

Segmentation of the Date in Entries of Historical Church Registers

M. Feldbach and K. D. Tönnies

Computer Vision Group, Department of Simulation and Graphics,
Otto-von-Guericke University, P.O. Box 4120, D-39016 Magdeburg, Germany,
{feldbach, klaus}@isg.cs.uni-magdeburg.de

Abstract. Handwriting recognition requires a prior segmentation of text lines which is a challenging task, especially for historical scripts. Exemplary for the date in entries of historical church registers, we present an approach which enables a segmentation by using additional knowledge about the word sequence. The algorithm is based on probability distribution curves and a neural network, which assesses local features of potential word boundaries. Our database consists of 298 different date entries from the 18th and 19th century which contain 674 word boundaries. The algorithm generates hypotheses for the expected date type, ordered by their probability. Tests resulted in an accuracy of 97% for the best four hypotheses.

Keywords: Handwriting recognition, word segmentation, document image processing

1 Introduction

Automatic reading of historical documents such as church registers would provide historians or sociologists with an efficient tool for extracting information. Vast amounts of such documents are stored unread in churches and archives. If automatic methods cannot be provided, most of those documents will never be transcribed. Apparently, automatic recognition of historical documents cannot be achieved at once but requires several preprocessing steps. We introduce an interactive system which provides new findings about detecting word boundaries as well as support for users regarding the identification of date entries.

Since the positions of word boundaries cannot be reliably found by analysing geometrical attributes only, we generate a list of hypotheses, sorted by their probability. Those hypotheses enable a specialised word respectively cipher recognizer, to process and recognise the isolated script objects. Essential preprocessing steps are the reconstruction and separation of text lines which have been described in [1] and [2].

If information about the characters is available, the analysis of character sequences leads to reliable predictions about word boundaries [3]. However, we don't have this information. In order to find word gaps, several distance measuring methods have been investigated, such as run-length Euclidean heuristic

distance [10], convex hull [7] etc. These methods analyse relations between adjacent connected components and find gap metrics to cope with various spacing styles [9,10]. Manmatha et al. [8] analyses characteristics of old scripts and segments words consisting of isolated and connected characters. Nonetheless, the examined scripts show straight text lines and well separated words. If the script, however, is characterised by gaps of different sizes, methods specialised on differentiating between inter-character gaps and inter-word gaps have been found successfully [5,4]. In old church registers, connections between adjacent words may occur. Therefore we must look for potential word gaps within text objects as well.

1.1 Date Types

The possible number and position of boundaries between date components can be restricted if the different date constituents are known a-priori. We examined a large number of entries in church registers and found that the date in all entries consisted of the elements ciphers (C), artefacts (A), and month names (M) with the following combinations being possible: *C-C-A-M*, *C-A-M*, *C-C-M*, *C-M*.

The artefacts after the ciphers are “te” or “ten” and indicate the date. Names of the months may be abbreviated.

1.2 Data Base

The date in a church register is user identified by marking the begin and end of the date entry. We created a database with a size of 298 different date entries from church registers of the county of Wegenstedt for development, training and test of our method. The entries contain 674 word boundaries and were from chronicles between 1719 and 1813. The following information is generated for each selected date from our pre-processing step:

- Skeletons of stroke segments (between stroke crossings or ends) that are certain or potential parts of the text in this line.
- Connectivity information between different segments.
- Line width of segments.
- Course of text lines (baseline and midline, baseline of the text line above and midline of the text line below).

The following information was added interactively to serve for training and test purposes:

- Day, month and year of the date (the year is not part of the date entry)
- Date type (such as, e. g., *C-A-M*).
- Position of boundaries between date elements.

2 Preprocessing

Script objects are identified based on the knowledge about the baseline and midline paths as well as the position of the left and right date boundaries. There

are two classes of objects: one class containing all objects which are *certainly* part of the date and one class containing all objects which are *potentially* part of the date [1]. Subsequently, by using objects only, which are certainly part of the date, interferences from adjacent lines or words are avoided.

In order to find potential word boundaries, preprocessing steps like slant correction or base line adjustment have to be performed. In the following, boundaries between all date elements (words or ciphers) are called word boundaries.

