Improving Handwriting Recognition by the Use of Semantic Information

Marcus Liwicki
DFKI
Germany
marcus.liwicki@dfki.de

Hassan Mohamed Abou Eisha
German University in Cairo
Egypt

Andreas Dengel
DFKI
Germany
andreas.dengel@dfki.de

ABSTRACT

This paper proposes a first attempt to include real semantic information into the process of handwriting recognition. We take advantage of the fact that the main topic of handwritten notes is often known beforehand like in annotation or reviewing tasks. Using state-of-the-art technologies from the knowledge management research area it is possible to store a semantic representation of the user's knowledge in a Personal Information Model (PIMO). This PIMO stores the relations between semantic concepts and documents on the computer. In this paper we extract texts from related documents and concepts of the PIMO. The vocabulary of these texts is then used to aid the recognizer. In our multi-writer experiments, a significant improvement of the recognition accuracy by 8% on the text line level has been achieved.

1. INTRODUCTION

More than 40 years ago the automatic recognition of handwritten text has become a research topic for computer scientists. During the following years recognition systems became more and more sophisticated and included *a-prioroi* knowledge on several layers. While the first recognizers have been developed for isolated characters or digits, later recognizers focused on complete words or even sentences. A current state-of-the-art handwriting recognition system uses language models to incorporate linguistic information [1, 7, 12], which is feasible for the recognition of text lines.

In this paper we propose a method which goes beyond those processing steps. Our system uses semantic information for the improvement of handwriting recognition. This information is extracted from a representation of a user's knowledge on the computer, a so-called Personal Information Model (PIMO) [10].

The described work is mainly motivated by the semantic desktop [2, 3] which contains a user-specific knowledge base. We assume that the handwritten informations are somehow related to other known documents and the relationship is known. This assumption is feasible because in most real-world cases a handwritten document belongs to a specific event (e.g., a project meeting or an interview) or corresponds to a document like in annotation and reviewing tasks.

Specifically, this work is part of the Semantic eInk research project [4]. The Semantic eInk system automatically processes online handwritten annotations on printed documents and interprets the semantic information of these annotations. This information will be expressed in the PIMO using the individual's vocabulary, and integrated into the Semantic Desktop. The integration makes this knowledge searchable, reusable, sharable and gives a context for its interpretation.

To the authors' knowledge this work is the first research incorporating semantic information into the recognition process. While title of Ref. [8] suggests that the topic of using semantic information for handwriting recognition has already been researched for several years, no higher-level knowledge is used there. Ref. [8] uses a general text corpus for the generation of a statistical model, which is nowadays known as the integration of a language model.

The rest of this paper is organized as follows. First, Section 2 briefly describes the Semantic eInk system as a usage scenario for our proposed recognizer. Next, Section 3 gives an overview of the general handwriting recognition approach and our contribution. Subsequently, the proposed methods are introduced in Section 4. Next, an experimental evaluation is performed in Section 5, and finally, Section 6 draws some conclusions and gives an outlook to future work.

2. SEMANTIC EINK

The idea of the Semantic eInk system was initially proposed at the DAS 2008 [4]. Semantic eInk allows a seamless integration of interactive paper technology into personal knowledge work. To be more specific, the workflow of printing a document, annotating it while reading, and integrating the new information into the personal knowledge base will be supported by an automated interpretation of user annotations. Therefore, the documents are printed onto the Anoto paper[1] and annotations are made with the digital pen. Then, a set of gestures and handwritten text is recognized and finally, the information is sent to the Semantic Desktop.

