

# Prototypical case mining from biomedical literature for bootstrapping a case base

Isabelle Bichindaritz

Published online: 22 June 2007  
© Springer Science+Business Media, LLC 2007

**Abstract** This article addresses the task of mining for cases from biomedical literature to automatically build an initial case base for a case-based reasoning (CBR) system. This research takes place within the Mémoire project, which has for goal to provide a framework to facilitate building CBR systems in biology and medicine. By analyzing medical literature, the ProCaseMiner system mines for medical concepts such as diseases, signs and symptoms, laboratory tests, and treatment plans in relationship with one another, and connects them together in a given medical domain. It then organizes these concepts in a higher-level structure called a case. This case mining component provides a definite help to bootstrap the creation of a biomedical CBR system case base, composed of both concrete cases and prototypical cases. Currently, most cases learnt correspond to prototypical cases, given the level of abstraction of their features. This article validates the approach by presenting a comparison between the prototypical cases learnt from stem-cell transplantation domain with those created by a team of experts in the domain.

**Keywords** Medical case-based reasoning · Case-based reasoning · Medical informatics · Text mining · Case mining

## 1 Introduction

Case-based reasoning (CBR) systems in biomedicine rely on patients' cases to propose diagnosis assessment and treat-

ment recommendations in case-based decision-support systems. Often these systems have reported that the cases readily available in electronic format are incomplete at best, and have resorted to multimodal reasoning systems to complement the cases with knowledge bases expressed in models and/or rules. Many times, cases are not even available in electronic format, which requires a tremendous amount of time entering data into CBR systems just to bootstrap them. This context prompts for the design of advanced automatic knowledge elicitation tools to provide CBR systems with the adequate knowledge they need for reasoning, without spending years eliciting this knowledge from experts. ProCaseMiner system presented in this article builds on a current trend to develop case mining systems to take advantage of electronically available knowledge sources that may be mined for cases usable by CBR systems with minimal human intervention.

The idea of mining cases from medical literature comes from Swanson [1, 2], who ignited the interest of researchers who followed in his tracks. Since then, text mining research from biomedical literature has developed as a promising new area to make sense of the exponentially increasing amount of information made available in particular through biomedical literature. Given the vast amount of information in an area, researchers are more and more constrained to specialize, thus abstracting themselves from other domains. Therefore text mining systems are needed to enable researchers to rapidly apprehend the knowledge in a domain, and to discover relationships between concepts from different domains. The goal of text mining from literature databases is to discover novel and interesting knowledge in the form of concepts, patterns, and relations [3–6]. Swanson [1, 2] for example describes a data mining system that brought forth seven medical discoveries later published in relevant medical journals. Following in his tracks, Weeber et al. [7] pro-

---

I. Bichindaritz (✉)  
University of Washington, Institute of Technology, 1900  
Commerce Street, Box 358426, Tacoma, WA 98402, USA  
e-mail: ibichind@u.washington.edu

posed a text mining system that proposes new target diseases for the drug thalidomide. The success of this system has revived the interest in Swanson's pioneering work, and recently many systems mining for relationships in biomedical literature have been developed [8–10].

The system presented here proposes to automate the process of mining for cases and more specifically prototypical cases from biomedical literature. It builds on a concept miner learning for relationships between concepts [9, 10], such as the relationship between caloric restriction and aging, and not for isolated concepts. The next section presents the *Mémoire* project and what a sample case-based knowledge base for a medical CBR system looks like. The third section introduces the Unified Medical Language System (UMLS) as the semantic network and associated natural language processing (NLP) tools guiding the discovery process. The fourth section sets forth the ProCaseMiner system architecture and different components. The fifth section presents an evaluation of the system. It is followed by a discussion and a conclusion.

## 2 Mémoire project

The goal of the *Mémoire* project [11] at the University of Washington is to provide a framework for the creation and interchange of cases, concepts, and CBR systems in biology and medicine. Its approach is to generalize from previous CBR systems built in biomedicine, among which Carepartner [12] has been a major source of inspiration. This project is notable because it was built over a four-year research effort, funded by a grant with the Agency on Health Care Policy and Research (AHCPR), and involved a multidisciplinary team composed of one case-based reasoning specialist, one agents specialist, three physicians, a physician assistant, two research nurses, one biostatistician, and two psychologists specialized in patient quality of life and outcomes measurement. This exceptional team and grant, in the premier cancer research center in stem-cell transplantation—Fred Hutchinson Cancer Research Center (FHCRC)—home of three Nobel prizes, came up with very interesting advances in CBR research as well. This section summarizes the kinds of knowledge and case structures that were found as effective in representing case knowledge. These form the structure that ProCaseMiner will be mining for.

### 2.1 Ontology

The system is known for its decision support results, including 94.5% of recommendations judged as clinically acceptable by the experts, figure growing over time [13]. These results come from the achievement of intensive knowledge

**Table 1** Carepartner's ontology, showing the number of objects represented in Carepartner system, in comparison with the standard medical nomenclature SNOMED v. 3.4. The prototypical cases correspond to the clinical pathways

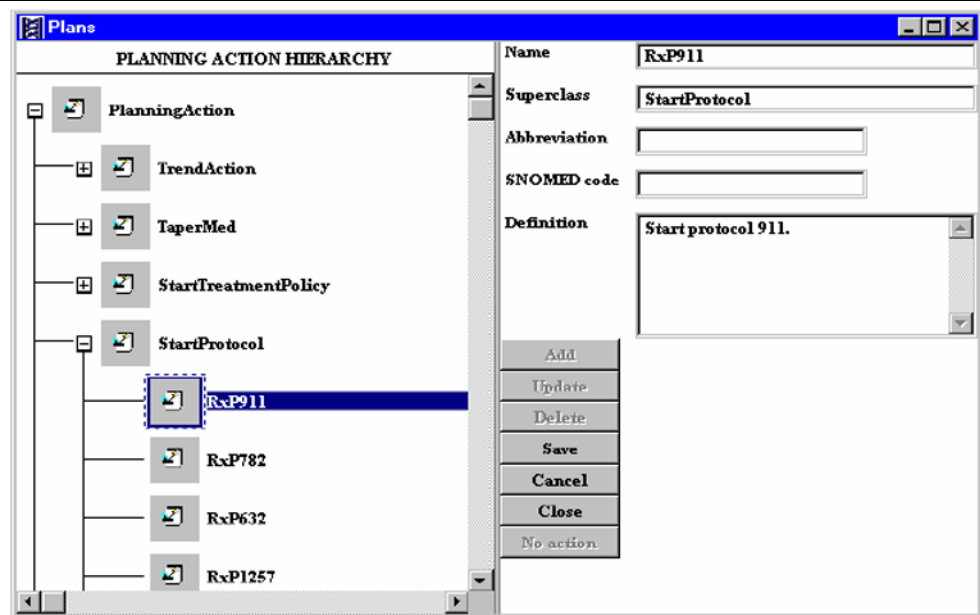
N	Care partner	SNOMED v. 3.4
Diseases	1109	35834
Functions	452	19221
Labs	1152	30723
Procedures	547	20105
Medications	2684	14846
Sites	460	5875
Terms	739439	
Relations	51	
Patient cases	4904	
Prototypical cases	91	

elicitation efforts to build the case base around a knowledge base of the domain. It was determined early on in the project that cases were not available in electronic format at a level of detail required for CBR. For instance, the patients database did not comprise treatments performed on the patients, nor most of the signs and symptoms, but only main events 'abstracted' from the paper charts. The project team had to come up with prototypical cases to bootstrap the system, which took over two years to develop at a level of thoroughness and consistency necessary for achieving the high accuracy in overall recommendations of over 94%. This system was also innovative because its proposed recommendations spanned not only diagnosis, but also lab results interpretation, and treatment planning. Table 1 presents the extent of the ontology built for this system.

### 2.2 Prototypical case

The cornerstone of the knowledge acquisition process has been the conception of prototypical cases, called clinical pathways in this system. This prototypical case structure is important for this article because it is also the prototypical case structure proposed in *Mémoire* as a generic prototypical case representation structure. Consequently, this is the kind of prototypical cases that ProCaseMiner is mining for. The clinical pathways, 112 of them having been implemented in the test version of the system (see Table 1), correspond to clinical diagnostic categories for the most part, some of them corresponding also to essential signs and symptoms requiring specific assessment or treatment actions. The clinical pathways are knowledge structures represented from the ontology described above, namely: all diseases, functions (also known as signs and symptoms), labs, procedures, medications, sites, and planning actions. Most of the terms naming these objects are standardized

**Fig. 1** Carepartner's planning actions ontology



using the Unified Medical Language System (UMLS) thesaurus and semantic network [14]. Only the terms not corresponding to objects in the UMLS have been added to the domain specific ontology. In particular, the planning actions used in the treatment part of a prototypical case did not exist in the UMLS and were all created for the system. They enhance the system with the representation of the actions performed by the clinicians in the framework of treatment plans, such as medications management, referral to specialists, and enrolling patients in specific protocols. Figure 1 shows some of the planning actions in the ontology, which enable its main functionality of treatment planning.

