Mining of words from undersegmented word images using holistic matching and tree-based search

(Bachelorproject)

Tom Gankema, s1875892, t.gankema@student.rug.nl,
Lambert Schomaker[*]

April 17, 2013

Abstract

To be able to recognize handwritten text, the text needs to be segmented. Without any recognition, errors are unavoidable during that segmentation, whereby images with multiple words can appear. Because these images do not contribute to create better word models, it is desirable to split these images. In our study we looked into undersegmented word images which are already transcribed. With a number of basic constraints it was possible to 'mine' new word instances in case one of the words in a multiple-word image did not have a model yet. This was achieved by building a segmentation graph containing all possible combinations of connected components that could lead to the pattern of the undersegmented image. Then a graph search is used to find the most likely sequence of wordzones in the undersegmented image. The study focused on both the development of this method as well as on creating a heuristic for handling new words.

1 Introduction

The recognition of handwritten text is a task which generally takes little effort for humans, regardless of the number of writers or the style of the handwriting, but automatic handwriting recognition (HWR) is up to this moment not close to this level. The difficulty of the task lies in the fact that the computer needs to find and recognize highly deformable patterns with no, or arguably very little, understanding of the meaning of these patterns. The consequence of this is that what seems to be obvious to humans may still be very challenging for computers.

Two methods are very common for the recognition of handwritten text, these are the analytic method and the holistic method (Srihar et al., 2007). The former treats a word as smaller sub units, such as characters, and is also known as the sliding-window technique. The latter is based on the general shapes of words and has more common with the way humans seem to be reading text, according to psychological studies of human reading (Madhvanath and Govindaraju, 2001; Serrano et al., 2007). In general it is true that analytical methods are more succesful for online HWR and holistic methods for offline HWR.

Online HWR is the name for recognition of handwriting that takes place on-the-fly and therefore has access to information about the temporal order of the handwriting. Offline HWR typically refers to the recognition of handwritten text in scanned documents and so has less information available than the online variant (Plamondon and Srihari, 2000). For the offline recognition of handwritten texts usually comes the need of preprocessing the handwritten documents, so that the relevant information is retained and is segmented in such a way that the recognition system should be able to recognize the segmented text. However, if the segmentation is not done correctly, this may cause problems later on in the recognition phase.

To solve the problems of segmentation, a lot of different segmentation methods for offline HWR have been proposed (Papavassiliou et al., 2010; Casey and Lecolinet, 1996), but so far no segmentation method has been found that can segment every

[*] University of Groningen, Department of Artificial Intelligence

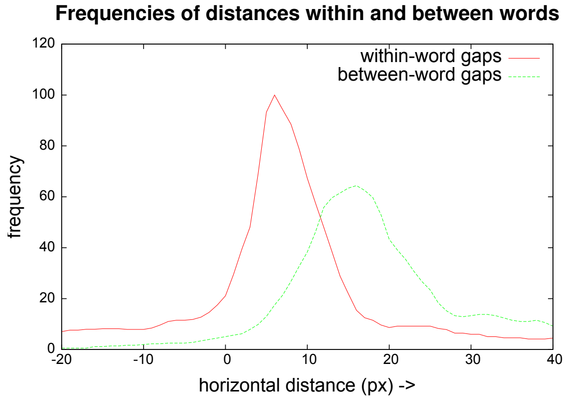

Figure 1: Comparison between frequencies of gaps within words and between words of a selection from the same dataset as used in our study. From this graph it can be derived that a perfect bottom-up segmentation is impossible (created by Dimitri Vrehen in an earlier study).

text into the wanted sub units. One of the reasons such a perfect method has not been found yet is that recognition seems to be needed to correctly segment a text, because the distances within words can be larger than the distances between words, as is shown in Figure 1. But the opposite is also true, segmentation seems to be needed to be able to recognize at all. This is an example of a classic chicken-and-egg problem. Recognition based segmentation methods try to get around this problem, by implementing some recognition in the segmentation phase, but for a holistic approach this would be a hard and very computationally intensive task (Lu and Shridhar, 1996).

