Rule-Based Information Extraction for Structured Data Acquisition using TEXTMARKER

Martin Atzmueller and Peter Kluegl and **Frank Puppe**
Department of Computer Science, University of Würzburg, Germany
{atzmueller, pkluegl, puppe}@informatik.uni-wuerzburg.de

Abstract

Information extraction is concerned with the location of specific items in (unstructured) textual documents, e.g., being applied for the acquisition of structured data. Then, the acquired data can be applied for mining methods requiring structured input data, in contrast to other text mining methods that utilize a bag-of-words approach.

This paper presents a semi-automatic approach for structured data acquisition using a rule-based information extraction system. We propose a semi-automatic process model that includes the TEXTMARKER system for information extraction and data acquisition from textual documents. TEXTMARKER applies simple rules for extracting blocks from a given (semi-structured) document, which can be further analyzed using domain-specific rules. Thus, both low-level and higher-level information extraction is supported. We demonstrate the applicability and benefit of the approach with two case studies of two real-world applications.

1 Introduction

Textual documents contain a lot of unstructured data. Information extraction systems are then applied in order to generate structure data using the source documents, i.e., for generating structured instances (cases) containing the extracted information. The data bases containing the structured instances can then be applied in multiple ways, e.g., for data mining or text mining methods that do not employ the common bag-of-words representation for textual data but structured instances.

The extracted instances can be considered at different levels of granularity: Corresponding to the quality of the features (of the instances) that we want to generate, there are different levels of difficulty when generating these features. The latter range from blocks of words, to sentences, phrases, and finally concepts. A general information extraction system should support all these different options in order to be broadly applicable for different domains. Another issue concerns the ease of use of the system and its applicability: Automatic information extraction systems are usually applied when there is a lot of labeled training data. Rule-based systems, for which the rules are manually or semi-automatically acquired, are commonly applied if there is not enough training data available, or if the considered domain is too difficult to handle using purely automatic methods.

In this paper, we propose a semi-automatic approach for rule-based structured data acquisition from text. The user can specify simple rules that consider features of the text, e.g., structural or syntactic features of the textual content. These rules are then applied by the TEXTMARKER system for information extraction from text. Using its flexible rule-based formalism TEXTMARKER supports both low-level information extraction tasks such as named entity recognition, but also higher-level tasks since the extracted concepts can also be processed using specialized rules.

Rules are especially suitable for the proposed information extraction task since they allow a concise and declarative formalization of the relevant domain knowledge that is especially easy to acquire, to comprehend and to maintain. Furthermore, in the case of errors, the cause can easily be identified by tracing the application of the individual rules. Especially the latter feature is rather important for an effective application of such an approach. Since the person applying the system does not necessarily need to be a specialist concering the rules for text extraction, a simple and intuitive way of signaling and tracing errors is necessary for supporting these types of users. In the past, we have considered other approaches for structured data acquisition from text, e.g., [Betz *et al.*, 2005]: The technique was applied sucessfully at the initial development stage. However, the maintenance of the formalized knowledge and the practical support of an inexperienced user in the case of errors proved to be a significant problem.

Therefore, we opted for a more robust alternative, and developed the TEXTMARKER system as a powerful system for rule-based information extraction. It can be applied very intuitively, since the used rules are especially easy to acquire and to comprehend. Using the extracted information, data records can be easily created in a post-processing step. So far, we have applied the system for two real-world projects: The first case study concerns the extraction of medical data from a phyisician's letter (discharge letter). The letter contains the observations and the diagnoses for a specific patient. After the relevant information have been extracted, a record for the patient can be created quite easily. The second project concerns a technical domain. TEXTMARKER is applied for generating structured data records from textual (Word-)documents.

The rest of the paper is organized as follows: Section 2 presents the process model for rule-based structured data acqusition from text. In Section 3 we introduce the TEXTMARKER system and discuss related work. Section 4 presents the two case studies of the presented approach for two real-world applications. Finally, Section 5 concludes the paper with a discussion of the presented work and promising directions for future work.

