Word Segmentation of Handwritten Dates in Historical Documents by Combining Semantic A-Priori-Knowledge with Local Features

Markus Feldbach and Klaus D. Tönnies

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

Abstract

The recognition of script in historical documents requires suitable techniques in order to identify single words. Segmentation of lines and words is a challenging task because lines are not straight and words may intersect within and between lines. For correct word segmentation, the conventional analysis of distances between text objects needs to be supplemented by a second component predicting possible word boundaries based on semantical information. For date entries, hypotheses about potential boundaries are generated based on knowledge about the different variations as to how dates are written in the documents. It is modeled by distribution curves for potential boundary locations. Word boundaries are detected by classification of local features, such as distances between adjacent text objects, together with location-based boundary distribution curves as a-priori knowledge. We applied the technique to date entries in historical church registers. Documents from the 18th and 19th century were used for training and testing. The data set consisted of 674 word boundaries in 298 date entries. Our algorithm found the correct separation under the best four hypotheses for a word sequence in 97% of all cases in the test data set.

1 Introduction

Vast amounts of old documents with historical content are stored unread in churches and archives. The work of historians and sociologists could be simpler if methods existed which automatically or semi-automatically extracted information from such documents. Church registers are a type of historical documents which record relevant events in a community over the centuries. Separate registers were kept for births, marriages and deaths. Manual extraction by humans is difficult because few people are able to read the old handwritings. The main goal of computer-assisted analysis is to extract names of persons involved in an event (birth, death, marriage) and dates when the event took place. Analysis consists of a segmentation step in order to separate the text into meaningful units and the automatic reading of names or dates from a segment.

Segmentation consists of three steps: *(1)* Separation of the text into blocks were each block describes a single event, *(2)* segmentation of the block into lines and *(3)* segmentation of lines into words and ciphers. In our current scenario for document analysis, we assume that a user interactively identifies the location of names and dates in the text. Then, the system attempts to read either names and dates and suggests interpretations. This is still more efficient than letting the user read the complete text from the registers as identification of names or dates is easier than reading handwritings of the 18th and 19th century.

In this simplified scenario, segmentation into blocks is not necessary as the user is expected to be able to differentiate between them. Segmentation into text lines was presented previously (see [1]) and will be mentioned only briefly in this paper. We will concentrate on the separation of the different components of the date provided that the user has specified a start and end point for the date entry. Separation between first and last name should follow the same principles. The output of this step will be a number of hypotheses which will serve as input for a date or name recognition step.

Church registers were written with lines being close to each other, with text of a given line possibly reaching into adjoining lines, connections between adjacent words, and gaps between characters belonging to one word. In text line segmentation, we followed a strategy which is independent of gaps between lines of text and straightness of text lines. Evidence stemmed from local minima defining

IEEE
COMPUTER
SOCIETY

locally straight baselines. This method was applied because horizontal or vertical continuity constraints such as those assumed in [4, 10] did not hold in our case.

For word boundaries, we have a similar situation. Word boundary detection schemes that rely on the predictions from the recognised characters itself [2] are not applicable as this information does not exist. Approaches which evaluate gap sizes for differentiating between word boundaries and character boundaries such as [3, 5, 6, 7, 8, 9] are more appropriate as they do not require text recognition. However, in old church registers, adjacent words may by connected. Thus, we have to expect word gaps within text objects as well as gaps between text objects which are not word boundaries. The performance of a data-driven approach will be poor. However, we can use the semantics of knowing that a selected text part is either a date or a name.

Dates and names are written in a limited number of ways and we attempt to find the best explanation of a sequence of text objects in a line with respect to this knowledge. The method is presented here using the date as an example and should be easily adapted for word separation in names.

Information about word boundaries for dates is supplied by distribution curves for potential boundary locations (we call this the global information). Information about the characteristics of word boundaries (which is called local information) is given by a feature vector that is classified by a backpropagation network.

The remainder of paper is organised as follows. First an overview about segmentation of text lines will be given and necessary pre-processing steps for word boundary detection will be explained. In the main part, computation and use of local and global information for boundary detection will be presented. Results of the detection step will be shown in the next section followed by concluding remarks.
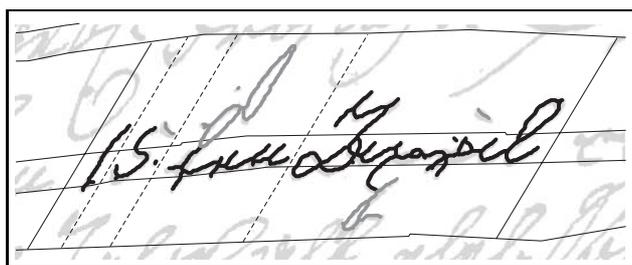
Figure 1. A marked date (C-C-A-M) with left and right boundaries (solid line) and found word boundaries (dashed line), the certain (black) and the potential (grey) parts of the date.

