Information Engineering Group

Gaining Process Information from Clinical Practice Guidelines Using Information Extraction

Extended version of Kaiser, K., Akkaya, C., and Miksch, S.: Gaining Process Information from Clinical Practice Guidelines Using Information Extraction. In: Proceedings of the 10th Conference on Artificial Intelligence in Medicine, 2005.

Katharina Kaiser, Cem Akkaya, and Silvia Miksch

Vienna University of Technology
Institute of Software Technology & Interactive Systems

Asgaard-TR-2005-4 May 2005

Authors: **Katharina Kaiser, Cem Akkaya, and Silvia Miksch**

{kaiser, silvia}@asgaard.tuwien.ac.at
http://ieg.ifs.tuwien.ac.at

Contact: **Vienna University of Technology**
Institute of Software Technology & Interactive Systems

Favoritenstraße 9-11/188
A-1040 Vienna
Austria, Europe

Telephone: +43 1 58801 18839
Telefax: +43 1 58801 18899
Web http://ieg.ifs.tuwien.ac.at

Contents

Abstract

Formalizing Clinical Practice Guidelines for a subsequent computer-supported processing is a cumbersome, challenging, and time-consuming task. But currently available tools and methods do not satisfactorily support this task.

We propose a new multi-step approach using Information Extraction and Transformation. This paper addresses the Information Extraction task. We have developed several heuristics, which do not take Natural Language Understanding into account. We implemented our heuristics in a framework to apply them to several guidelines from the specialty of otolaryngology. Our evaluation shows that a heuristic-based approach can achieve good results, especially for guidelines with a major portion of semi-structured text.

Chapter 1

Introduction

Computer-supported guideline execution is an important means to improve the quality in health care. To execute Clinical Guidelines and Protocols (CGPs) in a computer-supported way, the information in the guideline, which is in plain textual form, in tables, or represented in flow charts, has to be formalized. That means that a formal representation is required in order to make the information computable. Thus, several so called *guideline representation languages* have been developed to support the structuring and representation of various guidelines and protocols and to make possible different kinds of applications.

Many researchers have proposed frameworks for modeling CGPs in a computer-interpretable and -executable format (a comprehensible overview can be found at [10]). Each of these frameworks provides specific guideline representation languages. Most of these languages are sufficiently complex that a manual formalization of CGPs is a challenging project. Thus, research has to be directed in such a way where tools and methods are developed for supporting the formalization process. Anyway, for most of these tools and methods that are already developed or in developing process the human developer needs not only the knowledge about the formal methods, but also about the medical domain. Thereby, the formalization task is very challenging, time-consuming, and cumbersome.

Thus, we will look for new approaches that can facilitate the formalization process and support the developer by providing these kinds of knowledge as well as intelligent methods for a simplified guideline modeling processing.

Chapter 2

Related Work

In this Chapter, we present a short discussion of some relevant work describing guideline formalization tools as well as some examples of Information Extraction (IE) systems.

2.1 Guideline Formalization Tools

For formalizing clinical guidelines into a guideline representation language various methods and tools exist, ranging from simple editors to sophisticated graphical applications.

2.1.1 Markup-based Tools

Stepper [12] is a tool that formalizes the initial text in multiple user-definable steps corresponding to interactive XML transformations. The result of each step is an increasingly formalized version of the source document. Both the mark-up and the iterative transformation process are carried out by rules expressed in a new transformation language based on XML. Stepper documents of all activities. So the transformation process can easily be reviewed by other users. Stepper also provides an interface showing the interconnection between the source text and the model.

The GEM Cutter [11] transforms guideline information into the GEM format [13], showing the original guideline document together with the corresponding GEM document and makes it possible to copy text from the guideline to the GEM document.

The GEM Cutter is similar to the Document Exploration and Linking Tool / Addons (DELT/A), formerly known as Guideline Markup Tool (GMT) [18, 19], which supports the translation of HTML documents into an XML language. It provides two main features: (1) linking between a textual guideline and its formal representation, and (2) applying design patterns in the form of macros.

The DELT/A allows the definition of links between the original guideline and the target representation, which gives the user the possibility to find out where a certain value in the XML-language notation comes from. Therefore, if someone wants to know the origin of a specific value in the XML file the DELT/A can be used to jump to the correlating point in the text file where the value is defined and the other way round. By means of these features the original text parts need not be stored as part of the target representation elements. The links clearly show the source of each element in the target representation. Additionally, there is no need to produce a guideline in natural language from the target representation, since the original text remains unaltered.