The Semantic Desktop [2, 3] is a means for personal knowl-

[1]http://www.anoto.com

DAS '10, June 9-11, 2010, Boston, MA, USA

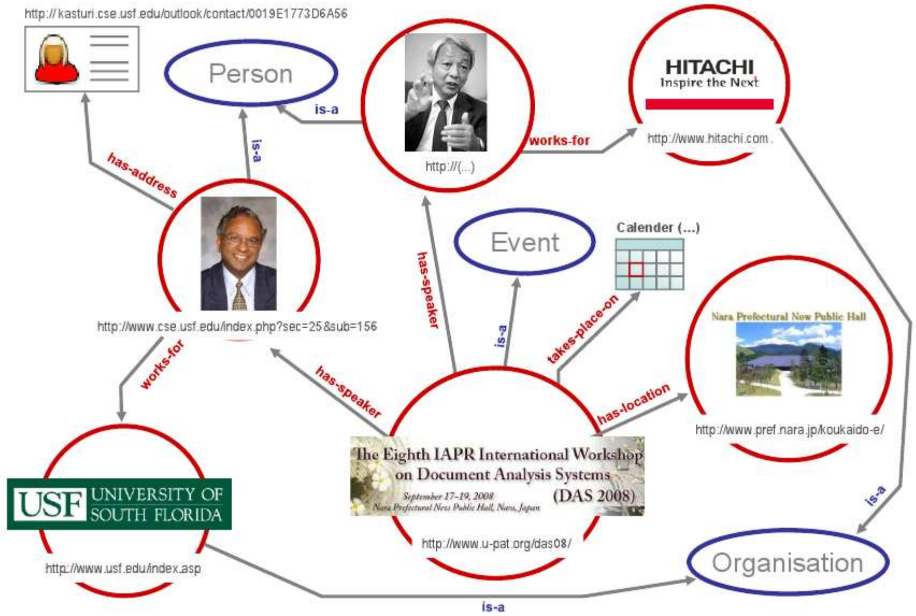


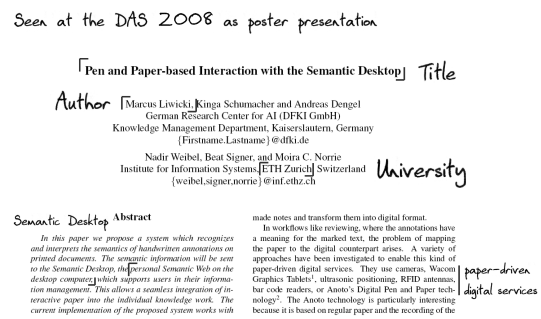
Figure 1: PIMO extract: example representation of the event "DAS 2008" and the keynote speakers [4]

edge management; it builds the personal Semantic Web on desktop computers. The consistent application of Semantic Web standards such as the Resource Description Framework[2] (RDF) and RDF Schema[3] (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) [10]. Figure 1 illustrates an extract of a PIMO which represents part of the information about the event "DAS 2008" and the keynote speakers of this conference. The figure shows some *ontological concepts* (classes like "Event" and instances like "USF"), which are related to the DAS conference and semantically describes the kind of relations, e.g., "take-place-on".

In the original prototype of Semantic eInk it is allowed to use three kinds of annotation (see Fig. 2). First, comments can be written at any place (The topmost handwritten text in Fig. 2). Second, the user can mark a short text passage and write a corresponding annotation ("Title" in Fig. 2). Third, a handwritten note can be added to a longer passage as a side-mark.

In a recent experimental study [5] we performed preliminary recognition experiments with the semantic eInk system. There about 84% of the words were recognized correctly. However, we observed that 98% of the correct words were present in the 5-best list and more than 99% were in the 10-best list. Furthermore, most of the incorrect words had nothing to do with the annotated document or with the knowledge domain around the document. Therefore we assume that integrating the semantic information of the document could help to improve the recognition results. Note that this assumption does not only hold for the Semantic eInk system, but in any other scenario where some knowledge about the handwriting is *a-priori* known.

[2]http://www.w3.org/RDF/
[3]http://www.w3.org/TR/rdf-schema/



Seen at the DAS 2008 as poster presentation

Pen and Paper-based Interaction with the Semantic Desktop Title

Author Marcus Liwicki, Kinga Schumacher and Andreas Dengel
German Research Center for AI (DFKI GmbH)
Knowledge Management Department, Kaiserslautern, Germany
{Firstname.Lastname}@dfki.de

Nadir Weibel, Beat Signer, and Moira C. Norrie
Institute for Information Systems, ETH Zurich Switzerland University
{weibel,signer,norrie}@inf.ethz.ch

Semantic Desktop **Abstract**

In this paper we propose a system which recognizes and interprets the semantics of handwritten annotations on printed documents. The semantic information will be sent to the Semantic Desktop, the personal Semantic Web on the desktop computer, which supports users in their information management. This allows a seamless integration of interactive paper into the individual knowledge work. The current implementation of the proposed system works with

made notes and transform them into digital format.