An example of a prototypical case is provided in Fig. 2. It shows that a prototypical case, mostly a diagnostic category or disease, such as here chronic graft versus host disease (CGVHD), which is a complication of stem-cell transplantation, comprises three parts:

1. A list of findings, corresponding to signs and symptoms.
2. A diagnosis assessment plan, which is a plan to follow for confirming (or informing) the suspected diagnosis.
3. A treatment/solution plan, which is a plan to follow for treating this disease when confirmed, or a solution when the pathway does not correspond to a disease.

The representation paradigm is that of object representation, such as each knowledge structure composing a clinical pathway is an instance of one of the classes defined in the ontology. The diagnosis assessment part and the treatment part can also be seen as simplified algorithms, since they use IF THEN ELSE structures, and LOOP structures, as well as SEQUENCE structures of actions in time, which, when instantiated with actual patients' data, provide a di-

agnosis assessment plan, or a treatment plan, tailored to a specific patient. In this way, this knowledge structure allows for sophisticated adaptation when reusing a prototypical case.

The goal of the prototypical case mining system has been to automatically learn the type of knowledge structure illustrated by clinical pathways. The formal representation of a prototypical case is presented here. The elements of the representation language are those of semantic networks:

- **A domain ontology**, which is the set of *class symbols* (also called concepts in the UMLS)  $\mathbf{C}$ , where  $C_i$  and  $C_j$  denote elements of  $\mathbf{C}$ , and the set of *relationship symbols* (also called relations in the UMLS)  $\mathbf{R}$ , where  $R_i$  and  $R_j$  denote elements of  $\mathbf{R}$ . Figures 3 and 4 provide examples of relationships from the UMLS. Examples of relationships are *isa*, *partOf*, *propertyOf*, and *coOccurs*. The classes are organized in a polyhierarchy of classes, in which a class may eventually have several superclasses. Several main categories can be described, such as *Functions*, *Diseases*, *Morphology*, and *Topography* for instance. SNOMED International v. 3.4 has been used to codify the ontology, whenever possible, hence the main categories listed. Also, many classes describe *Events*, *Time*, and *State* concepts.
- **A set of individual symbols** (also called instances)  $\mathbf{I}$ , where  $i$  and  $j$  denote elements of  $\mathbf{I}$ . Among these, some refer to instances of classes, others to numbers, dates, and other values. Instances of a class  $C_i$  are noted  $aC_i$ .
- **A set of operator symbols**  $\mathbf{O}$ , permits to form logical expressions composed of classes, instances and other values, and relationships. Pathways, guidelines and cases are expressed this way, and such a composition permits to

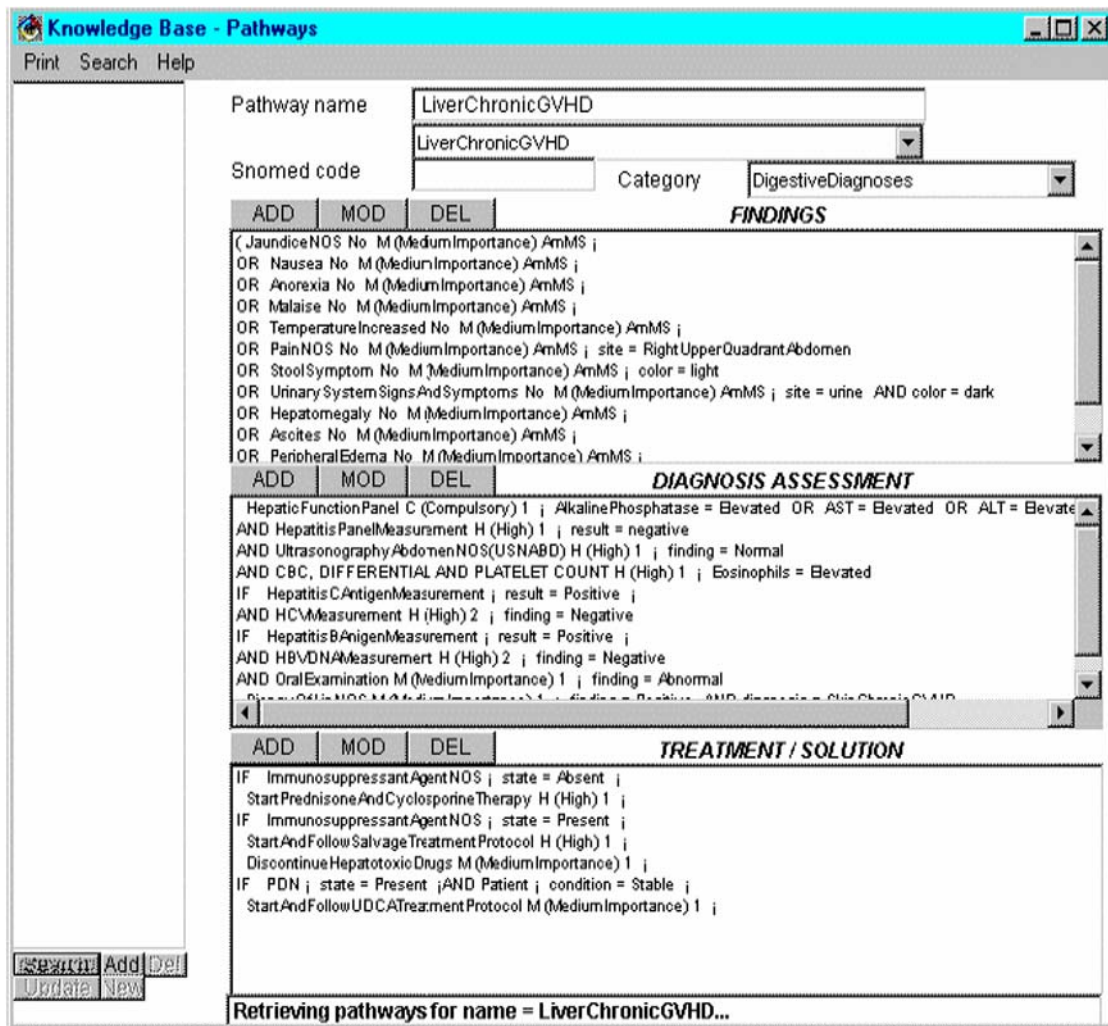
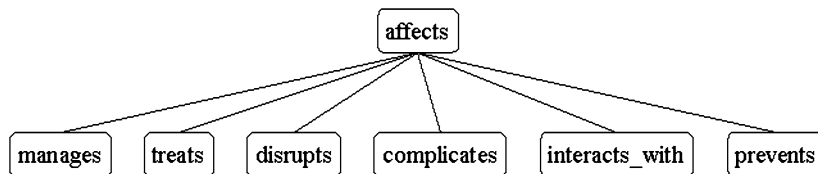


Fig. 2 A clinical pathway, corresponding to a prototypical case, for chronic graft versus host disease (CGVHD)

Fig. 3 Extract of UMLS relationships (from NLM’s UMLS project [14])



represent complex entities in a structured format. The set of operators comprises the following:  $\cap$  (AND),  $\cup$  (OR),  $\neg$  (NOT), ATLEAST  $n$ , ATMOST  $n$ , EXACTLY  $n$ , IF, THEN, ELSE,  $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $\neq$ .

