

From Handwriting Recognition to Ontologie-Based Information Extraction of Handwritten Notes

Marcus Liwicki¹, Sebastian Ebert^{1,2}, and Andreas Dengel^{1,2}

¹ DFKI, Trippstadter Str. 122, Kaiserslautern, Germany
Firstname.Lastname@dfki.de

² University of Kaiserslautern, Germany

Abstract. In this paper we discuss recent development of handwriting recognition (HWR). In particular, the transition from pure handwriting recognition to understanding of the handwritten notes is described. Therefore we first summarize the state-of-the-art in HWR. Next, two recent approaches in order to improve HWR and extracting knowledge are described. Experimental results on various data are reported and an outlook to future directions is given.

Keywords: OBIE, handwriting recognition, information extraction, knowledge management, personal information model

1 Introduction

Handwriting recognition (HWR) has been the topic of research for many decades. While the first recognizers have been developed for isolated characters or digits, later recognizers focused on complete words or even sentences [3, 11, 15]. Nowadays there exist solutions which have a quite good recognition performance (e.g., recognizers from Microsoft[©] and Vision Objects[©]).

However, the task of handwriting recognition cannot be assumed to be solved already. There is still room for improvement for the recognition performance, as well as handling different scripts and special environments. Currently, much research effort goes into the direction of improving recognizers in these use cases [1]. This approach will be shortly described in the first part of this paper (Section 4).

A second more recent approach is to understand the meaning of the written content instead of just recognizing the handwritten text. For many applications not only the ASCII transcription, but also the important content and concepts are of interest. This can be used to categorize the document or even to relate it to other documents and known concepts in the knowledge space of a person or a company. Section 5 will focus on the processes which are applied in order to understand the handwritten text.

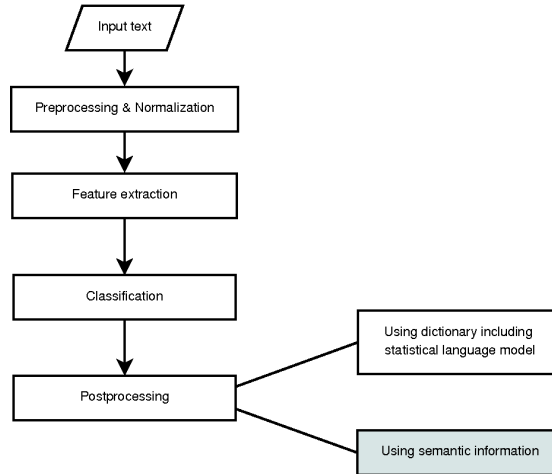


Fig. 1. General handwriting recognition and our main contribution: We include semantic information into the recognition process

2 State-of-the-Art

This section gives an overview of the handwriting recognition system in general and the main contribution of Section 4. The main steps performed in handwriting recognition are illustrated in Fig. 1, they consist of preprocessing, normalization, feature extraction, classification, and finally a postprocessing.

Preprocessing is the first step in the handwriting recognition system where the noise associated to the sample input is eliminated. This step often comprises line extraction, and sometimes word separation and character segmentation, depending on the recognition task. However, character segmentation is a very difficult problem. It is not possible to segment a word into characters before recognizing this word and on the other hand side the word can't be recognized correctly before being segmented into characters. This situation is known as Sayre's paradox [14].

Normalization decreases the effect of various writing styles by normalizing the input handwritten data. It can also be considered among the previous step. In normalization, the characters' skew, slant, height and width are adjusted.

Feature Extraction acquires the set of feature vectors from the input sample. This particular step is needed because the classifier usually needs numerical values as an input instead of using the raw point-sequence data.

Classification is the process where the feature vectors are fed to classifiers like Hidden Markov Models (HMMs) and Neural Networks (NNs) to obtain recognition candidates. Often, multiple alternatives are provided by the recognizer together with a recognition probability.