These tools all have in common that the authoring process has to be done manually by the human plan editor.

2.1.2 Graphic Tools

AsbruView [6] uses graphical metaphors to represent Asbru [10] plans. It is a tool to make Asbru accessible to physicians, and to give any user an overview over a plan hierarchy.

AREZZO and TALLIS [16] support the translation into PROforma [17] by means of graphical symbols representing the task types of the language. Protégé [4] is a knowledge-acquisition tool that supports the translation into guideline representation languages EON [9]and GLIF [10]. It uses specific ontologies for these languages, whereas parts of the formalization process can be accomplished with predefined graphical symbols. AREZZO, TALLIS, and Protégé represent the processes by means of flow charts.

All these methods and tools have in common that they support the formalizing process due to the complexity of both the clinical guideline and the guideline representation language only in a minor degree.

2.2 Information Extraction Systems

Information Extraction systems have been developed for various domains. For example, the BADGER system [15] is a text analysis system, which summarizes medical patient records by extracting diagnoses, symptoms, physical findings, test results, and therapeutic treatments based on linguistic concepts. At the University of Sheffield extensive research in Natural Language Processing, and especially IE, has been devoted (e.g., the AMBIT system [3] to extracting information from biomedical texts). In the legal domain, Holowczak and Adam developed a system that supports the automatic classification of legal documents [5].

Besides these domain specific systems, there are also other systems using Machine Learning techniques, which can be applied in various domains. WHISK [14], for example, learns rules for extraction from a wide range of text styles; its performance varies with the difficulty of the extraction task. RAPIER [2] uses pairs of sample documents and filled templates to induce pattern–match rules that directly extract fillers for the slots in the template.

Finally, different kinds of wrappers have been developed to transform an HTML document into an XML document (e.g., XWRAP [8] or LiXto, which provides a visual wrapper [1]). These methods and tools are very useful in the case of highly structured HTML documents or if simple XML files need to be extracted. However, CGPs are more complex and more structured XML/DTD files are needed in order to represent them.

Compared to the various methods and tools described above our approach for IE deals with very complex documents, consisting of both semi-structured and free text as well as tables. For this task we do not need to apply Natural Language Understanding, because the task demands not for understanding the text, but to detect patterns in the text. Likewise, we do not employ Machine Learning techniques due to the limited number of examples that we could use for the training of these techniques. We must also note the complexity of the information annotation task in the medical domain due to the complexity of the language used. That is why we use a knowledge engineering approach with a manual development of rules for the purposes of IE.

Chapter 3

Our General Idea: A Multi-step Transformation Process Based on Heuristic Methods

Most guideline representation languages are very powerful and thus very complex. They can contain many different types of information and data. We therefore decided to apply a multi-step transformation process (cf. Figure 3.1). It facilitates the formalization process by using various intermediate representations that are obtained by stepwise procedures.