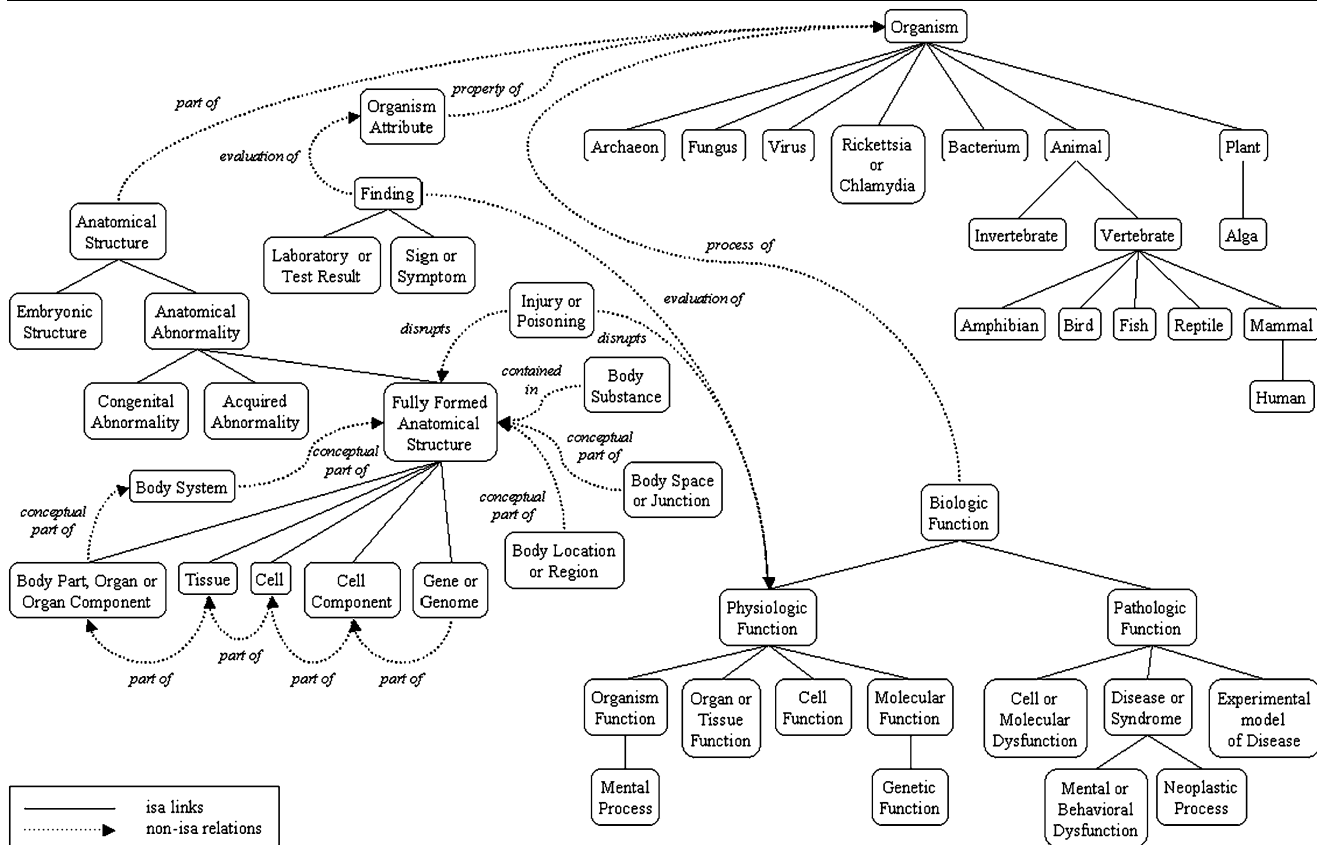
In this representation language, the attributes of a class are represented via the relationships. A ternary relationship, the arguments of which are an instance, and attribute, and a value mean that a certain attribute of a class has a certain value, or gets a certain value, such as in *propertyOf*(*HepaticFunctionPanel*, *AlkalinePhosphatase*, *Elevated*).

Cases and prototypical cases are expressed as  $\langle$ problem situation, solution $\rangle$ , where *problem situation* and *solution*

have the same representation, either a composition of instances with operators, or relationships between instances and values:

$$\begin{aligned}
 \text{problem situation} &= \Theta aC_i \\
 &\Theta R_j(aC_{j,1}, aC_{j,2}, \dots aC_{j,k}) \\
 \text{solution} &= \Theta aD_i \\
 &\Theta R_j(aD_{j,1}, aD_{j,2}, \dots aD_{j,k})
 \end{aligned}$$

with  $\Theta \in \{\cup, \cap\}$ , the default value being  $\cup$  for pathways, and  $\cap$  for cases and rules, and where  $aC_i$  and  $aD_i \in \mathbf{I}$ , and  $R_j \in \mathbf{R}$ . Cases are expressed using only  $\cap$ .



**Fig. 4** Extract of UMLS semantic network (from NLM’s UMLS project [14]). Some of the relationships illustrated are ‘part of’, ‘process of’, and ‘evaluation of’

This representation language is a form of description logic, and is compatible with the Web Ontology Language (OWL), as has been described in a previous article [13].

An example of an excerpt from *LiverChronicGVHD* prototypical case (see Fig. 2) represented in this language is the following:

```

problem situation = OR JaundiceNOS
                  OR Nausea
                  ...
                  OR UrinarySystemSignsAndSymptoms
                    propertyOf(UrinarySystemSigns
                               AndSymptoms,site,right
                               UpperQuadrantAbdomen)
                  OR StoolSymptom
                    = propertyOf(StoolSymptom, color,
                               light)
                  ...
solution          = AND HepaticFunctionPanel
                       propertyOf(HepaticFunction
                                   Panel, AlkalinePhosphatase,
                                   Elevated)
                       propertyOf(HepaticFunction

```

```

Panel, AST, Elevated)
propertyOf(HepaticFunction
Panel, ALT, Elevated)
AND
HepatitisPanelMeasurement
propertyOf(HepatitisPanel
Measurement, Result, Negative)
...
IF
ImmunosuppressantAgentNOS
propertyOf(Immunosuppressant
AgentNOS, State, Absent)
THEN
StartPrednisoneAnd
CyclosporineTherapy
...

```

### 3 UMLS project

In order to mine for prototypical cases, this project involves advanced text processing for mining from the literature, as is currently possible only in the biomedical domain. The main assets in this domain are the natural language processing knowledge bases and programs from the National Library of Medicine (NLM), which are presented in this section.

The “Unified Medical Language System” (UMLS) from the National Library of Medicine (NLM) [14], a specialized thesaurus and semantic network in biomedicine, provides standardized concepts for the creation of a controlled domain vocabulary. The UMLS provides a very powerful resource for rapidly creating a robust scientific thesaurus in support of precision searching. Furthermore, the semantic type descriptors for each concept and semantic network may offer some interesting opportunities for intelligent searching and mapping of concepts representing research findings, and their relationships [14].

Syntactic and semantic analysis tools for automated Natural Language Processing (NLP) are also provided by the National Library of Medicine’s UMLS project [15, 16]. UMLS ultimate goal is to facilitate the development of computer systems that behave as if they “understand” the meaning of the language of biomedicine and health. When this goal is achieved, UMLS will be an actual ontology where the meaning of terms can be automatically processed through reasoning processes.

By navigating the semantic network provided, it is possible to know which concepts extracted by the NLM tools from biomedical documents correspond to diseases, which correspond to findings, which correspond to medications, and so forth. It is also possible to know which relationships connect different concepts. There are a total of 135 semantic types. Figure 3 displays a few of the 54 relationships provided by the UMLS semantic network. Additionally, it is possible to extend the semantic network, both concepts and relationships, for instance for our purpose with a semantic network of planning actions that can be connected with a ‘treat’ relationship with other concepts.

The semantic network connects the UMLS concepts with one another using its set of relationships. It is extensible and allows enriching both the set of relationships and the set of concepts with domain specific concepts and relationships. NLP tools can be customized further to fit closely the needs of a particular biomedical domain. Figure 4 represents an extract from the UMLS semantic network.

#### 4 ProCaseMiner case miner

Based on the prototypical case structure presented in Sect. 2, and on the text processing knowledge bases and programs from the National Library of Medicine presented in Sect. 3, ProCaseMiner system mines for prototypical cases from biomedical literature. A selection of documents for a given medical domain is the input to this system. Pertinent documents may be literature articles, but also textual clinical practice guidelines, and medical case studies. It is important that such documents should all be related to a given domain, such as in our example stem-cell transplantation.

#### 4.1 Motivation

The knowledge embodied in cases represents the experience of expert clinicians. Therefore CBR systems need to have available records or traces of how expert clinicians solve clinical problems. Nevertheless, more often than not, this case knowledge is either not available in electronic format, or only partially available. For instance, in Mnaomia system [13], clinical cases were entirely recorded in paper-based medical records, which motivated the creation of prototypes by hand in order to bootstrap the reasoning process. It would not have been feasible within the timeframe of the project to wait until new cases were entered into the system to evaluate its efficiency, because cases were created at the rate of ten’s per year, which was not sufficient to cover a complex domain like eating disorders in psychiatry before decades.

In Carepartner system [12], although a database was available, it did not record the information required by the CBR decision-support system. The clinical research database recorded ‘abstracted’ data only, which was carefully selected by the clinical staff through several levels of review, so that the process took several weeks to achieve. This did not provide data in a timely manner, nor were all the data required ‘abstracted’. As a matter fact, the actual cases were recorded on paper only, and the creation of prototypical cases the only way of bootstrapping the CBR case base. Here also, it was not possible to wait to have enough cases in electronic format before evaluating the system, given that the rate of creation of new cases, including their long-term follow-up history, was slow.

