Pattern Recognition Letters 35 (2014) 204–213

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Bridging the gap between handwriting recognition and knowledge management

Marcus Liwicki [a,*], Sebastian Ebert [a,b], Andreas Dengel [a,b]

[a] German Research Center for Artificial Intelligence (DFKI) GmbH, Trippstadter Straße 122, 67663 Kaiserslautern, Germany
[b] Knowledge-Based Systems Group, Department of Computer Science, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany

ARTICLE INFO

Article history:
Available online 27 December 2012

Keywords:
Handwriting recognition
Information extraction
Ontology
OBIE

ABSTRACT

In this paper we introduce a new layer for the task of handwriting recognition (HWR), i.e., the use of semantic information in form of Resource Description Framework (RDF) knowledge bases. In particular, two novel processing stages are proposed for the first time in literature. The first stage is the inclusion of RDF knowledge bases into the HWR process, where we make use of a person's mental model. This process can be extended to use other ontological resource. The second stage is the transition from pure handwriting recognition to understanding the handwritten notes, i.e., the system extracts knowledge employing RDF knowledge-bases. This is also called ontology-based information extraction (OBIE). The task of our recognizer therefore is not only to recognize the ASCII transcription of the handwritten document, but also to identify the semantic concepts which appear in the text. For both novel approaches we performed a set of experiments on various data. First, the recognition rate of the HWR system is increased on several documents. Second, the performance of information extraction is also remarkable. By using the *k*-best word recognition alternatives in form of a lattice as an input for the OBIE system, the performance reaches a level which is very close to OBIE applied on pure ASCII text.

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 (Bunke, 2003; Plamondon and Srihari, 2000; Vinciarelli, 2002). 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 (Chaudhuri et al., 2010).

In this paper 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 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 this 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 (Adrian et al., 2009). The so-called *ontology-based information extraction* (OBIE) (Wimalasuriya and Dou, 2010) relies on general knowledge in form of an ontology. A user-specific knowledge base, for example, can be formalized in an RDF-graph[1] and made available in a Semantic Desktop (Decker and Frank, 2004; Dengel, 2007). 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). OBIE methods first segment the text into tokens, then identify their values and their corresponding instances of the ontology, and finally try to generate new facts based

* Corresponding author. Tel.: +49 0 631 205 75 1200; fax: +49 0 631 205 75 1020.
E-mail address: marcus.liwicki@dfki.de (M. Liwicki).

[1] The Resource Description Framework is described in <http://www.w3.org/RDF/>.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.12.012

on the text. To the authors' knowledge, in this paper OBIE is proposed for the first time in handwriting literature.

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. In our experiments we measure the number of correctly extracted instances. We found that considering more than just the top-candidate improved the performance.

Note that this article is an extended version of Ebert et al. (2010). However, while Ebert et al. (2010) focused on the overall system description and experiments, this article gives more background information of ontologies and handwriting recognition. Furthermore, the methods for improving handwriting recognition are described with more detail and experiments for this task are included. Finally, a novel set of experiments is performed on short handwritten notes in order to compare the behavior on handwritten notes to the behavior on handwritten texts.

The paper is organized as follows. First, Section 2 gives an overview of the general structure of HWR systems and introduces two examples that were used in our experiments. Furthermore, related work in the field of information extraction is presented. Second, Section 3 deals with the representation of knowledge and explains basic concepts that are used throughout the paper. Next, Section 4 shows how semantic information can be incorporated in the HWR process to increase the recognition performance and gives experimental results on this approach. Subsequently, Section 5 describes our approach on how to extract knowledge out of handwritten text. Experimental results are also reported in Section 5. Finally, Section 6 concludes the paper and gives directions for future work.

2. Background

2.1. State-of-the-art HWR systems

This section gives an overview about 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 step.

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. On the one hand side, it is not possible to segment a word into characters before recognizing this word and on the other hand side the word cannot be recognized correctly before being segmented into characters. This situation is known as Sayre's paradox (Sayre, 1973).

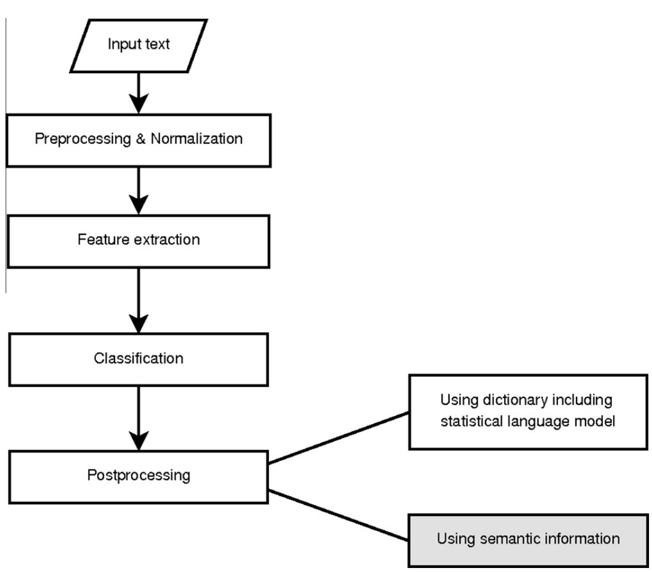


Fig. 1. General handwriting recognition and our main contribution: we include semantic information into the recognition process.