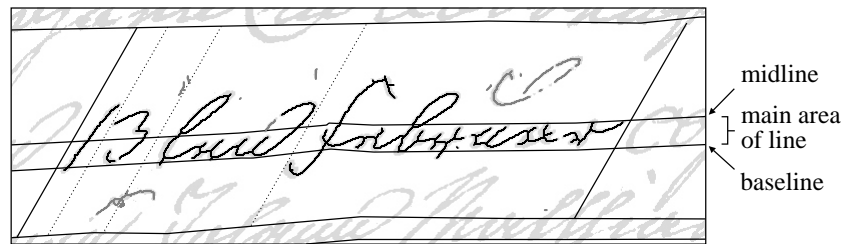


Fig. 1: A marked date (C-C-A-M) with left and right boundaries (solid line) and found word boundaries (dashed line), certainly (black) and potentially (grey) parts of the date

2.1 Slant Correction

There is a number of methods for slant correction which determine the best angle by searching for the most vertical strokes such as [11] but in our case the simple procedure, described in the following, corrects slants effectively.

We estimate the slant of the script by considering the single line segments as vectors. The average direction of all vectors, weighted by their length, is calculated. The average direction determines the slant angle. Every skeleton point is displaced vertically depending on the slant angle and the distance to the base line. Due to the calculation on the bases of discrete coordinate values, a gap is likely to appear between two adjacent skeleton points. These gaps will be closed again. Additionally, the skeleton segments are smoothed in order to eliminate artefacts which were caused by the correction.

2.2 Base Line Adjustment

The vertical distances of script objects and centre line are determined relatively to the baseline in order to improve the examination of the script object arrangement.

2.3 Removal of Punctuation Marks

Small objects between midline and baseline, potentially representing punctuation marks, are removed prior to searching word boundaries. An object is small if the vertical and horizontal extent of its skeleton is smaller than the stroke width.

3 Potential Boundary Search

In the following, objects are connected continua within the date zone. A date entity is either of the three classes cipher, month, or artefact (see Sect. 1.1).

A list of potential word boundaries between slant-corrected objects is generated. Boundaries may exist at horizontal gaps between objects as well as within an object. The latter is true when two date entities touch each other.

A height of the date $y_{\max}^{\text{high}}(x)$ is computed for each location x as the maximum height of the object at this location. A second measure $y_{\max}^{\text{low}}(x)$ is computed as the maximum height of strokes at location x which are below the midline.

Two types of potential word boundaries (pWB) are computed. A pWB of type I is generated at each gap in the text line main area between baseline and midline. This finds all potential boundaries between non-connected text parts as well as between those that are connected above the midline (such as touching capital letters and/or ciphers). A pWB of type II is generated at position x if there is only one line segment and $y_{\max}^{\text{low}}(x)$ has a local minimum. In order to find only relevant local minima, a vertical search window is used. The width of the window is two times the stroke width of the single segment.

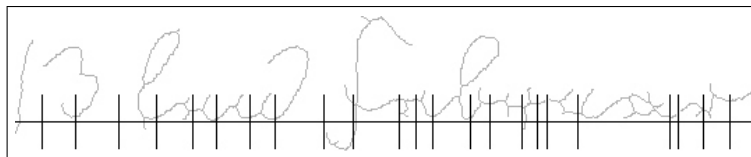


Fig. 2: Adjusted skeleton of the date entry with marked position of potential word boundaries.

4 Assessing Potential Word Boundaries

There exist eight different kinds of boundaries $g = 1 \dots 8$ for the four types of date entries (see Sect. 1.1). Date entries of our data base have between 5 and 25 potential boundaries (e. g. see Fig. 2). In the following, these boundaries are assessed regarding their positions and local attributes in order to derive hypotheses about the positioning of the word boundaries. For every word boundary, probability values are calculated for every kind of boundary by determining the average value of a probability distribution curve p_g^{DC} (see Sect. 4.1) and the normalized output value of a neural network p^{NN} (see Sect. 4.2).

4.1 Probability Distribution Curves

For every kind of boundary g , a probability distribution curve is generated. The position of a training boundary is normalized related to the date width

and it ranges therefore in the interval $[0, 1]$. The probability distribution curve which results from the samples is smoothed by a gaussian function. The variance decreases with an increasing number of samples (see (2)). An appropriate value for k was found experimentally at 0.04. The probability value $p_g^{\text{DC}}(x_i)$ of a potential boundary with position $x_i \in [0, 1]$ for boundary type g is calculated by

$$p_g^{\text{DC}}(x_i) = \frac{1}{N_g} \cdot \sum_{j=1}^{N_g} \exp\left(\frac{(x_i - x_{g,j})^2}{-2\sigma_g^2}\right) \quad (1)$$

$$\sigma_g^2 = k \cdot \frac{1}{N_g} \quad (2)$$

where N_g denotes the number of training boundaries $x_{g,j}$ for boundary type g and $j = 1 \dots N_g$. The probability distribution curves of the two word boundaries of date type C-A-M is shown in Fig. 3.

