Proceedings of the

International Joint Workshop

on

Knowledge Representation for Health Care

Process-Oriented Information Systems in Health Care

Extraction & Processing of Rich Semantics from Medical Texts


Vienna, Austria

June 24th, 2017
Preface

This document contains the joint proceedings of three of the workshops organized in the International Artificial Intelligence in Medicine Conference (AIME) in 2017, Vienna, Austria. These workshops were the 9th Int'l Workshop on Knowledge Representation for Health Care (KR4HC), the 10th Int'l Workshop on Process-Oriented Information Systems in Healthcare (ProHealth), and the 2nd Int'l Workshop on Extraction and Processing of Rich Semantics from Medical Texts (RichMedSem).

As part of medical informatics, KR4HC focuses on representing and reasoning with medical knowledge in computers as a means to support knowledge management, clinical decision-making, and health care modeling and medical simulation. The KR4HC community aims at developing efficient representations, technologies, and tools for integrating all the important elements that health care providers work with: Electronic Medical Records (EMRs) and healthcare information systems, clinical practice guidelines, and standardized medical vocabularies.

The ProHealth workshop focuses on using information technology to improve the management and quality of healthcare processes. This community aims both at adapting successful business process management (BPM) solutions to the healthcare domain and proposing novel approaches that address the specific needs of healthcare processes. Particular areas of improvement are organization, optimization, cooperation, risk analysis, flexibility, re-utilization, and integration of health care tasks and teams.

As a complement, RichMedSem addresses different aspects of accessing, extracting, processing, and filtering health related textual information. The aim of the workshop is to encourage researchers from the communities of medical natural language processing (NLP), information retrieval, knowledge representation and knowledge management to present novel issues and techniques related to the extraction and processing of rich semantics from medical texts, but more importantly to discuss current challenges and future steps towards new directions for gathering and processing rich semantics in the medical domain. One topic are applications in the medical domain that already make use of extracted rich semantics and how to make them more reliable and adaptable. The objective is to go beyond extraction of basic information such as diagnoses, symptoms and treatments, and focus instead on methods that extract quantitative and qualitative information from clinical records, medical guidelines, research literature and other medically relevant sources.

David Riaño
Mor Peleg
Richard Lenz
Manfred Reichert
Kerstin Denecke
Yihan Deng
Thierry Declerck
Frank van Harmelen
Organizing Committee

**KR4HC-ProHealth**

David Riaño, Universitat Rovira i Virgili, Spain  
Mor Peleg, University of Haifa, Israel  
Richard Lenz, University of Erlangen-Nuremberg, Germany  
Manfred Reichert, University of Ulm, Germany

**RichMedSem**

Kerstin Denecke, Bern University of Applied Sciences, Switzerland  
Yihan Deng, Bern University of Applied Sciences, Switzerland  
Thierry Declerck, Saarland University & German Research Center for AI, Germany  
Frank van Harmelen, Vrije Universiteit Amsterdam, The Netherlands

Program Committee

- Syed Sibte Raza Abidi, Dalhousie University
- Luca Anselma, Università di Torino
- Joseph Barjis, Delft University of Technology
- Arturo González-Ferrer, Instituto de Investigación Sanitaria San Carlos
- David Isern, Universitat Rovira i Virgili
- Stefan Jablonski, University of Bayreuth
- Vasileios Koutkias, INSERM
- Peter Lucas, Radboud University Nijmegen
- Wendy MacCaull, Dept of Math/Stats/Comp Sci, St. Francis Xavier University
- Mar Marcos, Universitat Jaume I, Castellón
- Stephanie Medlock, Academic Medical Center, University of Amsterdam
- Silvia Miksch, Vienna University of Technology
- Stefania Montani, University of Piemonte Orientale
- Øystein Nytrø, Norwegian University of Science and Technology
- Leon Osterweil, UMass Amherst
- Mor Peleg, University of Haifa
- Danielle Sent, AMC/UvA
- Brigitte Seroussi, Assistance Publique - Hôpitaux de Paris
- Yuval Shahar, Ben Gurion University
- Ton Spil, University of Twente
- Maria Taboada, University of Santiago de Compostela
- Annette Ten Teije, VU University Amsterdam
- Paolo Terenziani, Dipartimento di Informatica, Università del Piemonte Orientale "Amedeo Avogadro"
- Lucinéia Heloísa Thom, Federal University of Rio Grande do Sul
- Frank Van Harmelen, Vrije Universiteit Amsterdam
- Dongwen Wang, Department of Biomedical Informatics, Arizona State University
- Barbara Weber, University of Innsbruck
- Szymon Wilk, Poznan University of Technology
Day Program

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td>Welcome</td>
</tr>
<tr>
<td>9:10</td>
<td>Session I: Keynote</td>
</tr>
<tr>
<td>10:30</td>
<td>Coffee-Tea break</td>
</tr>
<tr>
<td>11:00</td>
<td>Session II: Temporal Analysis, Representation, and Simulation</td>
</tr>
<tr>
<td>12:30</td>
<td>Lunch break</td>
</tr>
<tr>
<td>14:00</td>
<td>Session III: Modeling and Decision Support, Data Analysis.</td>
</tr>
<tr>
<td>15:30</td>
<td>Session IV: Natural Language Processing and Deep Learning</td>
</tr>
<tr>
<td>17:00</td>
<td>Closure</td>
</tr>
</tbody>
</table>
Table of Contents

Preface
Organizing Committee
Program Committee
Day Program

Session I: Keynote
How Linked (Open?) Data can benefit Healthcare Systems .......................................................... 9
Frank van Harmelen

Session II: Temporal Analysis, Representation, and Simulation
Temporal Conformance Analysis and Explanation on Comorbid Patients. .......................... 11
Luca Piovesan, Paolo Terenziani, and Daniele Theseider Dupré.

Representing and reasoning with probabilistic temporal distances for guideline interaction analysis. ................................................................. 25
Paolo Terenziani, Antonella Andolina.

Utilizing Temporal Genomic Data for Healthcare. ................................................................. 31
Guenter Tusch and Shahrzad Eslamian.

Representation of Temporal Constraints in Clinical Guidelines using openEHR archetypes and GDL. ................................................................. 39
Lilian Mie Mukai Cintho, Claudia Moro, Lucia Sacchi, and Silvana Quaglini.

A Knowledge-Based Conceptual Model to Simulate Circulatory Shock. ................................. 45
David Riaño, José Ramón Alonso.

A Data- and Expert-driven Decision Support Framework for Helping Patients Adhere to Therapy: Psychobehavioral Targets and Associated Interventions. ................................. 53
Szymon Wilk, Dympna O’Sullivan, Martin Michalowski, Silvia Bonaccio, Wojtek Michalowski, Mor Peleg, and Marc Carrier.

A general framework for the distributed management of exceptions and comorbidities. ....... 67
A. Bottrighi, L. Piovesan, P. Terenziani.

A Label Taxonomy to Support Collaborative Formalization of Computer-interpretable Guidelines and Classification of Modeling Issues. ................................................................. 81
Arturo González-Ferrer, María Ángel Valcárcel, Henar González-Luengo, and Enea Parimbelli.

............................................................................................................................................. 89

*A Self-Enforcing Network for the Analysis of Fall Cases in a University Hospital.* ............................. 97

Christina Klüver and Christian Dahlmann.

**Session IV: Natural Language Processing and Deep Learning**

Towards automatic clinical operation encoding (CHOP) with convolutional neural networks. ................................................................. 105

Yihan Deng, André Sander, Lukas Faulstich, and Kerstin Denecke.

*Corpus Annotation for Aspect Based Sentiment Analysis in Medical Domain.* ................................. 115


Analyzing cancer forum discussions with text mining.......................... 127

Gerard van Oortmerssen, Stephan Raaijmakers, Maya Sappelli, Erik Boertjes, Suzan Verberne, Nicole Walasek, and Wessel Kraaij.

*Design of an Information Extraction Pipeline for German Clinical Texts.* ............................... 131

Madeleine Kittner, Bariya Bajwa, Damian Rieke, Mario Lamping, Johannes Starlinger, and Ulf Leser.
How Linked (Open?) Data can benefit Healthcare Systems

Frank van Harmelen
Knowledge Representation & Reasoning
Computer Science Department
Vrije Universiteit Amsterdam

Abstract. A steady progress in semantic technologies over the past decade and a half has resulted in stable syntactic and semantic models for publishing and interlinking datasets on the web. Such interlinked and interoperable datasets have had a significant impact on a large number of technology sectors: e-commerce, cultural heritage, science, media and publishing, just to name a few.

However, the impact of linked data on healthcare information systems has been limited so far in comparison with these other sectors. In this talk I will argue that Linked Data technologies can also be very useful for a variety of applications in healthcare information systems, and that this is even true (perhaps surprisingly) for Linked *Open* Data.
Temporal Conformance Analysis and Explanation on Comorbid Patients*

Luca Piovesan, Paolo Terenziani, and Daniele Theseider Dupré

DISIT, Computer Science Institute, Università del Piemonte Orientale
{lucapiovesan|paolo.terenziani|dtd}@uniupo.it

Abstract. The treatment of comorbid patients is one of the main challenges of modern health care, and many Medical Informatics approaches have been devoted to it in the last years. In this paper, we propose the first approach in the literature that analyses the conformance of execution traces with multiple Computer-Interpretable Guidelines (CIGs), as needed in the treatment of comorbid patients. This is a fundamental task, to support physicians in an a-posteriori analysis of the actually provided treatments. Notably, the conformance problem is very complex in this context, since CIGs may have negative interactions, so that in specific circumstances full conformance may be dangerous for patients. We thus complement our conformance analysis with an explanation approach, aimed at justifying deviations in case they had avoided some possible undesired interaction. Our approach is based on Answer Set Programming, and, to face realistic problems, devotes specific attention to the temporal dimension.

1 Introduction

Clinical practice guidelines are the major tool that has been introduced to grant both the quality and the standardization of healthcare services, on the basis of evidence-based recommendations. The adoption of computerized approaches to acquire, represent, execute and reason with Computer-Interpretable Guidelines (CIGs henceforth) can provide crucial additional advantages. Therefore, in the last twenty years, many different approaches and projects have been developed to manage CIGs (consider, e.g., the book [16] and the survey [8]). By definition, clinical guidelines address specific clinical circumstances (i.e., specific pathologies). However, individual patients may be affected by more than one pathology (comorbid patients). The treatment of such patients is one of the main challenges for modern health care, also due to the aging of population, and the increase of chronic pathologies. In fact, in comorbid patients the treatments of single pathologies may interact with each other, and the approach of proposing an ad-hoc “combined” treatment to cope with each possible comorbidity does

* This research is original and has a financial support of the Università del Piemonte Orientale.
not scale up: “Developing Clinical Practice Guidelines that explicitly address all potential comorbid diseases is not only difficult, but also impractical, and there is a need for formal methods that would allow combining several disease-specific clinical practice guidelines in order to customize them to a patient” [7]. Thus, new methodologies are required to study the interactions between treatments, and to combine treatments: “This sets up the urgent need of developing ways of merging multiple single-disease interventions to provide professionals’ assistance to comorbid patients” [13]. In the last years, several computer-based approaches have started to face this problem, aiming at providing physicians with different forms of support for managing multiple CIGs and their interactions.