Postprocessing comprises several steps which can be performed on the recognizer's output. Very often word lexicons or even grammars are used to improve the recognition result.

We use the *Microsoft Handwriting Recognizer*³. This recognizer extracts some online and offline features from oversegmented characters and applies TDNN classifier for the recognition. Dictionary information is integrated by using a trie-based approach. For more information about the recognizer, refer to [10].

The contribution of Section 4 is to enhance the postprocessing by the integration of semantic information. The semantic information is extracted from the Personal Information Model of the user (PIMO) present in his or her Semantic Desktop. More specific information about this approach is given in the next section.

Word spotting is the task to find a given word in a handwritten text [8]. Usually, the word is presented as a query of the user who wants to find those places where the specific word appears. At first glance word-spotting seems to be similar to ontology-based information extraction. However, in the latter case there is not only a single query. Ontology-based information extraction tries to find instances in an ontology. Furthermore, we do not apply just a search algorithm, but we furthermore take relations between the concepts in the ontology into account.

Document retrieval from handwritten documents became more and more popular in the last years. Here the task is to find (retrieve) or classify a given set of documents [9]. Even if also some kind of information extraction is performed, this topic is only loosely related to the topic addressed in this paper, since we do not consider whole documents, but entities appearing in those documents.

3 The Personal Information Model - A Way to Represent a User's Knowledge

The Semantic Desktop [4, 5] is a means for personal knowledge management; it builds the personal Semantic Web on desktop computers. The consistent application of Semantic Web standards such as the Resource Description Framework⁴ (RDF) and RDF Schema⁵ (RDFS) provides the identification of digital resources, i.e., text documents, e-mails, contacts, multimedia files, by unique URIs, across application borders. In contrast to current limitations in file and application based information management, the user is able to create his or her own classification system which reflects the way of thinking; it consists of projects, people, events, topics, locations, etc. Furthermore, the Semantic Desktop enables the user to annotate, classify and relate all resources, expressing his or her view in a *Personal Information Model* (PIMO) [13]. Figure 2 illustrates an extract of

³ The Microsoft Windows XP Tablet PC Edition SDK for our experiments. It is available for download at

<http://www.microsoft.com/windowsxp/tabletpc/default.msp>

⁴ <http://www.w3.org/RDF/>

⁵ <http://www.w3.org/TR/rdf-schema/>

Based on all available information, the dictionary is created once and is used for all handwritten phrases disregarding their specific topic. The dictionary is created as follows:

1. A graph (RDF-graph⁶) of the PIMO is extracted.
2. All textual information from the RDF statements are selected.
3. The texts contained in objects beyond these relations (electronic documents) are added to these information.
4. Finally, the dictionary is composed from the n most frequent words of the resulting text corpus.

Note that the static approach is similar to a database-driven recognition approach, where a database of the topic is at hand. However, using a PIMO is a broader approach, because all information is stored across conventional application borders. A typical database has no information about a persons contacts and the bookmarks in a web-browser, while this information is available in a well-structured PIMO.

4.2 Dynamic Approach

The dynamic dictionary also takes the topic of the input data into account. We perform a navigation through the RDF graph. The starting point is the object in the PIMO (the *main thing*) where the handwritten annotations are related to. Often this object can be easily determined. In the case of annotating or reviewing a document, for example, it will be the electronic document. In the case of meeting notes, it will be the project or the topic of the meeting. In the PIMO each object is identified by a unique URI, so we start at the URI of the main thing.

The algorithm is then similar to a breadth first search in the graph domain where the edges are given by connector relations (relations that connect topics to related ones). The depth of the search is a parameter which will be investigated in our experiments. The algorithm works as follows:

1. An RDF-graph is extracted.
2. Starting from the main thing, find all concepts related to that thing by connector relations.
3. Repeat Step 2 until the desired depth is reached.
4. All textual information from the RDF statements are added to the vocabulary.
5. The texts contained in objects beyond these relations (electronic documents) are also added to these information.
6. Finally, the dictionary is composed from the n most frequent words of the resulting text corpus.