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©*[2] for parts of our experiments. 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 the work of Pittman (2007).

As an alternative, the MyScript recognizer from Vision Objects© was used for the recognition.[3] The overall recognition system is built on the principles presented by Knerr et al. (1997). Furthermore, a state-of-the-art statistical language model as described by Perraud et al. (2006) is used.

The contribution of Section 4 is to enhance the postprocessing by the integration of semantic information. The semantic information is extracted from a representation of the user's mental model. More specific information about the mental models and their representation is given in the next section.

2.2. Related work in information extraction on handwritten documents

The contribution of Section 5 is to extract information from the handwritten notes. Several other research areas are related to this task. Word spotting, for example, is the task of finding a given word in a handwritten text (Manmatha et al., 1996). 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, since specific words are to be retrieved. However, in word spotting there is only a single query while ontology-based information extraction tries to find semantic instances given in an ontology, which might be very complex. Furthermore, we do not just apply a search algorithm, instead we also take relations between the concepts in the RDF knowledge base into account.

Another related task is the retrieval of documents out of a given document corpus. Document retrieval became more and more popular in the last years. Here the task is to find (retrieve) or classify a given set of documents (Pena Saldarriaga et al., 2010). Even if

[2] The Microsoft Windows XP Tablet PC Edition SDK© is available for download at <http://www.microsoft.com/windowsxp/tabletpc/default.mspx>.

[3] The MyScript Builder SDK© is available for purchase at <http://www.visionobjects.com/>.

some kinds of information extraction is performed, this topic is only loosely related to the topic addressed in Section 5, since we do not consider whole documents, but entities appearing in those documents.

Information extraction from documents, which were not handwritten, was proposed in (Adrian et al., 2009). In this paper we use the approach of Adrian et al. (2009) and enhance it for the task of handwriting recognition.

3. Representing knowledge in RDF

Knowledge can be represented by sets of attribute-value pairs. These sets are often very complex and contain several kind of information, e.g., concepts of events, entities (like persons), objects, or ideas; and spatiotemporal schemas for the contexts of concepts. Within the spatiotemporal schemas there are structural relations which define them. Those relations might be spatial, e.g., an i-dot is always placed above the i; social, e.g., I like the handwriting style of writer "A"; temporal, e.g., the signature is written at last; etc. Several contexts can also be brought into relation with one-another by linking the corresponding spatiotemporal schemas with causal or temporal information.

In the Semantic Web community standards, such as the Resource Description Framework[4] (RDF) and RDF Schema[5] (RDFS) have been introduced to represent the attribute-value pairs. In RDF, concepts are interlinked with one another via binary relations. This is a way of formalizing the above mentioned ideas about schemas.

3.1. Components of RDF knowledge bases

By using the RDFS vocabulary the main components of a domain ontology (also referred to as *input ontology* in this paper) O can be defined as $O(H_C, H_P, I, S, A)$ Adrian et al. (2009):

- The hierarchy of classes (H_C) which is the transitive closure of all `rdfs:subClassOf` expressions, where subsumptions of two classes c_1 and c_2 can be expressed by `rdfs:subClassOf` (c_1, c_2).
- The hierarchy of properties (H_p) which is the transitive closure of all `rdfs:subPropertyOf` expressions, where a specialization p_1 of a property p_2 can be expressed by `rdfs:subPropertyOf` (p_1, p_2).
- The instance base (I) consisting of resources i with `rdf:type` (i, c) where $c \in H_C \setminus \{\texttt{rdfs:Literal}^T\}$.
- The symbol base (S) consisting of resources s with `rdf:type` (s, c) where `rdfs:subClassOf`(c, `rdfs:Literal`).
- The assertion base (A) consisting of triple expressions in the form of $p(i, r)$ with $p \in H_P$ and $i, r \in H_C \bigcup I \bigcup S$.

3.2. Linked open data

Another important issue in representing knowledge is to identify digital resources, i.e., text documents, web sites, or multimedia files, by unique URIs. A very huge movement of using URIs and formal standards is the Linked Open Data (LOD) Community Project (Heath and Bizer, 2011). This community tries to make the web resources human- and machine understandable by describing HTTP-URIs with RDF and interlinking data from different sources using existing description standards.

Our proposed system is designed to work with RDF and therefore with LOD. Thereby, it is lifted to a generic system which works on thousands of knowledge bases available world-wide.

If the goal is to understand the meaning of handwritten notes there is need for a digital representation of a user's mental model. An approach towards this mental model is the *Personal Information Model* (PIMO), which is motivated by the Semantic Desktop. While the mental model is part of the cognitive system and thus individual and cannot be externalized thoroughly, the PIMO aims to represent parts of the mental model necessary for knowledge work (Sauermann et al., 2007).