In our previous work in this area, we have devised a set of methodologies to support physicians in the management of comorbid patients: physician-driven navigation of CIGs at different levels of abstraction, to focus on the parts that are relevant for potential interactions [9]; knowledge-based detection of interactions between actions [1]; mixed-initiative management of detected interactions [10]; merging the CIGs into a unique treatment for a given patient [11].

In this paper, we ground on the above mentioned previous work and on an approach to check a-posteriori conformance of the treatment of a patient with respect to one clinical guideline [15], to face the comorbidity problem from a new perspective (that, to the best of our knowledge, has not been considered yet): we explore the interplay between CIGs from the viewpoint of a posteriori conformance analysis [5], intended as the adherence of an observed CG execution trace to the CIGs executed on a (comorbid) patient. Our goal is not to provide an evaluation of whether the treatment was appropriate or not; rather, we identify actions that have been executed or could have been executed, following the individual CIGs, and may interact, according to the approach in [1]; and we allow for interpreting the actual trace according to possible ways for managing interactions (defined in [10]). Therefore, the trace may deviate from a strict application of the individual CIGs under execution. Notably, we also identify and signal cases in which the execution log is conformant with all the CIGs, but some interaction has not been avoided (since, in such cases, the adherence to the CIGs might not have been the best option). Reasoning on interactions in a-posteriori conformance analysis may complement reasoning (on the same knowledge) to support CIG execution: for example, it may be used (see [12]) at discharge time to support documenting the patient care process, while at execution time physicians may not have time to interact with a support tool.

Significant conclusions can be drawn from this conformance analysis if the trace is reasonably correct and complete as regards patient data and executed actions. Missing events in a trace, or missing data in the description of an event (or even the incorrect recording of events), can indeed be hypothesized to have occurred or to hold, in order to make conformance analysis more flexible [3]; however, a high number of incorrect or missing events in the trace with respect to actual events, in combination with several ways (studied in [15] and in the present paper) for explaining discrepancies between CIG and trace, would make the space of possible explanations quite large. In this paper we concentrate on explaining
discrepancies in terms of management of possible interactions; discrepancies that cannot be explained in this way are considered cases of non-conformance, which may be due to incompleteness or incorrectness of the trace, even though, in general, there is no way for distinguishing. On the other hand, it is not realistic to assume that at execution time explicit information is recorded on the fact that the actions that have actually been executed deviate from the recommendations of an individual CIG: detecting this is part of conformance analysis.

To make our approach more precise in the analysis of non-conformances due to possible interactions, we take into account the temporal dimension. Indeed, patient data hold at specific (intervals of) time; the timing of actions should respect temporal constraints in the CIGs, and interactions occur (or do not occur) in time. This makes conformance analysis more complex but richer: for instance, the CIG constraints may be violated in order to temporally avoid undesired interactions. Temporal analysis requires that at least imprecise temporal information on actions is provided; imprecision could lead an interaction, and then an explanation, to be considered possible (due to possible overlap of effects in time) in cases where more precise temporal information would not.

In this paper, we propose a general methodology to cope with the above issues, that is mapped to Answer Set Programming (ASP), which, as shown in [15] for the case of a single CIG, is quite useful to analyse conformance, since from one side it supports the non-monotonic forms of reasoning naturally used by physicians in this context and, from the other side, naturally supports the search for alternative explanations (solutions).

We demonstrate the potential of our approach on a relatively simple - but explicative - example. As a future work, we plan to make a deep experimentation of our approach, considering actual traces containing also imprecise (and possibly missing) data.

2 Preliminaries

At least four different types of data/knowledge sources should be considered to analyze compliance: patient data, traces of execution, CIG models, and general knowledge about action effects, intentions and interactions between such elements (henceforth, called ontological knowledge).

For patient data we mean patients’ findings, i.e., data which are usually collected in patients’ electronic health records (EHR). In particular, as discussed in the introduction, we intend that available data represent all known information that is relevant for treating the patient, and that such pieces of information are temporally tagged (possibly with imprecision). Also, the available execution trace (trace for short) is considered as all that is known on the clinical actions executed on the patient, The occurrence of actions is temporally tagged with the start and end time. We also assume that, in the case of decisions, the log explicitly indicates the chosen alternative.

As concerns the CIG model, our approach is not biased towards any specific CIG formalism; however, we will use the GLARE formalism [17] as a concrete
example, due to its specific attention to the temporal aspects. Indeed, we simply consider the possibility of distinguishing between atomic and composite actions, and of specifying (therapeutic and diagnostic) decisions. CIGs specify the control flow of actions and include temporal constraints between them. Additionally, actions may have preconditions, and temporal constraints between the time when preconditions hold and the time when the related action must be executed can be specified. Actions are considered for execution as follows:

– When the control flow indicates that an action \( a \) will have to be executed (in a time window dependent on the temporal constraints in the CIG), we say that \( a \) becomes scheduled. This means that, in case we are considering a sequence of actions followed by a decision, all the actions in the sequence (and the decision) are scheduled (while the actions following the decision are not, since they belong to alternative paths, and, at this point of the analysis, the physician could not know a-priori which alternative she will take)
– When an action is reached by the control flow, i.e., the previous action ends, it becomes candidate and its execution proceeds according with the execution model described in [15]. A candidate action could become active or discarded; if active, it could either be completed or aborted. Here, we just remark that: An action should start (become active) at a time such that all preconditions, with their temporal constraints, enable the action, if such time exists.

Ontological knowledge. Possible interactions between CIGs, and between CIG recommendations and the patient status have to be identified. To do so, we have to explicitly consider also the effects and intentions of actions, and the time window in which such elements can occur. Additionally, we must rely on a knowledge base that models the possible interactions between action effects and intentions, which is CIG and patient independent. To address such a need, we take advantage of the temporal ontology we have provided in [1] to devise a decision support system helping physicians in the treatment of comorbid patients.

3 A general approach to conformance analysis and explanation

A high-level view of our general methodology is graphically shown in Figure 1. For the sake of simplicity, we assume the execution of two CIGs and binary interactions (i.e., between pairs of actions), which is the case explicitly considered in [10]. Our approach can be easily generalized to multiple CIGs as long as we only consider at each time the interaction of two of them, while [10] (and then the approach in this paper) could be generalized to non-binary interactions.

We perform conformance analysis for the times when some change occurred, either in the state of actions (e.g., an action becomes candidate, or a candidate action becomes active) or in the state of the patient (e.g., a new value for a finding is detected). At each such time (indicated as Reference Time - RT - henceforth), we consider as input of our analysis:
Fig. 1. The possible interaction between two actions relevant for a reference time RT may be used to explain the rest of the log.

1. The set of all actions that are scheduled, candidate or active, each one with its “execution window” (i.e., the range of time within which the action must be started and/or completed, given the constraints in the respective CIG);
2. The set of all the possible effects of such actions, each one with its “existence window” (i.e., the range of time in which the effect may start and end, given knowledge in our ontology);
3. The set of past actions with effects whose “existence window” includes RT;
4. The status of the patient at RT;
5. The trace of execution.

The sets (1) and (3) are the relevant actions for RT. Considering (1), (2), and (3), and ontological knowledge modeling possible interactions [1] we detect whether interactions are possible between an action in (1) from one CIG and an action in (1) or (3) from the other CIG; i.e., at least one action should not have started or not be completed, so that some modification can be applied in order to manage interactions. In practice, for a given pair of actions, it is enough to perform the detection at the earliest RT for which they are both relevant. The interaction detection follows the general methodology devised in [1] and takes into account the direct and indirect effects of the actions and whether some of such effects interact and may overlap in time. We consider two cases:

- No interaction is possible; in such a case, no deviation from the CIGs can be justified (by the analysis of interactions). We check (slightly extending the methodology we proposed in [15] to cope with a single CIG) whether the (proper part of the) execution trace is conformant with the CIGs, and report, as non-justified, any non-conformance.
- One interaction is possible (the case of multiple interactions is significantly more complex – as the analysis of multiple faults, and we plan to address it as
future work). In such a case, deviations from the CIGs might be explained as a way to manage the interaction (e.g., to avoid it). In our previous work, we have identified different modalities used by physicians to manage interactions [10]. This issue is described in more detail in the next section.

4 Explaining non-conformance

Physicians adopt different methodologies to manage interactions (e.g., avoid undesired interactions) between (the effects of) CIG actions. After a careful analysis of the medical literature, in [10] we have identified a set of “modalities” to achieve such a goal. Notably, such options are not mutually exclusive: indeed, in several practical cases, many options are possible, and the physicians have to choose between them. In the following, we describe how we check whether one of such modalities may be used to explain a non-conformance to the original CIGs aimed at managing a possible interaction. The first modalities aim at avoiding an interaction.

Replanning. One of the interacting actions is substituted by a new plan (set of actions), achieving the same goal (or a similar one, according to [10]) but avoiding the interaction. Such an option explains cases of non-conformance in which an action in a CIG is not executed (while it should have been, given the conditions and constraints in the CIG), and one or more actions, not present in the CIGs, are executed (are present in the trace). In order to justify such a non-conformance with the application of the replanning modality, our approach first detects whether an interaction would have occurred (following the CIG), and then uses ontological knowledge to check whether the actions in the trace but not in the CIGs reach the same goals (intentions) of the “substituted” action.

Temporal avoidance. Interactions can be temporally avoided. To do so, interacting actions can be executed at times such that the interaction cannot actually occur (i.e., their effects do not overlap in time). Such a modality can be used in order to explain cases of non-conformance due to the violation of some of the temporal constraints in one or more CIGs. To do it, our approach checks whether, in case the CIG constraints had been respected, an interaction would have possibly occurred, while such interaction was not possible given the execution times in the trace.

Medical knowledge indicates that not all undesired interactions strictly need to be avoided. In some cases, CIGs can be adjusted to manage the situations in which the interactions arise. We support three main management options to this purpose: dosage adjustment (for drug interactions), effect monitoring, and interaction mitigation.

Dosage adjustment. Interactions can be mitigated through a variation of dosage with respect to the ones recommended in the CIGs. Such a pattern can be easily identified (by comparing the dosage in the trace with the one in the CIG) and explained (by identifying the potential interaction, and using the ontology to check whether the sign of the dosage variation is the proper one for mitigating the interaction).
Effect monitoring. In some cases, monitoring the effects of the interaction is enough. In particular, if an interaction causes a change of some parameters of the patient, they have to be monitored and evaluated by the physicians during the span of time in which the interaction occurs. Obviously, if a serious risk is detected, other management options can be finally applied. The effect monitoring modality explains traces in which (i) interacting actions present in the CIGs are indeed executed, but (ii) they are followed by a monitoring action (not present in the original CIGs) and a decision action (not present in the original CIG) to evaluate the state of the patient and decide whether to continue the current therapy or not. In the latter case, the trace must contain another management or the CIG must be suspended.