In workflows like reviewing, where the annotations have a meaning for the marked text, the problem of mapping the paper to the digital counterpart arises. A variety of approaches have been investigated to enable this kind of paper-driven digital services. They use cameras, Wacom Graphics Tablets[1], ultrasonic positioning, RFID antennas, bar code readers, or Anoto's Digital Pen and Paper technology[2]. The Anoto technology is particularly interesting because it is based on regular paper and the recording of the

paper-driven digital services

Figure 2: Annotating documents with the Semantic eInk system

3. RECOGNITION SYSTEM OVERVIEW

This section gives an overview of the handwriting recognition system. The main steps performed in handwriting recognition are illustrated in Fig. 3, they consist of preprocessing, normalization, feature extraction, classification, and finally a postprocessing.

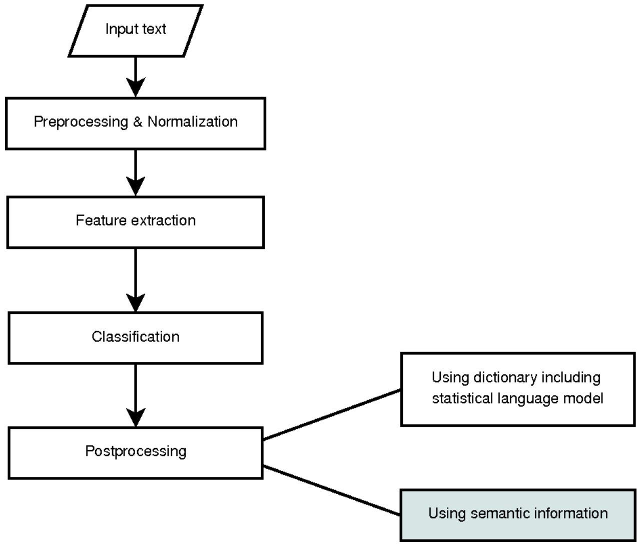


Figure 3: General handwriting recognition and our main contribution: We include semantic information into the recognition process

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 [11].

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.

As in our previous work [5] we use the *Microsoft Handwriting Recognizer*[4]. 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 [6].

The contribution of this paper 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.

4. METHODOLOGY

The aim of this paper is to use semantic information to enhance automated recognition of handwritten texts and annotations. Therefore we use a wordlist obtained from the user's PIMO. The information is extracted from the user's PIMO as it reflects the user domain and considered to be part of the cognitive system of the user [9]. The wordlist extracted represents the semantic information that will be used to support and improve the recognition process.

At a first glance, it seems to be a very simple attempt to improve the performance by just altering the recognition dictionary. However, as will be shown in the experiments section (see Section 5), this approach is already very helpful. Furthermore, the methodology of how to extract the wordlist will be important. We will describe this methodology in the remainder of this section.

Two approaches are used for extracting the dictionary, a static and a dynamic approach. While the static approach uses the information of the whole PIMO, the dynamic approach takes the relations of the semantic concepts into account.

4.1 Static Approach

For the static approach we extracted all data present in the user's PIMO. This data comprise all known concepts (persons, projects, documents), specific entities, electronic documents, and their relations between each other.

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-graph[5]) 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

[4]The Microsoft Windows XP Tablet PC Edition SDK for our experiments. It is available for download at http://www.microsoft.com/windowsxp/tabletpc/default.mspx

[5]the Resource Description Framework is described in http://www.w3.org/RDF/

Recognition mode	System dictionary	Wordlist
User dictionary	yes	yes
Application dictionary	no	yes
Default	yes	no

Table 1: The usage of dictionaries in different recognition modes

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 (see also Fig. 4 for an illustration.

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.

4.3 Including Wordlist into Recognition

As stated above, the Microsoft Handwriting Recognizer is used for the recognition. Usually this recognizer uses a system dictionary that includes all common words in the language.[6] However, there exists means for extending this dictionary through a user dictionary or an application dictionary.

Table 1 shows the properties of the different recognition modes. The user dictionary is a dictionary that contains

[6]This dictionary is unknown to the authors of this paper since it is proprietary.