Even with the adoption of electronic medical records, there will be a significant wait period before a comprehensive enough case base can be readily available to, or at least ready to be mined for, case-based reasoning. Moreover the issue of incompleteness of the knowledge recorded in electronic medical records will still remain an issue in the future. Other issues will deal with changing the format of the electronic medical record or data entry guidelines, to which the CBR system will have to adapt.

Therefore the need to provide case-based knowledge in a timely manner to a case-based reasoning system in clinical practice is an important problem to solve. This article proposes as a solution to this problem to mine for prototypical cases, to the extent that a prototypical case represents knowledge—from practice or from theory—in a format that can be processed by a case-based reasoner.

#### 4.2 Problem statement

Bootstrapping case-based reasoning systems in biomedical domains can be time consuming. It is advantageous to propose methods for automatically building case libraries either from resources in electronic format, such as electronic medical records, or from the literature. There exists

biomedical literature reporting how medical cases can be handled—in journals and other resources such as guidelines and textbooks. Taking advantage of these sources of cases can assist case-based reasoning tremendously as is reported in a new trend in case-based reasoning called case mining.

ProCaseMiner proposes to mine for cases from biomedical literature to automatically build a start-up case base for a case-based reasoning system. The approach is to mine for concepts and relationships between concepts, before connecting these in the structured representation of a case and/or prototypical case.

Although the system goal is not to eliminate completely human expert participation in building the case base, it is nevertheless to minimize this task. Therefore, the present system limits itself to completely automatic methods, while human expert intervention could be solicited at a later stage to refine what the system has learnt, in addition to the automatic enrichment of the case base inherent to the CBR process.

### 4.3 Architecture

ProCaseMiner core component is the RelationshipMiner, which mines for triples  $\langle \text{concept-1}, \text{relationship-1,2}, \text{concept-2} \rangle$  from a document. It also attaches a condition to a triple when it finds it to represent the information that IF a condition occurs, then an action or test is undertaken. This can be represented as  $\langle \text{concept-1}, \text{relationship-1,2}, \text{concept-2} \rangle$  IF  $\langle \text{concept-3}, \text{relationship-3,4}, \text{concept-4} \rangle$ . An example can be  $\langle \text{Patient}, \text{startTreatment}, \text{PrednisoneAndCyclosporineTherapy} \rangle$  IF  $\langle \text{absent}, \text{property\_of}, \text{ImmunosuppressantAgentNOS} \rangle$ . This structure is called a triple pair.

ProCaseMiner interprets the results from RelationshipMiner by successively mining for diagnoses in DiagnosisMiner, findings in FindingMiner, assessments in AssessmentMiner, and treatments in TreatmentMiner. Diagnoses constitute the title or identification of the prototypical cases, while findings, assessments, and treatments constitute the three main parts of each prototypical case. Following, it builds cases from these results in Prototypical-CaseBuilder. In some cases, learnt relationships will be associated with conditions, and in others there will not be any of these conditions. Generally, from medical articles and clinical practice guidelines, the learnt artifact will be a prototypical case, although it is possible that from the particular type of document called a clinical case study, the learnt artifact could be a practice or clinical case. The system in its current stage focuses on learning prototypical cases. A natural extension of this will be learning practice cases. The main difference between the two will be the selection of the articles passed as input to the system.

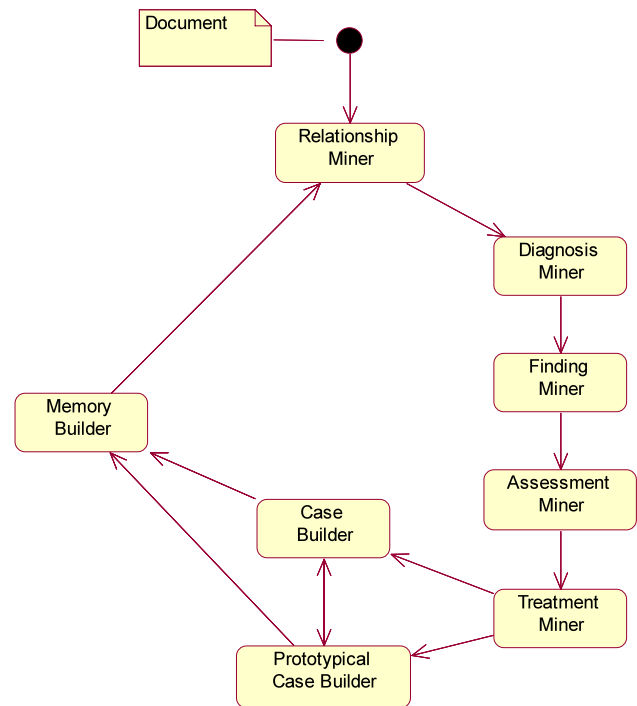


Fig. 5 ProCaseMiner architecture

The previous steps deal with prototypical cases and practice cases built from scratch from a single document. A next step is to consolidate learning results across documents. This step is called MemoryBuilder.

Figure 5 represents the architecture of the system with its different components.

### 4.4 Relationship miner

The RelationshipMiner component is a precursor system of ProCaseMiner developed for the Memoire project. The idea of mining for relationships from medical literature started from a common work with Telemakus project [17], which consists of a set of domain documents (current focus is the biology of aging), a conceptual schema to represent the main components of each document, and a set of tools to query, visualize, maintain, and map the set of documents through their concepts and research findings [17]. For that purpose, Telemakus system mines and maps research findings from research literature. At present, knowledge extraction resorts to systems with both manual and automated components. A key area of current work is to move towards automating the research concept identification process, through data mining [17]. This is exactly why RelationshipMiner was originally developed, before being moved from Telemakus project to be integrated in Memoire project.

The author's research team developed an automated system to mine for concepts linked by relationships from biomedical literature [9]. This system originally kept only the

pairs of concepts in relationship, for the indexing purposes of Telemakus project, and not the relationships between these. The system since then has been improved in its concept mining feature, and a relationship mining feature has been added [10].

Fuller et al. [17] have developed a novel text mining methodology to solve some of the issues related to mining for concepts from the literature. They base their work on the assumption that the research findings brought by an article can be found primarily in the figure and table legends of a document. Another assumption is that researchers are mostly interested in the concepts associated in relationships, and not those that are isolated. Since RelationshipMiner was first developed for Telemakus [17], this system uses the two assumptions above as heuristics to prune the search space of relationships between concepts in a biomedical document. Other heuristics are used during ranking, as is presented later in this article. The quality of the mined relationships is very dependent on the ability of the system to sift through all the possible relationships between concepts in a document to select only those that are important, like a human expert would do. This approach has been successfully validated and is presented in [9, 10].

The RelationshipMiner system involves two knowledge bases, UMLS database, and domain specific database (DSDB), which in particular stores the pre-processed documents that will serve as the input to the system. Within DSDB, the domain specific thesaurus represents the standardized vocabulary of the domain. Concept mining involves processing articles already stored in domain-specific database (DSDB).

These articles comprise the full text of the original articles, parsed in several parts, such as title, summary, section part, figure and table legends, and so forth.

An example of the process flow is provided in Fig. 6, and the components of RelationshipMiner are presented in Fig. 7. The components of the system are the following:

1. *Data Access Component*, which extracts the document to mine from the texts database.

2. *Syntactic Analyzer*, which analyzes the syntax of a document by parsing its sentences and extracting lexical information. Each sentence is parsed and grammatical structures are extracted. From the concept association perspective, each sentence is made up of a connector phrase, called a *trigger phrase*, and the two phrases connected by that trigger phrase. An example of trigger phrase shown on Fig. 6 is “effects of”. These trigger phrases are usually prepositions, but human experts from Telemakus project have also provided special phrases that act as triggers, such as “effect of”. A trigger phrase may contain a connector phrase that separates the remaining part of the sentence into two phrases. After a trigger is found in a sentence, the remaining sentence is split into two phrases optionally connected by a connector

phrase. This phase of the system is called *syntactic analysis* in a broad sense. The connector word and two phrases together are called a *triple*.

Currently, the system uses a basic parser API called *Specialist Text Tools* API that is an open source java implementation [15]. This parser is a minimal commitment barrier category parser. The minimal commitment analysis assigns underspecified syntactic analysis to lexically analyzed input. The parser package contains a shallow parser that extracts minimal phrases from sentences. Using the *Specialist lexicon*, the part of speech and other syntactic information are analyzed. This analysis is specific to biomedical field. The *Specialist Text Tools* tokenizer package tokenizes text into words, sentences, and sections. It can handle free text and Medline citation formats. Sentences are found by looking for sentence bounding punctuation for the most part, and looking at the capitalization of the next word that follows. By the end of processing, an analyzed sentence contains all the tokens that make up the sentence, along with their character offsets back to the original document. The result of this phase is a list of words associated with their grammatical type (see Fig. 6).