Interaction mitigation. Some interactions cause undesired but tolerable side effects. In such cases, a new action (or set of actions) that mitigates such effects can be added to the interacting CIGs. Of course, the new action is a deviation with respect to the two CIGs. To explain such a deviation as an application of the interaction mitigation modality, the additional action must mitigate the effects of an occurring interaction.

Not all interactions between CIGs are negative or undesired. This is the case when two actions in the two CIGs pursue the same or similar goals. In such a case, intention alignment can be applied by physicians.

Intention alignment. In the case of intention alignment, the physician may want to “merge” two actions of two different CIGs in a unique one, executing it in a time that respects the temporal constraints of both CIGs, or to substitute them with a new action, which pursues the same (or similar) goals of the two actions. This modality can be used to explain the occurrence in the trace of an action which is not present in any of the two CIGs, instead of two CIG actions. The ontological knowledge is used to check whether the new action can achieve both the goals of the actions it substitutes.

Besides the above modalities, other modalities have been identified in [10]. Such modalities are practically very useful, but detecting them is less useful in an a-posteriori analysis.

Safe alternative and interaction alignment. Such modalities consist in the avoidance (safe alternative) or enforcement (interaction alignment) of an interaction through the choice of alternative paths (when alternative therapeutic actions or paths of actions are specified in the CIG) in the CIGs. Given the trace, we can just recognize the paths chosen by the physicians. In principle, it could be hypothesized that, at the time of some decision, other paths have been disregarded to avoid or enforce interactions, but this is not useful to explain any deviation from the CIGs (since, indeed, paths in the CIGs have been carried on). Notably, in our current approach, we do not even try to detect whether some interaction (which has motivated some deviation from the CIGs) could have been avoided through the application of the safe alternative option (i.e., by selecting, a-priori, different paths from the CIGs). Such an analysis would be quite complex, and scarcely useful in an a-posteriori conformance analysis, because, in the
clinical practice, it is not realistic to expect that physicians consider all the pos-
sible future consequences of their therapeutic choices, exploring in the CIGs all
the paths stemming from each decision, and analysing all possible interactions
between them; notably, such an analysis, though complex, may be very useful in
the context of decision support.

Due to space constraints, we only briefly describe how our overall approach
is represented in ASP. A choice rule:

\[
\begin{align*}
1 \{ & \text{management(Cg1,A,Cg2,B, no\_management)}; \\
 & \text{management(Cg1,A,Cg2,B, replanning)}; \\
 & \text{management(Cg1,A,Cg2,B, temporal\_avoidance);} \ldots \} 1 :- \\
 & \text{possiblyInteractScheduled(Cg1,A,Cg2,B,\ldots,S)}, \\
 & \text{relevant(Cg1,A,S), relevant(Cg2,B,S),} \\
 & \{\text{ended(A,S1) : S1<=S; ended(B,S2) : S2<=S}\} 1.
\end{align*}
\]

where all management modalities are considered in the conclusion, allows the
ASP solver to consider a candidate answer set for each such modality. In general,
the rule applies at a reference time S if: the actions are scheduled, they are not
both completed, and they possibly interact, given the state of the execution at
S and the temporal constraints in the CIGs. This is verified with the predicate
possiblyInteractScheduled, which is defined based on the knowledge about ef-
ects and actions exported by the knowledge server in Figure 1, and temporal
reasoning implemented in ASP. From the knowledge server we export the fact
that the fifth and sixth arguments of possiblyInteractScheduled are effects of
actions A and B, and they may potentially interact; in possiblyInteractScheduled
we check that may actually overlap in time, considering temporal indeterminacy
(at S) of the execution time of actions, which have not necessarily started, and
of effects with respect to the actions.

For each of the modalities above, there are rules to define (i) necessary con-
ditions for their applicability in a specific log, in order to prune the candidate
explanations not supported by the log and (ii) how the CIG execution can be
modified according to such modality. Among the remaining answer sets, as in
[15], optimization statements are used to select the answer set(s) with a mini-
mum number of discrepancies with respect to the log. In the following section,
we provide more details of our approach on specific examples.

5 A concrete example

We consider the concurrent execution of a CIG for peptic ulcer (PU) and a
CIG for venous thromboembolism (VT). Figure 2 shows the two simplified CIGs
at a high level of detail, using the GLARE representation. In the CIGs, the
action “Amoxicillin therapy” (AT), belonging to PU, interacts with the action
“Warfarin therapy” (WT, belonging to VT), which has the intention of avoiding
the development of clots. Such an interaction is usually avoided in the medical
practice, since it increases the anticoagulant effect of warfarin, raising the risk of
bleedings. In Figure 2 some temporal constraints are reported on delays between
actions, and on action duration.

18
Fig. 2. CIGs for peptic ulcer (PU) and venous thromboembolism (VT). Circles are atomic actions, hexagonal nodes are composite actions, diamond nodes are decisions.

Fig. 3. Graphic representation of the three execution logs considered. Repeated arcs for the actions VTst and IntD are drawn only for the first execution of VT.

We applied our approach to three different logs for the two CIGs above. First, we describe a log in which no management has been applied, then we consider a log in which warfarin has been replaced with heparin, and finally we consider a situation in which the beginning of the warfarin therapy has been postponed until the end of the amoxicillin one. Figure 3 represents the three execution logs: the first row shows the log of PU (common to the three executions), while the rows 3-5 represents the different executions of VT. The second row represents the timeline, and an arrow from an action in a log to the timeline indicates that such an action has been executed (or started/ended) in that particular timepoint (e.g., action PUst has been executed at day 0). For durative actions, we indicate with Act_s and Act_e the starting and ending points of the action Act.

In all the examples, in the execution of the VT CIG, the anticoagulant therapy is selected (IntD); at the time of IntD, WT becomes scheduled, and its interaction is detected with AT, which is being executed (i.e., it is active).

Henceforth, we focus only on three options: no_management, replanning and temporal_avoidance. The most relevant rules for such options are described below. The following rule recognizes scenarios in which an interaction was possible and no management for it has been carried on.

\[
\text{info(possibly\_interacting\_actions\_executed,A,B):} = \begin{cases} 
\text{possiblyInteract(Cg1,A,Cg2,B,...),started(A,\_),} \\
\text{started(B,\_),\{management(Cg1,A,Cg2,B,M): M>no\_management\}} 
\end{cases}
\]
The first three conditions in the premise require that two actions A and B, which possibly interact considering temporal constraints, have started. The predicate \( \text{possiblyInteract} \) is analogous to \( \text{possiblyInteractScheduled} \), except that it uses the actual execution time of the actions.

The following set of rules are relative to the replanning modality.

1: \{\text{substitute(Cg,A,C):hasEffect(C,E,_,_,_),causes(E,I),C<>A}\} :- \text{management(Cg,A,\_\_, replanning), aimsTo(Cg,A,I,I_s,I_e)}.
2: :- \text{substitute(Cg,A,C), started(A,\_)}.
3: :- \text{substitute(Cg,A,C),hasEffect(C,E,E_{si},E_{sd},E_{ed},E_{ei}), started(C,T_{s}),aimsTo(Cg,A,I,I_s,I_e),causes(E,I), ended(C,T_{e}),1}\{T_{s}+E_{sd}>I_{s}; T_{e}+E_{ed}<I_{e}\}.
4: \text{block(A,S) :- substitute(\_\_,A), relevantS(\_\_,A,S)}.
5: succ(Cg,C,Anext) :- \text{substitute(Cg,A,C), succ(Cg,A,Anext)}.
6: succ(Cg,Apred,C) :- \text{substitute(Cg,A,C), succ(Cg,Apred,A)}.

The predicates \( \text{hasEffect(Act,E,E_{si},E_{sd},E_{ed},E_{ei})} \), \( \text{causes(A,B)} \), \( \text{aimsTo(Cg,A,I,I_{s},I_{e})} \) are exported from the knowledge server and model, respectively, the facts that an action \( \text{Act} \) has a particular effect \( \text{E} \), starting between \( E_{si} \) and \( E_{sd} \) time units after \( E \), and ending between \( E_{ed} \) and \( E_{ei} \) time units after \( E \); that the effect/intention \( \text{A} \) causes \( \text{B} \); that the action \( \text{A} \) in the CIG \( \text{Cg} \) has the intention \( \text{I} \) that must occur between \( I_{s} \) and \( I_{e} \).

Basically, the first rule creates a candidate answer set for each possible action \( \text{C} \) having an effect achieving the intention of one of the interacting actions. Then, candidate answer sets considering the replacement of an action which execution is reported in the log are discarded (rule 2). On the other hand, rule 3 discards candidates in which the replacing action \( \text{C} \) is executed in a time not compatible with the temporal constraints (if any) of the intention \( \text{I} \) of the original action \( \text{A} \) in the CIG. Finally, rules 4-6 replace the original action \( \text{A} \) with \( \text{C} \) in the CIG.

The following rule is relative to temporal avoidance.

\[- \text{management(Cg1,A,Cg2,B, temporal_avoidance),1}\{\text{not started(A,\_)}; \text{not started(B,\_);possiblyInteract(Cg1,A,Cg2,B,\_\_,\_\_)}.\]

The rule discards the temporal avoidance option if, in the execution log, one of the two actions has not been executed or they have been executed in times in which the interaction is still temporally possible. Further rules model the fact that, when justifying temporal avoidance, deviations from the temporal constraints in the CIGs are allowed.

Example 1. Consider the log of VT in the third row of Figure 3, in which (i) WT starts during AT and (ii) no deviations with respect the original CIGs are present. Given (ii), replanning is ruled out, while (i) discards temporal avoidance. The only remaining alternative is \text{no_management}, which is reported as output together with the warning that a possible interaction could have occurred (i.e., \text{info(possibly_interacting_actions_executed,wt,at)}).

Example 2. The log in the fourth row of Figure 3 shows an example in which “Heparin therapy” (HT), which is not present in the original CIGs, has been executed, while WT is not present. This last fact discards the option of temporal avoidance, while the \text{no_management} option does not match the log and is then...
discarded by minimization, since the log can be explained by the replanning option. In fact, in our ontological model [10,1]. HT has the effect of decreasing the blood coagulation, which avoids the development of clots (i.e., the intention of WT). The resulting answer set contains the facts management(vt,wt,pu,at, replanning), substitute(vt,wt,ht).

Example 3. In the log in the last row of Figure 3, the beginning of WT is delayed after the end of AT. In our knowledge model, the two actions interact due to the interaction between the “Anticoagulation” effect of WT and the “Platelet aggregation Inhibition” effect of AT, where the latter ends with the ending of AT. Thus, administering WT after AT\_e is “safe” and the predicate possiblyInteract does not hold considering this log scenario, so that temporal avoidance is supported (and the resulting answer set contains the predicate management(vt,wt,pu,at, temporal_avoidance)). On the other hand, replanning is not supported because WT and AT are present in the log, while no management does not explain the fact that the log violates the temporal constraint [0,3] days between actions IntD and WT.