Table 1. Performance of the approach

Doc. Mode	Dynamic Dict.		Static Dict.		Default
	AD	UD	AD	UD	
d1 Line	77.9	76.2	77.2	74.5	69.7
d1 Word	76.6	76.3	75.8	74.0	72.9
d2 Line	81.5	81.0	79.7	75.9	72.2
d2 Word	81.3	79.4	76.7	75.2	74.8

4.3 Results

In this section we will summarize the experiments performed on realistic annotations of a person using the semantic desktop and his own PIMO. Table 1 shows the results which correspond to the second task in [1]. In these experiments we have also tested the recognizer on the word basis, i.e., without linguistic information. This is motivated by the fact that very often real handwritten annotations make not much sense from a linguistic point of view (often they contain just one or two words have been written as annotation). These results are averaged over the writers. The depth for the dynamic dictionary is fixed to 4. Without the use of any semantic information the Microsoft recognizer performs better in the word-level task than in the line recognition task. This supports the assumption that the annotations make not much sense.

Using semantic information was always very useful and lead to a significant improvement of the recognition rate. On the text line level, the absolute improvement of the recognition accuracy is more than 8% which is statistically significant. On the word level the recognition rate increases by about 4% .

It is an interesting observation that the application dictionary mode performed better in these experiments. In this mode only the extracted wordlist was used as a dictionary. This also can be explained by the fact that annotations usually tend to be shorter than complete notes, because they often only reflect a short anchor or reminder for the writer.

Another interesting result (not given in the table) is that the recognition accuracy on the text line level for the original writer increased by about 15%. On these real annotations the Microsoft recognizer only performed with 75%, but the final recognizer achieved more than 90%.

5 Ontology-Based Information Extraction from Handwritten Text

In the previous section, semantic information has been used in order to improve the handwriting recognition process. In this section we go one step further. Instead of just recognizing the handwritten text, we try to understand the meaning of the written content. For many applications not only the ASCII transcription, but also the important content and concepts are of interest. This can be used

⁶ the Resource Description Framework is described in <http://www.w3.org/RDF/>

to categorize the document or even to relate it to other documents and known concepts in the knowledge space of a person or a company.

Considering the process of note-taking, for example, the person would write down newly acquired knowledge about instances which might appear already in his or her personal knowledge space. Our proposed system can extract the information and identify the new knowledge based on the written content. Finally, the user just needs to check the correctly identified information. This would decrease the work-load of the person significantly, because usually these information has to be typed into the computer and formalized manually.

Recent advance in knowledge management allows to extract information from unstructured text which is available in ASCII format [2]. The so-called *ontology-based information extraction* (OBIE) relies on general knowledge in form of an ontology. A user-specific knowledge base, for example, can be formalized in a RDF-graph⁷ and made available in a semantic desktop [4, 5]. OBIE uses this formalized knowledge and identifies the concepts which appear in the handwritten text. Based on this information, new knowledge can be generated, which just needs to be shortly confirmed by the user (instead of typing the new information explicitly by the keyboard).

Our proposed system performs a seamless integration of the handwriting recognition into the OBIE process. Instead of just applying OBIE on the recognized text, we designed an integrated process which also takes the top- k alternatives for each word into account.

The output of the recognizer are alternatives for each word. This can be illustrated in a recognition lattice, which contains the word alternatives as nodes and the word transitions as directed edges. An example for a handwritten sentence about ABBA is given in Figure 3.

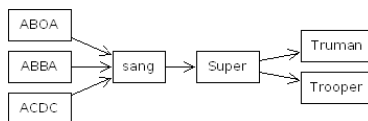


Fig. 3. Recognition lattice for alternates of “*ABBA sang Super Trooper*”

Now we extract all possible paths of a given length n in this recognition lattice. Herby, n can be seen as a parameter which controls the maximum length of labels of extracted instances. Note that a higher value would result in a longer processing time.