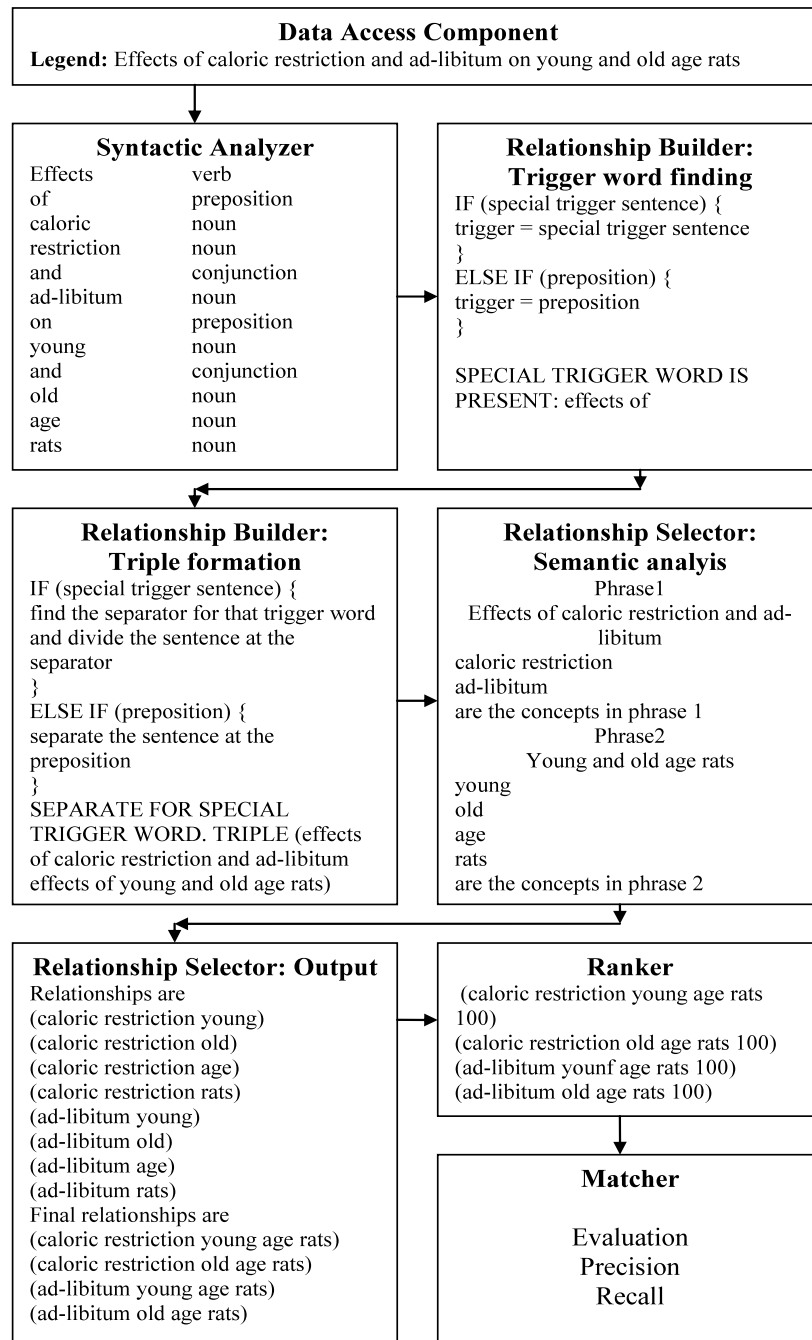
3. *Relationship Builder*, which takes the lexical information from above, locates a trigger phrase for a relationship from each sentence, and forms from there a triple composed of two phrases and a trigger phrase. Each triple represents a relationship between two concepts. Each sentence can generate several triples. An example is *<caloric restriction and ad-libitum, effects\_of, young and old age rats>*.

4. *Relationship Selector*, which semantically analyzes each phrase in each triple by accessing the UMLS, and extracts from each phrase its main concepts. An example issued from the previous triple is *<caloric restriction, effects\_of, young age rats>*. Achieving this result requires in reality recombining pieces from a previous expression, such as combining the triples *<caloric restriction, effects\_of, young>*, *<caloric restriction, effects\_of, age>*, and *<caloric restriction, effects\_of, rats>*.

Semantic analysis is performed on the results of syntactic analysis of the legends to determine the meaning of the words in the sentence. In this step, the semantics of each word or phrase is evaluated. Though there are several choices for performing semantic analysis on free text, this project uses *MMTx* tool [16] as it is specifically developed for the biomedical field. The main purpose of *MMTx* semantic analysis is to find out the phrases and their variants and then match these to the phrases or words in the UMLS database. The words or phrases successfully mapped to the UMLS database can be considered as concepts in the biomedical or health field. In the particular example of Fig. 6, the terms are not found directly in UMLS, but in a dictionary added to UMLS for *MMTx* processing. The concept mapping process performed by *MMTx* can be summarized as follows (see Fig. 6 for an example):

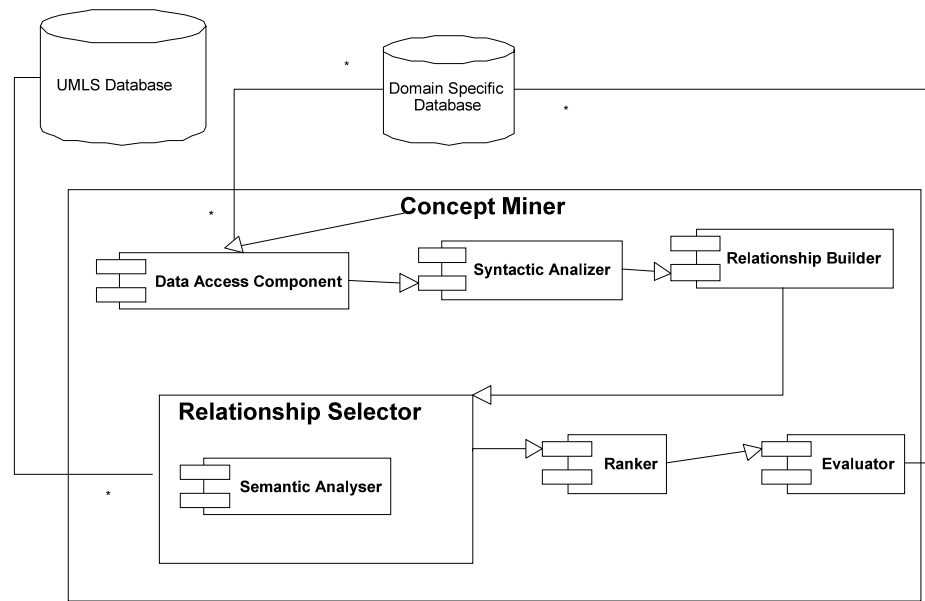


**Fig. 6** RelationshipMiner process flow



- Parse the text into noun phrases and perform the remaining steps for each phrase;
- Generate the variants for the noun phrase where a variant essentially consists of one or more noun phrase words together with all of its spelling variants, abbreviations, acronyms, synonyms, inflectional and derivational variants, and meaningful combinations of these;
- Form the candidate set of all Meta strings containing one of the variants;
- For each candidate, compute the mapping from the noun phrase and calculate the strength of the mapping using an evaluation function. Order the candidates by mapping strength;
- Combine candidates involved with disjoint parts of the noun phrase, recompute the match strength based on the combined candidates;
- Select those having the highest score to form a set of best Meta mappings for the original noun phrase.

**Fig. 7** RelationshipMiner architecture



MMTx provides as output both the Meta string candidates, and the best Meta mapping recommended by this system, such as for string “Obstructive Sleep Apnea” for example [16]:

Phrase: “Obstructive Sleep Apnea”

Meta Candidates (7)

1000 Sleep Apnea, Obstructive [Disease or Syndrome]

901 Apnea, Sleep (Sleep Apnea Syndromes) [Disease or Syndrome]

827 Apnea [Pathologic Function, Sign or Symptom]

827 Obstructive (Obstructed) [Functional Concept]

827 Sleep [Mental Process]

755 Sleeplessness [Disease or Syndrome, Sign or Symptom]

755 Sleepy [Finding]

Meta Mapping (1000)

1000 Sleep Apnea, Obstructive [Disease or Syndrome]

6. *Ranker*, which ranks the relationships extracted, so that not all triples are kept, but only those who contain the type of concepts interesting for the system, namely associations between diagnoses, findings, assessments, and treatments. The semantic type of each concept in a triple is searched for in the UMLS, and only the relationships between concepts related to findings (signs and symptoms), diseases, and treatments, and their attributes, are kept. This permits to further eliminate a lot of relationships from the ones found.