6 Related work and conclusions

In this paper, we propose the first approach that addresses the problem of analysing the conformance of execution traces with multiple CIGs, as needed in the treatment of comorbid patients. The importance of this task stems from the fact that full conformance to individual CIGs may be dangerous for comorbid patients: deviations are sometimes necessary to avoid undesired interactions between the CIGs. We thus identify cases in which traces are conformant, but undesirable interactions have not been avoided, and complement our conformance analysis with an explanation approach, aimed at justifying deviations in case they had avoided some possible undesired interaction. Additionally, two other main features of our approach, distinguishing it from the others in the literature, are:

(i) our attention to the temporal dimension and
(ii) our adoption of ASP to model and reason with the problem.

Until now, the two main tasks that we homogeneously dealt with in our approach have been only pursued in isolation by the approaches in the literature: (1) from one side, there are approaches to the conformance to CIGs, but considering just one CIG; (2) from the other side, there are approaches to manage multiple CIGs and their interactions, to cope with comorbid patients, but none of such approaches face the conformance problem. In the following, we separately consider the approaches in (1) and in (2), focusing (for the sake of space constraints) only on the ones more closely related to our one.

In the last years, several AI approaches have been developed for supporting the treatment of comorbid patients (see the survey [4]). For the detection of interactions between CIGs, the most closely related approach is the one in [19]. It provides a CIG-independent conceptual model for medical actions and
reasoning forms operating on it. Moreover, in such a work general rules are pro-
posed in order to identify different types of interactions on the basis of such a
knowledge. Several approaches have been devoted to the generation of integrated
CIGs. Some of them are not “conservative”: adopting different techniques, they
use the input CIGs as a starting point to build a mostly new CIG which has no
undesired interactions (e.g.,[14], using an agent-based approach and hierarchi-
ical planning). Others, including the one we devised within GLARE [11], adopt
more conservative techniques: since CIGs are evidence-based, physicians rely on
them, so that interactions are managed with the minimum possible deviations
from the original CIGs. Within such approaches, the one in [18], uses constraint
logic programming to identify and address adverse interactions, while [13] pro-
poses a model-based approach for the combination of CIGs, [6] a semantic-web

A limited number of approaches have dealt with verifying conformance of
a trace of actions with recommendations in a CIG. In [5], differences between
actual actions and CG prescriptions are detected and analyzed, e.g. by com-
paring, for a non-compliant actions, actual findings with findings that support
the action according to the CIG. [2] focuses on the interaction between clinical
guidelines (CGs) and the basic medical knowledge (BMK) in the light of the
conformance problem. In the last years, we also focused on the interaction be-
tween BMK and a CIG [15], using ASP, and aiming at providing a justification
for non-conformances. While in this paper we exploit the model of CIG action
execution developed in [15], the rest of the approach is different. First of all, in
[15] only one CIG is considered, and non-conformance can be explained on the
basis of a “general” basic medical knowledge, which may trigger new actions in
case problems not considered in the original CIG arise in the patient. In this
paper two or more CIGs are considered, and more specific rules (the modalities)
are used to explain non-conformances (in a context in which interactions are
possible). Additionally, a central point of the current approach is the analysis of
interactions between CIGs, while interactions (not even between the CIG and
the actions suggested by the basic medical knowledge) were not taken into ac-
count in [15]. As a consequence, the overall process of detecting and analysing
non-conformances, as outlined in Figure 1, is completely different from the con-
formance analysis carried on in [15]. The two approaches can be integrated, to
consider also the case of comorbid patients deviations explained by the arising
of new problems related to the patient status.

References

1. Anselma, L., Piovesan, L., Terenziani, P.: Temporal detection and analysis of guide-
2. Bottrighi, A., Chesani, F., Mello, P., Montali, M., Montani, S., Terenziani, P.: Con-
formance Checking of Executed Clinical Guidelines in Presence of Basic Medical
Representing and reasoning with probabilistic temporal distances for guideline interaction analysis

Paolo Terenziani¹, Antonella Andolina²

¹DISIT, Institute of Computer Science, Università del Piemonte Orientale, Alessandria, Italy
paolo.terenziani@uniupo.it
²ITCS Sommeiller, Corso Duca degli Abruzzi, 10100, Torino, Italy
antoando@libero.it

Abstract. The treatment of patients affected by multiple diseases (comorbid patients) is one of the main challenges of the modern healthcare, involving the analysis of the interactions of the guidelines for the specific diseases. However, practically speaking, such interactions occur in time. The GLARE project explicitly provides temporal representation and temporal reasoning methodologies to cope with such a fundamental issue. In this paper, we propose a further improvement, to take into account that, often, the effects of actions have a probabilistic distribution in time, and considering such probabilities further enhance the support for interaction detection.

Keywords. Computer-interpretable clinical guidelines, Comorbidities, Analysis of temporal interactions, probabilistic temporal reasoning.

1 Introduction

Clinical practice guidelines are the major tool that has been introduced to grant both the quality and the standardization of healthcare services, on the basis of evidence-based recommendations. The adoption of computerized approaches to acquire, represent, execute and reason with Computer–Interpretable Guidelines (CIGs henceforth) provides crucial additional advantages so that, in the last twenty years, many different approaches and projects have been developed to manage CIGs (consider, e.g., the book [1] and the recent survey [2]). One of such approaches is GLARE (Guideline Acquisition, Representation and Execution) [3], and its successor METAGLARE [4].

By definition, clinical guidelines address specific clinical circumstances (i.e., specific pathologies). However, specific patients may be affected by more than one pathology (comorbid patients). The treatment of such patients is one of the main challenges for the modern health care, also due to the aging of population, and the increase of chronic pathologies. The problem is that, in comorbid patients, the treatments of single pathologies may interact with each other, and the approach of proposing an ad-hoc “combined” treatment to cope with each possible comorbidity does not scale up. In the last years, several computer-based approaches have started to face this problem. We have started to extend GLARE to cope with comorbid patients. In this paper we focus on interaction detection. In our previous work, we have developed an
ontology for interactions, and we have complemented it with detection algorithms [5]. Interactions between CIGs occurs in time. For instance, the effects of two actions taken from different guidelines may potentially conflict, but practical conflicts happen only if the times of execution of such actions are such that their effects overlap in time. As a consequence, in our recent work, we have focused on an explicit treatment of temporal constraints and of temporal reasoning [6]. However, such an approach still presents a limitation: it does not consider the fact that, indeed, temporal constraints may have different probabilities\(^1\), and such probabilities may be important for physicians in order to correctly analyse and manage interactions.

**Example 1.** Consider, for instance, a patient affected by gastroesophageal reflux (GR) and by urinary tract infection (UTI).

The CIG for GR may recommend calcium carbonate administration (CCA), to be administered within three hours. CCA has the effect of decreasing gastric absorption (DGA). Considering as granularity units of 15 minutes, DGA can start after 1 unit with probability 0.4, after 2 with probability 0.4, and after 3, with probability 0.2. Additionally, the duration of DGA may be 4 units (probability 0.1), 5 (0.3), 6 (0.4), 7 (0.1), or 8 (0.1). The CIG for UTI may recommend Nalidixic acid administration (NAA), to be administered within two hours. NAA has as effect Nalidixic acid gastric absorption (NAGA), starting after 1 unit (probability 0.4) or 2 (probability (0.6). The duration of NAGA may be 1 (probability 0.05), 2 (0.05), 3 (0.15), 4 (0.15), 5 (0.25), 6 (0.25), 7 (0.05), 8 (0.05).

In order to support physicians in the study the interaction between CCA and NAA, one must take into account not only the temporal constraints, but also their probabilities. This is essential in order to answer physician’s queries such as:

(Q1) If I perform on the patient CCA in unit 1 or 2 (i.e., in the following 30 minutes), and NAA in units 1 or 2 (i.e., in the following 30 minutes), what is the probability that the effects of such two actions intersect in time (i.e., what is the probability of the interaction between CCA and NAA)?

In the following, we sketch our ongoing approach to support physicians in the management of probabilistic temporal interaction detection. This is the first approach in the literature managing such a challenging task (see Section 4).

\(^1\) Several aspects influence the absorption of a drug, and therefore its effects. In particular, they are influenced by the methods of administration (e.g., enteral, parenteral, transcutaneous...) of the drug, by its mechanisms of absorption and elimination, and by the targets of the administered substance. The fields in medicine that study such mechanisms are the pharmacokinetics and pharmacodynamics. Defining accurately the probabilities of the effects of a drug along the time could be difficult. As a first approximation, to do it, we considered the models of the plasma concentrations of the drugs, their half-life (i.e., the time to reduce the substance amount in the blood of 50%) and the type of the considered effect, and we approximate the probabilities with the help of an expert. However, more specific information and models could be considered, and we plan to cope with them as a future work.
2 Probabilistic temporal reasoning

To face the above problems, we extend temporal constraints to support the possibility to associate probabilities to temporal constraints, and to reason and query them.

Temporal Representation. As many approaches to temporal representation and reasoning (see, e.g., [7]), we base our approach on the notion of distance between time points. Indeed, we base our approach on STP (Simple Temporal Problem) [8], in which constraints are of the form \( P_1 [l,u] P_2 \), representing the fact that the distance between the time points \( P_2 \) and \( P_1 \) is between \( l \) (minimum distance) and \( u \) (maximum distance). We extend such a representation by associating, for each distance \( d \) (\( l \leq d \leq u \)), its probability. For example, the temporal constraint between CAA and the starting point DGAStart of its effect can be represented, in our approach, by the probabilistic temporal constraint:

\[
CCA < (1,0.4),(2,0.4),(3,0.2)> DGAStart
\]

In general, in our approach, the meaning of a constraint \( t_i <(d_1,p_1),\ldots,(d_n,p_n)> t_j \) is that the distance \( t_j - t_i \) between \( t_j \) and \( t_i \) is \( d_i \) with probability \( p_i \), or \( \ldots \) or \( d_n \) with probability \( p_n \).