A second parameter is the number of alternatives which is used for the lattice, k . A higher value of k would include more recognition alternatives. This could, on the one hand side, lead to the acceptance of instances which were not correctly recognized. On the other hand side, other instances could be found which only have a short edit-distance to the written word. Note that setting

⁷ the Resource Description Framework is described in <http://www.w3.org/RDF/>

$k = 1, n =$ number of words corresponds to the naive approach which just passes the recognition result to the OBIE system.

The number of all paths with length n , considering the top- k alternatives would be:

$$k^n * (\text{word count} - n + 1) \quad (1)$$

This is the upper limit of the processing time of the algorithm. However, the Vision Objects recognizer often outputs only one word when the confidence for this word is very high.

The following example illustrates the behavior of the algorithm. Let us consider the recognition of a text about ABBA:

Handwritten text: “*ABBA sang Super Trooper*”

Available instances: *ABBA, ACDC, Super Trooper*

HWR output: (*ABOA, ABBA, ACDC*), (*sang*), (*Super*), (*Trooper, Truman*)

The recognition lattice is depicted in Figure 3.

The resulting n-grams for $n = 1$ and $n = 2$ and $k = 1, 2, 3$ are listed in Table 2. Using the combination of $n = 1, k = 1$ no instance would be found. By setting $n = 2$, the song title *Super Trooper* would be found, but ABBA is still unknown. If we use $n = 2$ and $k = 2$ ABBA and Super Trooper would be found. When $k = 3$, there could be a misinterpretation, because the instance ACDC would also be found, which corresponds to a *false positive*, because ACDC did not sing *Super Trooper*. This problem is tackled by the OBIE process, i.e., the probability (*belief value*) for ACDC would be significantly lower than that for ABBA.

Table 2. n-grams for the example in Fig. 3

	n=1	n=2
k=1	ABOA, sang, Super, Trooper	ABOA sang, sang Super, <i>Super Trooper</i>
k=2	ABOA, <i>ABBA</i> , sang, Super Trooper, Truman	ABOA sang, <i>ABBA</i> sang, sang Super, <i>Super Trooper</i> Super Truman
k=3	ABOA, <i>ABBA</i> <i>ACDC</i> , sang Super, Trooper, Truman	ABOA sang, <i>ABBA</i> sang, <i>ACDC</i> sang, sang Super, <i>Super Trooper</i> , Super Truman

After extracting the instance candidates we reorder the instances based on the belief values of the OBIE system and the confidence of the handwriting

recognizer. The reordering of word alternatives is done by calculating a weighted sum s_p for every path of the recognition lattice.

$$s_p = W * \sum_{i=1}^n c(w_i^{(p)}) + \sum_{j=1}^{ic} b(i_j^{(p)}) \quad (2)$$

where W is a factor which weights the influence of the recognizer's confidence, $c(w_i^{(p)})$ is the handwriting recognition confidence of the i -th word in the path, ic is the instance count in the current path and $b(i_j^{(p)})$ is the belief value of the j -th instance in the path. In [6] experiments of ontology-based information extraction (OBIE) have shown that the proposed extension performs better than a simple approach which just feeds the output of the recognizer as an input to the OBIE system. Furthermore, using the ontology instances to alter the recognition lexicon increased the f-measure up to 69.67% which is close to the performance on electronic text which is already available in ASCII.

6 Conclusions and Future Directions

In this paper we have proposed several ways bridge the gap between handwriting recognition and understanding. First, an approach to include semantic information into the recognition phase has been described. Assuming that the main topic of the handwritten note is often known beforehand, state-of-the-art technologies from the knowledge management research area are used to improve the recognizer. The basic idea is to alter the word lexicon used during recognition in order to add valuable information about the terms a writer normally uses.