1.1 Problem definition

Suppose we have a collection of preprocessed images containing handwritten words, these images are segmented and some segments are transcribed. Then a learning mechanism is applied to transcribe more of the word images based on the available transcriptions and thereby creating better prototypes of the words. This learning happens in a continuous process.

After the segmentation one would wish to only have images containing single words, but in reality two things can go wrong in the segmentation process. The segments may be segmented too thorough, resulting in images with less than one word. And some segments may be segmented a bit to loose, with the result of images containing multiple words (Louloudis et al., 2009). These respectively over and under segmentated images are a problem, because they can not be used to create better prototypes of these words and therefore are not helpful in the learning process.

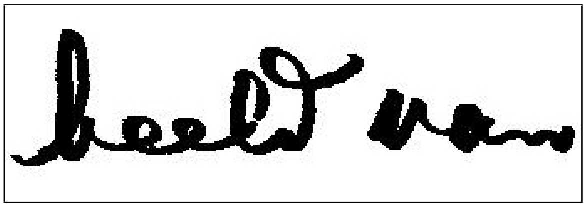

Figure 2: Example of an undersegmented image, with an unknown word (beeld) and a known word (van), the goal is to datamine the word 'beeld' by cutting it in between the two words using the prototype model of the word 'van'.

The purpose of this paper is to propose a method that is able to datamine new words from undersegmented word images. These images are already transcribed with an underscore between the words, but not all transcribed words have a prototype model available. In the simplest case one has an image with two words, of which both are transcribed, for example 'the_queen'. But only one of these words already has a prototype model, let's say 'the'. The idea is to use this model of the word 'the' to extract the unknown word 'queen' from the undersegmented image, by basically cutting the image between the words. This has the positive side effect that also more instances of known words will become available, resulting in better performance. In some way this word splitter is similar to a recognition based segmentation as is used in some character segmentation methods (Daifallah et al., 2009). The most important difference is that now the undersegmented image is already transcribed, which makes it a lot less computationally intensive than such a method usually would be on the word level.

1.2 Segmentation graph

Both in speech and handwriting recognition, graphs are widely used as an intuitive data structure to recombine segments from a segmentation process.

In speech recognition these segments are parts of the speech signal, whereas in HWR these segments can be words, characters or parts of characters, but in all cases the general concept of recombining segments using a graph is the same (Sternby and Friberg, 2005; Ortmanns et al., 1997). These graphs usually hold all possible combinations of the segments and a search is done to find the best path or the n-best paths through this graph. The method proposed in this paper makes use of such a segmentation graph.

Although graphs are used a lot for these kind of tasks, it is not always straightforward how to use them. One of the main questions when implementing a graph structure is which combinations of segments are allowed in the graph and which are not. In the case of handwriting recognition one obvious constraint is the direction of reading, for western languages usually only connections from left to right would be possible. But questions may arise. For example one can argue about whether it should be possible to jump backwards in the image. This may seem strange, but such an alleviation of constraint may improve the performance since two consecutive words can overlap (Mahadevan and Nagabushnam, 1995). Although less controversial, a similar discussion can arise about the amount of space that should be allowed between two consecutive words. Both constraints are reflected by a parameter in the system and will be explored in this study to be able to find the most optimal settings for the envisaged word splitter.

Another main question is whether to use probabilities or distances as scores in the graph and how to propagate them. The most accepted way of dealing with this problem is to use a multiplication of probabilities according to Bayes' theorem (Malakoff, 1999). However there are arguments against such a conjunctive rule. For example a recognition algorithm may return a probability close to zero because of a low likelihood hypothesis in a chain of evidence. This may result in ignoring the segment altogether, which is not desirable. An addition of probabilities will likely score better in such cases, because it is less restrictive on the collection of evidence. As for distances, it is certainly more intuitive to add values than to use multiplication. These hypotheses are tested, combining both probabilities and distances with both the discussed propagation methods.