Temporal Reasoning. Typically, in Artificial Intelligence [7], temporal reasoning is performed through the application of transitive closure: given the constraint \( C(i,k) \) between two time points \( i \) and \( k \), and the constraint \( C(k,j) \) between \( k \) and \( j \), the constraint \( C(i,j) \) between \( i \) and \( j \) is inferred by first composing (denoted by “@”) \( C(i,k) \) and \( C(k,j) \), and then intersecting (“\( \cap \)”) the result with the previous constraint (if any) between \( i \) and \( j \), i.e.,

\[
C(i,j) \leftarrow C(i,j) \cap (C(i,k)@ C(k,j))
\]

In the case of STP constraints [8], \( [l_1,u_1] \cap [l_2,u_2] = [\max(l_1,l_2), \min(u_1,u_2)] \), and \( [l_1,u_1]@[l_2,u_2] = [l_1+l_2,u_1+u_2] \). For STP constraints, temporal reasoning is usually performed through an application of Floyd-Warshall’s algorithm, which performs the transitive closure for each triplets \( i,k,j \) of time points [7]. The result of the application of such an algorithm is the minimal network [8], i.e., the tightest equivalent STP, i.e., an STP where the minimum and maximum implied distances between each pair of points are made explicit. Floyd-Warshall’s algorithm is correct and complete on STP, i.e., it performs all and only the correct inferences while propagating the STP constraints [8]. Its temporal computational cost is cubic in the number of time points [8].

Our representation model is basically an extension of STP in order to include probabilities. We can thus perform temporal reasoning as in STP, using Floyd-Warshall’s algorithm. However, we have to adapt it to apply to our probabilistic constraints. In order to achieve such a goal, we need to provide suitable definitions of the intersection (\( \cap \)) and of the composition (\( @ \)) operators, to propagate both distances and probabilities.

Notation. We adopt the following notation: given two constraints \( c_1 \) and \( c_2 \), we indicate with \( p_1d \) and \( p_2d \) the probability of \( d \) in the first and in the second constraint.
respectively.

In our approach, the composition @ between two constraints
\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})>\) and \(<(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})>\) is obtained as follows. As in
STP [8], the output distances are evaluated as the pairwise sum of the input distances.
For each output distance \(d'\), its probability \(p_{d'}\) is obtained by summing up the product
\(p_{1d_1u} \times p_{2d_2v}\) of the probabilities of each pair of distances \(d_{1u}\) (from the first input con-
straint) \(d_{2v}\) (from the second constraint) such that \(d_{1u}+d_{2v}=d'\). More formally:

**Definition. Composition (@).** Given two probabilistic constraints
\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})>\) and \(<(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})>\), their composition is defined
as follows:

- let \(\{d'_1,\ldots,d'_r\} = \{d'\in\mathbb{R}^+ \mid d' = d_{1u} + d_{2v} \text{ and } d_{1u} \in \{d_{11},\ldots,d_{1n}\} \text{ and } d_{2v} \in \{d_{21},\ldots,d_{2m}\}\}\),
  and let \(p_{d'} = \sum_{d_{1u}+d_{2v}=d'} (p_{1d_1u} \times p_{2d_2v})\), then

\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})> \odot <(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})> = <(d'_1, p_{d'_1}),\ldots,(d'_r, p_{d'_r})>\)

**Example (composition).** For example, the composition of the constraint between
NAA and the start of NAGA (represented by NAGA S) with the one between the start
and the end (NAGA E) of NAGA gives as result the constraint between NAA and
NAGA E:

\(<(1,0.4),(2,0.6)> \odot <(1,0.05), (2,0.05), (3,0.15),
(4,0.15), (5,0.25), (6,0.25), (7,0.05), (8,0.05)> \odot \Rightarrow<br/>
NAA <(2,0.02), (3,0.05),
(4,0.09), (5,0.15), (6,0.19), (7,0.25), (8,0.17), (9,0.05),(10,0.03)> \odot \Rightarrow
NAGA E

In our approach, the intersection between two constraints
\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})>\) and
\(<(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})>\) is obtained by computing the intersection between the
two input sets of distances (as in STP [8]). For each intersecting distance, its prob-
ability is evaluated as the product of the probabilities of such a distance in the first and in
the second constraint (since it is an “AND combination” of the two cases). Formally:

**Definition. Intersection (∩).** Given two constraints
\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})>\) and
\(<(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})>\) their intersection is defined as follows:

- let \(\{d'_1,\ldots,d'_k\} = \{d_i \in \{d_{11},\ldots,d_{1n}\} \cap \{d_{21},\ldots,d_{2m}\}\}\),
  then

\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})> \cap <(d_{21},p_{21}),\ldots,(d_{2m},p_{2m})> = \text{Normal}(\langle d_i', p_{d'_i} \rangle, \ldots, (d_k', p_{d'_k})),\rangle\) where Normal(\(<(d_{11},p_{11}),\ldots,(d_{1n},p_{1n})>\) normalizes \(p_{11},\ldots,p_{1n}\)
in such a way that their sum is 1.

**Example (Intersection).** For example, \(<(1,0.4),(2,0.6),(3,0.15),(4,0.09),\ldots,(10,0.03)> \odot \Rightarrow<br/>
NAGA E \cap \Rightarrow \langle(8,0.132),(9,0.775),(10,0.093)> \odot \Rightarrow
NAGA E
3 Query Answering Facilities

Temporal reasoning can be used in order to evaluate the minimal network of a set of probabilistic temporal constraints, i.e., the strictest probabilistic temporal constraints between each pair of points. For example, the inferred constraint between DGAS and NAGAS is: DGAS <(-13, 0.000003), (-12, 0.00006), (-10, 0.00218), (-9, 0.0067), (-8, 0.01618), (-7, 0.0328), (-6, 0.05775), (-5, 0.08877), (-4, 0.11784), (-3, 0.1353), (-2, 0.13862), (-1, 0.12964), (0, 0.10739), (1, 0.07651), (2, 0.04689), (3, 0.025239), (4, 0.01167), (5, 0.00444), (6, 0.00127), (7, 0.000234), (8, 0.000002)) > NAGAS.

However, to cope with interaction detection, physicians must also be provided with facilities in order to query the resulting minimal network. In particular, it is fundamental to provide physicians with tools to ask queries like Q1 in Example 1. To manage queries like Q1, we define the predicate INTERSECTS(e1, e2) between (durative) events e1 and e2, and define a way to evaluate its probability.

\[
\text{Prob(INTERSECTS(e1, e2))} = 1 - \text{Prob(NON-INTERSECTS(e1, e2))}
\]

\[
\text{Prob(NON-INTERSECTS(e1, e2))} = \text{Prob(End(e1) > Start(e2))} + \text{Prob(End(e2) > Start(e1))}
\]

Given two any time points t1 and t2, and given the temporal constraint t1 <(d1, p1), …,(dk, pk)> t2, we define

\[
\text{Prob}(t2 > t1) = \sum_{d_i > 0} p_i
\]

The query Q1 can be easily expressed as the hypothetical query in the following (where \(X_0\) is used to denote the current time)

\[
\text{Prob(INTERSECTS(DGA, NAGA))}
\]

IF \{X0 <(1.0, 0.5), (2.0, 0.5)> CAA, X0 <(1.0, 0.5), (2.0, 0.5)> NAA\}

Queries like Q1 are managed by our approach in three steps:

1. First, the “hypothesized” temporal constraints (“IF” part of the query) are added to the minimal network (through the operation of intersection with previous temporal constraints).
2. Temporal reasoning is performed (through Floyd-Warshall algorithm), producing a new minimal network.
3. The probability of INTERSECTS is evaluated in the new minimal network, on the basis of the above definition.

In our example, the result of query Q1 is 0.9486. Notably, after Q1, physicians might ask a query like Q2 (to check the probability of interaction in case NAA is executed in the first 30 minutes, and CAA between two and three hours from the current time):

\[
\text{(Q2) Prob(INTERSECTS(DGA, NAGA))}
\]

IF \{X0 <(9.0, 0.25), (10.0, 0.25), (11.0, 0.25), (12.0, 0.25)> CAA, X0 <(1.0, 0.5), (2.0, 0.5)> NAA\}

The probability is 0.02455, suggesting to the physician that the probability of interaction sharply decreases if she delays the execution of CAA. Notably, using a “standard” temporal reasoner (i.e., not considering probabilities), the physician could only infer that an interaction may occur, both in case CAA is executed within the first 30 minutes, and in case it is executed after two or three hours.
Conclusions and Related Work

When managing comorbid patients, interactions between CIG actions must be studied. Indeed, since such interaction occur (or do not occur) in time, temporal representation and temporal reasoning is required. However, current approaches do not take into account the fact that the different temporal distances may have different probabilities. In this paper, we start to define an approach to overcome such a limitation. Our approach is fully original in the panorama of Artificial Intelligence [7]. Starting from 2000’s, several researchers have pointed out that, to cope with the complexity of several application domains, temporal constraints must be augmented with preferences (see, e.g., [9]) or probabilities (see, e.g., [10]). However no other approach has paired quantitative STP constraints (i.e., distances between time points) with probabilities. In this paper, we provide a first step towards such a goal, which is of paramount importance to cope with problems such as the one presented in Section 1. The probabilistic temporal reasoning module described in this paper has been already implemented and successfully tested. On the other hand, its integration within GLARE is still future work.

References

Utilizing Temporal Genomic Data for Healthcare

Guenter Tusch and Shahrzad Eslamian

1 Grand Valley State University, Allendale, MI, USA
tuschg@gvsu.edu, eslamias@mail.gvsu.edu

Abstract. Genomic data become more frequently part of clinical practice. Novel tools and methods are required to transform information from increasingly voluminous genomic databases into actionable data for health care. In the era of precision medicine, the development of high-throughput technologies and electronic health records resulted in a paradigm shift in healthcare. However, the treatment of temporal data still remains a challenge. Recent efforts propose temporal models for the electronic health record, but not for genomic data.

One frequently employed model for temporal data in healthcare is temporal abstraction, a model based on conversion of expression values into an interval-based qualitative representation expressing the amount of change over time. The challenge is to find a domain specific mapping to create those representations. This study explores the feasibility of a hybrid AI-statistical model where the amount of change is determined by statistical significance the most reliable measure to determine biological significance. We propose to use empirical Bayes methods to determine differences in consecutive time points. Comparisons across platforms are done by comparing p-values. For DGE count data we use the voom transformation allowing RNA-seq data to be analyzed in a similar way. We demonstrate this approach in the framework of our SPOT software.

Keywords: Temporal representation and reasoning, statistics, decision support, microarray, RNA-seq.

1 Introduction

The increasing number of genomically targeted therapies and immunotherapy in the last decade has significantly changed the practice of oncology. At the same time, the electronic health record (EHR) has become a standard in the oncology clinic. Further integration of cancer genomic data into EHR systems will improve clinician decision making and eventually patient care. Essential is here an understanding of biological pathways, networks and molecular systems. It will require information from the genetic and the gene-product level in order to ultimately assign all genes and their products to functional pathways. Therefore, researchers are interpreting biological phenomena in terms of groups of genes, or gene sets, instead of looking at individual genes. (A gene set is any set of genes that share some common feature, for instance a molecular function or a correlated gene expression). This clinical view has led to the discovery of completely new therapies. On the other hand, gene expression profiling
tests have been recommended for clinical practice in addition to conventional clinical markers in order to identify those patients who might not benefit from adjuvant treatment. This approach can reduce toxicity and cost to the health care system.