3.3. PIMO and the Semantic Desktop

The Semantic Desktop (Decker and Frank, 2004; Dengel, 2007) 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 RDF and 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 reflecting 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 PIMO (Sauermann et al., 2007). Fig. 2 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 "Organization" and instances like "USF"), which are related to the DAS conference and semantically describes the kind of relations, e.g., "take-place-on".

It is obvious that many things that a user can think of are already implicitly represented in his resources. Typical knowledge workers have already many entries in their address books and files structured in folders on their computers. There exist many algorithms and approaches to automatically generate or extend ontologies based on data in text files or other data sources.[6]

4. Improving handwriting recognition by the use of semantic information

In order to enhance automated recognition of handwritten texts and annotations we use a word list obtained from the knowledge base. The word list 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 straightforward attempt for improving the performance by just altering the recognition dictionary. However, experiments have shown that this approach is already very helpful (see below). Furthermore, the methodology of how to extract the word list is 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 knowledge base (domain), 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 knowledge base. This data comprise all known concepts (persons,

[4] <http://www.w3.org/RDF/>.
[5] <http://www.w3.org/TR/rdf-schema/>.

[6] Aperture is a Java framework for generating data and metadata <http://aperture.sourceforge.net/>.

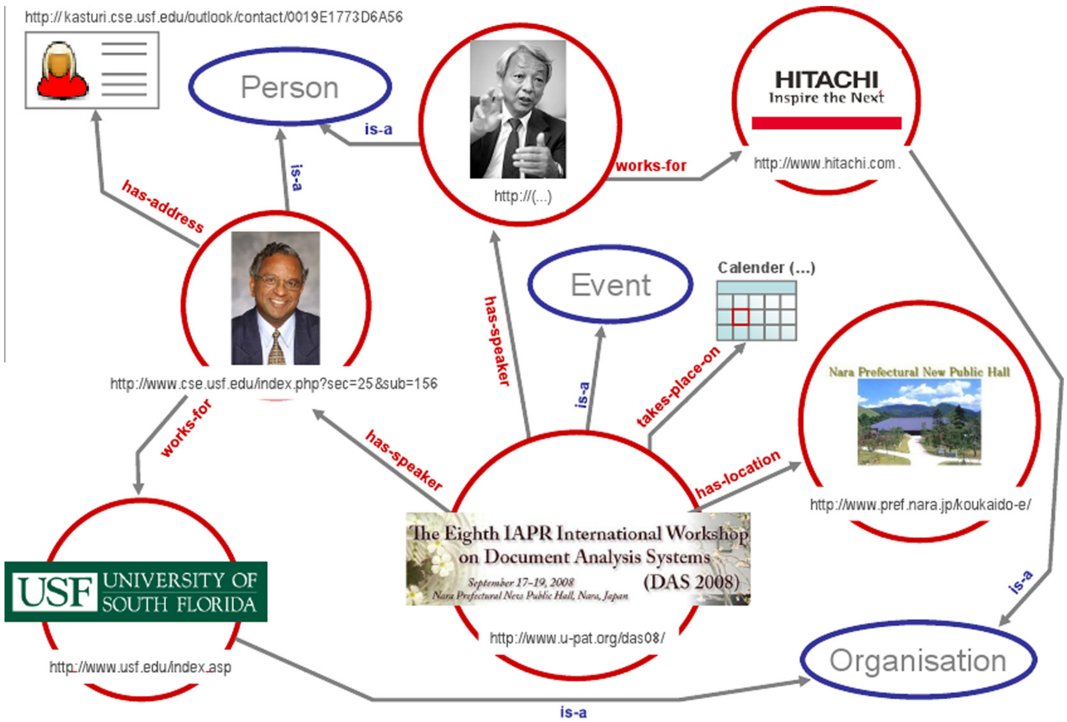


Fig. 2. PIMO extract: example representation of the event "DAS 2008" and the keynote speakers.

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. (i) All textual information from the RDF statements are selected. (ii) The texts contained in objects beyond these relations (electronic documents) are added to this information. (iii) 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 knowledge base described in RDF is a broader approach, because all information is stored across conventional application borders. A typical database, for example, has no information about a person's 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 knowledge base (the *mainthing*) 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. As stated before, in the RDF knowledge base each object is identified by a unique URI, so we start at the URI of the *mainthing*.

The algorithm is then similar to a breadth first search in the graph domain (also called spreading activation) where the edges are given by connector relations (relations that connect concepts to related ones). The depth of the search is a parameter being investigated in our experiments. The algorithm works as follows:

1. An RDF-graph is extracted.
2. Starting from the *mainthing*, find all concepts related to it by connector relations.

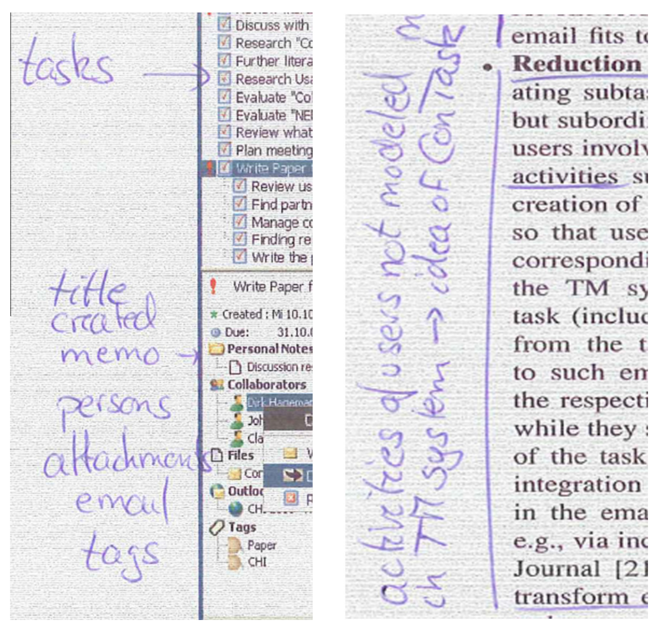


Fig. 3. Annotation examples for Set 2.

3. Repeat step 2 until the desired depth is reached.
4. All textual information from the RDF statements is added to the vocabulary.
5. The texts contained in objects beyond these relations (electronic documents) are also added to this information.
6. Finally, the dictionary is composed from the *n* most frequent words of the resulting text corpus.

4.3. Discussion

Note that both, the static and the dynamic approach make use of several formal RDF semantics of RDF knowledge bases. Since a

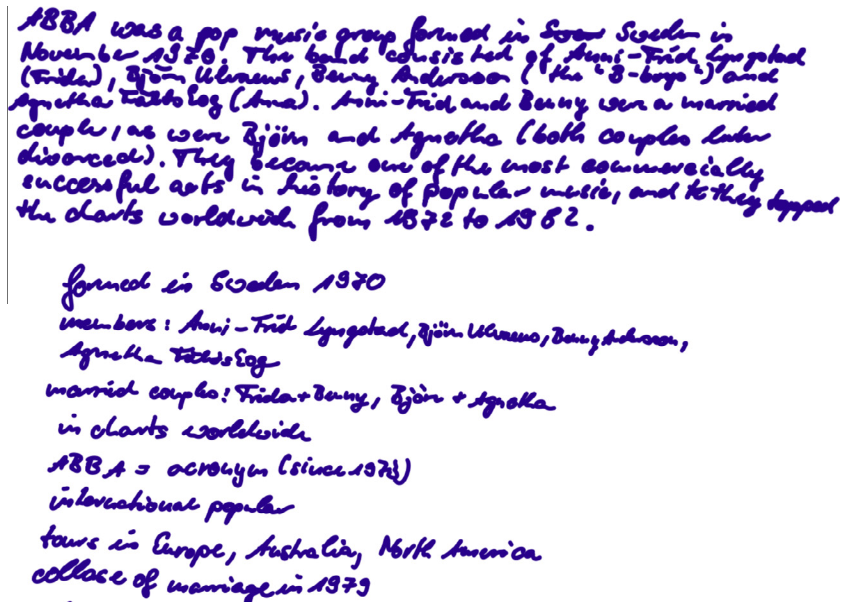


Fig. 4. Text (top) and notes (bottom) examples for Set 3.

specific knowledge base is taken, the domain and range is limited to the specific application area. Thus both approaches differ from simpler ad hoc approaches which make use of online dictionaries like WordNet.[7] Furthermore, the dynamic approach makes use of specific properties of ontologies, i.e., the subsumptions. In the PIMO a subsumption is represented as a connector relation.[8] As the dynamic approach only increases the set of concepts based on the connector relations, it grows more slowly than if all connections by WordNet would be taken into account.

4.4. Experimental data

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

To reflect a realistic situation, we have used two data sets based on the PIMO of a real person who has been using the NEPOMUK Semantic Desktop (Sauermann et al., 2005) as a personal knowledge management tool over years. The knowledge base contains about 7000 instances and many properties resulting in more then 50,000 words. The instances are the real projects, persons, topics, documents, and other concepts, the person deals with in his personal knowledge work.

This person uses the Anoto pen[9] 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 (1775 words in total). We manually generated the ground truth for these documents to compare it with the recognizer's output. All relations of the documents 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. In order to make sure that no optimization has been done, an independent synthetic data set has been used during the creation of the algorithm.

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. Examples can be seen in Fig. 3. 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 1000 annotations written by six writers. Again, no parameters were optimized on this set.

The third and fourth data set contains documents from the domain of music of the 20th century. These experiments have been performed in order to show the generality of our approach, i.e., that we are not bound to the PIMO but that we can apply the strategy on any ontology-guided recognition tasks. Both sets are based on an RDF knowledge base which has general information about the existence of specific bands, e.g., ABBA, Madonna, and Tina Turner. The sets were generated by taking a subset of the DBPedia[10] which contains articles related to music. Note that the DBPedia has been chosen in order to show that the approaches work on LOD which usually contains some noise. The used instance base consists of 4312 instances, that contain 5562 datatype property values (e.g., names, song titles, album titles). Further 17 different datatype and object properties (e.g., foaf:name, rdfs:label) were used. The classes and properties were chosen to cover the most important concepts of the music domain which is considered in our OBIE task.

