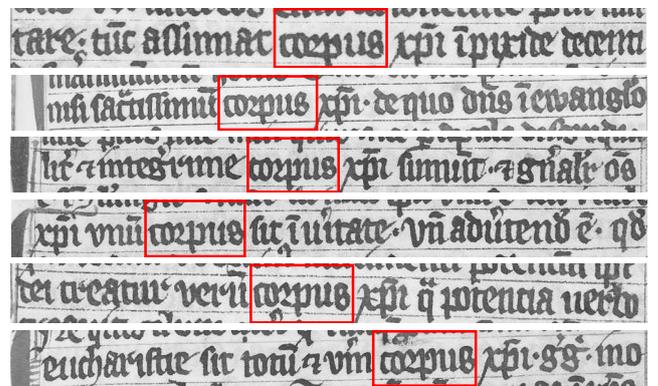


Figure 9: Example of occurrences of the word corpus. The template has been fitted to the text line by dynamic time warping. (Section 3.4.3)

N best matches. The results are presented as ROC curves in figures 10, 11 and 12.

4.4 Word statistics

Besides searching, word spotting is useful to generate various kinds of agglomerated information about a manuscript. To present an overview of a document to a user, we have generated graphs of word occurrence indexed by page number, analogous to plots generated by Google Books Ngram Viewer [1] that plots word occurrences as a function of year. In figure 13, Ngram-like plots can be seen.

5. CONCLUSIONS

The present implementation of word spotting clearly improves on the state of the art as regards performance. It also represents a more robust variety of word spotting, as it does not rely on accurate word segmentation and text lines of similar height to work well.

We have also successfully extended the use of word spotting to new varieties of handwriting. Our data cover two dis-

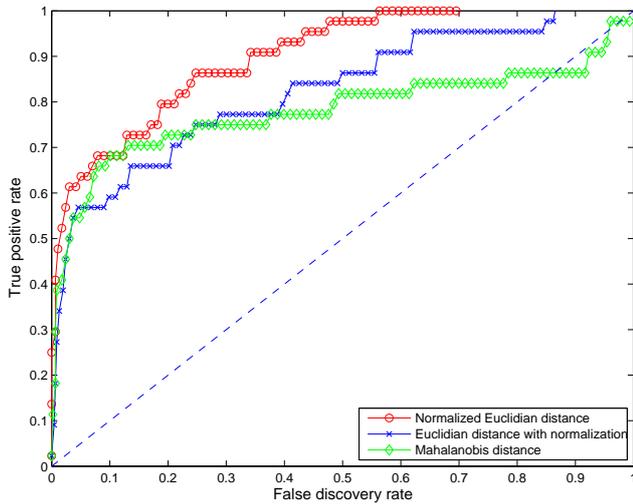


Figure 10: ROC curve for results when searching for the $\chi\rho\bar{i}$ abbreviation of *Christi* in C64. (Section 4.3.)

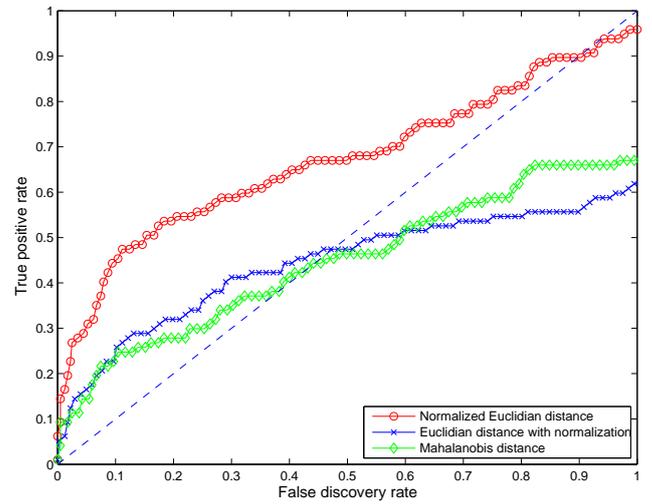


Figure 12: ROC curve for results when searching for the word *och* in C61. (Section 4.3.)

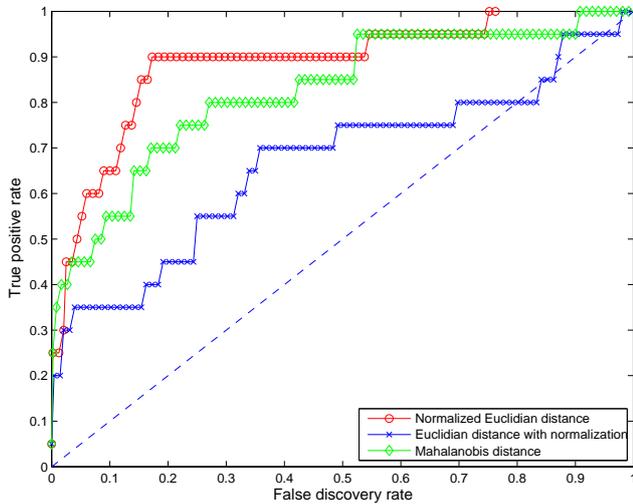


Figure 11: ROC curve for results when searching for the word *corpus* in C64. (Section 4.3.)