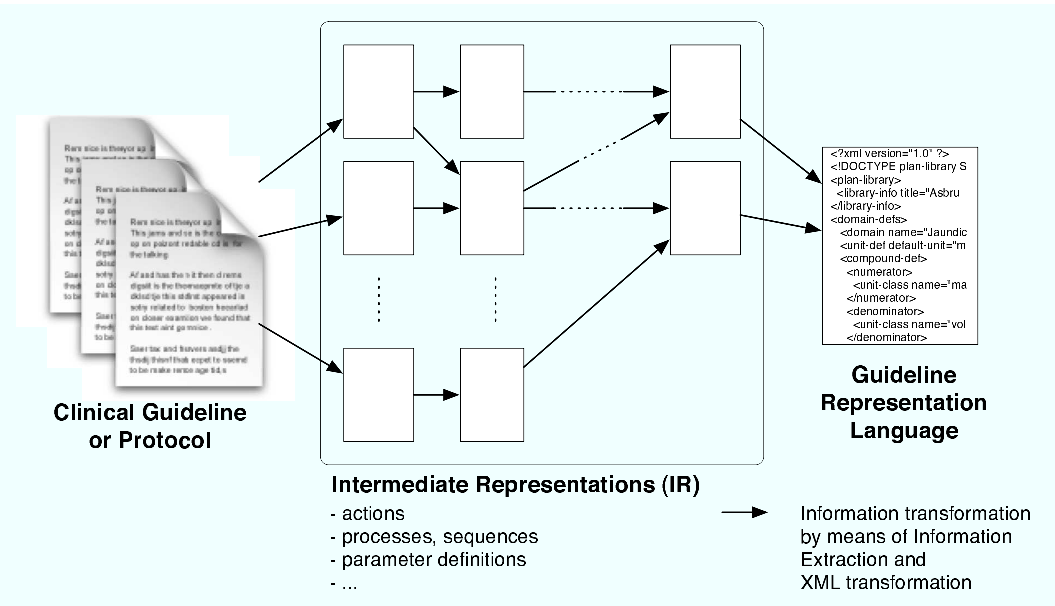

Figure 3.1: Guideline transformation process. A multi-step process using intermediate representations to transform clinical guidelines and protocols (CGPs) into a formal representation language.

The benefits of the intermediate representations are:

- Concise formalization process

- Different formats for various kinds of information
- Separate views and procedures for various kinds of information
- Application of specific heuristics for each particular kind of information
- Simpler and more concise evaluation and tracing of each process step

To process as large a class as possible of documents and information we need specific heuristics. These are applied to a specific form of information, for instance:

- **Different kinds of information**
  Each kind of information (e.g., processes, parameters) needs specific methods for processing. By presenting only one kind of information the application of the belonging method is simpler and easier to trace.

- **Different representations of information**
  We have to take into account various ways in which the information might be represented (i.e., structured, semi-structured, or free text).

- **Different kinds of guidelines**
  CGPs exist for various diseases, various user-groups, various purposes, various organizations, and so on, and have been developed by various guideline developers' organizations. Therefore, we can speak about different classes of CGPs that may contain similar guidelines.

To transform information by applying Information Extraction (IE) methods, we generated specific templates that can present the desired information. Heuristic methods detect relevant information, which is filled into the templates' slots for subsequent processing. In the next section we present a method that extracts process information from clinical guidelines for otolaryngology using heuristic algorithms. The output of this method is a unified format, which can be transformed into the final representation.

Chapter 4

Process Extraction from Clinical Guidelines and Protocols

CGPs present effective treatment processes. One challenge when authoring CGPs is the detection of individual processes and their relations and dependencies. We try to detect these by means of IE. CGPs consist of semi-structured and free text. The resulting output can subsequently be processed to yield refined representations, leading ultimately to the representation in the guideline representation language.

By analyzing treatment processes of otolaryngology contained in the guidelines we detected the following processes:

- Sequential processes
- Processes without temporal dependencies
- Processes which exclude each other (i.e., one process has to be selected from several)
- Processes containing subprocesses
- Recurring processes

We have therefore developed heuristics in order to gain treatment processes from the CGPs, where a process is described by at least one sentence. That means that a sentence, for instance, *'Take acetaminophen or ibuprofen.'*, presents only one process and not a selection of two processes. The heuristics are categorized in three groups: (1) heuristics for detecting the relevant sentences, (2) heuristics for detecting whether the sentence is the description of a process, an annotation, or a negative action, and (3) heuristics for detecting relations between processes (see Figure 4.1 for details). The heuristics are developed for XHTML-conform documents. Before going into details with our heuristics we describe our dictionary that they use.