For the third data set we have extracted 75 texts about 15 bands and asked 15 writers to copy these texts. As such, we simulate the scenario that a person writes down information about specific bands, which might enlarge the person's knowledge base. An example text seen in Fig. 4 on top.

[7] <http://wordnet.princeton.edu/>.

[8] represented by the relation "is-a" (note that "is-a" denotes subsumption, not only instantiation.

[9] family <www.anoto.com>.

[10] <http://dbpedia.org>.

Table 1
Recognition accuracies for text line recognition on Set 1 using the different word lists and different recognition modes.

Doc.	Dictionary	# Words in dictionary	Accuracy in %
D1	Default		70.3
	Depth 1	39	71.9
	Depth 2	2340	71.2
	Depth 3	9997	71.9
	Depth 4	24,510	73.0
	Depth 5	49,987	73.3
	Static	50,000	70.2
D2	Default		77.3
	Depth 1	283	77.3
	Depth 2	3111	77.3
	Depth 3	18,956	76.1
	Depth 4	49,983	79.1
	Static	50,000	78.5
D3	Default		63.0
	Depth 1	2826	63.0
	Depth 2	18,333	62.0
	Depth 3	33,721	63.3
	Depth 4	49.983	61.4
	Static	50,000	63.0

For the fourth data set, we asked the same writers to write down some notes about their band. These notes are just bullet points containing some information about the band taken from the same texts. Each writer wrote down 12 bullet points in average with 8–20 words, each. An example text seen in Fig. 4 on the bottom. This scenario corresponds to quickly writing down information during an interview or a fast research about a specific person.

4.5. Results

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 by using the Levenshtein (Gusfield, 1997) distance between the recognition result and the ground truth. For measuring the results for word recognition (as for the annotations in the second experiment) also the word recognition rate is used, which just counts the number of correct words and divides it by the number of words in total.

The summarized results of the three meeting notes documents (Set 1) appear in Table 1. For each document the recognition accuracy for different parameters is given. Since the Microsoft© recognizer has been used for these experiments, the general english dictionary is always considered for the recognition. The generated dictionary can be seen as adding additional words which are then recognized with a higher probability. The default classifier uses no additional word list. The number of words included in the word list for each depth of the search algorithm are given in column 3. As can be seen, there is no significant difference between using the static dictionary and the default recognizer. However, the usage of a dynamic word list is beneficial for the recognition.

Note that one might argue that using a dynamic word list 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 s, while the recognition of each text line takes about 1 s. Since there are at least 10 text lines in each document, the search time is negligible.

Table 2 shows the recognition results on the annotations of the two documents described above (Set 2). In these experiments we additionally tested the recognizer on the word basis, i.e., without linguistic information. This is motivated by the fact that very often real handwritten annotations do not make much sense from a linguistic point of view (often they contain just one or two words that have been written as annotation). These results are averaged over the writers. The depth for the dynamic approach 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 do not make much sense.

Table 2
HWR accuracy on the second data set in %.

Document	Mode	Dynamic dictionary	Static dictionary	Default
1	Line	77.9	77.2	69.7
1	Word	76.6	75.8	72.9
2	Line	81.0	79.7	72.2
2	Word	81.3	76.7	74.8

Table 3
Handwriting recognition results on Sets 3 and 4.

	Set 3		Set 4	
	Rec. rate (%)	Accuracy (%)	Rec. rate (%)	Accuracy (%)
Default	79.6	78.3	74	72
Static dictionary	81.4	80.2	76	75

Using semantic information was always very useful and lead to a significant improvement of the recognizer's performance. 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%.

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

Table 3 shows the handwriting recognition performances on Sets 3 and 4. Since only a small knowledge base was used, the static dictionary approach has been used in these experiments. As can be seen, the HWR performance could also be increased. This shows that the approach is generic and can be applied to any existing RDF knowledge base and with some extensions to any ontology.

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 for categorizing the document or even relating it to other documents and known concepts in the knowledge space of a person or a company.

5.1. Ontology-based information extraction

We use an OBIE system to extract relevant instances from handwritten text. By means of information extraction, unstructured text is stepwise transformed into formal knowledge relating it to the originating ontology and instance base. This is done by a pipeline of cascading extraction tasks (Adrian et al., 2009). Conceiving the extraction pipeline as black box algorithm, mandatory input parameters are the input ontology and the instance base as described in Section 3.1.

Ontology and instance base are analyzed during a preceding processing and training phase. This training has been performed on other data in previous work. Results are index structures (e.g., suffix arrays, B*-trees) and learning models (e.g., conditional random fields, k-nearest neighbor classifiers) that can be used by efficient extraction tasks inside the extraction pipeline (for details refer to Adrian et al. (2009)):

5.1.1. Segmentation

Partitions text into segments: paragraphs, sentences, and tokens. These tokens are called n-grams, as they are letter sequences of length up to a defined n. A part-of-speech tagger tags each token with its part of speech (POS).