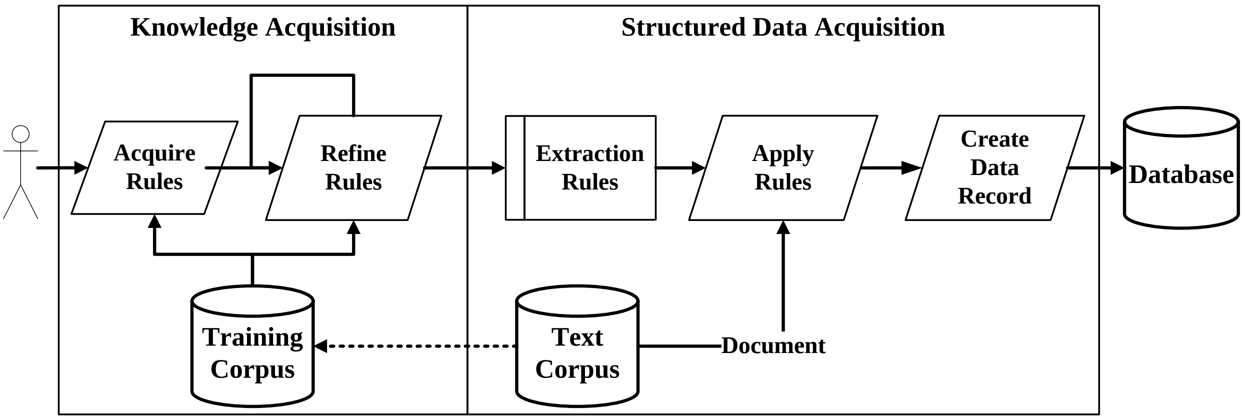


Figure 1: Process Model: Semi-Automatic Rule-Based Structured Data Acquisition from Texts

2 Process Model for Rule-Based Structured Data Acquisition

In the following section, we describe the semi-automatic process model for rule-based text extraction for generating structured data records. We utilize the TEXTMARKER system (c.f., Section 3) as the core component of the process. TEXTMARKER is a flexible integrated system for the extraction of textual information from unstructured or semi-structured documents.

In general, the proposed process consists of two phases: The knowledge acquisition phase, and the structured data acquisition phase. The knowledge acquisition phase necessarily precedes the structured data acquisition phase: In a semi-automatic process, the necessary knowledge for structured data acquisition from text can be formalized by a domain specialist, and can be tested on a training corpus containing a set of typical documents from the domain. The knowledge is given by a set of specific extraction rules. The process is incremental such that the extraction performance can be used for optimizing the set of extraction knowledge. The obtained knowledge provided by a set of extraction rules are then applied by the TEXTMARKER system described below.

In the structured data acquisition phase the formalized knowledge, i.e., the formalized rules, are applied on the (new) documents in a straight-forward manner. Given a document, for example, a set of segments (blocks of words) can be extracted, and further specialized rules can be applied for extracting specific concepts. Then, the structured data is created and inserted into the database.

Alltogether, the process for rule-based text extraction and acquisition of structured data considers the following steps that are shown in Figure 1:

1. **Knowledge Acquisition Phase**:
   (a) *Acquire Extraction Rules*: Using a set of training documents, usually an initial set of extraction rules is formalized by a domain specialist, based on the features of the training documents and the concepts to be extracted. Therefore, the training documents should ideally capture typical characteristics of the documents encountered in the practical application.
   (b) *Refine Rules*: Using the given rules, the user can tune and refine these in incremental fashion. In this way, also extensions and changes for the document corpus can be easily included.

2. **Data Acquisition Phase**:
   (a) *Apply Rules*: After the knowledge acquisition phase a set of extraction rules is available. These can then be applied for each document of the text corpus, and the output can be created. For this step, for example, segments or words of the document can be considered, but also *annotations*, that were generated during the process, can also be utilized. In this way both low-level and high-level information extraction tasks can be implemented. In general, the output is domain-specific, but in the context of the presented work usually attribute–value pairs will be considered.
   (b) *Create Data Record*: In this step, the set of attribute–value pairs (concepts) is applied for creating the final data record. The specific implementation of this step is domain-dependent, and can vary from a simple matching of concepts to more sophisticated natural language processing techniques. In the case studies in Section 4 we discuss some exemplary techniques.

The *input* of the process, i.e., the knowledge acquisition phase is usually given by a set of training documents that are used for optimization and refinement of the set of extraction rules. Although the domain specialist could also provide a set of rules directly, in practice validating these with a set of typical documents will usually increase the performance of the system. The applied training corpus can consist of the complete text corpus, but usually also a representative sample of these documents is sufficient for obtaining valid results. However, in practice the text corpus usually grows over time, therefore the process can also be iterated including further documents in the training corpus.

The *output* of the process is a set of structured data records to be integrated in a database. In general, the output of the data acquisition phase can be specified quite flexible: Since TEXTMARKER provides several output options including direct textual output, modifying the input document, and also the connection with programming languages, the user can provide flexible solutions that are also easily extensible. In the context of the presented work, the output for the creation of structured data records will usually consist of attribute–value pairs that are subsequently included in the created data records.

3 Information Extraction using TextMarker– An Overview

In manual information extraction humans often apply a strategy according to a *highlighter metaphor*: First relevant headlines are considered and classified according to their content by coloring them with different highlighters. The paragraphs of the annotated headlines are then considered further. Relevant text fragments or single words in the context of that headline can then be colored. In this way, a *top-down analysis and extraction* strategy is implemented. Necessary additional information can then be added that either refers to other text segments or contains valuable domain specific information. Finally the colored text can be easily analyzed concerning the relevant information.

The TextMarker system[1] tries to imitate this manual extraction method by formalizing the appropriate actions using *matching rules*: The rules mark sequences of words, extract text segments or modify the input document depending on textual features. The current TextMarker implementation is based on a prototype described by [von Schoen, 2006] that supports a subset of the TextMarker language described below. The present TextMarker system is currently being extended towards a rich client application and an integration as a UIMA component [Götz and Suhre, 2004; Ferrucci and Lally, 2004]. This enables a feature rich development environment with powerful debugging capabilities and also an easy reusability of the components.