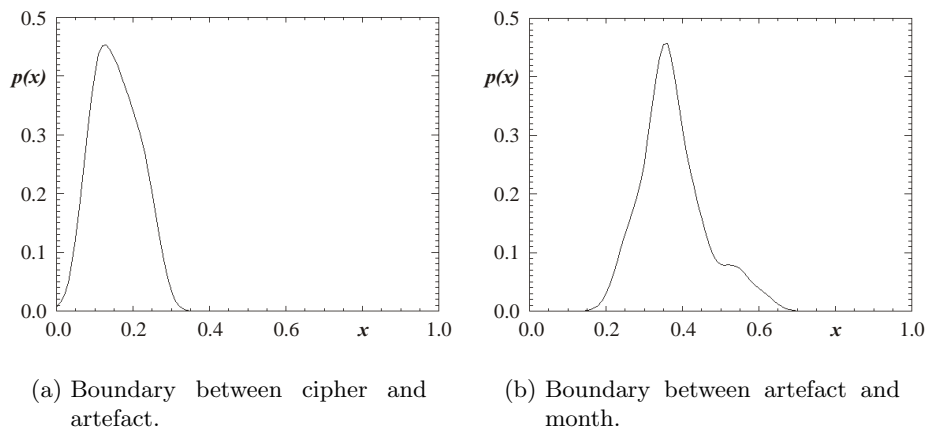


Fig. 3: Probability distribution curves for the date type C-A-M.

4.2 Local Features

Four features are extracted and propagated by a neural network in order to further examine a potential boundary. We used a multi-layer perceptron with four input neurons, eight hidden layer neurons and one output neuron. It was trained in a way that a value of 0.2 is expected at the output neuron for a wrong boundary and a value of 0.8 for a correct boundary.

Due to the sigmoid transfer functions of the neurons, values of 0 and 1 are inappropriate [6]. For further processing, the output o of the network is normalised to $p^{\text{NN}}(x_i) \in [0, 1]$:

$$p^{NN}(x_i) = \begin{cases} 0 & o < 0.2 \\ (o - 0.2) / 0.6 & 0.2 \leq o \leq 0.8 \\ 1 & o > 0.8 \end{cases} \quad (3)$$

The four features we used are boundary width, number of crossings respectively touchings with script objects, height of the script left to the boundary, height of the script right to the boundary.

Boundary Width. We estimate the beginning $x_{\min}(b_i)$ and the end $x_{\max}(b_i)$ of the interval which contains the boundary b_i . In case of boundaries of type I, $x_{\min}(b_i)$ is the beginning and $x_{\max}(b_i)$ the end of the gap in the main area of line. In case of type II, $x_{\min}(b_i)$ is the position of the next local maximum of $y_{\max}^{\text{low}}(x)$ left to the boundary and $x_{\max}(b_i)$ the position of the next local maximum right to the boundary. In order to minimise the influence of the script width, we normalise the boundary width related to the date width w_{total} and the number of potential boundaries N_{pb} . The boundary width $w(b_i)$ is therefore

$$w(b_i) = \frac{(x_{\max}(b_i) - x_{\min}(b_i)) \cdot N_{\text{pb}}}{w_{\text{total}}} \quad (4)$$

Number of Object Touchings. The more often a potential object boundary touches a script line, the less likely it is a true object boundary. This relation is implemented by counting the number of cuts.

Height of the Script Next to a Boundary. Inter-word boundaries and inter-character boundaries differ also by the shape of adjacent characters. For this reason, the height of the characters left and right to the boundary is included as a feature.

The width of a character is about two times the distance of two potential boundaries. Thus, we calculate the width w_a of the interval to be examined by

$$w_a = \frac{2N_{\text{pb}}}{w_{\text{total}}} \quad (5)$$

The left interval has a range of $\max(0, x_{\min} - w_a)$ to w_a and the right interval has a range of x_{\max} to $\min(w_{\text{total}}, x_{\max} + w_a)$.

In each of those intervals, the maximum height of the script objects $y_{\max}^{\text{high}}(x)$ is calculated and normalised by dividing by the distance $y_{ml}(x)$ between baseline and midline.

5 Generating Hypotheses

Since the type of a date is unknown, for each of the four possible types, containing $k = 1 \dots 3$ word boundaries, $\binom{N_{\text{pb}}}{k}$ hypotheses are generated from the combination of the N_{pb} potential boundaries.