Gene expression databases provide a wealth of transcriptomic data on the Internet. Two of the most popular ones are GEO (Gene Expression Omnibus) at the NCBI that has more than 32,000 sets of gene-expression data collected from over 2 million samples (as of April 2017) of over 1,700 platforms and ArrayExpress at the EBI that consists of over 2.2 million assays and some 70,000 experiments (or 45.3 TB of archived data). When translating information captured in these increasingly voluminous biomedical and genomic databases into actionable data for health care or prevention, treatment of temporal data still remains a challenge. The challenge is not so much in modeling complex pattern of intervals like in clinical domains (see [1]), but more in classifying different types of intervals in terms of trends, as “increasing”, “decreasing”, “constant” etc. (see [2]). Biologists typically rely on statistical measures for those purposes. Frequently time-course experiments allow to study the dynamics of transcriptomic changes in cells exposed to different stimuli. The most common resources to retrieve "curated gene sets" are the Gene Ontology (http://www.geneontology.org) framework defining concepts to describe gene function and their relationships and the Kyoto Encyclopedia of Genes and Genomes (http://www.genome.jp/kegg/) database integrating information on genomic, chemical and systemic function.

This paper explores the feasibility of searching for these particular temporal effects or temporal patterns in gene sets or pathways across data from different studies based on temporal modeling through knowledge-based temporal abstraction (KBTA). KBTA allows for conversion of gene expression values into an interval-based qualitative representation expressing the amount of change over time. It also allows to compare studies where the experimenter chose different pattern of time points. Change in these types of studies is typically determined by statistical significance. Assume a clinical researcher conducted a clinical study where he/she discovered for a specific groups of patients peaks or valleys in a set of genes in the first 24 hours after an intervention. He or she wants to know what other experiments in NCBI GEO potentially even in animal experiments showed the same effect, e.g., to potentially find a different application for an existing drug. Although different studies address similar questions a comparative search through public databases is further exacerbated by the use of heterogeneous platforms and analysis methods in those studies. We describe a hybrid AI-statistical model to help alleviate those problems.

Those data repositories are essential in finding clusters of similar gene expression patterns of gene sets potentially across species.

2 Methodology

Databases that contain temporal gene expression data are organized in a way that data are recorded at the particular time point the measurement took place, i.e., the tissue sample was taken. These time points are not standardized but change from experiment to experiment and are determined by the experimenter as he/she sees best fit for the biological question that will be answered by the experiment.
1.1 Temporal Abstraction

We use Knowledge Based Temporal Abstraction (KBTA) to transform the raw data into a qualitative representation of temporal change based on time intervals, not discrete temporal data. KBTA is the task of summarizing large amounts of time-oriented data using domain-specific knowledge (see Shahar [2]). The KBTA method is based on a formal model of input and output entities, their relations, and the domain-specific properties that are associated with these entities - called the KBTA ontology. Shahar describes four different output types, state, gradient or trend, rate, and pattern abstractions. For the domain of gene expression studies predominantly the gradient type representing temporal trends is important. It might represent increasing, decreasing or constant values with an associated specific time interval for that this trend persists. The temporal intervals then can form patterns using Allen’s approach of temporal relationships [3]. For example, a “peak” can be defined as an interval with increasing trend immediately followed by one with a decreasing trend. Thus peaks can be found in a pattern search even if experiments use different time point patterns. Conversely, a “valley” can be defined.

There is a variety of implementations of the KBTA method in different domains, many clinical. Almost all of them focus on describing individual patient courses for therapeutic purposes, e.g. [1], but there are a few genomic applications, e.g., [4]. The key to this methodology is to use domain-specific knowledge to determine, if values are changing or remaining constant. Several packages and tools are currently available, for instance the virtualArray software package in Bioconductor can combine raw data sets using almost any chip types based on current annotations from NCBI GEO. No such tool is currently available for temporal studies.

The goal of this study is to make temporal information that can be found in publicly available repositories accessible and thus ultimately available in clinical settings. We use as an example for a repository NCBI GEO that contains data from both microarray and RNA-seq gene expression studies. For microarray studies, it contains in most of the cases both raw data sets and curated data. To accommodate for potential bias based in experimental conditions data typically have to be cleaned and normalized before they can be used for analysis. There are different normalization procedures, which are chosen by the authors of the publication to their best knowledge. The curated data sets (Genomic Data Structure - GDS - in NCBI terms) are normalized. Although this includes a subjective element, we chose to use GDS data whenever available, because the best represent the statistical results communicated in the corresponding publications. To implement KBTA in gene expression studies we use mostly the same methodology that a biologist would use to determine trends in high-throughput data, i.e., by means of statistical significance.

There are two main approaches: to treat each time point in the time series independently (see, e.g., [4]) or use the entire time series as one entity (see, e.g., [5,6]). For the first approach extensions of methods developed for static (non time-course) experiments can be utilized. However, they ignore the sequential nature of time-course data and the resulting time dependent correlation structure, which could result in a loss of statistical power. We use here the first approach for its simplicity, but will show in the discussion how our method can be modified and adapted to the second approach.
For our model we assume that each time series consists of \( n \) time points and \( k \) replicates (independent measurements) for each time point \((k>1)\). Furthermore, we assume a point of reference for each series (typically the starting point \( t=0 \) or possibly a series with identical time points and a reference treatment). We determine for each time point the p-value (see below) tested against the reference. We also determine a significance threshold (typically \( .05 \)). Next we identify dominant points, i.e., data points that characterize a trend change following that point. A non significant p-value or an absolute difference of p-values smaller than a given threshold \( \delta \) represents a “constant” trend, a decreasing significant p-value represents an “increasing” trend and an increasing significant p-value represents a “decreasing” trend. (For two-color microarray data, where the data might consist of log ratios, for the negative values decreasing p-values would mean “decreasing”). Because the time series we investigated were small (the median of all time points/study was 4), we chose \( \delta=0 \).

The dominant points separate the time series into distinct intervals, each with a different trend. Therefore, we use de facto a piecewise linear approximation of the time series based on significant p-values. For a database search all time series have to be separated into intervals. Then those can be combined using Allen’s relationships. For instance, a peak is an interval labeled “increasing” that is immediately followed (Allen relationship: meets) by an interval labeled “decreasing” (see below section 3). That can be specified for the entire gene set in a query.

1.2 Domain Specific Knowledge

Two aspects are critical for our implementation: increasing the power of the statistical test by empirical Bayes methods and p-value adjustment. Due to the relative high cost of high-throughput sequencing technology sample sizes in gene expression studies are in general small resulting in little statistical power. To accommodate for that, empirical Bayes methods are employed. For microarray studies one approach uses the Bioconductor limma software to model the normalized expression values in the framework of the linear model in statistics. Using the empirical Bayes methodology, a moderated t-statistic [7] is calculated that reduces the polled variance by borrowing information across all genes of the particular chip. We use the moderated paired t-test to determine, if there are significant differences in consecutive time points. If the difference is significant, we label the interval as increasing or decreasing depending on the direction of change. We don’t adjust for the length of the interval assuming that, if a biological signal is present, it does not depend on the length of the interval. While this approach assumes that the experiment is based on one particular platform, comparison across platforms can be done by comparing p-values assuming that p-values accurately measure biological effects, and those don’t depend on the platform. Therefore, in our model the p-values and the direction of the change inform the temporal abstraction, e.g., no significance means “constant”.

1.3 RNA-seq Studies

It is expected that emerging digital gene expression (DGE) technologies will overtake microarray technologies in the near future for many functional genomics applications. In contrast to microarrays, RNA-seq array require a computing intensive reassembly
step for up to 300M reads with subsequent steps, typically using the Tuxedo protocol [8]. One of the fundamental data analysis tasks, especially for gene expression studies, involves determining whether there is evidence that counts for a transcript or exon are significantly different across experimental conditions [9]. There are at least five different competing approaches for differential expression (DE) analysis, most of them based on assuming a negative binomial distribution of the log-counts. We, however, use the voom transformation [11] that estimates the mean-variance relationship of the log-counts, generates a precision weight for each observation and thus allows RNA-seq data to be analyzed the same way as microarray data. This allows us to have a unified approach and use basically the same parts of the program for microarrays and RNA-seq data. Our chosen edgeR/limma/voom approach with TMM normalization seems to give reliable and trustworthy results (see, e.g., [10]).

1.4 P-value Adjustments

For DE studies many genes have to be tested for differential expression on the same data set, which leads to a depreciation of the nominal p-value. Therefore, the p-value has to be adjusted. Two methods are typically applied, the Bonferroni correction or the false discovery rate (FDR) approach. Both approaches have their drawbacks and are missing some significant genes. While the focus of the FDR is on controlling the false positives while potentially missing many significant genes, the Bonferroni correction is very conservative in assigning significance. Since the FDR is the most common approach, we use that adjustment (also to have a more likely match with published results and biological verification). Significant genes are verified, e.g. by quantitative polymerase chain reaction (qPCR), in almost all publications.

3 Implementation

For performance reasons most of the data are preprocessed and stored in a MySQL database. We plan to migrate to an Apache Hadoop environment using Apache Hive as soon as the numbers of the available time series data increases significantly. As of May 2017 we found in ArrayExpress 350-1000 data sets and in NCBI GEO 930 microarray data sets and 125 RNA-seq datasets. For microarray studies we use the GDS format with annotation files, which are normalized, and then extract the data matrix. For high throughput sequencing RNA-seq studies data are preprocessed as described above and normalized resulting in a data matrix. We use R Bioconductor (BioC - bioconductor.org) with limma and the voom transformation for RNA-seq. The necessary databases are accessed using standard BioC tools like GEOmetadb. This implementation has been integrated into the SPOT web application [12] via HTML, JavaScript and PHP that feeds into Protégé. Complex time patterns can be modelled using Protégé (http://protege.stanford.edu) with the help of Allen’s temporal logic as has been described in an earlier publication [12]. We used as an example a microarray study [14] with differentially expressed genes grouped into three peak clusters: “early”, “delayed” and “late”. We were able to reproduce the results of the authors of the study (see below an implementation of one specific pattern in Protégé).
Example of the temporal concept hasEarlyPeak as implemented in Protégé’s SWRL Tab (https://github.com/protegeproject/swrltab-plugin). Output types “INCREASE” and “DECREASE” are modelled with above described techniques, while the time span of less than one day is realized by intersection of intervals created with Allen’s temporal logic (see [12]).

\[
\text{Tissue(?tissue) } \land \text{hasExperiment(?tissue, ?exp) } \land \text{hasGene(?exp, ?gene)} \land \text{hasGeneName(?gene, ?geneName)} \land \text{swrlb:equal(?outputType, "INCREASE") } \land \text{temporal:hasValidTime(?gene, ?tVT)} \land \text{hasGene(?exp, ?gene2)} \land \text{hasGeneName(?gene2, ?geneName2)} \land \text{swrlb:equal(?geneName2, ?geneName)} \land \text{swrlb:equal(?outputType2, "DECREASE") } \land \text{temporal:hasValidTime(?gene2, ?tVT2)} \land \text{temporal:meets(?tVT, ?tVT2, "days")}\land \text{temporal:hasStartTime(?tVT, ?startTime)} \land \text{temporal:hasFinishTime(?tVT2, ?finishTime)} \land \text{swrlx:lessThanOrEqual(?finishTime, 1)} \text{\swrlx:createOWLThing(?hbVT, ?exp)} \rightarrow \text{\swrlx:ValidPeriod(?hbVT)} \land \text{temporal:hasStartTime(?hbVT, ?startTime)} \land \text{temporal:hasFinishTime(?hbVT, ?finishTime)} \land \text{hasEarlyPeak(?exp, ?hbVT)}
\]

The above described approach works for searches within a species across platforms, if all platforms involved in the search share the same genes. For searches across species we determine the orthologs, i.e., genes in different species that evolved from a common ancestral gene by speciation, from the InParanoid database and translate genes in between platforms of different species.