The default input for the TextMarker system is semi-structured text, but it can also process structured or free text. Technically, HTML is often the input format, since most word processing documents can be converted to HTML. Additionally, the TextMarker systems offers the possibility to create a modified output document.

In the following sections we first give a short conceptual overview on the TextMarker language and introduce its core concepts. After that, we discuss the syntax and the semantics of the TextMarker language in detail, and provide several illustrating examples. Next, we present special characteristics of the language that distinguishes the TextMarker system from other rule based information extraction systems, and discuss related work.

3.1 Core TextMarker Concepts

As a first step in the extraction process the TextMarker system uses a tokenizer (scanner) to tokenize the input document and to create a stream of basic symbols. The types and valid annotations of the possible tokens are predefined by a taxonomy of *annotation types*. Annotations simply refer to a section of the input document and assign a type or concept to the respective text fragment.

Figure 2 shows an excerpt of a basic annotation taxonomy: *CW* describes all tokens, for example, that contains a single word starting with a capital letter, *MARKUP* corresponds to HTML or XML tags, and *PM* refers to all kinds of punctuations marks.

By using (and extending) the taxonomy, the knowledge engineer is able to choose the most adequate types and concepts when defining new *matching rules*, i.e., TextMarker rules for matching a text fragment given by a set of symbols to an annotation. If the capitalization of a word, for example, is of no importance, then the annotation type *W* that describes words of any kind can be used.

The initial scanner creates a set of basic annotations that may be used by the matching rules of the TextMarker language. However, most information extraction applications require domain specific concepts and annotations. Therefore, the knowledge engineer is able to extend the set of annotations, and to define new annotation types tuned to the requirements of the given domain. These types can be flexibly integrated in the taxonomy of annotation types.

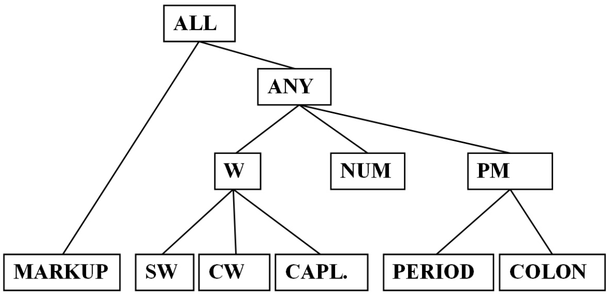


Figure 2: Part of a taxonomy for basic annotation types (W=Word, NUM=Number, PM=Punctuations, SW=Word without capitals, CW=Word starting with a capital letter).

3.2 Syntax and Semantics of the TextMarker Language

One of the goals in developing a new information extraction language was to maintain an easily readable syntax while still providing a scalable expressiveness of the language. Basically, the TextMarker language contains expressions for the definition of new annotation types and for defining new matching rules. The rules are defined by expressions containing a list of rule elements headed by the type of the rule.

The purpose of the different rule types is to increase the readability of rules by making their semantic intention explicit. Each rule element contains at least a basic matching condition referring to text fragments or already specified annotations. Additionally a list of conditions and actions may be specified for a rule element. Whereas the conditions describe necessary attributes of the matched text fragment, the actions point to operations and assignments on the current fragments. These actions will then only be executed if all basic conditions matched on a text fragment or the annotation and the related conditions are fulfilled. Table 1 summarizes the TextMarker grammer for defining matching rules and annotations. It contains an excerpt of the TextMarker syntax in Backus-Naur-Form (BNF) concerning the rule definitions.

Due to the limited space it is not possible to describe all of the various conditions and actions available in the TextMarker system. However, the usage of the language and its readability can be demonstrated by simple examples:

```
ADDTYPE CW{INLIST,animals.txt}(MARK,animal)
ADDTYPE animal 'and' animal
        (MARK,animalpair,0,1,2)
```

The first rule looks at all capitalized words that are listed in an external document *animals.txt* and creates a new annotation of the type *animal* using the boundaries of the matched word. The second rule searches for an annotation of the type *animal* followed by the literal *and* and a second *animal* annotation. Then it will create a new annotation *animalpair* covering the text segment that matched the three

[1] textmarker is a common german word for a highlighter

Rule	→ RuleType RuleElement+
RuleType	→ ’ADDTYPE’ | ’DEFAULT’ | . . .
RuleElement	→ MatchType Conditions? Actions?’+’?
MatchType	→ Literal | Annotation
Annotation	→ ’ALL’|’ANY’|’MARKUP’|’W’|. . .
Conditions	→ ’{’ Condition (’;’ Condition)* ’}’
Condition	→ ’-’? CondType (’,’ parameter)*
CondType	→ ’PARTOF’|’CONTAINS’|’NEAR’|. . .
Actions	→ ’(’ Action (’;’ Action)* ’)’
Action	→ ActionType (’,’ parameter)*
ActionType	→ ’MARK’|’FILTER’|’REPLACE’|. . .