Finally, because the goal of this study is to find new words in a datamining process, it is also important how to handle the words of which no word model is available yet. In this study a heuristic, which maximizes the probabilities and minimizes the distances of segments with a width close to the expected width of a word, is developed and tested.

2 Methods

2.1 Data and preprocessing

The data is drawn from a collection of scanned (300 dpi) historical handwritten documents from the Dutch National Archive, all written by a single writer. These documents are preprocessed and segmented according to the methods described in (Schomaker, 2008). All instances of the data that are selected for this study are already transcribed by humans and contain multiple words, most images contain only two words, but also cases of images containing three or four words do occur in this selection. For a data example see Figure 2.

Then, before building the segmentation graph, the undersegmented images are further cut into segments using connected components. The result of this process are wordzone images, all of which contain a single connected component. These images are temporarily stored along with their original position in the image and the width and height of the wordzone images self. Also all images consisting of multiple connected components (up to a maximum of nine) are stored in the same way, making sure there always is an image containing a single full word, assuming the connected components algorithm cut the original image at least somewhere between the words and does not consist of more than nine connected components.

2.2 The word splitter

The number of words which can be in the wordzone images is limited to the number of words from the label of the original image. Therefore it is now possible to match the words from the labels with all wordzone images using a recognizer. The recognizer that is used for this study is based on a Kohonen self-organized map and returns a distance measure. This distance has a value between 0 and

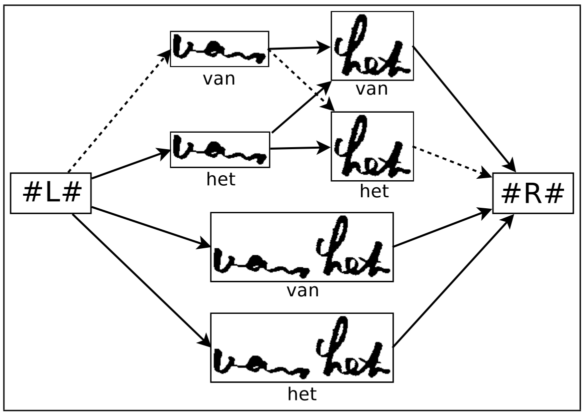

Figure 3: Basic example of a graph of an image labeled 'van_het' with only two connected components, showing all nodes and all edges in the graph. The correct path is indicated by the dotted arrows.

1 or a non-match value and can be converted into a measure that has similarities with probability, using e^{-d}, where d is the distance measure returned by the Kohonen SOM. If $d = 0$ then $e^{-d} = 1$ and if $d \to \infty$ then $e^{-d} \to 0$, which is desirable. In the remainder of this article, this measure is meant when referring to pseudo-probability.

These scores are used when building the graph. Each node in the graph represents a combination of a wordzone image, a word that could be in that image and the score returned by the recognizer. Then all possible combinations of two word nodes in the graph are considered and, if an edge between these nodes is possible according to the specified rules, this edge is added to the graph. This way a graph, as in Figure 3 is built.

After the graph is created, all paths from the start node to the end node are evaluated, using a depth first search. It will calculate the cumulative path quality by propagating (or sometimes called aggregating) the scores in the wordnodes, this process is referred to as 'propagating scores' in the remainder of this article. Eventually the path with the best score is returned. This path should consist of the nodes containing the words from the input label and the correct corresponding wordzone images. For words of which no word model is available a heuristic is used which estimates whether it is plausible that a word is in a certain wordzone image. This heuristic calculates the probability that a word with a certain amount of characters is in a wordzone image with a certain width. It does so by estimating the width of the word according to Formula 2.1. The probability that a word with this expected width is in a wordzone image can then be calculated by comparing it with the actual width of the wordzone image, assuming a discrete normal distribution. If the probability exceeds a certain threshold, the node in the graph gets assigned a score of 1 when using pseudo-prbabilities and a score of 0 when using distances and vice versa if the threshold is not exceeded.