7. *Matcher*, which pairs together triples connected by a WHEN or an IF type of connector. This step takes into account sentences comprising a conditional preposition. For example the triples  $\langle concept-1, relationship-1,2, concept-$

$2 \rangle$  and  $\langle concept-3, relationship-3,4, concept-4 \rangle$  are associated into a triple pair  $\langle concept-1, relationship-1,2, concept-2 \rangle$  IF  $\langle concept-3, relationship-3,4, concept-4 \rangle$  if there exists a conditional preposition between the original phrases or sentences. Examples of conditional prepositions are *if, when, after, and following*.

#### 4.5 Diagnosis, finding, assessment, and treatment miner

From the relationships triples and pairs of triples discovered by RelationshipMiner, diagnoses are extracted by mapping the concepts within these relationships with the UMLS semantic network ‘disease’ concept type. Similarly, findings are mapped to the UMLS semantic network ‘finding’ concept type, assessment with ‘laboratory or test’ concept type, and treatment with ‘clinical drug’, ‘substance’, ‘food’, or ‘planning action’.

#### 4.6 Prototypical case builder

After triples are built, and if some triple pairs are associated with a specific diagnosis, a prototypical case can be built for this diagnosis by connecting in a prototypical case structure the triples associating a diagnosis with its list of findings, list of assessment results, and list of treatment actions. A summarized example can look like the clinical pathway provided on Fig. 2, although some elements are not learnt now, such as the importance of a finding. The corresponding prototypical pathway learnt by the system is provided in Fig. 8.

The linking of the different parts of a prototypical case is made possible by the addition of the triple pairs. The relationship between triple pairs is transitive, therefore it is possible to connect together triples on surface not related. The

**Fig. 8** Prototypical case mined

LiverChronicGVHD		
Connector	Finding Name	(Properties, Values)
	Nausea	
AND	Anorexia	
AND	PainNOS	RightUpperQuadrantAbdomen
AND	Stool	light
AND	ImmunosuppressantAgentNOS	absent

Diagnosis Assessment		
Connector	Procedure Name	(Properties, Values)
	HepaticFunctionPanel	AlkalinePhosphataseMeasurement(ALKP) elevated ASTMeasurement(AST) elevated ALTMeasurement(ALT) elevated LDHMeasurement(LDH) elevated
AND	UltrasonographyAbdomenNOS(USNABD)	normal
AND	HepatitisCAntigenMeasurement	negative
AND	OralExamination	abnormal
AND	RequestGIConsult	LiverChronicGVHD

Treatment Plan	
Condition/Connector	Planning Action Name
	StartPDNCSPTherapy

relationships having led to building the sample prototypical case shown in Fig. 8 are provided in Fig. 9.

#### 4.7 Memory builder

Memory builder is the processing step that takes place after prototypical cases have been built from a document. Generally, prototypical cases can be described in different documents, often partially. For example, one document describes the symptoms of a disease, while another document describes the assessment process, while yet another describes the treatment plan. If a prototype for a disease already exists in memory, the following operations take place:

- Merge the list of findings that are not contradictory. If some findings are contradictory, connect them with an *OR*.
- Choose the list of assessment recommendations which is the most complete, or which comes from the most authoritative source. Often, assessment recommendations are associated with *IF* conditions, such as for example in: *<GIConsult, has-for-result, LiverChronicGVHD> IF <LiverChronicGVHD, causes, RequestGIConsult>*. These types of assessment recommendations can be merged from different documents since the *IF* condition in front will allow to select competing recommendations

based on the clinical context. When none of the heuristics above can determine how to solve conflicting recommendations, they can simply be added connected with an *OR* connector.

- Choose the list of treatment recommendations which is the most complete, or which comes from the most authoritative source. Most often, treatment recommendations are associated with *IF* conditions, such as for example in: *<StartPDNCSPTherapy, treats, LiverChronicGVHD> IF <absent, property-of, ImmunosuppressantAgentNOS>*. These types of treatment recommendations can be merged from different documents since the *IF* condition in front will allow to select competing recommendations based on the clinical context. When none of the heuristics above can determine how to solve conflicting recommendations, they can simply be added connected with an *OR* connector.

Another heuristic used by the system is to process the different documents in an order that facilitates a progressive building of the prototypical cases. For example there exist some documents which obviously contain the type of information the system is looking for. These documents are processed by the system in priority. Among these documents, we can list the clinical practice guidelines devoted to a particular ailment, or the articles the title of which in-

**Fig. 9** Set of triples from which the LiverCGVHD prototypical case shown on Fig. 8 is built

```

<LiverChronicGVHD, causes, Nausea>
<LiverChronicGVHD, causes, Anorexia>
<LiverChronicGVHD, causes, PainNOS>
<RightUpperQuadrantAbdomen, property-of, PainNOS> IF
<LiverChronicGVHD, causes, PainNOS>
<LiverChronicGVHD, causes, light Stool>
<ImmunosuppressantAgentNOS, treats, LiverChronicGVHD>
<LiverChronicGVHD, causes, elevated
AlkalinePhosphataseMeasurement>
<LiverChronicGVHD, causes, elevated ASTMeasurement>
<LiverChronicGVHD, causes, elevated ALTMeasurement>
<LiverChronicGVHD, causes, elevated LDHMeasurement>
<LiverChronicGVHD, causes, normal UltrasonographyAbdomen>
<LiverChronicGVHD, causes, negative HepatitisCAntigenMeasurement>
<LiverChronicGVHD, causes, abnormal OralExamination>
<GIConsult, has-for-result, LiverChronicGVHD>IF <LiverChronicGVHD
causes, RequestGIConsult>
<StartPDNCSPTherapy, treats, LiverChronicGVHD> IF <absent,
property-of, ImmunosuppressantAgentNOS >

```

dicates that they are related to diagnosing and/or treating a particular ailment. We can list as examples: ‘Liver Chronic Graft Versus Host Diseases clinical guideline, or ‘Diagnosis and Treatment of Liver Chronic Graft Versus Host Disease’. The system processes these documents first in order to build the skeleton of its prototypical cases, and the remainder of the documents base comes to enrich the prototypical cases with additional details or more up-to-date clinical information.

Indeed automatically building the case memory could in a straight forward manner evolve into automatically updating a case memory, and this is not a minor advantage of the method advocated here.

## 5 Evaluation

The ProCaseMiner system was evaluated by comparison with the knowledge-base developed for the FHCRC Carepartner system [12]. This knowledge-base comprises in particular 112 prototypical cases developed over the course of two years.

The documents processed are all related to the hematopoietic stem-cell transplantation domain. Moreover, only the authors from FHCRC were selected, and topics related to diagnosis and treatment, meta analysis articles, and clinical practice guidelines available online. A total of 500 articles were selected from over 5000 for their pertinence and coverage of the task at hand.

The success of the system is determined by the recall and precision ratios. Precision is the ratio of matching features to the total number of features identified. Recall is the ratio of matching features to the total number of features identified by the manual process. The precision and recall are calculated both at the prototypical case level, and at the level

of features matched within each prototype. For example, a match between a prototypical case entitled LiverChronicGVHD mined by the system and LiverChronicGVHD acquired from the human experts corresponds to a prototypical case match; a match between JaundiceNOS between the actual LiverChronicGVHD clinical pathway, and the learnt LiverChronicGVHD prototypical case corresponds to a feature match. More precisely, (1) and (2) provide the formulae to calculate precision and recall at the prototypical case level, and (3) and (4) to calculate precision and recall at the feature level.