Table 1: Extract of the TEXTMARKER language definition in Backus-Naur-Form

rule elements (the digit parameters refer to the number of matched rule element).

```
ADDTPYE W(MARK,firstname,firstnames.txt)
ADDTYPE firstname CW(MARK,lastname)
LOGGING paragraph{VOTE,firstname,lastname}
        (LOG,’Found more firstnames than
        lastnames’)
```

In this example, the first rule annotates all words that occur in the external document *firstnames.txt* with the type *firstname*. The second rule creates a *lastname* annotation for all capitalized word that follow a *firstname* annotation. The last rule finally processes all *paragraph* annotations. If the *VOTE* condition counts more *firstname* than *lastname* annotations, then the rule writes a log entry with a predefined message.

```
ADDTYPE ANY{PARTOF,paragraph,ISINTAG,
        font,color=red}(MARK,delete,+)+
ADDTYPE firstname(MARK,delete,0,1) lastname
DEFAULT delete(DEL)
```

Here, the first rule looks for sequences of any kind of tokens except markup and creates one annotation of the type *delete* for each sequence, if the tokens are part of a *paragraph* annotation and colored in red. The + signs indicate this greedy processing. The second rule annotates first names followed by last names with the type *delete* and the third rule simply deletes all text segments that are associated with that *delete* annotation.

3.3 Special Features of the TEXTMARKER Language

The TEXTMARKER language features some special characteristics that are usually not found in other rule-based information extraction systems or even shift it towards scripting languages. The possibility of creating new annotation types and integrating them into the taxonomy facilitates an even more modular development of information extraction systems than common rule based approaches do. Beside others, there are two features that deserve a closer look in the scope of this work: The robust extraction by filtering the token or annotation set and the usage of scoring rules for uncertain and heuristic extraction.

Robust extraction using filtering

Rule based or pattern based information extraction systems often suffer from unimportant fill words, additional whitespace and unexpected markup. The TEXTMARKER System enables the knowledge engineer to filter and to hide all possible combinations of predefined and new types of annotations. Additionally, it can differentiate between every kind of HTML markup and XML tags. The visibility of tokens and annotations is modified by the actions of rule elements and can be conditioned using the complete expressiveness of the language. Therefore the TEXTMARKER system supports a robust approach to information extraction and simplifies the creation of new rules since the knowledge engineer can focus on important textual features. If no rule action changed the configuration of the filtering settings, then the default filtering configuration ignores whitespaces and markup. Using the default setting, the following rule matches all four types of input in this example (see [von Schoen, 2006]):

```
DEFAULT ’Dr’ PERIOD CW CW
```

```
Dr. Peter Steinmetz
Dr . Peter      Steinmetz
Dr. <b><i>Peter</i> Steinmetz</b>
Dr.PeterSteinmetz
```

Heuristic extraction using scoring rules

Diagnostic scores are a well known and successfully applied knowledge formalization pattern for diagnostic problems [Puppe *et al.*, 2001]. Single known findings valuate a possible solution by adding or subtracting points on an account of that solution. If the sum exceeds a given threshold, then the solution is derived. One of the advantages of this pattern is the robustness against missing or false findings, since a high number of findings is used to derive a solution. For more information on the diagnostic score pattern see, e.g., [Puppe, 2000].

The TEXTMARKER system tries to transfer this diagnostic problem solution strategy to the information extraction problem. In addition to a normal creation of a new annotation, a *MARK* action can add positive or negative scoring points to the text fragments matched by the rule elements. If the amount of points exceeds the defined threshold for the respective type, then a new annotation will be created. Further, the current value of heuristic points of a possible annotation can be evaluated by the *SCORE* condition. In the following, the heuristic extraction using scoring rules is demonstrated by a short example:

```
ADDTYPE paragraph{CONTAINS,W,1,5}(MARK,
        headline,5)
ADDTYPE paragraph{CONTAINS,W,6,10}(MARK,
        headline,2)
ADDTYPE paragraph{CONTAINS,emph,80,100,%}
        (MARK,headline,7)
ADDTYPE paragraph{CONTAINS,emph,30,80,%}
        (MARK,headline,3)
ADDTYPE paragraph{CONTAINS,CW,50,100,%}
        (MARK,headline,7)
ADDTYPE paragraph{CONTAINS,W,0,0}(MARK,
        headline,-50)
ADDTYPE headline{SCORE,10}(MARK,realhl)
LOGGING headline{SCORE,5,10}(LOG,
        ’Maybe a headline’)
```

In the first part of this rule set, annotations of the type *paragraph* receive scoring points for a *headline* annotation, if they fulfill certain *CONTAINS* conditions. The first condition, for example, evaluates to *true*, if the paragraph contains one word up to five words, whereas the fourth conditions is fulfilled, if the paragraph contains thirty up to eighty percent of *emph* annotations. The last two rules finally execute their actions, if the score of a *headline* annotation exceeds ten points, or lies in the interval of five and ten points, respectively.