Formula to calculate the expected width of a word for the heuristic for unknown words. W is the width of the total image, N the total number of characters in this image, N_{spaces} the total number of spaces in the image, W_{space} the average width of a space and N_{chars} the total number of characters of the new word.

$$W_{expected} = \frac{W-(N_{spaces}*W_{space})}{N} * N_{chars} \tag{2.1}$$

2.3 Experiments

The graph search will be in a free modus for all experiments, except for the last one. This means that, during the search, the depth of a path through the graph will not be constrained to a minimum or maximum and it is not checked whether the words in the nodes are correct according to the label of the image. This to avoid ceiling effects in the performance evaluation. In these tests a path is considered correct if the output is equal to the input label. It must be noted that this is not always strictly spoken true, because theoretically it is possible that the correct output label will be given with the incorrect corresponding wordzones, although this will rarely occur. Even so, the free modus will always result in an equal or worse performance compared with the full modus, because with the described constraints only paths which would result in an incorrect answer will be removed.

The first goal is to find the best method to propagate scores along the graph. These scores are distances or pseudo-probabilities and will be propagated by multiplication or addition. All four combinations of methods and scores are considered, because the scores are expected to influence the way of propagating. In case of the addition method, the final scores are divided by the number of nodes in the path to normalize the output scores between 0 and 1. For this experiment only instances of which all words have a prototype model available will be used. The method with the best recognition performance will be used in further experiments.

One of these next experiments will be about the possible connections in the graph. First, the maximum gap, which determines the maximum number of pixels from left to right between two consecutive segments, will be under study. Afterwards a maxiumum overlap, which determines the maximum number of pixels two consecutive segments may overlap, is tested. To avoid loops in the graph, it is at all times ensured that the successive segment ends on a larger x-position than its predecessor. And as for the propagation experiment, only instances of which all words are known by the system are used.

The next experiment concerns a heuristic for unknown words (UWH), this experiment will evaluate two variables. The first one is a deviation which is estimated by using a percentage of the expected width. This estimation is used instead of a calculation of the standard deviation, because to calculate the standard deviation, the widths of the individual words need to be known, which is exactly what we are looking for in this study. The second variable is a threshold, if this threshold is exceeded, the wordzone image will be accepted as possibly containing the unknown word and gets assigned a score as explained in section 2.2. For this experiment exactly one word model will be ignored from each example randomly, thereby mimicking the behaviour as if one of the words is unknown. The results are compared with a control experiment in which instead of the UWH, random scores between 0 and 1 are assigned to nodes with unknown words.

Finally an experiment will be performed to measure the exact performance of the system for datamining new words. For this last experiment more constraints are added during the search, using the number of words to determine the minimum and maximum depth of a correct path through the graph and checking for each node whether the word in the node is the same as in the word in the label at the corresponding depth in the graph. This way the search space can be limited considerably. The method can be seen as a graph pruning during the search, cutting all of the paths which do not satisfy the constraints. For this experiment, exactly one word model will again be ignored randomly from each instance and a path is only considered correct if it does not just return the correct output label, but also the correct corresponding wordzones, to check this the output wordzones are checked by hand.

3 Results

3.1 Propagation of evidence

Each of the four methods of propagating scores along the graph was tested on the same dataset with 861 instances, the results are shown in Table 1. The values of the maximum gap and maximum overlap were both chosen (based on a pilot experiment) at 40 pixels. Addition of pseudo-probabilities had the highest recognition performance (71.1%) and therefore this method is used in the experiments of Sections 3.2, 3.3 and 3.4. A χ^2-test was done on the frequency counts of correct and incorrect classifications and shows a strong effect of the propagation method on the classification performance ($p < 10^{-6}$), but shows no significant effect ($p > 0.25$) of the unit of similarity on the classification performance.

Table 1: Percentages of correctly classified instances and their 95% confidence intervals for the 4 propagation methods (N=861).

propagation	unit	
	pseudo-probabilities	distances
multiplication	56.4 % (53.1%–59.7%)	60.3 % (57.0%–63.6%)
addition	71.1 % (68.1%–74.1%)	70.9 % (67.9%–74.0%)

3.2 Maximum gap and overlap

The maximum number of pixels between two consecutive segments was tested by evaluating it on a test set with 93 instances, using an addition of probabilities to propagate scores along the graph.