*case level precision*

$$= \frac{\text{Count}(\text{matching prototypical cases})}{\text{Count}(\text{ProCaseMiner found prototypical cases})}, \quad (1)$$

*case level recall*

$$= \frac{\text{Count}(\text{matching prototypical cases})}{\text{Count}(\text{human found prototypical cases})}, \quad (2)$$

*feature level precision*

$$= \frac{\text{Count}(\text{matching features})}{\text{Count}(\text{ProCaseMiner found features})}, \quad (3)$$

$$\text{feature level recall} = \frac{\text{Count}(\text{matching features})}{\text{Count}(\text{human found features})}. \quad (4)$$

The system is evaluated for all the 500 articles. The average values of recall and precision for these documents are shown in Table 2. It shows very encouraging results, even though the process of learning prototypical cases is very complex. These results show that the system definitely learns structures closely related to what the experts came up with—and so in a much shorter time (one day of processing versus 2 years with the knowledge elicitation work with the team members). Precision and recall are known as quality measures varying in opposite direction, meaning that if we

**Table 2** Precision and recall ratios

Number of documents	Prototypical case		Feature	
	Recall	Precision	Recall	Precision
500	95%	62%	70%	65%

try to increase the precision, then the recall will be less good. For example here, the precision of 62% for prototypical case precision means that the system has learnt many different prototypical cases from what the humans have determined. The recall of 95% means that the system has learnt 95% of the prototypical cases found by humans. Therefore we could think of fine-tuning the system to be more selective about which prototypical cases are to be kept—by restricting the ranking threshold considered as significant. Nevertheless, by doing so, the system would lose some of its ‘good’ prototypical cases. Consequently, the precision would increase but the recall would decrease. The same comments can be applied to the features, although here there is definitely additional work to accomplish to attempt to raise both precision and recall. The area of improvement will be the manner by which the system reconciles contradictory features from different documents, which is currently too strict.

These results also show that if the system does not learn as much as the experts, it would provide an excellent starting point for further refinements with the medical team. The time saved is very significant, and moreover the approach presents major advantages.

A major advantage is that the system can be trained continuously and thus could regularly update its knowledge base—a case base maintenance issue [18]. In fact, an explanation of why the results are somewhat different from the clinical team prototypes is that knowledge has evolved since then, and it may simply be that the current prototypes are more current than the ones from the knowledge-base, which date five years now. Another explanation is that not all the articles were processed, and that a more judicious choice or more complete choice would provide better results. Finally, there is no evidence that the prototypes provided by the human experts are always better. What would be an interesting supplemental evaluation would be to compare recommendations from the system for both knowledge bases.

## 6 Discussion

Important previous work has been attempted in CBR to either retrieve textual cases [19], or to apply CBR to information retrieval [20]. Early work has also studied how to extract expertise from cases [21]. In reality, the present work does not deal with textual cases, and thus is not directly related to textual CBR literature. The system extracts cases from documents, through a text mining process, but does not actually

reason from the documents. As a matter of fact, the documents themselves may not refer at all to ‘cases’. The prototypical case structure is a knowledge representation artifact suited for merging well with a case base. A future extension of this work is to apply CBR to biomedical cases described in textual format, where textual CBR will become very pertinent. This is the proposed work of mining for individualized clinical cases from clinical cases studies published in the literature, although here also the author may prefer to first mine for the individualized cases through text mining, instead of directly working from cases in textual format. Other CBR researchers have been applying CBR to information retrieval, such as [20]. This work will be also pertinent for our future work on retrieving textual cases. These authors [20] follow an approach inverse to ours: showing that CBR can be pertinent for intelligent information retrieval. Other CBR researchers [22] have studied similarity measures used in information retrieval, such as the cosine measure, and how they relate to CBR retrieval. This article does not deal with similarity measures, but with case mining.

Pertinent previous work [23] relates to case mining, feature mining, and prototype mining. These are addressed in successive paragraphs.

Case mining refers to the process of mining potentially large data sets for cases [24]. Researchers have often noticed that cases simply do not exist in electronic format, that databases do not contain well-defined cases, and that the cases need to be created before CBR can be applied. Another option is to start CBR with an empty case base. When large databases are available, preprocessing these to learn cases for future CBR permits to capitalize on the experience dormant in these databases. Qiang and Cheng [24] propose to learn cases by linking several database tables. *Clustering* and *SVM* techniques permit to mine for cases in [24].

Feature mining refers to the process of mining data sets for features. Many CBR systems select the features for their cases, and/or generalize them. Wiratunga et al. [23] note that transforming textual documents into cases requires dimension reduction and/or feature selection, and shows that this improves the classification and CBR accuracy—and efficiency. These authors induce a kind of decision tree called boosted *decision stumps* because they have only one level in order to select features, and induce *rules* to generalize the features. In biomedical domains, in particular when data vary continuously, the need to abstract features from streams of data is particularly prevalent. Recent, and notable, examples include [25, 26] who reduce their cases time series dimensions through *Discrete Fourier Transform*. Niloofer and Jurisica [27] propose an original method for generalizing features. Here the generalization is an abstraction that reduces the number of features stored in a case. Applied to the bioinformatics domain of micro arrays, the system uses both *clustering* techniques to group the cases into

clusters containing similar cases, and feature selection techniques.

Generalized case mining refers to the process of mining databases for generalized and/or abstract cases. Generalized cases are called in varied ways, such as prototypical cases, abstract cases, prototypes, stereotypes, templates, classes, categories, concepts, ossified cases, and scripts. Although all these terms refer to slightly different concepts, they represent structures that have been abstracted or generalized from real cases either by the CBR system, or by an expert. When these prototypical cases are provided by a domain expert, this is a knowledge acquisition task. Many authors mine for *prototypes*, and simply refer to *induction* for learning these, such as CHROMA [28]. Bellazzi et al. [29] organize their memory around prototypes. The prototypes can either have been acquired from an expert, or induced from a large case base. Schmidt and Gierl [30] point that prototypes are an essential knowledge structure to fill the gap between general knowledge and cases in medical domains. The main purpose of the prototype learning step they propose is to guide the retrieval process and to decrease the amount of storage by erasing redundant cases. A generalization step is necessary to learn the knowledge contained in stored cases. They use several threshold parameters to adjust their prototypes, such as the number of cases the prototype is filled with, and the minimum frequency of each contraindication for the antibiotic therapy domain. Others specifically refer to *generalization*, so that their prototypes correspond to generalized cases. An example of system inducing prototypes by generalization is a computer aided medical diagnosis system interpreting electromyography for neuropathy diagnosis [31]. A similar connectionist approach is proposed by Reategui et al. [32]. Portinale and Torasso [33] in ADAPTER organize their memory through E-MOPs learnt by generalization from cases for diagnostic problem-solving. E-MOPs carry the common characteristics of the cases they index, in a discrimination network of features used as indices to retrieve cases. Mougouie and Bergmann [34] present a method for learning generalized cases, called the Topkis–Veinott method. Maximini et al. [35] have studied the different structures induced from cases in CBR systems. They point out that several different terms exist, such as generalized case, prototype, schema, script, and abstract case. The same terms do not always correspond to the same type of entity. They define three types of cases: point cases, and two forms of generalized cases. Yet other authors refer to *abstraction* for learning abstract cases. Branting [36] proposes case abstractions for its memory of route maps. The abstract cases, which also contain abstract solutions, provide an accurate index to less abstract cases and solutions. Perner [37] learns prototypes by abstracting cases as well for improving image pattern recognition.

Finally, many authors learn *concepts*, and refer to conceptual clustering as their learning methodology [37–41]. Diaz-Agudo and González-Calero [42] use *formal concept analysis* (FCA)—a mathematical method from data analysis—as another induction method for extracting knowledge from case bases, in the form of *concepts*.

The abundance of literature in case, feature, and prototype mining shows that this question is essential to CBR, as a machine learning methodology. ProCaseMiner is mostly related to case mining, but differs from previous approaches [24] by mining from literature. It does not mine from textual cases as in [23]. One of the main advantages of the method proposed here is that it will facilitate the bootstrapping of CBR systems in biomedicine by providing a starting case base of mostly prototypical cases, which will render the methodology readily applicable to a much wider range of domains, in particular those where electronic cases are not available, like Carepartner [12].

## 7 Conclusion

Case mining and prototypical case mining from medical literature is a very promising approach to building case bases. It has the potential of spreading the development of CBR systems in many domains where either electronic cases are not available, or they are incomplete, which is most frequent, or experts are not available for months or years of knowledge elicitation tasks. Moreover, it offers new opportunities for updating case bases from recent medical advances, and for leveraging multiple domains CBR. The ProCaseMiner system presented in this article focuses on learning prototypical cases from biomedical literature. A natural extension of this work will involve learning individualized clinical cases whenever pertinent literature such as clinical case studies is available. Research ahead in this direction involves automatically selecting the body of documents the most adequate for feeding the case mining system, learning more complex case structures and features, combining case mining from databases and from literature, and studying the knowledge discovery process in itself from both the case-based standpoint, the rule-based standpoint, and the model-based standpoint.