3.4 Related Work and Discussion

Information extraction and the related acquisition of structured data are part of a widespread and still growing scientific community that originates a multiplicity of new systems, tools and approaches. Many systems for processing structured and semi-structured texts can be found in the area of web information extraction systems [Kaiser and Miksch, 2005].

One of these systems is the LAPIS system (*Lightweight Architecture for Processing Information Structure*) [Kuhlins and Tredwell, 2003] that executes self-explanatory script-like edit operations on the input document. Providing a graphical user interface with an integrated browser, this system allows to revise the extraction results in a HTML view. But its original purpose as innovative text editor causes a lack of some essential concepts like the definition of new types and the representation of uncertainty that is necessary for the effective text extraction.

Another system for extraction information especially for text extraction from the web is the LIXTO SUITE [Baumgartner *et al.*, 2001b] with its LIXTO VISUAL WRAPPER. This system provides a graphical user interface for a semi-automatic generation of wrappers. The supervised learning approach uses manual annotations and decisions of the user to learn and refine rules of the ELOG language [Baumgartner *et al.*, 2001a]. Therefore there is no knowledge about the language representation or HTML structure needed to created an appropriate wrapper. But its visual programming approach seems to prefer simple conditions instead of complex ones that would increase the robustness of the wrapper.

The PHOENIX system [Betz *et al.*, 2005] uses a rule set in a proprietary XML syntax to recursively segment relevant blocks of text. The rules are composed of a set of XPATH statements that make a preliminary selection of interesting text segments and use constraining conditions to increase their precision. This system was especially developed to extract case information from documents created by common word processing programs, but depends significantly on a predefined structure of the documents by which the robustness of its extraction process is affected significantly.

Various tools and approaches are available to extract information from semi-structured texts and for the creation of structured data records (e.g., [Mustafaraj *et al.*, 2007]). A prominent example is given by the DISCOTEX system by [**?**] that applies a learning component for generating the information extraction system. After textual documents have been processed, a data mining component can then be applied for the specific knowledge discovery step. While DISCOTEX also proposes a process model for text extraction and mining, the process presented in this work is more general. It focuses on the semi-automatic acquisition of extraction functionality, using rules, that is applicable for both low-level and high-level text extraction systems.

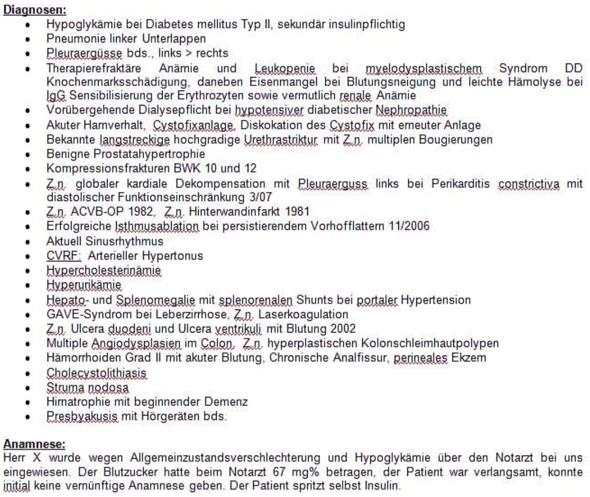
In summary, no system fulfilled all requirements for the core system of this process. This motivated the new development of the TEXTMARKER system with the described features: The modeling of extraction knowledge using rule-based patterns, the intuitive knowledge acquisition supported by graphical editors, the powerful features of the TEXTMARKER language and its extensibility prove crucial when developing efficient and effective text extraction approaches for structured data record creation.

4 Case Studies

In the following sections we describe two real-world case studies applying the presented approach. The first case study considers the generation of structured data records given semi-structured medical discharge letters. The second case study is concerned with high-level information extraction and data acquisition in a technical domain.

4.1 Generating Structured Discharge Letters

The first case study considers the generation of data records from semi-structured medical discharge letters. These letters are written by the physicians when a patient has been diagnosed and leaves after a hospital stay. The letters are typically written by the responsible physicians themselves and are stored as Office (Word) documents. These contain the observations, for example, the history of the patient, results from certain examinations, measurements of laboratory parameters, and finally the inferred diagnoses of the patient. Figure 3 shows an example of the diagnoses and the history part of an (anonymized) discharge letter. The available electronic discharge letters provide the basis for various purposes, for example, for quality control with respect to a hospital information system, for medical evaluations, or for creating case-based training sessions. However, the manual formalization and record creation using these is quite costly. Therefore, we applied the presented approach for structured data acquisition from the textual documents.