4.1 Dictionary for Extracting Processes: CGPs of Otolaryngology

The dictionary we are using for this purpose includes various classes, such as

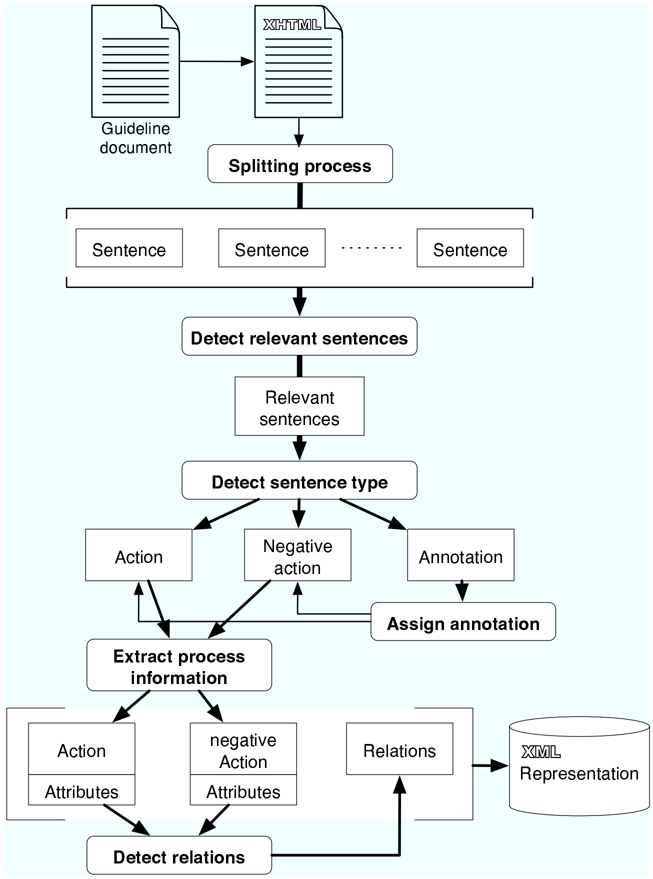

Figure 4.1: Process Extraction. Various steps in gaining process information out of clinical practice guidelines.

- *Medical* terms (i.e. drug agents, surgical procedures, and diagnosis terms)
- *Action* terms (mainly verbs; e.g., ’activate’, ’perform’, ’prescribe’, ’treat’, ’integrate’, ’receive’)
- *Condition* terms (i.e. regular expressions describing a condition, such as ’if [^,:\.]+’, ’in case(s)? [^,:\.]+’, ’for .*<diagnosis-term>’)
- *Dose unit* terms (e.g. ’(m|d|c)?(l|g)(/kg/day)?’, ’drop(s)?’, ’teaspoon(s)?’, ’tsp’)
- *Time unit* terms (e.g. ’(m(illi)?)?sec(ond)?(s)?’, ’min(ute)?(s)?’)
- *Relation* terms (e.g. ’after’, ’before’, ’during’, ’while’)
- *Negative action* terms (i.e. expressions describing actions that should not be performed; e.g., ’no .*(benefit|advantage)’, ’not to perform’, ’not be (used|treated)’)

The medical terms are based on a subset of the Medical Subject Headings (MeSH)[1] of the United States National Library of Medicine. We adapted them according for missing terms, different wordings, acronyms, and varying categorization.

4.2 Task 1: Detecting Relevant Sentences

Detecting relevant sentences is a challenging task, which we undertake in two steps: (1) detecting irrelevant sentences to exclude them from further processing and (2) detecting relevant sentences.

In both steps we use special keywords to detect whether a sentence is irrelevant or relevant. Keywords describing irrelevant sentences are ’history’, ’diagnosis’, ’criteria’, ’symptom’, ’clinical assessment’, ’risk factor’, ’complicating factor’, ’etiology’, and so on. These terms point out that the following paragraph does not describe treatment processes, but that it describes symptoms, demonstration of diagnoses, and so on. If such a term appears within a caption the corresponding section is removed.

Detecting relevant sentences is not a trivial task. First we parse the entire document and split it into sentences. Then we process every sentence with regard to its context within the document and its group affiliation. Thereby, the context is obtained by captions (e.g., *’Acute Pharyngitis Algorithm Annotations | Treatment | Recommendations:’*) and a group contains sentences from the same paragraph or the same list, if there are no sublists (see Algorithm 1 for details). Furthermore, we mark explicitly described annotations (i.e., sentences or paragraphs starting with *’Note’* or *’Notice’*).

[1]http://www.nlm.nih.gov/mesh/

Algorithm 1. Parsing the document and splitting into sentences.

```
 1.  for each Paragraph p and each List-Entry l do {
 2.     split p and l into sentences
 3.     for each Sentence s do {
 4.         if s.context contains an action-, agent-term, or
            surgical procedure and s is part of a list-entry
 5.             set s.listEntry to true
 6.         if previous paragraph pp contains the expression
            'measure|remedies' and s is part of a list-entry {
 7.             add pp to s.context
 8.             set s.listEntry to true
 9.         }
10.     }
11. }
```

Each sentence is now checked for relevance. Relevant sentences are identified by the occurrence of an agent term or surgical procedure or their context contains specific terms (cp. Algorithm 2). These terms (i.e., 'measure', 'remedies', 'medication', 'treatment') are important, because they specify actions that may not contain agent terms or surgical procedures (e.g., *'Maintain adequate hydration (drink 6 to 10 glasses of liquid a day to thin mucus)'*). Furthermore, we assign a therapy list to each sentence, which contains all agent terms and surgical procedures contained in the sentence.