## References

1. Swanson DR (2001) Information discovery from complementary literatures: categorizing viruses as potential weapons. *J Am Soc Inf Sci* 52(10):797–812
2. Swanson DR, Smalheiser NR (1997) An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artif Intell* 9:183–203
3. Han J, Kamber M (2006) *Data mining concepts and techniques*, 2nd edn. Kaufmann, San Mateo

4. Hearst MA (1999) Untangling text data mining. In: Dale R, Church K (eds) Proceedings of the 37th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Morristown, pp 3–10
5. Nasukawa T, Nagano T (2001) Text analysis and knowledge mining system. In: Knowledge management Special Issue, IBM Syst J, 40:967–984
6. Dorre J, Gerstl P, Seiffert R (1999) Text mining: finding nuggets in mountains of textual data. In: Chaudhuri S, Madigan D, Fayyad U (eds) Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 398–401
7. Weeber M, Vos R, Klein H, de Jong-van den Berg LTW, Aronson AR, Lolema G (2003) Generating hypotheses by discovering implicit associations in the literature: a case report of a search for new potential therapeutic uses for thalidomide. *J Am Med Inf Assoc* 10(3):252–259
8. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics* 17(1):S74–S82
9. Bichindaritz I, Akkineni SS (2005) Concept Mining for Indexing Medical Literature. In: Perner P, Ahlemeyer-Stubbe A (eds) Machine learning and data mining (MLDM), industrial conference on data mining (ICDM) 2005. Lectures notes in artificial intelligence. Springer, Leipzig, pp 682–691
10. Bichindaritz I (2006) Named relationship mining from biomedical literature. In: Perner P, Ahlemeyer-Stubbe A (eds) Industrial conference on data mining (ICDM) 2006. Lectures notes in artificial intelligence. Springer, Leipzig, pp 64–75
11. Bichindaritz I (2004) Mémoire: case-based reasoning meets the semantic web in biology and medicine. In: Funk P, González Calero PA (eds) Proceedings of ECCBR 2004. Lecture notes in artificial intelligence, vol 3155. Springer, Berlin, pp 47–61
12. Bichindaritz I, Kansu E, Sullivan KM (1998) Case-based reasoning in CARE-PARTNER: gathering evidence for evidence-based medical practice. In: European workshop on case-based reasoning. Lectures notes in artificial intelligence, vol 1488. Springer, Berlin, pp 334–345
13. Bichindaritz I (2006) Semantic interoperability of case bases in biology and medicine. In: Bichindaritz I (ed) Special issue on case-based reasoning in the health sciences. *Artif Intell Med*, 36(2):177–192
14. National Library of Medicine (2005) The Unified Medical Language System. <http://umls.nlm.nih.gov>. Last access: 2005-04-01
15. National Library of Medicine (2004) The Specialist NLP Tools. <http://specialist.nlm.nih.gov>. Last access: 2005-04-01
16. National Library of Medicine (2005) MetaMap Transfer (MMTx). <http://mmtx.nlm.nih.gov>. Last access: 2005-04-01
17. Fuller S, Revere D, Bugni P, Martin GM (2004) A knowledgebase system to enhance scientific discovery: telemakus. *Biomed Digit Libr*. Sep 21 1(1):2
18. Wilson DC, Leake DB (2001) Mainting case-based reasoners: dimensions and directions. *Comput Intel J*, 17(2)
19. Ashley KD, Lenz M (eds) (1998) Textual case-based reasoning, AAAI-98 Workshop. Technical Report WS-98-12, AAAI, Menlo Park
20. Daniels JJ, Rissland EL (1995) A case-based approach to intelligent information retrieval. In: Proceedings of SIGIR. ACM, New York, pp 238–245
21. Beck HW (1991) Language acquisition from cases. In: Bareiss R (ed) Proceedings of a workshop on case-based reasoning (DARPA). Kaufmann, San Mateo, pp 159–169
22. Wilson DC, Bradshaw S (2000) CBR textuality. *Expert Updat* 3(1):28–37
23. Wiratunga N, Koychev I, Massie S (2004) Feature selection and generalisation for retrieval of textual cases. In: Funk P, González Calero P (eds) Proceedings of ECCBR 04. Lecture notes in artificial intelligence, vol 3155. Springer, Berlin, pp 806–820
24. Qiang Y, Cheng H (2003) Case mining from large databases. In: Ashley K, Bridge DG (eds) Proceedings of ICCBR 03. Lecture notes in artificial intelligence, vol 2689. Springer, Berlin, pp 691–702
25. Montani S, Portinale L, Bellazzi R, Leornardi G (2004) RHENE: a case retrieval system for hemodialysis cases with dynamically monitored parameters. In: Funk P, González Calero P (eds) Proceedings of ECCBR 04. Lecture notes in artificial intelligence, vol 3155. Springer, Berlin, pp 659–672
26. Nilsson M, Funk P (2004) A case-based classification of respiratory sinus arrhythmia. In: Funk P, González Calero P (eds) Proceedings of ECCBR 04. Lecture notes in artificial intelligence, vol 3155. Springer, Berlin, pp 673–685
27. Niloofar A, Jurisica I (2004) Maintaining case-based reasoning systems: a machine learning approach. In: Funk P, González Calero P (eds) Proceedings of ECCBR 04. Lecture notes in artificial intelligence, vol 3155. Springer, Berlin, pp 17–31
28. Armengol E, Plaza E (1994) Integrating induction in a case-based reasoner. In: Keane M, Haton J-P, Manago M (eds) Proceedings of EWCBR 94. Acknosoft, Paris, pp 243–251
29. Bellazzi R, Montani S, Portinale L (1995) Retrieval in a prototype-based case library: a case study in diabetes therapy revision. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR 98. Lecture notes in artificial intelligence, vol 1488. Springer, Berlin, pp 64–75
30. Schmidt R, Gierl L (1995) Experiences with prototype designs and retrieval methods in medical case-based reasoning systems. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR. Lecture notes in artificial intelligence, vol 1488. Springer, Berlin, pp 370–381
31. Malek M, Rialle V (1994) A case-based reasoning system applied to neuropathy diagnosis. In: Keane M, Haton J-P, Manago M (eds) Proceedings of EWCBR. Acknosoft, Paris, pp 329–336
32. Reategui E, Campbell JA, Borghetti S (1995) Using a neural network to learn general knowledge in a case-based system. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR. Lecture notes in artificial intelligence, vol 1010. Springer, Berlin, pp 528–537
33. Portinale L, Torasso P (1995) ADAPTER: an integrated diagnostic system combining case-based and abductive reasoning. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR. Lecture notes in artificial intelligence, vol 1010. Springer, Berlin, pp 277–288
34. Mougouie B, Bergmann R (1995) Similarity assessment for generalized cases by optimization methods. In: Craw S, Preece A (eds) Proceedings of EWCBR 02. Lecture notes in artificial intelligence, vol 2416. Springer, Berlin, pp 249–263
35. Maximini K, Maximini R, Bergmann R (2003) An investigation of generalized cases. In: Ashley KD, Bridge DG (eds) Proceedings of ICCBR 03. Lecture notes in artificial intelligence, vol 2689. Springer, Berlin, pp 261–275
36. Branting KL (1997) Stratified case-based reasoning in non-refinable abstraction hierarchies. In: Leake D, Plaza E (eds) Proceedings of ICCBR. Lecture notes in artificial intelligence, vol 1266. Springer, Berlin, pp 519–530
37. Perner P (1995) Different learning strategies in a case-based reasoning system for image interpretation. In: Smyth B, Cunningham P (eds) Proceedings of ECCBR. Lecture notes in artificial intelligence, vol 1488. Springer, Berlin, pp 251–261
38. Bichindaritz I (1995) A case based reasoner adaptive to several cognitive tasks. In: Veloso M, Aamodt A (eds) Proceedings of ICCBR 95. Lecture notes in artificial intelligence, vol 1010. Springer, Berlin, pp 391–400
39. Bichindaritz I (1995) Case-based reasoning and conceptual clustering: for a co-operative approach. In: Watson I, Fahrir M (eds) Lecture notes in artificial intelligence, vol 1020. Springer, Berlin, pp 91–106

40. Lebowitz M (1986) Concept learning in a rich input domain: generalization-based memory. In: Michalski RS, Carbonell JG, Mitchell TM (eds) *Machine learning: an artificial intelligence approach*, vol 2. Kaufmann, Los Altos, pp 3–25
41. Perner P (2003) Incremental learning of retrieval knowledge in a case-based reasoning system. In: Ashley K, Bridge DG (eds) *Proceedings of ICCBR 03. Lecture notes in artificial intelligence*, vol 2689. Springer, Berlin, pp 422–436
42. Diaz-Agudo B, González-Calero P (1995) Classification based retrieval using formal concept analysis. In: Aha D, Watson I (eds) *Proceedings of ICCBR 01. Lecture notes in artificial intelligence*, vol 2080. Springer, Berlin, pp 173–188