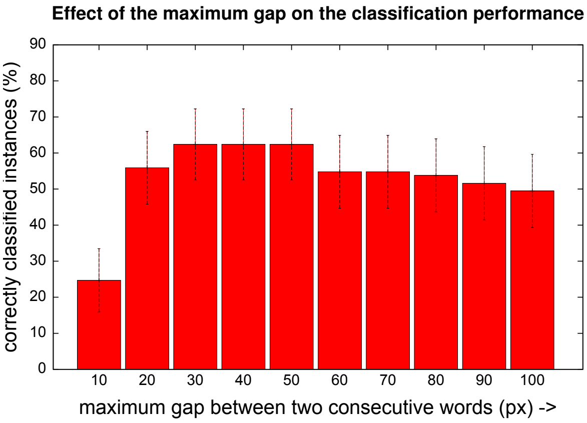

Figure 4: Performance of the word splitter for a different maximum gap between two consecutive words.

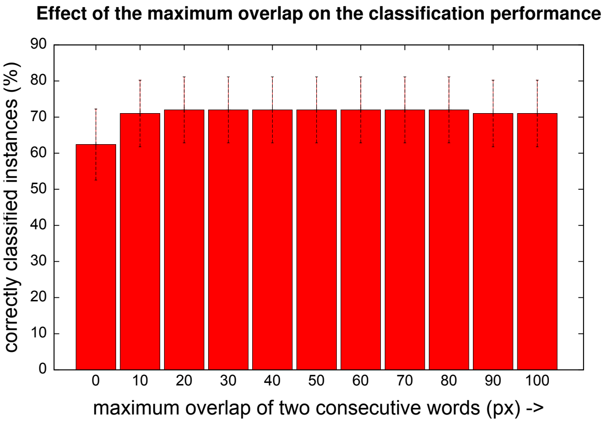

Figure 5: Performance of the word splitter for a different maximum overlap of two consecutive segments.

Figure 4 shows the results of this experiment for different values of the maximum gap, and indicates an optimal value for the parameter.

The maximum number of pixels two consecutive segments may overlap was tested in a similar way on the same dataset. Figure 5 shows the results of this experiment and indicates that a value larger than zero can improve the performance.

3.3 UWH

Table 2: Performance of the word splitter when one word model is ignored in each instance, varying the parameters of the UWH.

dev	threshold				
	10^{-5}	10^{-4}	10^{-3}	0.01	0.05
0.01	21.5%	23.7%	22.6%	21.5%	24.7%
0.03	20.4%	28.0%	37.6%	46.2%	39.8%
0.05	11.8%	39.8%	52.7%	41.9%	39.3%
0.1	5.4%	34.4%	40.9%	34.4%	19.4%
0.2	7.5%	25.8%	10.8%	7.5%	7.5%

The heuristic to classify unknown words was tested on a test set with 93 instances, for each instance one of the word models was ignored at random. The tested parameters were a deviation as rate of the expected width of a word and a threshold which influences the strictness of the heuristic. The highest measured classification performance was 52.7% (95% confidence interval ±10.15). This was reached by a deviation of 0.05 (5 percent of the expected width) and a threshold of 10^{-3} as is shown in Table 2. When instead of the UWH a random number (between 0 and 1) was generated and assigned to the node with the unknown word, the classification performance on the same test set was 12.9% (95% confidence interval ±6.81).

3.4 Datamining new words

For the last experiment a test set of 500 instances was used and more constraints were added to the search comparing to the previous experiments as described in the Methods section. One word model was again ignored from each instance randomly and scores were propagated through the graph by addition of pseudo-probabilities. The maximum gap and maximum overlap were both set at 40 pixels and the UWH was used with a deviation of 0.05 and a threshold of 0.01.