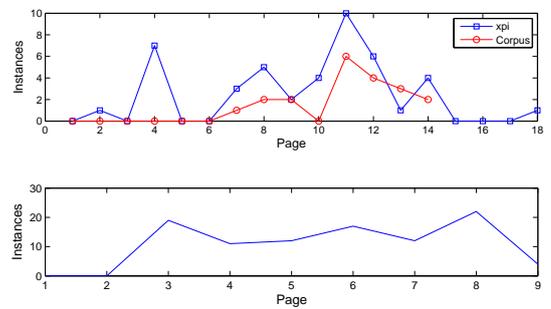


Figure 13: Counts for words over a sequence of pages: $\chi\rho\bar{i}$ and *corpus* from C64 (upper plot) and the word *och* from C61 (lower plot). (Section 4.4.)

tinctly different gothic script types, one formal (Textualis, C64) and one cursive (Cursiva, C61). One could assume greater difficulties with the cursive variant, as the letters here tend to vary more due to faster execution, which is also seen in the figures 10, 11 and 12.

From the point of view of our long-term goal of developing data mining and transcription tools, the present results are encouraging. As it stands, our word spotting algorithm would be of value in a tool for computer-aided transcription, given that it has been trained on frequent word forms. It is also clear that not all occurrences of the words are found. The method is thus useful for interactive browsing but cannot be fully trusted for generation of word statistics, unless we make assumptions about independency. Nevertheless, graphs like that in figure 13 allow us to explore the material interactively and will help the user to form hypotheses about

the material. In Manuscript C64, the word count statistics for the words *corpus* and $\chi\rho\bar{i}$ suggests that the text discusses the eucharist on pages 6–14. At the same time, the common word *och* (*and*) is evenly distributed across C61. Since large amounts of written material from medieval Europe are written in scripts of these types, it is very important for future research on HTR for medieval manuscripts to develop methods for both kinds of script.

We envisage that further success in HTR for medieval manuscripts requires solutions involving an ensemble of several kinds of image classifiers, among which word spotters would be important components. Other classifiers would target letters and even smaller writing components, which are difficult to find using spotting techniques based on dynamic time warping. The investigation of more advanced word-models based on Hidden Markov Models, similar to the developments in speech processing, is also interesting for future research. Interpretive decisions are also likely to benefit from the use of language models (cf. [3]), which estimate the probability that a certain letter or word sequence would occur, given, e.g. the language, the document type, or the context.

To be able to collect word forms automatically directly from a manuscript written in a cursive variant of the gothic script, in our case *Cursiva Recentior*, must be considered a major breakthrough for the study of the languages from the Middle Ages. An enormous amount of written material in this script type exists from this period, and a large part of this is not available in printed editions. For historical lexicographic studies (for instance preparation of dictionaries), our method would make it possible to collect word form tokens directly from unedited manuscripts. As long as word investigations are only performed on works available in print, our knowledge of the language from the period in question will remain limited. This work would thus combine the efforts of image analysts, philologists, and computational linguists and points in the direction of an empirically systematic computational palaeography.

6. ACKNOWLEDGMENTS

We would like to thank the Deans of the Faculties of Arts and Languages, Jan Lindegren and Björn Melander, for funding, advice and encouragement; Ewert Bengtsson from the Centre for Image Analysis, for funding and advice. Britt-Inger Johansson and the SALT team for coordination; Per Cullhed and Uppsala University Library for access to medieval documents; Stina Fallberg Sundmark for her work on C64; and Monica Hedlund for her expertise in Latin palaeography;

7. REFERENCES

- [1] Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331(6014):176–182, Jan. 2011.
- [2] M. Andersson-Schmitt and M. Hedlund. *Mittelalterliche Handschriften der Universitätsbibliothek Uppsala: Katalog über die C-Sammlung: Bd. 2. C 51–200*. Almqvist & Wiksell International, Stockholm, 1989.
- [3] B. Broda and M. Piasecki. Correction of medical handwriting OCR based on semantic similarity. In H. Yin, P. Tino, E. Corchado, W. Byrne, and X. Yao, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2007*, volume 4881, pages 437–446. Springer Berlin / Heidelberg, 2007.
- [4] A. Cappelli. *Lexicon Abbreviaturarum*. Verlagsbuchhandlung von J. J. Weber, Leipzig, 1928.
- [5] A. Derolez. *The Palaeography of Gothic Manuscript Books*. Cambridge University Press, 2003.
- [6] M. Diem and R. Sablatnig. Recognition of degraded handwritten characters using local features. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 221–225, July 2009.
- [7] S. Kane, A. Lehman, and E. Partridge. Indexing George Washington’s handwritten manuscripts. 2001.
- [8] V. Kluzner, A. Tzadok, Y. Shimony, E. Walach, and A. Antonacopoulos. Word-based adaptive OCR for historical books. In *International Conference on Document Analysis and Recognition*, pages 501–505, 2009.
- [9] V. Lavrenko, T. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*, pages 278 – 287, 2004.
- [10] Y. Leydier, F. Lebourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recogn.*, 40:3552–3567, December 2007.
- [11] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *Int. J. Doc. Anal. Recognit.*, 9:123–138, April 2007.
- [12] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten archives. *Intelligent multimedia information retrieval*, 1997.
- [13] D. Nain, S. Haker, W. E. L. Grimson, E. R. Cosman, Jr., W. W. Wells, III, H. Ji, R. Kikinis, and C.-F. Westin. Intra-patient prone to supine colon registration for synchronized virtual colonoscopy. In *Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part II, MICCAI '02*, pages 573–580, London, UK, UK, 2002. Springer-Verlag.
- [14] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:521, 2003.
- [15] H. Sakoe and S. Chiba. Dynamic programming optimization for spoken word recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, 1980.