2 Preprocessing

Four different boundaries describe the shape of a line of text: The ascender line, the descender line, the baseline and the centre line. The former two are not so important for the segmentation and recognition and they are not well pronounced because ascenders and descenders occur too infrequently and their height has a large variance. Thus, we restrict our search to that of the baseline and the centre line.

The baseline is found based on the local minima of all script objects in y-direction. They are assumed to be locally straight even though lines of text curve over the complete width of the page. Local minima indicate, for the most part, points on the baseline and on the descender line with the majority stemming from baseline minima. Thus, the only line stretching over the whole width of the page and being made up of local minima from script objects that are close enough together and locally straight, should be the baseline. To a lesser extent, the same argument holds for finding the centre line based on the local maxima.

The knowledge about the path of the line and the position and size of the script objects is used for carrying out the segmentation. Each object has a fixed and known relation to any base and centre line. It may be situated either above or below this line or it intersects the line. This information is used to allocate the script objects to lines of text.

For every line the script objects are assigned to one of three classes: The continuum is *certainly*, *potentially*, or *not* part of the line.

Script objects containing to the date are identified based on the knowledge about the baseline and midline paths as well as the position of the left and right date boundaries. 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, slant correction, base line adjustment, and the removal of punctuation marks have to be performed. The slant of the script will be estimated 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. Base line adjustment means the recalculation of the vertical position of every object point related to the base line. In the following, boundaries between all date elements (words or ciphers) are called word boundaries.

3 Date Segmentation

The search of boundaries between words will be made by analysing local features of potential boundaries (as e. g. [7, 8]). Additionally, we assume that some strokes within a script object may constitute a word boundary between two objects being connected. As this decreases the performance of segmentation, we use semantic information
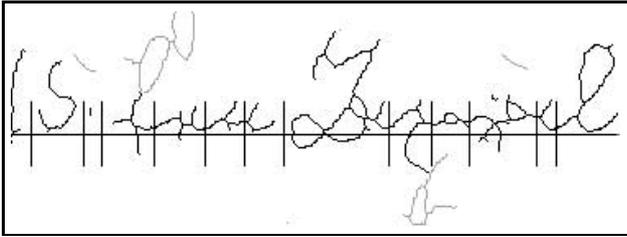
Figure 2. Adjusted skeleton of the date entry with marked position of potential word boundaries.

about expected word boundaries in a date entry to supplement the a-fore-mentioned knowledge. This is done using probability distribution curves for probable separation between different parts of a date.

3.1 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.

3.2 Potential Word Boundaries

A date entity is either of the three classes cipher, month, or artefact.

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 $y_{\max}^{\mathrm{high}}(x)$ is computed for each location x as the maximum height of the object at this location. A second measure $y_{\max}^{\mathrm{low}}(x)$ 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 $y_{\max}^{\mathrm{low}}(x)$ 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.

There exist eight different kinds of boundaries $g = 1 \ldots 8$ for the four types of date entries (see Sect. 3.1). 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 $\mathrm{P}_g^{\mathrm{DC}}$ (see Sect. 3.4) and the normalised output value of a neural network p^{NN}.

3.3 Local Features

We trained a multi-layer perceptron with four input neurons, eight hidden layer neurons and one output neuron in order to further examine a potential boundary. 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 (distance between the bounding boxes). In case of type II, $x_{\min}(b_i)$ is the position of the next local maximum of $y_{\max}^{\mathrm{low}}(x)$ 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

$$w(b_i) = \frac{(x_{\max}(b_i) - x_{\min}(b_i)) \cdot N_{\mathrm{pb}}}{w_{\mathrm{total}}} \qquad (1)$$

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 size of the intervals right and left to the boundary position is determined by the character width. This

value is about two times the average distance of two adjacent potential boundaries.