Algorithm 2. Checking each sentence upon relevance.

```
 1. for each Sentence s do {
 2.     if (s.listEntry is true and s.context contains expression
        'measure|remedies|medication') or s.annotation is true
 3.         s is relevant
 4.     if s.listEntry is true and s.lastContextEntry contains
        expression 'treatment'
 5.         s is relevant
 6.     if s.content contains agent-terms or surgical procedures
 7.         s is relevant
 8.     store all agent terms and surgical procedures in the
        therapy-list of s
 9. }
10. store relevant sentences
```

Now we have collected the relevant sentences from the guideline, we can proceed with the next step.

4.3 Task 2a: Detecting Whether a Sentence is the Description of a Process, an Annotation, or Describes a Negative Action

First, we check, whether a sentence describes an action or a negative action (cf. Algorithm 3). Negative actions are instructions that an action should not be performed, often under specific conditions (e.g., *'Do not use aspirin with children and* *teenagers because it may increase the risk of Reyes syndrome.*'). Most guideline representation languages will handle such actions by inverting the condition. Languages may exist which will handle these in other ways. Therefore we will provide a representation for such actions that can be used in a general way.

Algorithm 3. Checking each sentence whether it describes an action or a negative action.

```
for each Sentence s do {
    if s.listEntry is true
        s is relevant
    split s.content in subclauses
    for each SubClause sc do {
        if sc contains a negative-action-term and an (agent
        term or surgical procedure) {
            if the negative-action-term contains 'not '
                set s.notAction to true
            else {
                remove all agents or surgical procedures in sc
                from the therapy-list
            }
        }
        if s.listEntry is false
            if sc contains an (agent term or surgical procedure)
            and an action term {
                s is relevant
                mark sc as mainClause
            }
        }
    }
}
store relevant sentences
```

Detecting explicitly described annotations has already been described in Section 4.2. To detect implicitly described annotations we have developed a special heuristic (cf. Algorithm 4). This is done by checking whether the agent terms or surgical procedures in a sentence also appear in processes belonging to the same group appearing above this sentence in the text. If this happens, the sentence is added as an annotation to all these processes.

Algorithm 4. Checking each sentence whether it describes a new process or an annotation of a process.

```
for each Sentence s do {
    create new Process p
    set note to false
    find condition clauses in s.content
    remove agents and surgical procedures appearing in condition
    clauses from therapy-list
    for each Process pp where pp.group-id == s.group-id {
        if pp.therapy-list contains all agents in
        s.therapy-list {
```

```
            set pp.annotation to s.content
            if s has mainClause
                find additional information in s.mainClause
            else
                find additional information in s.content
            add additional information to pp
            set note to true
        }
    }
    if note is false {
        if s has mainClause
            find additional information in s.mainClause
        else
            find additional information in s.content
        add additional information to p
    }
    store p
}
```

Explicitly mentioned annotation sentences are added to each process of the same group appearing before the annotation sentence. If there are no sentences in front of the annotation, the annotation sentences are added to each process of the parent group.

If the sentence is not an annotation, it is added as a new process and augmented by additional information. The additional information contains conditions, the duration and iterative aspects of the process, as well as the dosage in case of a drug administration. The information obtained is presented as single-slot values. As we decided to one sentence as one process, a multi-slot system is not necessary at this time. If there are processes with multiple filled slots, the processes can be refined in subsequent steps, if this is wanted.

4.4 Task 2b: Detecting Relations between Processes

The default relationship among processes is that there is no synchronization in their execution. But as we determined when we analyzed the guidelines, there are more kinds of relations (see begin of Chapter 4).

To group processes to a *selection* they must fulfill the following requirements: (1) the processes have to belong to the same group, and (2) agents or surgical procedures have to be associated to the same category. For instance, processes describing the administration of *Erythromycin*, *Cephalexin*, and *Clindamycin* within one group are combined in a *selection*, as all these agents are antibiotics. If processes are grouped in a selection, one of these processes has to be selected to be executed. This algorithm works well for CGPs of otolaryngology. There may be other specialties where this algorithm has to altered for other agent categories.

Furthermore, we try to detect relations between processes that are explicitly mentioned within the text as well as relations that are implicitly given by the document structure. The former is very difficult to detect, as we often cannot detect the reference of the relation within the CGP (e.g., *'After 10 to 14 days of failure of first line antibiotic ...'*). Nevertheless, we found heuristics that arrange processes or process groups if the reference is unambiguously extractable out of the text. These heuristics can be grouped in two categories: (1) detecting sentences describing relations between processes, and (2) detecting the processes that are described in the preceding heuristic.