4 Discussion and Future Aspects

Besides limma, there are other common algorithms to determine significance in time-series gene expression data, as for instance SAM, EDGE or BETR. Typically, they assign a p-value to an entire gene set or entire time series, not an individual time point, often incorporating a sophisticated weighting schema (see [6]). Our approach could be adapted as follows: If both methods show significance, use our method. If our method shows no significance, but the other method does, decrease the p-values in our method accordingly and use it with at least one significance.

There is a challenge with the validation of our approach with biological data. Typically, when a microarray study or RNA-seq study finds significant gene expression changes, those are verified through fluorescent, one-step reverse transcription-polymerase chain reaction (RT-PCR) or quantitative polymerase chain
reaction (qPCR). Since every study potentially uses a different method to identify differentially expressed genes and similarly for clustering the gene sets as for example STEM, GQL (see [6]) or TimeClust [4], the significant genes that our algorithm finds, may not be verified. One solution could be implementing different standard pipelines and giving the user the choice. Given, that we describe an exploratory and not modelling approach as in [13], this might be of minor concern.

While microarray DE studies are pretty much standardized, there is quite a variety of different pipelines for the analysis of RNA-seq data sets. Therefore, the results from our unified approach may differ from the actual confirmed results in the corresponding publications due to use of different analysis pipelines [10].

Therefore, we will carefully compare our results with those published while we continue to evaluate our program on a representative set of sample studies.

References

Representation of Temporal Constraints in Clinical Guidelines using openEHR archetypes and GDL

Lilian Mie Mukai Cintho1, Claudia Moro1, Lucia Sacchi2 and Silvana Quaglini 2

1 Health Technology Graduate Program, Pontifical Catholic University of Paraná, Brazil
2 Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Italy

c.moro@pucpr.br

Abstract. To facilitate the adherence of clinical guidelines (CG), they must be integrated with the Electronic Health Record (EHR). Many CG are composed by Temporal Constraints (TC), and their computational representation is a challenging task to develop clinical decision support systems (CDSS), as is necessary to map clinical data and to rules. The archetype architecture, one of the EHR interoperability standards, enables the same computational format to represent the CDSS rules and EHR data, avoiding this mapping. In this case, rules are specified applying Guideline Definition Language (GDL) that is based on archetypes. This work aims to represent TC in clinical guidelines with archetypes and GDL to incorporate decision support in EHRs. An example of each TC class representation (qualitative, quantitative and mixed) in CG was identified for the management of Atrial Fibrillation and represented by archetypes and GDL. To ensure semantic interoperability of clinical data, SNOMED CT was used. The rules created were performed with data from 20 patients randomly selected, and were manually validated. The OpenEHR and GDL archetypes allowed to represent TCs contained in the clinical guideline.

Keywords: Archetypes. Guideline Definition Language. Decision Support System. Temporal Constraint.

1 Introduction

To represent Temporal Constraints (TC) contained in clinical guidelines is a challenging task due to the difficulty of specifying and understanding the expressiveness of temporal formalisms and the complexity of the correct and complete reasoning of the algorithms that operate on them [1].

Since the end of the 1990s, there were proposals for clinical guidelines representation with decision support and TC specification using formalisms and tools, as GLARE [1]. However, questions related to integration of decision support in an Electronic Health Record (EHR) were not addressed, especially how to map from EHR data to CDSS rules. The integration of CDSS into EHR based in the archetypes architecture, that is a standard to system interoperability and make possible to represent data and rules of CDSS, avoiding mapping.
Proposals for integrating clinical guidelines with decision support in an EHR applying archetypes as the architecture model were present [2-4]. Decision rules are specified based on archetypes using the Guideline Definition Language (GDL) [5] and an inference engine [6]. This solution is fundamental for sharing information between different EHR, however does not address issues related to TC.

This work aims to represent with archetypes the TCs contained in clinical guidelines, and to use the GDL as unifying language for representing both EHR CDSS. TCs are essential parts of clinical guidelines related to diagnostic and therapeutic algorithms. They are defined as an order among clinical actions, and can be classified as qualitative (as an example, A begins before B) and quantitative (e.g., dates, delays and duration), or mixed. TCs may also involve temporal granularities and periodicity or repetition [1].

Interoperability can be defined as the ability to exchange information between different systems and organizations. ISO/TR 20514 [7] specifies that exchange of information between EHRs must occur in a functional and semantic way. Functional or syntactic interoperability corresponds to the exchange of information between two or more systems in a way that is readable by the receiver. In the semantic form, this information exchange must be understood at the level of the domain concepts defined, so that the information is computable by the receiving system [8]. The level of semantic interoperability depends on both the terminologies used and the content of the information to be exchanged.

Several international standards organizations work with the definition of an interoperable EHR architecture. The standard of ISO/EN 13.606, specifies the use of archetypes to share clinical information. Archetypes are computable expressions of a domain-level concept in the form of structured constraint statements based on a reference model, allowing standardization of knowledge representation in an EHR [9].

2 Methods

From each TC class (qualitative, quantitative and mixed), an example was identified in the Guideline for the Management of Patients with Atrial Fibrillation of the European Society Cardiology. In Table 1 these classes are illustrated with examples. The example of duration was the same used for the mixed TC, since it contemplates these two restrictions.

Based on these classifications of TC, examples of duration, repetition, delays and mixed were represented in archetypes and GDL. TC of data and qualitative are represented in the example of quantitative and qualitative (mixed).

For each example, clinical data and related decision rules were identified, which were modelled by openEHR archetypes and rules in GDL, respectively. Before archetypes modelling, we checked in the openEHR Foundation repository the ones that could be reused or specialized, while the others were elaborated from scratch.

The clinical terms used to represent the data in the archetypes were represented with the terminology of SNOMED-CT for example, “cardioversion (procedure), code
The terms were used for representing the data in the archetypes as to binding for the codification of the terms.

Decision rules have been defined and created using GDL Tool. The rules created were performed with data from 20 patients randomly selected, and were manually validated.

### Table 1. TC classes and examples from AF clinical guideline.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Subclasses</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualitative</td>
<td>Dates</td>
<td>&quot;NOACs can be restarted as soon as effective hemostasis has been achieved&quot;</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>&quot;Date of scheduled cardioversion&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;For patients with AF of &gt;= 48h duration, or when the duration of AF is unknown, OAC therapy is recommended for &gt;= 3 weeks prior to and for &gt;= 4 weeks after cardioversion ....&quot;</td>
</tr>
<tr>
<td>Quantitative</td>
<td>Delays</td>
<td>Acute Coronary Syndrome and/or undergo percutaneous coronary intervention, a period of triple therapy is needed (OAC plus aspirin plus clopidogrel), followed by the combination OAC plus single antiplatelet drug and, after one year, management can be with OAC alone in stable patients&quot;</td>
</tr>
<tr>
<td></td>
<td>Periodic</td>
<td>&quot;annually in patients with normal (CrCl &gt;=80 mL/min) or mild (CrCl 50-79 mL/min) renal impairment, and perhaps 2-3 times per year in patients with moderate (i.e. CrCl 30-40 mL/min) renal impairment)&quot;</td>
</tr>
<tr>
<td></td>
<td>Repeated</td>
<td>&quot;For patients with AF of &gt;= 48h duration, or when the duration of AF is unknown, OAC therapy is recommended for &gt;= 3 weeks prior to and for &gt;= 4 weeks after cardioversion, regardless of the method (electrical or pharmacological)&quot;</td>
</tr>
<tr>
<td>Mixed</td>
<td>Qualitative</td>
<td>&quot;For patients with AF of &gt;= 48h duration, or when the duration of AF is unknown, OAC therapy is recommended for &gt;= 3 weeks prior to and for &gt;= 4 weeks after cardioversion, regardless of the method (electrical or pharmacological)&quot;</td>
</tr>
<tr>
<td></td>
<td>Quantitative</td>
<td>&quot;NOAC- Novel Oral Anticoagulant; AF- Atrial Fibrillation; OAC- Oral Anticoagulant; VKA- Vitamin A Antagonist; CrCl- Creatinine Clearance&quot;</td>
</tr>
</tbody>
</table>

3 Results

The results are the three examples of TC (duration/mix, delays and periodic/repetition) represented by archetypes and GDL rules.

Two rules created to represent duration/mix class of TC are presented in Figure 1, related to pre-cardioversion and post-cardioversion. To represent the rules of quantitative constraints of delays, it was necessary the creation of two rules, as shown in Figure 2. For representing the period of evaluation for renal function, eight rules, two of them are shown in Figure 3. For all the cases, the results of the rules execution are the same comparing to the medical decision.
Archetypes and the rules used for the representation of TC analyzed in this study may be included in the EHR based on archetypes that integrates decision support, proposed by [2].
4   Discussion

According to [10], the representation of TC contained in clinical guidelines is not a trivial task. Understanding and interpreting the expressiveness of the rule often requires other clinical guidelines to support some decisions. In this study, it was also necessary to analyses the Chronic Kidney Disease the use of Novel Oral Anticoagulant (NOAC) therapy. In other cases, the specialist should evaluate the interpretation result as in some guidelines it is described that the patient should be stable; however, this condition will depend on several clinical factors not represented in the guideline.

The reuse of available archetypes in the openEHR repository for the representation of different clinical guidelines in an EHR allowed optimizing the customization of EHR considering the time reduction for the specification of an EHR, and consequently its cost reduction [11].

Different approaches propose the representation of TC using complex algorithms, tasks in plans and flowcharts [12], but there is no interaction with the clinical EHR database and not all the approaches cover so complex temporal issues as those illustrated above [13].

The rules created in this work using archetypes and GDL showed the possibility of representing TC through conditions (IF) and actions (THEN), both in qualitative and quantitative events.

The use of GDL facilitated structuring the rules in computational format using the reference models and knowledge of the archetypes without the need to resort to external tools. Independent tools [14-15] difficult information sharing, so is necessary to include an information mapping to allows the interface between the clinical guidelines and EHR.