All wordzone images were checked by hand to see whether the word that was written on the image was what the word splitter said that was written on it. 436 out of the 500 instances were splitted correctly, which is a classification performance of 87.2% (95% confidence interval ±2.9). From the 64 that were not splitted correctly, 44 times it was caused by a connected component consisting of multiple words, so an oversegmentation would be required at those points, 11 times it was caused by

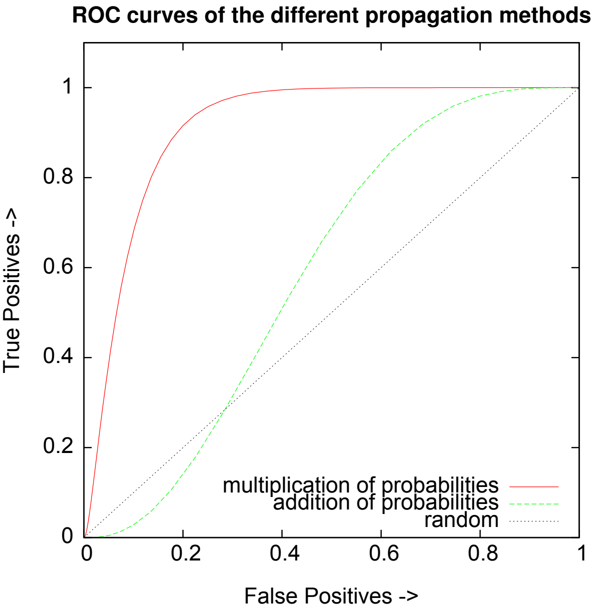

Figure 6: ROC curves of both multiplication and addition of pseudo-probabilities. The area under the curve of the multiplication method is 0.91 and of the addition method 0.59.

an error of the UWH and 9 times there was another cause, for example inaccurate word models.

3.5 Additional analysis

Additional analysis on the results in section 3.1 show there is a noticeable difference between the ROC-curves of the different propagation methods. These ROC-curves are shown in the graph of Figure 6. The area under the curve (AUC) of the multiplication method is 0.91 and the AUC of the addition method is 0.59. The latter is comparable to the AUC scores of both multiplication of distances (0.62) and addition of distances (0.65) which are not shown in the graph.

4 Discussion

4.1 Propagation of evidence

The experiments regarding different propagation methods show, contrary to expectations, that there is no significant difference between the use of pseudo-probabilities or distances. This is remarkable because of the fundamental differences between distance and probability measures. But can possibly be explained by the fact that the pseudo-probability measure that is used here is not a proper probability measure, but a measure derived from the distance measure. The small performance differences between both methods can possibly be explained by the fact that both measures are so closely related.

However, the results do show a significant difference between the multiplication method and addition method. If scores are added, higher recognition rates are measured than when scores are multiplied. This corresponds with our hypothesis that noise in one of the segments has a larger influence on the classification when multiplication is used than when addition is used. And argues against a conjunctive rule such as provided by Bayesian theory.

However a remark has to be made regarding these results, because additional analysis of the results show that the area under the ROC-curve (AUC) of the multiplication of pseudo-probability method is much larger than the AUC's of the other methods. A large AUC means a high true positive rate and a low false positive rate. The true positive rate in the ROC-curve is equal to the surface under the normal distribution of the scores of correct classifications up to a certain threshold. The corresponding false positive rate is equal to the surface under the normal distribution of the scores of incorrect classifications up to the same threshold (assuming that correct classifications have lower scores than incorrect classifications, otherwise the surfaces up from a threshold need to be calculated). Therefore the true positives are a rate of the total amount of correct classifications and the false positives are a rate of the total amount of incorrect classifications. In the situation of section 3.1 all classifications are accepted and therefore these methods have a true positive rate of 1 and also a false positive rate of 1. The multiplication of pseudo-probabilities method has a relatively high true positive rate and a relatively low false positive rate for the most optimal threshold. This means that the normal distributions of scores of correct and incorrect classifications show little overlap and that it is easier to distinguish between correct and incorrect classifications when using this method than when using the other methods. Therefore a threshold, above or under which classifications are rejected, would improve the classification performance of the multiplication of pseudo-probabilities method more than it would improve the other methods. How large this improvement will be is still to be studied.