Algorithm 5. Checking each sentence if it describes relations between processes.

```
for each Sentence s do {
    if s contains relation terms {
        create new Relation r
        if relation terms appears at begin of s {
            split s by ’,’ in 2 clauses
            set inMainClause to true
        } else {
            split s by relation term in 2 clauses
            set inMainClause to false
        }
        for each agent term in clause 1 do {
            if inMainClause is true
                set r.source to agent
            else
                set r.destination to agent
            set relation type of r
        }
        for each agent term in clause 2 do {
            if inMainClause is true
                set r.destination to agent
            else
                set r.source.to agent
            set relation type of r
        }
    }
}
```

Algorithm 5 describes the heuristic to detect sentences describing relations. A relation is mainly identifiable by a relation term (e.g., ’before’, ’after’, ’while’, ’during’). If such a term appears, we are searching for agent terms, as these describe most of our processes. After we detected these terms, we search for processes containing the particular agent terms (see Algorithm 6). If we have found both the source process and the destination process we can create a new relation.

Algorithm 6. Checking each process for being part of a relation.

```
for each Relation r do {
    if r.type is ’PRECEDING’ {
        for each agent s in r.source do {
            search for Process p containing agent s
            set toID to p.process_id
            for each agent d in r.destination do {
                search for Process p containing agent d
                set fromID to p.process_id
                store SingleRelation
            }
```

```
        }
    } else {
        for each agent d in r.destination do {
            search for Process p containing agent d
            set fromID to p.process_id
            for each agent s in r.source do {
                search for Process p containing agent s
                set toID to p.process_id
                store SingleRelation
            }
        }
    }
}
```

Implicitly given relations by the document structure we detect by patterns of the document structure (e.g., *'Further Treatment'* appears `after` *'Treatment'* or *'Treatment'* appears `before` *'Follow-Up'*). Thereby, we developed structure patterns that are used to determine the relations between several groups.

Chapter 5

Evaluation

To evaluate our heuristics we developed a framework and used several guidelines from otolaryngology. We obtained 18 CGPs from the National Guideline Clearinghouse (NGC)[1] that describe the treatment and management of various diseases of otolaryngology. The CGPs were developed from eight different organizations in North America and Europe. We split the CGPs into two groups: (1) six guidelines for developing and improving the heuristics and (2) twelve guidelines for testing our heuristics. To select the CGPs for the two groups was not trivial. Organizations that develop guidelines do not always use the same hierarchical structure. Therefore, we were unable to select the CGPs according to the organization that developed them. However we used the complexity of the hierarchical structures as selection criteria and distributed them evenly to each group.

Before applying our heuristics we have to carry out some pre-processing. This consists of making the documents XHTML-conform and applying additional structuring elements. The latter is done by converting paragraphs and their corresponding headings to list elements. In this way, we obtained a unique document format containing lists and sublists as well as paragraphs.

We then detected relevant sentences in Task 1 as explained in Section 4.2. Task 2 summarizes the detection of the kind of sentence, additional information, and the relations between processes as explained in Sections 4.3 and 4.4. We evaluated our heuristics by means of Recall and Precision measures. The *Recall* score measures the ratio of correct information extracted from the texts against all the available information present in the text. The *Precision* score measures the ratio of correct information that was extracted against all the information that was extracted [7].

[1] http://www.guidelines.gov/

[2] Centers for Disease Control and Prevention

[3] Institute for Clinical Systems Improvement

[4] University of Michigan Health System

[5] Scottish Intercollegiate Guidelines Network

[6] Cincinnati Children's Hospital Medical Center

[7] American Academy of Family Physicians; American Academy of Otolaryngology–Head and Neck Surgery; American Academy of Pediatrics;

[8] American Academy of Pediatrics

[9] Finnish Medical Society Duodecim

[10] Practice Guidelines Initiative

Table 5.1: Evaluation measures of Task 1 according to the used clinical practice guidelines.