5.1.2. Symbolization

Recognizes symbols s in the text. The similarity function $sim(s, s_e)$ matches phrases s of the text with symbols of the input ontology. For example, assuming the existence of the RDF-triple <: DFKI > `rdfs:label` "DFKI" and the text "DFKI was founded in 1988", "DFKI" is recognized as content symbol of type `rdfs:label`.

By applying existing gazetteers and regular expressions, symbolization also performs named entity recognition and structured entity recognition. By performing noun phrase chunking, noun phrases expressing candidates for names without any structure in syntax (e.g., names) are detected.

5.1.3. Instantiation

Resolves instances of the instance base for each recognized symbol s_e. Note that for a content symbol s_e several instances might exist, i.e., for each assertion $p(i_e, s_e) \in A$ i_e is a recognized instance.

In the above example, "DFKI" would now be resolved as rdfs:label of instance:DFKI. An instance candidate recognition resolves possible candidates for recognized datatype property values. Here, ambiguities may occur if more than one instance possesses the same datatype property values (e.g., first names of several persons). Candidates are disambiguated by counting resolved instances in the domain model that are related directly with an object property. As result, the ambiguous instance with a higher count of related and recognized instances is taken. Note that with this particular step, we begin to go beyond the conventional search.

5.1.4. Contextualization

Extracts facts (RDF triples) about resolved instances. Assume that $P_Q \subseteq HP$ is a set of queried properties inside the extraction template Q. Recognized facts are assertions of type $p(i_1, i_2)$ or $p(i_1, s)$ with $i_1, i_2 \in I$, $p \in P_Q$, and $s \in S$. At first, a fact candidate extraction computes all possible facts between resolved instances. Then, a set of fact selectors rates these facts according to heuristics. A known fact selector heightens rates of extracted facts that exist as triples inside the domain model.

5.1.5. Population

Creates scenario graphs in RDF format. They contain extracted values of resolved instances with those datatype property values that match with text sequences and RDF triples about object properties between these resolved instances. Scenario graphs can be filtered and ordered by confidence values in range between zero and one.

In the experiments of this paper, the evaluation will be done up to the step of instantiation. For the next steps we have applied no changes to OBIE for electronic documents (Adrian et al., 2009). Note that the task of handwriting recognition can be seen as a special symbolization task where the similarity function $sim(s, s_e)$ should also take the possible handwriting recognition errors into account. However, our approach performs the fusion of HWR and OBIE at the instantiation step in order to make more use of the knowledge base.

5.2. Fusion of HWR and OBIE

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 is a set of 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 John Lennon is given in Fig. 5.

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 k is the number of alternatives which is used for the lattice. A higher value of k would include more recognition alternatives. This could, on the one hand, lead to the acceptance of instances that were not correctly recognized. On the other hand, other instances could be found which only have a short edit-distance to the written word. Note that setting $k = 1$, $n =$ 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:

$$k^n * (\text{word count} - n + 1) \quad (2)$$

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.

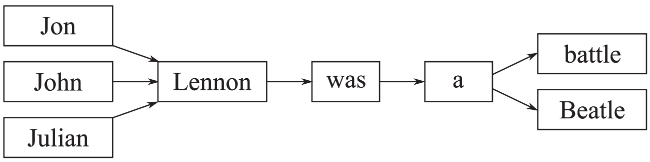
The following example illustrates the behavior of the algorithm. Let us consider the recognition of a text about John Lennon:

> Handwritten text: *"John Lennon was a Beatle"*.
> Available instances: *Beatles*, *Beatle*, *John Lennon*, *Julian Lennon*.
> HWR output: (*Jon*, *John*, *Julian*), (*was*), (*a*), (*battle*, *Beatle*).

The recognition lattice is depicted in Fig. 5.

The resulting n-grams for $n = 1$ and $n = 2$ and $k = 1, 2, 3$ are listed in the bottom of Fig. 5. Using the combination of $n = 1$, $k = 1$ no instance would be found. By setting $k = 2$, the group name *Beatle* would be found, but John Lennon is still unknown. If we use $n = 2$ and $k = 2$ Beatle and John Lennon would be found. When $k = 3$, there could be a misinterpretation, because the instance Julian Lennon (the son of John) would also be found, which corresponds to a *false positive*, because Julian has not been a member of the Beatles. This problem is tackled by the OBIE process, i.e., the probability (*belief value*) for Julian Lennon would be significantly lower than that for John Lennon.

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.



	n=1	n=2
k=1	Jon, Lennon, was, a, battle	Jon Lennon, Lennon was, was a, a battle
k=2	Jon, John, Lennon, was, a, battle, *Beatle*	Jon Lennon, *John Lennon*, Lennon was, was a, a battle, a *Beatle*
k=3	Jon, John, Julian, Lennon, was, a, battle, *Beatle*	Jon Lennon, *John Lennon*, *Julian Lennon*, Lennon was, was a, a battle, a *Beatle*

Fig. 5. Recognition lattice for alternates of *"John Lennon was a Beatle"* (top) and resulting n-grams (bottom).

Table 4
Reference f-measures.