Diagnosen:

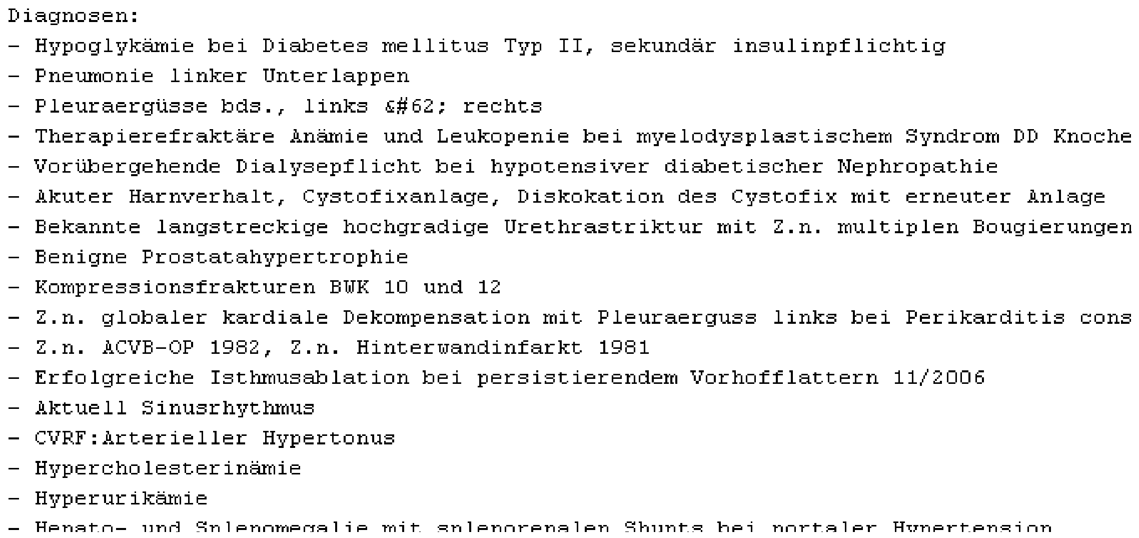
- Hypoglykämie bei Diabetes mellitus Typ II, sekundär insulinpflichtig
- Pneumonie linker Unterlappen
- Pleuraergüsse bds., links > rechts
- Therapierefraktäre Anämie und Leukopenie bei myelodysplastischem Syndrom DD Knochenmarksschädigung, daneben Eisenmangel bei Blutungsneigung und leichte Hämolyse bei IgG Sensibilisierung der Erythrozyten sowie vermutlich renale Anämie
- Vorübergehende Dialysepflicht bei hypotensiver diabetischer Nephropathie
- Akuter Harnverhalt, Cystofixanlage, Diskokation des Cystofix mit erneuter Anlage
- Bekannte langstreckige hochgradige Urethrastriktur mit Z.n. multiplen Bougierungen
- Benigne Prostatahypertrophie
- Kompressionsfrakturen BWK 10 und 12
- Z.n. globaler kardiale Dekompensation mit Pleuraerguss links bei Perikarditis constrictiva mit diastolischer Funktionseinschränkung 3/07
- Z.n. ACVB-OP 1982, Z.n. Hinterwandinfarkt 1981
- Erfolgreiche Isthmusablation bei persistierendem Vorhofflattern 11/2006
- Aktuell Sinusrhythmus
- CVRF: Arterieller Hypertonus
- Hypercholesterinämie
- Hyperurikämie
- Hepato- und Splenomegalie mit splenorenalen Shunts bei portaler Hypertension
- GAVE-Syndrom bei Leberzirrhose, Z.n. Laserkoagulation
- Z.n. Ulcera duodeni und Ulcera ventrikuli mit Blutung 2002
- Multiple Angiodysplasien im Colon, Z.n. hyperplastischen Kolonschleimhautpolypen
- Hämorrhoiden Grad II mit akuter Blutung, Chronische Analfissur, perineales Ekzem
- Cholecystolithiasis
- Struma nodosa
- Hirnatrophie mit beginnender Demenz
- Presbyakusis mit Hörgeräten bds.

Anamnese:

Herr X wurde wegen Allgemeinzustandsverschlechterung und Hypoglykämie über den Notarzt bei uns eingewiesen. Der Blutzucker hatte beim Notarzt 67 mg% betragen, der Patient war verlangsamt, konnte initial keine vernünftige Anamnese geben. Der Patient spritzt selbst Insulin.

Figure 3: Example of a discharge letter (in German): The screenshot shows the diagnoses ("Diagnosen: ...") and the history part ('Anamnese: ...'). The segments corresponding to these need to be extracted for the case creation.

We started with a training corpus of 43 discharge letters. The goal was then to process these and to extract the relevant information (observations, diagnoses) in order to create data records for later evaluation and text mining. Discharge letters usually follow certain formalization patterns: The document is started by the salutation, the diagnosis part, the history of the patient, textual paragraphs describing the results of various examinations like computer tomography (CT), and the result of laboratory examinations, i.e., the measured parameters. Since this structure is followed quite strictly, we were able to utilize this feature for extracting the relevant observations.



```
Diagnosen:
- Hypoglykämie bei Diabetes mellitus Typ II, sekundär insulinpflichtig
- Pneumonie linker Unterlappen
- Pleuraergüsse bds., links &#62; rechts
- Therapierefraktäre Anämie und Leukopenie bei myelodysplastischem Syndrom DD Knoche
- Vorübergehende Dialysepflicht bei hypotensiver diabetischer Nephropathie
- Akuter Harnverhalt, Cystofixanlage, Diskokation des Cystofix mit erneuter Anlage
- Bekannte langstreckige hochgradige Urethrastriktur mit Z.n. multiplen Bougierungen
- Benigne Prostatahypertrophie
- Kompressionsfrakturen BWK 10 und 12
- Z.n. globaler kardiale Dekompensation mit Pleuraerguss links bei Perikarditis cons
- Z.n. ACVB-OP 1982, Z.n. Hinterwandinfarkt 1981
- Erfolgreiche Isthmusablation bei persistierendem Vorhofflattern 11/2006
- Aktuell Sinusrhythmus
- CVRF:Arterieller Hypertonus
- Hypercholesterinämie
- Hyperurikämie
- Hepato- und Splenomegalie mit splenorenalen Shunts bei portaler Hypertension
```