These promising result motivate further research to include semantic information into handwriting recognition. We plan to perform experiments on a larger set of writers using more and different documents and PIMOs.

Another interesting point for future research is to investigate the include of semantic information in similar areas. Recent research focused on whole book recognition [16]. There the authors alter the word recognition probabilities based on previous observations. An extension of this research would be to (semi-)automatically build a knowledge base of the recognized book and use this gained knowledge during the recognition. Note that this approach would be similar to natural reading, where the reader gains more knowledge during reading. This knowledge is then not only used for recognizing previously unknown terms, but also understanding the content.

Second, a method for extracting relevant information from handwritten texts has been presented. We use the recently introduced ontology-based information extraction (OBIE) and extend it to the task of handwritten texts. In our experiments we have shown that our system achieves an f-measure of nearly 70% which is close to the performance on electronic text. This result can be interpreted as follows: The OBIE system was able to correctly understand the handwritten instances with an accuracy of 70% .

While the results are already promising and useful for the application in practice, there is still some room for improvement. In the current system we did not take the recognition confidences for the words and phrases into account when feeding them as an input to the OBIE system. Future work will enhance the algorithm to respect these confidences as well. Another interesting task is to decrease the processing time, i.e., by the use of Hash-maps.

References

1. 11th Int. Conf. on Frontiers in Handwriting Recognition (2008)
2. Adrian, B., Hees, J., van Elst, L., Dengel, A.: iDocument: using ontologies for extracting and annotating information from unstructured text. In: KI 2009: Advances in Artificial Intelligence. LNAI, vol. 5803, pp. 249–256. Springer (9 2009)
3. Bunke, H.: Recognition of cursive Roman handwriting – past present and future. In: Proc. 7th Int. Conf. on Document Analysis and Recognition. vol. 1, pp. 448–459 (2003)
4. Decker, S., Frank, M.R.: The networked semantic desktop. In: WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web (2004)
5. Dengel, A.R.: Knowledge technologies for the social semantic desktop. In: In Proceedings of the Second International Conference on Knowledge Science, Engineering and Management (2007)
6. Ebert, S., Liwicki, M., Dengel, A.: Ontology-based information extraction from handwritten documents. In: 12th International Conference on Frontiers in Handwriting Recognition. pp. 483–488 (2010)
7. Liwicki, M., Schumacher, K., Dengel, A., Weibel, N., Signer, B., Norrie, M.C.: Pen and paper-based interaction with the Semantic Desktop. In: DAS (2008)
8. Manmatha, R., Han, C., Riseman, E.: Word spotting: A new approach to indexing handwriting. In: Proc. Computer Vision and Pattern Recognition. pp. 631–637 (1996)
9. Pena Saldarriaga, S., Viard-Gaudin, C., Morin, E.: Combining approaches to on-line handwriting information retrieval. In: Proc. SPIE Document Recognition and Retrieval XVII. vol. 7534, pp. 753403–753403–8 (2010)
10. Pittman, J.A.: Handwriting recognition: Tablet pc text input. *Computer* 40(9), 49–54 (2007)
11. Plamondon, R., Srihari, S.N.: On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(1), 63–84 (2000)
12. Sauermann, L., Bernardi, A., Dengel, A.: Overview and outlook on the semantic desktop. pp. 1–19 (2005)
13. Sauermann, L., van Elst, L., Dengel, A.: PIMO – a framework for representing personal information models. In: Proceedings of I-SEMANTICS 2007. pp. 270–277 (2007)
14. Sayre, K.M.: Machine recognition of handwritten words: a project report. *Pattern Recognition* 5(3), 213–228 (1973)
15. Vinciarelli, A.: A survey on off-line cursive script recognition. *Pattern Recognition* 35(7), 1433–1446 (2002)
16. Xiu, P., Baird, H.S.: Towards whole-book recognition. In: Document Analysis Systems (2008)