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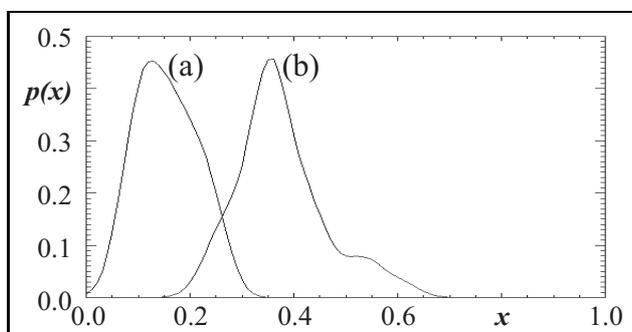
3.4 Probability Distribution Curves

For every kind of boundary g, a probability distribution curve is generated. The position of a training boundary is normalised with respect to the width of the date 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 Equation (3)). An appropriate value for k was found experimentally at 0.04. The probability value $p_g^{DC}(x_i)$ of a potential boundary with position $x_i \in [0, 1]$ for boundary type g is calculated by

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

$$\sigma_g^2 = k \cdot \frac{1}{N_g} \qquad (3)$$

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

Figure 3. Probability distribution curves for the date type C-A-M for the boundary between cipher and artefact (a) and the boundary between artefact and month (b).

3.5 Generating Hypotheses

Since the type of a date is unknown, for each of the four possible types, containing $k = 1 \ldots 3$ word bounda-

ries, $\binom{N_{pb}}{k}$ 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 $1 \leq x < y < z \leq N_{pb}$.

$$p(h_i) = \frac{p(b_x)[+p(b_y)[+p(b_z)]]}{k_i} \qquad (4)$$

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

For the best four hypothesis the resulting objects are created which are potential ciphers or words by assigning the strokes to those objects.

4 Results

The date in a church register is user identified by marking the beginning and end of the date entry. We created a database with 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. They were made by six different writers. The following information is generated for each selected date from our preprocessing step:

- Skeletons of stroke segments (between stroke crossings or ends) that are the certain or the 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.

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. 10 runs with different training and test sets resulted on average in 97% of the correct word boundary combinations being included in the best four hypotheses (see Fig. 4). This is in the range of results of techniques which require gaps for detecting word boundaries (e. g. 87.6% in [7], 95.6% in [8]). 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%.

COMPUTER SOCIETY

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 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 [9] might solve this problem.
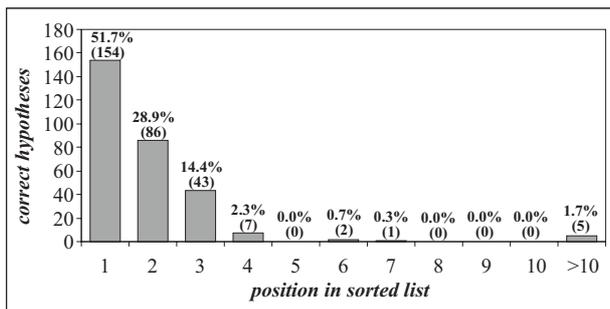
Figure 4. Positions of 298 correct hypotheses in sorted list related to their probability.

5 Conclusions

Data-driven methods, which only analyse local features, are not efficient for old scripts such as in historical church registers. The use of additional knowledge about the given structure of the word sequence improves the segmentation. We present a method for detecting and segmenting the date in entries of historical church registers. We considered scripts of which lines and word boundaries are not characterised by obvious gaps as well as scripts containing word touchings. This was achieved by analysing the positions of boundaries between the words in word sequences with a limited number of variations. The technique was demonstrated by segmenting of words in date entries from church registers.

Currently, our algorithm uses only the certain parts of line. We plan to improve the segmentation by considering the parts which are classified as the potential part of line and are close to the certain parts. Furthermore, tests with other distance measurements (such as in [9]) could show a better performance.

The output of the algorithm are hypotheses about potential cipher or word objects which can be used as the input of a suitable 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] 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.

[3] G. Kim and V. Govindaraju. Handwritten Phrase Recognition as Applied to Street Name Images. *Pattern Recognition*, 31(1):41–51, January 1998.

[4] G. Kim, V. Govindaraju, and S. N. Srihari. An Architecture for Handwritten Text Recognition Systems. *International Journal on Document Analysis and Recognition*, 2(1):37–44, February 1999.

[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] 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.

[7] 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.

[8] 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.

[9] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41–52, January 1994.

[10] M. Shridar and F. Kimura. Segmentation-Based Cursive Handwriting Recognition. In H. Bunke and P. S. P. Wang, editors, *Handbook of Character Recognition and Document Image Analysis*, pages 123–156. World Scientific, February 1997.