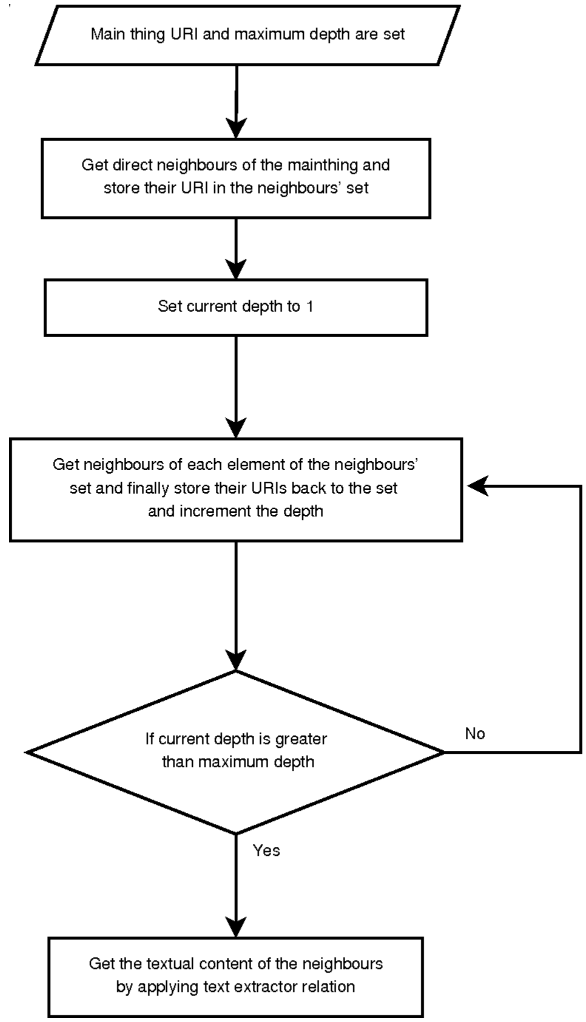


Figure 4: Illustration of the dynamic dictionary creation.

words added by the user. If an application dictionary is used, the recognizer is restricted to return words in the application dictionary as a result. In the default mode, no specific word list is used.

The application dictionary can enhance the performance as it contains words that are very likely to occur in an application (like the wordlist in our case). Since the system dictionary is completely ignored, general English terms might be misrecognized in this mode.

Applying the wordlist as user dictionary overcomes this problem. However, it can also have disadvantages. Words present in the system dictionary with a small edit distance to the ground truth of the handwritten data could make the recognizer misrecognize the words even if they appear in the wordlist. Unfortunately, the Microsoft Handwriting Recognizer does not allow prioritizing the confidence of words from the wordlist over those in the system dictionary. Thus there is no chance to have a parameter controlling the influence of the wordlist in a more fine-grained manner.

5. EXPERIMENTS

We have performed several experiments in order to asses the the influence of the different dictionary extraction methods and wordlist inclusion approaches. These approaches have been investigated on two data sets with different properties.

5.1 Data

To reflect a realistic situation, we have used two data sets. Both data sets are based on the PIMO of a real person who is using the NEPOMUK Semantic Desktop [9] as a personal knowledge management tool over years. This person uses the Anoto pen for taking notes during meetings and connects them to the concept of the meeting in the Semantic Desktop. Therefore, the handwritten information and the relation to the PIMO are known, making this data very useful for our experiments.

The first data set consists of three meeting notes, each filling about one A4-page (1,775 words in total). We manually generated the ground truth for these documents to compare it with the recognizer's output. All relations of the document in the PIMO have been investigated and removed if they were based on the annotations (like relations to Persons whose names were written down). The concepts (e.g., persons) themselves, however were kept in the PIMO if there also existed other relations from these concepts. This step has been performed to make sure that no ground-truth data exists in the PIMO in order to reflect a real-world situation. Note that this database is quite small, but still very useful, because no optimization of any parameter has been performed on this set. Algorithm parameters like which relations to choose as edges in the dynamic search (see Section 4) have been empirically set beforehand on synthetic data.

For the second data set we asked that person to write annotations on research papers (two documents extracted from the PIMO which were not annotated beforehand) using the Anoto pen, pretending that a perfect Semantic eInk system would exist. Afterwards, we asked five other writers to copy the annotations line by line, in order to make the experiments writer-independent. Altogether this dataset comprises about 1,000 annotations written by 6 writers. Again, not parameters were optimized on this set.

5.2 Evaluation

The recognizer was applied on each text line. The recognition performance is measured by the **Accuracy** using the following formula:

$$\text{Accuracy} = \frac{\text{No. of hits} - \text{No. of insertions}}{\text{No. of ground truth elements}} \quad (1)$$

where the number of hits and insertions are calculated using Levenstein distance between the recognition result and the ground truth. In the second experiments also the word recognition rate is used, which just counts the number of correct words and divides it through the number of words in total.