4.2 Maximum gap and overlap

The parameter sweeps over the parameters which control the maximum allowed gap between words and the maximum allowed overlap of words show that both parameters have an influence on the classification. The maximum gap parameter is an important parameter in the system, because the performance varies significantly over different values of the parameter. But the parameter is not sensitive as the highest recognition rates are stable between values of 30 to 50 pixels. At higher values of the parameter the performance gradually drops, which does not mean that gaps wider than 50 pixels between two consecutive words do not occur. But allowing connections between two words which had more than 50 pixels space between them lead more often to an incorrect classification than to a correct classification.

Also it appears that the parameter controlling the maximum overlap of two consecutive words is less important than the parameter controlling the maximum gap, but it is still true that allowing overlap of words can improve the performance. However it can not be stated that it always will, as the differences are not significant. Figure 7 shows an example containing the overlapping words 'de Tweede', which will only be classified correctly if overlap of two consecutive segments is allowed. When not allowing overlap when classifying this example, the result will at best be that either the right part of the 'e' from the word 'de' will be ignored or that the left part of the 'T' of the word 'Tweede' will be ignored. This example shows that allowing overlap can lead to an increase of the correct classifications and advocates to allow overlap.

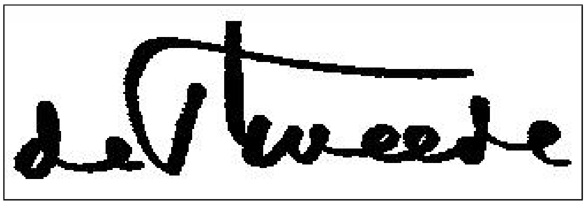

Figure 7: Example with the words 'de' and 'Tweede'. This image will only be classified correctly by the system if overlap of two consecutive words is allowed.

4.3 UWH

The heuristic that is used in our study to classify unknown words uses a very simple strategy to classify unknown words based on the widths of image segments. Yet the performance gains from this heuristic is already eminent compared to when random scores are assigned to unknown words. This shows that even a simple heuristic can improve the system significantly. The UWH does however also introduce two extra parameters into the system and does not yet take into account that not all characters are evenly wide. Therefore a more advanced method may improve the system even more.

Such an advanced method could for example calculate the average widths of individual characters based on gathered information about words and their widths. If there is enough data available it is possible to calculate the average widths of all characters. If for example the average width of the words 'abc' and 'abcd' is known, it is possible to determine the average width of the letter 'd'. That information can in turn be used to get more information from the remaining data. If such methods are applied in a smart way, one can make an accurate approximation of the average widths of individual characters. That information can then contribute to a better heuristic for unknown words.

4.4 Datamining new words

The experiment regarding the classification of unknown words shows that it is possible to find words in undersegmented images using the developed system. All correct classifications did not only output the correct expected words but also the correct unknown corresponding word images. That result is relevant because it concerns words that are not yet known by the system and enables the system to learn these words. The described word splitter has similarities with recognition based segmentation methods in which it is not necessary that all words are recognized by the system to achieve a cor-

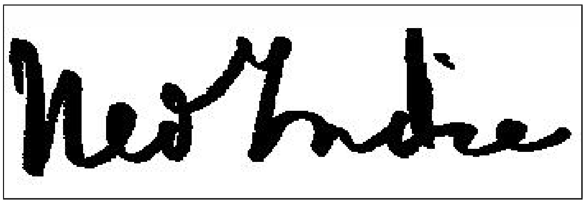

Figure 8: Example with the words 'Ned' and 'Indie' consisting of a single connected component.

rect segmentation. However on the word level this was usually considered computationally too heavy. By making use of the transcription of the words and applying an appropriate heuristic, our study has shown that it is very well possible to apply the same principles on the word level to learn unknown words.