Title	COR	POS	ACT	REC	PRE
Acute otitis media: management and surveillance in an era of pneumococcal resistance[2]	4	7	4	0.57	1.00
Acute pharyngitis[3]	46	53	49	0.87	0.94
Acute rhinosinusitis in adults[3]	10	11	10	0.91	1.00
Acute sinusitis in adults[3]	56	65	56	0.86	1.00
Allergic rhinitis[4]	18	21	18	0.86	1.00
Diagnosis and management of childhood otitis media in primary care.[5]	8	18	8	0.44	1.00
Diagnosis and treatment of obstructive sleep apnea[3]	39	76	39	0.51	1.00
Diagnosis and treatment of otitis media in children[3]	53	72	53	0.74	1.00
Evidence based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age[6]	19	27	20	0.70	0.95
Evidence based clinical practice guideline for medical management of otitis media in children 2 months to 6 years of age[6]	17	18	18	0.94	0.94
Management of sore throat and indications for tonsillectomy.[5]	15	22	15	0.68	1.00
Otitis media[4]	7	7	7	1.00	1.00
Otitis media with effusion[7]	8	9	8	0.89	1.00
Pneumococcal vaccination for cochlear implant candidates and recipients: updated recommendations of the Advisory Committee on Immunization Practices[2]	6	9	6	0.67	1.00
Reduction of the influenza burden in children[8]	3	3	4	1.00	0.75
Rhinitis[3]	50	56	56	0.89	0.89
Sore throat and tonsillitis[9]	17	20	18	0.85	0.94
Symptomatic treatment of radiation-induced xerostomia in head and neck cancer patients[10]	3	3	3	1.00	1.00
Summary	379	497	392	**0.76**	**0.97**

Scoring Key:

POS – Number of sentences according to the key target template

ACT – Number of sentences identified by the system

COR – Number of correctly identified sentences by the system

REC – Ratio of COR slot fillers to POS slot fillers

PRE – Ratio of COR slot fillers to ACT slot fillers

Table 5.2: Evaluation measures of Task 2 according to the used clinical practice guidelines.

Title	COR	POS	ACT	REC	PRE
Acute otitis media: management and surveillance in an era of pneumococcal resistance[2]	14	14	14	1.00	1.00
Acute pharyngitis[3]	99	101	111	0.98	0.89
Acute rhinosinusitis in adults[3]	24	24	34	1.00	0.71
Acute sinusitis in adults[3]	188	196	227	0.96	0.83
Allergic rhinitis[4]	35	37	44	0.95	0.80
Diagnosis and management of childhood otitis media in primary care.[5]	26	27	27	0.96	0.96
Diagnosis and treatment of obstructive sleep apnea[3]	70	70	83	1.00	0.84
Diagnosis and treatment of otitis media in children[3]	78	83	89	0.94	0.88
Evidence based clinical practice guideline for children with acute bacterial sinusitis in children 1 to 18 years of age[6]	51	54	66	0.94	0.77
Evidence based clinical practice guideline for medical management of otitis media in children 2 months to 6 years of age[6]	52	63	62	0.83	0.84
Management of sore throat and indications for tonsillectomy.[5]	21	22	23	0.95	0.64
Otitis media[4]	20	23	25	0.87	0.80
Otitis media with effusion[7]	23	23	24	1.00	0.96
Pneumococcal vaccination for cochlear implant candidates and recipients: updated recommendations of the Advisory Committee on Immunization Practices[2]	11	17	15	0.65	0.73
Reduction of the influenza burden in children[8]	7	9	7	0.78	1.00
Rhinitis[3]	76	76	90	1.00	0.84
Sore throat and tonsillitis[9]	45	52	53	0.87	0.85
Symptomatic treatment of radiation-induced xerostomia in head and neck cancer patients[10]	7	7	7	1.00	1.00
Summary	847	898	1011	**0.94**	**0.84**

Scoring Key:

POS – Number of slot fillers according to the key target template

ACT – Number of slot fillers generated by the system

COR – Number of correct slot fillers generated by the system

REC – Ratio of COR slot fillers to POS slot fillers

PRE – Ratio of COR slot fillers to ACT slot fillers

Table 5.3: Evaluation measures of Task 1 and Task 2.

	Recall	**Precision**
Task 1	0.76 (0.9174)	0.97 (0.9092)
Task 2	0.94 (0.9763)	0.84 (0.9073)

Task 1 (see Table 5.1) provides promising results, although the lower recall score implies that detecting relevant sentences has to be improved. The higher precision score shows that irrelevant sentences are barely categorized as relevant.

The input to Task 2 (see Table 5.2) consists of the sentences detected by Task 1. The task's recall score is very high, which means that only a few slots were spuriously not detected. The precision score implies there are some incorrect slot fillers. These arise from not always detecting the correct type of sentence and especially assigning annotations to their particular actions, which has to be improved. The overall evaluation results are presented in Table 5.3.