5.3 Meeting Notes

The results of the three meeting notes documents appear in Table 2. For each document the recognition accuracy for different parameters is given. The default classifier uses no additional wordlist. In the tables the abbreviations *AD* and *UD* denote application dictionary and user dictionary. The results of the dynamic approach are given for each depth of the search algorithm and the number of words included in the wordlist are given in Column 3.

Table 2: Recognition accuracies (in %) for text line recognition on the three documents using the different wordlists and different recognition modes

Doc.	Dictionary	# words in AD	AD	UD
D1	default		70.3	70.3
	depth 1	39	33.3	71.9
	depth 2	2,340	52.8	71.2
	depth 3	9,997	62.1	71.9
	depth 4	24,510	64.9	73.0
	depth 5	49,987	70.9	73.3
	static	50,000	70.2	-
D2	default		77.3	77.3
	depth 1	283	50.3	77.3
	depth 2	3,111	62.0	77.3
	depth 3	18,956	74.2	76.1
	depth 4	49,983	62.8	79.1
	static	50,000	78.5	-
D3	default		63.0	63.0
	depth 1	2,826	54.3	63.0
	depth 2	18,333	63.0	62.0
	depth 3	33,721	63.9	63.3
	depth 4	49.983	62.8	61.4
	static	50,000	63.0	-

Table 3: 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

As can be seen, there is no significant difference between using the static dictionary and the default recognizer. However, using the user dictionary setting (see Table 1) the usage of a dynamic wordlist is beneficial for the recognition.

One might argue that using a dynamic wordlist increases the computation time needed for the recognition. However, this search has to be performed only once for each document. In our experiments the time for the search was less than 5 seconds, while the recognition of each text line takes about 1 second. Since there are at least 10 text lines in each document, the search time is negligible.

5.4 Annotations

Table 3 shows the result of the annotation task. 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%.

A deeper analysis of the results have shown that in both experiments many improvements are due to the use of specific project or person names. However, even some terms generally used by the writer could be corrected by our approach.

6. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach to include semantic information into the recognition of handwritten texts. 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.

In our experiments we have shown that the performance of the recognizer has always been improved when semantic information is incorporated into the recognition. In the experiments on whole meeting notes the recognition accuracy was improved by 2%. In the recognition experiments on document annotations, the accuracy gain was more than 8%.

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. It will also be interesting use a recognizer where we can directly control the influence of the wordlist.

Currently we are developing more sophisticated approaches, where not only the wordlist is altered. We are investigating methods where the knowledge information is directly included into the recognition.

Another interesting point for future research is to investigate the include of semantic information in similar areas. Recent research focused on whole book recognition [13]. 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.

7. ACKNOWLEDGMENTS

This research has been supported by the Stiftung Rheinland-Pfalz für Innovation.

8. REFERENCES

[1] H. Bunke. Recognition of cursive Roman handwriting – past present and future. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 448–459, 2003.

[2] S. Decker and M. R. Frank. The networked semantic desktop. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.

[3] A. R. Dengel. Knowledge technologies for the social semantic desktop. In *In Proceedings of the Second International Conference on Knowledge Science, Engineering and Management*, 2007.

[4] M. Liwicki, K. Schumacher, A. Dengel, N. Weibel, B. Signer, and M. C. Norrie. Pen and paper-based interaction with the Semantic Desktop. In *DAS*, 2008.

[5] M. Liwicki, M. Weber, and A. Dengel. Automatic recognition and interpretation of pen- and paper-based document annotations. In *Advances in Artificial Intelligence*, volume 5803 of *LNAI*, pages 664–671, 2009.

[6] J. A. Pittman. Handwriting recognition: Tablet pc text input. *Computer*, 40(9):49–54, 2007.

[7] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[8] T. Rose, L. Evett, and R. Whitrow. The use of semantic information as an aid to handwriting recognition. In *Proc. 1st Int. Conf on Document Analysis and Recognition*, pages 629–637, 1991.

[9] L. Sauermann, A. Bernardi, and A. Dengel. Overview and outlook on the semantic desktop. pages 1–19, 2005.

[10] L. Sauermann, L. van Elst, and A. Dengel. PIMO – a framework for representing personal information models. In *Proceedings of I-SEMANTICS 2007*, pages 270–277, 2007.

[11] K. M. Sayre. Machine recognition of handwritten words: a project report. *Pattern Recognition*, 5(3):213–228, 1973.

[12] A. Vinciarelli. A survey on off-line cursive script recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[13] P. Xiu and H. S. Baird. Towards whole-book recognition. In *Document Analysis Systems*, 2008.