Figure 4: Intermediate results: Exemplary segments (for the diagnoses part) for the discharge letter shown in Figure 3.

For extracting the relevant information, we developed a set of rules that take the structure of the document into account. As discussed above, a discharge letter needs to follow a certain standard structure. For applying the TEXTMARKER system, we could therefore focus on these building blocks of the document. In this way, we developed a set of rules for extracting segments of the letter first, for example, considering the diagnosis block (c.f., Figure 4). After that, those segments were split up further, for example, considering the fact that individual diagnoses are almost always contained in separate lines within these segments. Some examples of the applied extraction rules for the diagnoses are shown below.

```
DECLARE diagnosisStartMarker
ADDTYPE CW{PARTOF,paragraph;-PARTOF,
        diagnosisStartMarker;INLIST,
        SynonymeAdditional.txt}(MARK,
        diagnosisStartMarker,0,1)
        CW{INLIST,SynonymeDiagnosis.txt,
        5,relative}
ADDTYPE CW{PARTOF,paragraph;-PARTOF,
        diagnosisStartMarker;INLIST,
        SynonymeDiagnosis.txt,5,relative}
        (MARK,diagnosisStartMarker)
DECLARE historyStartMarker
ADDTYPE CW{REGEXP,[A-Za-zÄÖÜäöüß]*(anamnese
        |Anamnese)}(MARK,historyStartMarker)
DECLARE startMarker
ADDTYPE diagnosisStartMarker(MARK,startMarker)
ADDTYPE historyStartMarker(MARK,startMarker)
ADDTYPE highlightedParagraph{TOTALCOUNT,
        historyStartMarker,0,0}
        (MARK,startMarker)
```

The rules concern the definition of several markers for a set of diagnoses (block of diagnoses in the discharge letter) and the start of the history section. Both the *diagnosisStartMarker* and the *StartMarker* are used for marking the start of an *interesting* paragraph, i.e., block of content. The last rule considers the case that the history, i.e., the *historyStartMarker* is missing, since for extracting the diagnoses we consider the content between the *diagnosisStartMarker* and the following *startMarker*. The last rule was added for increasing the robustness of the system. However, the case that the history was missing occurred only in a minority of cases. The diagnoses block is then split up into segments concerning the individual diagnoses.

An example of the intermediate output of the extraction phase is shown in Figure 4. After the segments have been extracted we apply a post-processing phase in which the segments are matched with specific diagnoses and observations using a lexicon and a synonym-list. This matching step can be easily implemented using the TEXTMARKER system, focusing on a set of interesting concepts (observations and diagnoses). The result of the application is a set of attribute–value pairs that can be directly applied for structured data record creation. Considering the diagnoses, for example, we create pairs for binary attributes regarding the extracted diagnoses. The resulting database of dissection records is then available, e.g., for knowledge discovery and quality monitoring.

4.2 High-Level Information Extraction

The second case study describes the application of the presented process for a high level information extraction and automatic content segmentation and extraction task. Unfortunately, we can only describe the case study in a very general way due to non-disclosure terms. Therefore, we will outline and summarize the general setting and ideas, but we will not show specific screenshots or technology.

As a general setting, word processing documents in common file formats are the input of the described system. These documents, initially[2] consisting of common Microsoft Word or OpenOffice documents need to be mined for project-like information with temporal margins, e.g., information similar to facts commonly contained in curriculums vitae.

In the concrete application, the input documents feature an extremly heterogeneous layout and are each written by a different person. Interesting text fragments may relate from plain text to structured tables, combinations of these, or parts of them. Additionally, the layout is not sufficient enough for a correct classification, since also domain dependant semantics may change the relevance of a fragment in its specific context. The output of a document are a set of templates that contain exact temporary information, the exact text fragment related to the template and various domain specific information, e.g., the responsible position or a title phrase, in our curriculum vitae analogy.