Thus, refining and enhancing our heuristics, especially for documents with a minor portion of semi-structured text, will be one of our next steps in order to provide a good basis for subsequent transformation steps.

Chapter 6

Conclusion and Future Work

We have shown that it is possible to extract process information from CGPs applying heuristics. Our heuristics use patterns in the structure of the document as well as of specific expressions. Thus, we do not need to use Natural Language Understanding.

We applied a framework in order to evaluate our heuristics that can cope with both semi-structured and free text of the documents. The resulting information is filled in single-slot templates, which can represent processes and their relations. This information extracted can then be used for further transformations to finally generate a representation in a guideline representation language.

Our next step is to improve our heuristics and to enhance them for guidelines with processes of higher complexity. Furthermore, we want to support the modeling process by giving the plan modeler the ability to evaluate and trace after each step, making the process traceable. We will therefore implement *links* that connect the origin of the information and the extracted or transformed information, thereby making it possible to evaluate the individual steps by means of the DELT/A [19], which can visualize these links.

Acknowledgements

This project is supported by ”Fonds zur Förderung der wissenschaftlichen Forschung FWF” (Austrian Science Fund), grant P15467-INF.

Bibliography

[1] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the Conference on Very Large Databases (VLDB)*, 2001.

[2] M. E. Califf and R. J. Mooney. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 328–334, Orlando, FL, July 1999.

[3] R. Gaizauskas, M. Hepple, N. Davis, Y. Guo, H. Harkema, A. Roberts, and I. Roberts. AMBIT: Acquiring medical and biological information from text. In S. J. Cox, editor, *Proceedings of the UK e-Science All Hands Meeting*, Nottingham, UK, 2003.

[4] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human Computer Studies*, 58(1):89–123, 2003.

[5] R. D. Holowczak and N. R. Adam. Information Extraction based Multiple-Category Document Classification for the Global Legal Network. In *Proc. of the 9th Conference on Innovative Applications of Artificial Intelligence*, pages 992–999. AAAI Press/MIT Press, 1997.

[6] R. Kosara and S. Miksch. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131, May 2001.

[7] W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. Evaluating an Information Extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6), 1994.

[8] L. Liu, C. Pu, and W. Han. XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources. In *Intern. Conference on Data Engineering (ICDE)*, pages 611–621, 2000.

[9] M. A. Musen, S. W. Tu, A. K. Das, and Y. Shahar. EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Information Association (JAMIA)*, 3(6):367–388, 1996.

[10] M. Peleg, S. W. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, and M. Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2003.

[11] K.-A. Polvani, A. Agrawal, B. Karras, A. Deshpande, and R. Shiffman. *GEM Cutter Manual*. Yale Center for Medical Informatics, 2000.

[12] M. Růžička and V. Svátek. Mark-up Based Analysis of Narrative Guidelines with the Stepper Tool. In K. Kaiser, S. Miksch, and S. W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101: Studies in Health Technology and Informatics, pages 132–136, Prague, Czech Republic, 2004. IOS Press.

[13] R. N. Shiffman, B. T. Karras, A. Agrawal, R. Chen, L. Marenco, and S. Nath. GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association (JAMIA)*, 7(5):488–498, 2000.

[14] S. Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, 1999.

[15] S. Soderland, D. Aronow, D. Fisher, J. Aseltine, and W. Lehnert. Machine Learning of Text Analysis Rules for Clinical Records. Technical report, University of Massachusetts, Amherst, MA, 1995.

[16] R. Steele and J. Fox. Tallis PROforma Primer – Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.

[17] D. R. Sutton and J. Fox. The syntax and semantics of the pro*forma* guideline modeling language. *Journal of the American Medical Informatics Association (JAMIA)*, 10(5):433–443, Sep / Oct 2003.

[18] P. Votruba, S. Miksch, and R. Kosara. Linking Clinical Guidelines with Formal Representations. In *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME 2003)*, pages 152–157, Berlin, 2003. Springer.

[19] P. Votruba, S. Miksch, and R. Kosara. Facilitating knowledge maintenance of clinical guidelines and protocols. In M. Fieschi, E. Coiera, and Y.-C. J. Li, editors, *Proc. from the Medinfo 2004 World Congress on Medical Informatics*, pages 57–61. IOS Press, 2004.