The probability $p(h_i)$ of hypothesis h_i is calculated by the average of probabilities $p(b_x)$, $p(b_y)$, $p(b_z)$ of word boundaries b_x, b_y, b_z with $1 \leq x < y < z \leq N_{pb}$.

$$p(h_i) = \frac{p(b_x)[+p(b_y)[+p(b_z)]]}{k_i} \quad (6)$$

Finally, a list with all hypotheses is created and sorted according to their probability $p(h_i)$.

6 Results

We extracted 298 date entries and manually defined the correct word boundaries from church registers of the 18th and 19th century. Because of the insufficient size of the data base, we performed several runs where we used 90% (268) as training data and 10% (30) as test data. After 10 runs with different training and test sets, 97% of the correct word boundary combinations are included in the best four hypotheses (see Fig. 4). In case of using the local features only, we obtain 88%, while the processing of the distribution curve without the local features results in 69%. This shows the advantage of combining the two kinds of property – position and local features.

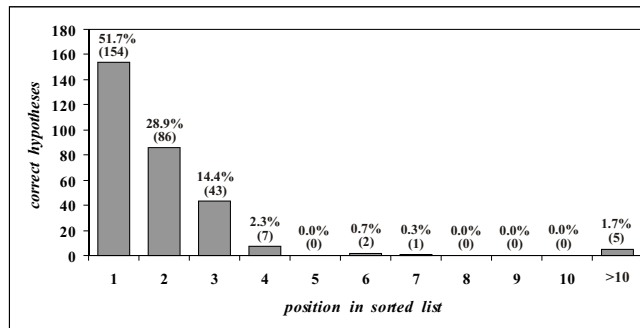


Fig. 4: Positions of 298 correct hypotheses in sorted list related to their probability.

A fast automatic evaluation of the tests was possible since information about the correct word boundaries was also available for the test data.

Wrongly assessed hypotheses may have different reasons. In many cases, mistakes are due to the fragmentation of actually connected word segments caused by bleached ink. These segments could thus not be classified as certain part of the date. Since only the certain script objects were considered in the method described above, the shape of the script was changed. An additional source of error was punctuation which was not removed due to its size. A more elaborated removal technique such as the one applied in [10] might solve this problem.

7 Conclusions

In this paper, we presented a method which delivers hypotheses about the position of word boundaries. We considered scripts whose word boundaries are not characterised by obvious gaps as well as scripts containing word touchings. This was achieved by using additional information about the word sequence to be examined as shown exemplarily for date entries from old church registers.

We plan to improve the segmentation by reducing sources of error such as incomplete removal of punctuation marks. The output of the algorithm are hypotheses about the number and positioning of word boundaries. These hypotheses about date components will then serve as prior information for a date recogniser.

References

1. M. Feldbach and K. D. Tönnies. Line Detection and Segmentation in Historical Church Registers. In *Sixth International Conference on Document Analysis and Recognition*, pages 743–747, Seattle, USA, September 2001. IEEE Computer Society.
2. M. Feldbach and K. D. Tönnies. Robust Line Detection in Historical Church Registers. In *Pattern Recognition, 23rd DAGM Symposium*, pages 140–147, Munich, Germany, September 2001. Springer-Verlag.
3. D. Kazakov and S. Manandhar. A hybrid approach to word segmentation. In D. Page, editor, *Proceedings of the 8th International Conference on Inductive Logic Programming*, volume 1446, pages 125–134. Springer-Verlag, 1998.
4. G. Kim and V. Govindaraju. Handwritten Phrase Recognition as Applied to Street Name Images. *Pattern Recognition*, 31(1):41–51, January 1998.
5. S. H. Kim, S. Jeong, G.-S. Lee, and C.Y.Suen. Word Segmentation in Handwritten Korean Text Lines Based on Gap Clustering Techniques. In *Sixth International Conference on Document Analysis and Recognition – ICDAR 2001*, pages 189–193. IEEE Computer Society, September 2001.
6. H. Kruse, R. Mangold, B. Mechler, and O. Pengler. *Programmierung Neuronaler Netze: Eine Turbo Pascal Toolbox*. Addison-Wesley, 1991.
7. U. Mahadevan and R. C. Nagabushnam. Gap Metrics for Word Separation in Handwritten Lines. In *International Conference on Document Analysis and Recognition*, pages 124–127, Montreal, Canada, 1995.
8. R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten documents. In *Scale-Space Theories in Computer Vision*, pages 22–33, 1999.
9. U. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *Sixth International Conference on Document Analysis and Recognition*, pages 159–163, Seattle, USA, September 2001. IEEE Computer Society.
10. G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41–52, January 1994.
11. A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2001.