Reference	f-measure (%)
OBIE on Set 3 (unstructured text)	73.83
n = no. of words, $k = 1$	57.92
OBIE on Set 4 (handwritten notes)	63.59
n = no. of words, $k = 1$	51.42

$$s_p = W * \sum_{i=1}^{n} c(w_i^{(p)}) + \sum_{j=1}^{ic} b(i_j^{(p)}) \quad (3)$$

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 ith word in the path, ic is the instance count in the current path and $b(i_j^{(p)})$ is the belief value of the jth instance in the path.

5.3. Experiments

For the experiments described in this section we mainly used the third data set as described in Section 4.4. Furthermore, we have validated the general behavior on the fourth data set.

Within the texts of Set 3 there is a total of 402 instances that could be recognized by the system. Based on the algorithm described in Section 5.2 we tested different parameter combinations of $n = 1, 2, 3, 4, 5$ (size of the n-gram), $k = 1, 2, 3$ (number of recognition alternatives) and $W = 0.1, \ldots, 5.5$ (weighting factor in Eq. 3). Furthermore, the case $k = 1,\ n =$ number of words was tested as a reference system, which just passes the recognizer's output to the OBIE process. To assess the upper bound of the algorithm, we also tested the information extraction performance on the ground-truth of the handwriting recognition process.

5.4. Results

The OBIE results were measured by means of precision and recall. Precision is the percentage between the number of correct retrieved instances and the number of all retrieved instances. Recall is the percentage between the number of correct retrieved instances and the number of correct instances within the text. To be able to compare different test results, balanced f-measure is used, that takes precision as well as recall into account.

$$\text{precision} = \frac{\text{No. of correct retrieved information}}{\text{No. of retrieved information}} \quad (4)$$

$$\text{recall} = \frac{\text{No. of correct retrieved information}}{\text{No. of overall correct information}} \quad (5)$$

$$\text{f-measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

The reference values of the information extraction experiments on Sets 3 and 4 appear in Table 4. The upper bound for the f-measure on Set 3 would be 73.83%. While this value seems to be quite small, it is already practically useful, since the user finally evaluates the gathered information and only chooses those instances which are correct in his or her point of view. The naive approach of just passing the HWR result to the OBIE process yields a performance of 57.92%. The main reason is that nearly 22% of the words were not correctly recognized. Note that the performance of OBIE

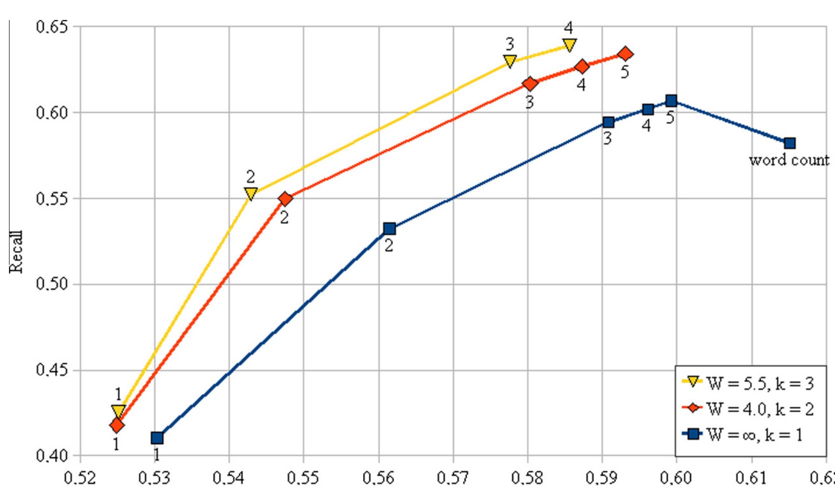


Fig. 6. F-measures, recision, and recall on Set 3 for different parameter configurations.

on Set 4 is about 10% lower than the performance on handwritten texts. This might be due to the general nature of notes where natural language processing is more difficult.

The table in Fig. 6 shows the f-measure on Set 3 for different n and k. As can be seen, the f-measure depends on the size of n. Further, increasing k improves the f-measure as well. The best system uses the 2 best recognition alternatives and 5-grams. It performs with 59.68%. Note that we did not test 5-grams with the 3 best recognition alternatives, because it becomes computationally very expensive (more than one minute per text), which is practically not useful anymore. Future work will be to increase the processing speed of the system.

Detailed results are shown in Fig. 6. There, the precision and recall values for the three parameter configurations are given. All data points are labeled with the size of n. The figure demonstrates, that n has a big effect on both values (precision and recall). This is mainly because some instances have a longer size. Note, however, that if we increase the value of n, the processing time also increases (cf. Eq. 2).

On Set 4 the general observation, i.e., that n has a big effect on the performance, has been confirmed. However, the best value was quite small, i.e., $n = 2$. This again shows that handwritten notes have a different nature than handwritten texts.

5.5. Results of combined approach

In a final experiment we measured the performance of a combined approach, i.e., first applying the improved HWR described in Section 4 and then applying the methods described in Section 5. The final f-measure of the integrated HWR/OBIE system on text could be increased to 69.67%. This value is a remarkable increase of the performance and comes close to the reference result, i.e., the 73.83% when information extraction was applied on the clean ASCII-Text. A similar observation has been made on the handwritten notes. Thus, we can conclude that the performance of our integrated approach is comparable to the performance of OBIE on clear ASCII text.