[2] The input documents are converted to HTML

Although the application is still under development, it already involves 479 rules and contains several domain specific dictionaries with up to 80000 entries. During the process, the TEXTMARKER system basically tries to imitate the human perception of text blocks when processing the documents. For this purpose interesting named entities, e.g., temporal information, are recognised. Then, the application identifies text structures of different types of complexity and size, e.g., a headlined paragraph or a row of a table. These overlapping text fragments are then compared both in a top-down and a bottom-up manner. If one of these text fragments or a set of text fragments of the same type contains a significant pattern of interesting named entities, then they are marked as a relevant block of text. Finally additional rules find the domain specific information which is also used to refine the found segments. This outline describes the basic functionality of the system. According to this specification, extraction rules were defined by the knowledge engineer using test documents provided by the domain specialist.

In the current state the described TEXTMARKER application was evaluated on correct text fragments and temporary data only. In this setting, it already achieved an F1 measure of about 89% tested on 58 randomly selected documents with 783 relevant text fragments. These results seem to indicate potential for further improvements, however, in order to obtain more reliable results we need to perform more evaluations together with our project partners first.

5 Conclusions

This paper presented an effective rule-based approach for the generation of structured data records from text: We have proposed a semi-automatic process that featured the TEXTMARKER system as the core-component for the text extraction. The paper provided a conceptual overview on the TEXTMARKER application, and described the core concepts, the TEXTMARKER language, and the acquisition and application of extraction rules. For demonstrating the applicability, benefit and effectiveness of the approach, the paper discussed two cases studies from two real-world applications.

The results and the experiences so far show, that the proposed process and the TEXTMARKER system are quite capable for implementing difficult text and information extraction tasks. Then, the application of the versatile TEXTMARKER system can also be applied as a preprocessing step for the structured data acquisition task.

In the future, we aim to consider automatic learning methods for the (semi-)automatic acquisition of extraction rules. Then, the acquisition of extraction knowledge can be supported by the system, e.g., by proposing appropriate templates for the extraction. Furthermore, we plan to extend the TEXTMARKER language in order to further simplify the creation of domain-specific annotations. Additionally, we aim to completely integrate the TEXTMARKER system with other natural-language processing tools using UIMA (Unstructured Information Management Architecture) [Ferrucci and Lally, 2004], such that the extraction process can be enhanced using further specialized tools.

Acknowledgements

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

References

[Baumgartner *et al.*, 2001a] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. The Elog Web Extraction Language. In *LPAR '01: Proceedings of the Artificial Intelligence on Logic for Programming*, pages 548–560, London, UK, 2001. Springer-Verlag.

[Baumgartner *et al.*, 2001b] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual Web Information Extraction with Lixto. In *The VLDB Journal*, pages 119–128, 2001.

[Betz *et al.*, 2005] Christian Betz, Alexander Hörnlein, and Frank Puppe. Authoring Case-Based Training by Document Data Extraction. In *Proc. 10th Intl. Workshop on Chemical Engineering Mathematics*, 2005.

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Nat. Lang. Eng.*, 10(3-4):327–348, 2004.

[Götz and Suhre, 2004] Thilo Götz and Oliver Suhre. Design and Implementation of the UIMA Common Analysis System. *IBM Syst. J.*, 43(3):476–489, 2004.

[Kaiser and Miksch, 2005] Katharina Kaiser and Silvia Miksch. Information extraction. a survey. Technical Report Asgaard-TR-2005-6, Vienna University of Technology, Institute of Software Technology and Interactive Systems, 2005.

[Kuhlins and Tredwell, 2003] Stefan Kuhlins and Ross Tredwell. Toolkits for Generating Wrappers – A Survey of Software Toolkits for Automated Data Extraction from Web Sites. In Mehmet Aksit, Mira Mezini, and Rainer Unland, editors, *Objects, Components, Architectures, Services, and Applications for a Networked World*, volume 2591 of *Lecture Notes in Computer Science (LNCS)*, pages 184–198, Berlin, October 2003. International Conference NetObjectDays, NODe 2002, Erfurt, Germany, October 7–10, 2002, Springer.

[Mustafaraj *et al.*, 2007] Eni Mustafaraj, Martin Hoof, and Bernd Freisleben. Knowledge Extraction and Summarization for an Application of Textual Case-Based Interpretation. In *Case-Based Reasoning Research and Development, Proc. 7th Intl. Conference on Case-Based Reasoning (ICCBR 2007)*, volume 4626 of *Lecture Notes in Computer Science*, pages 517–531, Berlin, 2007. Springer.

[Puppe *et al.*, 2001] Frank Puppe, Susanne Ziegler, Ulrich Martin, and Jürgen Hupp. *Wissensbasierte Diagnosesysteme im Service-Support [Knowledge-Based Systems for Service-Support]*. Springer, Berlin, 2001.

[Puppe, 2000] Frank Puppe. Knowledge Formalization Patterns. In *Proc. PKAW 2000*, Sydney, Australia, 2000.

[von Schoen, 2006] Patrick von Schoen. Textmarker: Automatische Aufbereitung von Arztbriefen für Trainingsfälle mittels Anonymisierung, Strukturerkennung und Teminologie-Matching [TextMarker: Automatic Refinement of Discharge Letters for Training Cases using Anonymization, Structure- and Terminology-Matching]. Master's thesis, University of Wuerzburg, 2006.