Of the errors in this experiment most were caused by a connected component consisting of multiple words. To be able to classify these images correctly it would be necessary to split words on other criteria than just the connected components. In other words the images need to be oversegmented to avoid these errors. Figure 8 shows an example of such an image.

Other errors by the system were caused by the UWH that sometimes assigned a high recognition score to a segment which did not contain the (entire) correct word while at the same time it did assign a low recognition score to the correct segment. Figure 9 shows an example of such an error. In that example the UWH expects the word 'eervol' to have a smaller width than the actual width of the word, which in this case causes an error because the leading 'e' of the word 'eervol' is smaller than the maximum gap that is allowed by the maximum gap parameter. Possibly would the suggestion in section 4.3 to improve the heuristic lead to a decrease of this type of errors.

The remaining errors had an undefined cause, this includes errors following from the word models and the Kohonen SOM recognizer, such as errors caused by inaccurate word models. The exact cause of these errors is hard to trace back, because the recognizer that is used is a neural network based classifier.

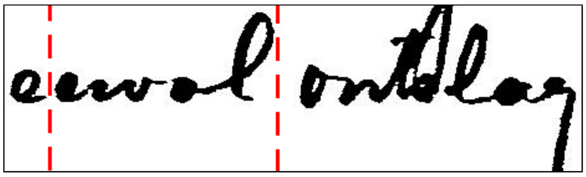

Figure 9: Example with the words 'eervol' and 'ontslag'. If the model of the word 'eervol' is unknown in this example, the UWH recognizes the word 'eervol' as indicated by the dotted lines, which is incorrect in this example.

4.5 Conclusion

Our study shows that it is possible to find new words from undersegmented images in a datamining experiment. These results can help to improve recognition rates of handwriting recognition systems, because it makes it possible to gather more examples of known words as well as to learn new words from existing data. The described method avoids the problems following from undersegmentation under the limitation that the text image is already transcribed. With this method it is also possible, although not yet practically tested, to split longer sequences of words. Furthermore is it theoretically possible to apply the principles of the system on character level as well.

References

R.G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):690 –706, jul 1996.

K. Daifallah, N. Zarka, and H. Jamous. Recognition-based segmentation algorithm for on-line arabic handwriting. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 886–890, jul 2009.

G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition*, 42(12): 3169–3183, 2009.

Y. Lu and M. Shridhar. Character segmentation in handwritten words an overview. *Pattern Recognition*, 29(1):77–96, 1996.

S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):149 –164, feb 2001.

U. Mahadevan and R.C. Nagabushnam. Gap metrics for word separation in handwritten lines. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 124–127, 1995.

David Malakoff. Bayes offers a 'new' way to make sense of numbers. *Science*, 286(5444):1460–1464, 1999.

S. Ortmanns, H. Ney, and X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech & Language*, 11 (1):43 –72, 1997.

V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis. Handwritten document image segmentation into text lines and words. *Pattern Recognition*, 43(1):369–377, 2010.

R. Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):63 –84, jan 2000.

L. R. B. Schomaker. Word mining in a sparsely labeled handwritten collection. In Berrin A. Yanikoglu and Kathrin Berkner, editors, *Proceedings of Document Recognition and Retrieval XV, IS&T/SPIE International Symposium on Electronic Imaging*, pages 6815 –6823. SPIE, 2008.

J.I. Serrano, A. Iglesias, and M.D. del Castillo. Modeling human reading in conceptual networks for text representation and comparison. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 613 –618, aug 2007.

Rohini Srihar, Shravya Shetty, Sargur Srihari, Rohini K. Srihari, Shravya Shetty, and Sargur N. Srihari. Use of language models in handwriting recognition, 2007. http://www.cedar.buffalo.edu/~srihari/papers/TR-06-07.pdf.

J. Sternby and C. Friberg. The recognition graph - language independent adaptable on-line cursive script recognition. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, volume 1, pages 14 – 18, aug 2005.