6. Conclusion

In this paper we have proposed several ways to bridge the gap between handwriting recognition and knowledge management. First, an approach to include semantic information into the recognition phase has been described. Assuming that the main topic of the handwritten notes 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 results motivate further research on including semantic information into handwriting recognition. We plan to perform experiments on a larger set of writers using more and different documents and knowledge bases. It will also be interesting to use a recognizer where we can directly control the influence of the word list.

Another interesting point for future research is to investigate the inclusion of semantic information in similar areas. Recent research focused on whole book recognition (Xiu and Baird, 2008). 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 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 RDF instances to alter the recognition lexicon increased the f-measure up to 69.67% which is close to the performance on electronic text.

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. The main bottleneck currently is the string matching between the n-grams and the instances of the ontology, which could be speeded up by the use of Hash-maps.

The approach proposed in this paper can be useful for many note-taking scenarios. Furthermore, please note that the output of the method could be regarded as the topic of the document (and it is relation to known concepts). This system therefore also solves the problem of document classification and categorization, just on a higher level, i.e., the ontology-level.

Noteworthy, the approaches proposed in this paper are very general. The same methodologies can be easily used for speech recognition as well. Furthermore, the usage of ontologies represented in RDF format allows the system to be applied in any area where structured RDF data exist, e.g., any of the knowledge bases in the LOD-cloud.

An interesting topic for future research is to automatically decide which domain ontology has to be chosen based on the handwriting given as an input in scenarios where the context is not available beforehand. An idea would be to apply a hybrid approach which first applies a simple OBIE approach using a general knowledge base, such as the DBPedia. Based on the intermediate results, a more specific knowledge base might be selected and the handwriting recognition and information extraction results might be improved.

Acknowledgments

We would like to thank Mennatallah Ghoneim and Hassan Mohamed Abou Eisha for performing parts of the experiments and Markus Weber for his support during implementing a usable system.

References

Adrian, B., Hees, J., van Elst, L., Dengel, A., 2009. iDocument: Using ontologies for extracting and annotating information from unstructured text. In: KI 2009: Advances in Artificial Intelligence. LNAI, vol. 5803. Springer, pp. 249–256.

Bunke, H., 2003. Recognition of cursive Roman handwriting – past, present, and future. In: Proc. 7th Internat. Conf. on Document Analysis and Recognition, vol. 1. pp. 448–459.

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

Dengel, A.R., 2007. Knowledge technologies for the social semantic desktop. In: Proc. of the 2nd Internat. Conf. on Knowledge Science, Engineering and Management.

Ebert, S., Liwicki, M., Dengel, A., 2010. Ontology-based information extraction from handwritten documents. In: 12th Internat. Conf. on Frontiers in Handwriting Recognition, pp. 483–488.

Gusfield, D., 1997. Algorithms on Strings, Trees, and Sequences. Cambridge University Press.

Heath, T., Bizer, C., 2011. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool.

Chaudhuri, B.B., Srihari, S.N., Schomaker, L., 2010, Internat. Conf. on Frontiers in Handwriting Recognition, 16–18 November, 2010. IEEE Computer Society, Kolkata, India.

Knerr, S., Anisimov, V., Baret, O., Gorsky, N., Price, D., Simon, J., 1997. The A2iA INTERCHEQUE system: Courtesy amount and legal amount recognition for French checks. Internat. J. Pattern Recognition Artificial Intell. 11 (4), 505–548.

Manmatha, R., Han, C., Riseman, E., 1996. Word spotting: A new approach to indexing handwriting. In: Proc. Computer Vision and Pattern Recognition, pp. 631–637.

Pena Saldarriaga, S., Viard-Gaudin, C., Morin, E., 2010. Combining approaches to on-line handwriting information retrieval. In: Proc. SPIE Document Recognition and Retrieval XVII, vol. 7534, pp. 753403–753403–8.

Perraud, F., Viard-Gaudin, C., Morin, E., 2006. Language independent statistical models for on-line handwriting recognition. In: Internat. Workshop on Frontiers in Handwriting Recognition, pp. 435–440.

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

Plamondon, R., Srihari, S.N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intell. 22 (1), 63–84.

Sauermann, L., Bernardi, A., Dengel., A., 2005. Overview and outlook on the semantic desktop. In: 6th Internat. Semantic Web Conference, pp. 1–19.

Sauermann, L., van Elst, L., Dengel, A., 2007. PIMO – a framework for representing personal information models. In: Proc. of I-SEMANTICS 2007, pp. 270–277.

Sayre, K.M., 1973. Machine recognition of handwritten words: A project report. Pattern Recognition 5 (3), 213–228.

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

Wimalasuriya, D.C., Dou, D., 2010. Ontology-based information extraction: An introduction and a survey of current approaches. J. Inf. Sci. 36, 306–323.

Xiu, P., Baird, H.S., 2008. Towards whole-book recognition. Document Analysis Systems, pp. 629–636.