The evaluation of the proposed system was performed by testing all possible combination of data entries in each rule created in the GDL. The interface of the GDL Tool allowed to verify the conformity of the expected results for each executed action.

Considering that the different classes of TC can be represented by archetypes and GDL rules, and there are already solutions based on archetypes to integrate CDSS into the EHR [2], it is possible to develop EHR integrating decision support that includes all information, restriction and specifications presented in clinical guidelines.

5   Conclusion

It was possible to represent the TCs contained in clinical guidelines in their various classes of events based on openEHR archetypes and GDL.

The use of the GDL Tool enables the knowledge engineers together with health professionals to create the rules directly in the tool, taking advantage of the structures of the archetypes used to represent the clinical guideline with its TC through the conditions and their actions, facilitating the structuring of the logic of the rules.
Acknowledgments

This study was supported by CAPES (Brazilian Coordination for the Improvement of Higher Education Personnel) PhD scholarship

References


A Knowledge-Based Conceptual Model to Simulate Circulatory Shock

David Riaño¹, José Ramón Alonso²

¹Research Group on Artificial Intelligence, Universitat Rovira i Virgili, Tarragona, Spain
david.riano@urv.cat
²Emergency Department, Hospital Universitari Clínic de Barcelona, Barcelona, Spain
jralonso@clinic.cat

Abstract. Shock is a prevalent critical care and emergency issue requiring complex clinical interventions and skills. Training young physicians to manage such cases is better approached with the use of simulators to reduce risks and stress. In the past we implemented and used a patient simulator able to emulate the long term evolution of patients for seven sorts of shocks. Here we reformulate and provide a new conceptual model that fixes two limitations of the previous simulator: handling null treatments and time continuous evolution.


1 Introduction

Knowledge-based computer simulation of patients is an underexplored area of medical informatics research that deals with the construction of knowledge models that emulate the evolution of patients with respect to a concrete disease or group of diseases, or their reaction to specific treatments. In health care, this field is closely related to data synthetization [1-3] and prognosis [4-6] in the sense that, once validated, these knowledge models can be used to synthesize clinical data records or to foretell the expected evolution of a patient's condition.

In [7], Lippert et al. enumerate the major types of simulation in medicine as manikin-based simulation, part-task trainers, virtual reality trainers, screen-based simulators, simulated patients, and hybrid simulations. Notice that their focus is on the sort of interface provided, which can be a manikin or artificial body parts, standard or virtual reality software, and virtual patients or actors. Behind that interfaces, the simulation engine may use mathematical models such as differential equations [8], statistical models such as regression or Bayesian networks [9], or knowledge representation structures such as decision tables [12], among others.

Simulators are useful in the context of medical research and education [10-11] where the direct action on real patients is difficult; e.g., when there is a shortage of cases, frailty critical conditions, sensitive treatments, quick response time, and unavailability of second chances. This is the case of shock patient management in hosp-
tal emergency units where younger doctors and residents are trained. In [12], we proposed a knowledge-based simulator of patients having a shock. There, knowledge about patient evolution was represented as decision tables [13] storing units of information of the form \((S, A; S')\) where (1) \(S\) is a description of patient condition in terms of her vital signs, (2) \(A\) is the clinical action (treatment) prescribed for that patient, and (3) \(S'\) is the expected evolution of the patient condition in terms of her vital signs after twenty minutes receiving the \(A\) treatment when the patient's original state was \(S\).

This simulator was incorporated in an education program of the residents at the University Clinical Hospital in Barcelona (Spain) showing a successful learning curve and a good user’s satisfaction that were published in [14]. However, some of the design decisions on the simulator were controversial. First, the simulator was constructed under a *reactive approach*, which means that, according to the knowledge contained in the decision tables, the patient's condition evolves from \(S\) to \(S'\) as a reaction to the actions of the proposed treatment \(A\). Consequently a null treatment \(A=\emptyset\) will cause no change in the patient's condition (i.e., \((S, \emptyset; S)\)), contrarily to what we should expect. A second issue is that the system was conceived to represent a *discrete evolution* in which clinical decisions and their consequences occur every 20 minutes. So, the knowledge about patient evolutions in the decision tables was conditioned by this fact and it was unable to forecast patient evolutions for arbitrary times.

Here, we propose a new knowledge-based conceptual model of our simulator resulting from the adaptation of our previous system to also deal with null treatments and providing continuous time simulation. In section 2 we formalize the new simulation model, and in section 3 we describe the simulation cycle. Section 4 has a short discussion on the next steps to conclude the construction of the simulator.

## 2 The Simulation Model for Shock Patients

Shock is a common condition in critical care, affecting about one third of patients in ICUs. It is described as the clinical expression of circulatory failure that results in inadequate cellular oxygen utilization [15]. Shock patients are considered critical cases that have to be stabilized urgently. Our knowledge-based simulator inherits the same sorts of shocks targeted, the clinical context of these shocks (i.e., vital signs and clinical actions), the patient personalization profile (i.e., normality ranges and sensitivities), and the treatment response variability model described in [12], but it incorporates knowledge about pharmacodynamics, includes a new knowledge-base about patient-treatment evolution, and redefines the simulation cycle to allow patient evolution under a null treatment, time continuous simulation, and exacerbations.

### 2.1 Clinical Context and Parameters of the Simulator

Seven sorts of shocks are simulated: anaphylactic shock, cardiac tamponade, cardiogenic shock, hemorrhagic shock, neurogenic shock, shock due to acute pulmonary embolism, and septic shock. The clinical condition of a patient is described in terms
of the $n_s=7$ shock-related vital signs $s_i$ in $S^1$. Treatments are combinations of actions from a set of the $n_a=18$ clinical actions $a_j$ in $A^2$.

Simulated patients have an individual profile $(N_p, S_p, T_p)$ where $N_p$ is the *normality parameters* of the patient $p$ with respect to all the vital signs in $S$; i.e., $(s_i, m_i, l_i, h_i, M_i)$ with $m_i$, $l_i$, $h_i$, and $M_i$ indicating the min, low, high, and max values of the sign $s_i$ for $p$. Values below $m_i$ (or above $M_i$) are not admissible for this patient, while values between $l_i$ and $h_i$ are considered normal for $p$. Sign $s_i$ is said to have low (or high) values for $p$ when its value is between $m_i$ and $l_i$ (or between $h_i$ and $M_i$), respectively.

$S_p$ stands for the *sensitivity* of patient $p$ to all the clinical actions in $A$. So, $(a_j, sens_j)$ in $S_p$ shows the sensitivity of $p$ to action $a_j$ as a percentage $sens_j$. $sens_j=100\%$ means normal sensitivity to $a_j$, $sens_j>100\%$ shows hypersensitivity to $a_j$, and $sens_j<100\%$ resistance to $a_j$. If a patient is $110\%$ hypersensitive to dobutamine and $80\%$ resistant to antibiotics, the effects of dobutamine in the vital signs of the patient $p$ are increased $10\%$ above the expected effects on a regular patient, and the effects of antibiotics reduced $20\%$. Effects on regular patients are inferred by the knowledge-base (see 2.2).

$T_p$ is the patient *tendency* with respect to each one of the signs in $S$; i.e., a set of functions $t_i(t)$ describing the natural evolution of the values of the sign $s_i$ along time.

### 2.2 Representing Patient Evolution with a Rule-Based System

The units of knowledge $(S, A; S')$ described in section 1.1 are represented as evolution rules such as "IF ${\{s_k, r_k}\}_{k \in S}$ AND ${\{a_k, I_k\}_{k \in A}}$ THEN ${\{s'_k, D_k\}_{k \in S'}}$" 4, with $r_k$ one of the discretized values 'low', 'normal', or 'high', and $I_k$ either a true/false value representing whether action $a_k$ is applied or not, or $I_k$ a dosage range $[L_k, U_k]$ if $a_k$ is a drug administration. This sort of rules is *activated* for all the patients whose clinical condition includes the sign-value pairs ${\{s_k, v_k\}_{k \in S}}$ with:

- $m_k \leq v_k < l_k$ ($m_k$ and $l_k$ the min and low values for $s_k$ in $N_p$), if $r_k = \text{low}$;
- $l_k \leq v_k < h_k$ ($l_k$ and $h_k$ the low and high values for $s_k$ in $N_p$), if $r_k = \text{normal}$;
- $h_k \leq v_k \leq M_k$ ($h_k$ and $M_k$ the high and max values for $s_k$ in $N_p$), if $r_k = \text{high}$;

and whose current treatment includes ${\{a_k, d_k\}_{k \in A}}$ with:

- $d_k = I_k$ (representing whether $a_k$ is part of the current treatment and $I_k$ of $a_k$ is true, or $a_k$ is not part of the treatment and $I_k$ is false in the rule).

---

1 Vital signs $S$ are: systolic/diastolic blood pressures, arterial blood pressure, heart rate, central venous pressure, superior vena cava oxygen saturation, and finding of hematocrit.

2 Clinical actions $A$ are: antibiotic therapy, antihistamine, atropine, diuretic, epinephrine bolus, thrombolytic, hydrocortisone, insertion of intra-aortic balloon, pericardiocentesis, reperfusion, resuscitation using intravenous fluid, transfusion of plasma, transfusion of red blood cells, vasodilators, dobutamine, dopamine, epinephrine, and norepinephrine.

3 $S$ and $S'$ being subsets of $S$, and $A$ a subset of $A$.

4 In the current implementation, rules are adapted from a former simulator and, therefore, they affect one single sign and action; i.e., "IF $(s_x, m_x)$ AND $(s_y, I_y)$ THEN $(s_z, D_z)$".
• \( L_k \leq d_k \leq U_k \) (representing the patient is receiving a dosage \( d_k \) ml/h of drug \( a_k \) in the range \( l_k = [L_k, U_k] \) ml/h);

All the activated rules are triggered. In a former version of the simulator, if none evolution rule was triggered, the patient condition would remain unchanged. If an evolution rule is triggered, the values \( \Delta_k \) in the right hand side of the rule are the expected increments (or decrements) of the vital signs \( s_k \) in \( S' \). If several rules \( R \) of the knowledge-base are triggered for a simulated patient \( p \), then all the increments \( \Delta_k' \) suggested by each rule for the same vital sign \( s_k \) are accumulated, and the final value \( \Delta_k = \sum_{r \in R} \Delta_k' \) is used to calculate the expected evolution of \( s_k \).

2.3 Pharmacodynamics Knowledge

In pharmacodynamics [16,17], the effect of a drug can be measured as a percentage of the maximal response expected. This percentage (or effectiveness) may vary along time. In our system, this variation is represented as a vector \((t_0, t_1, t_2, t_3)\) of increasing times \( t_i \) (in minutes), synthesizing the function in Fig. 1. Time \( t_0 \) is the time required for the drug to start taking some effect, \( t_1 \) is the time the drug reaches maximal effect (100% effectiveness), \( t_2 \) when the effect starts vanishing, and \( t_3 \) when the observed effect disappears.

![Fig. 1. Times of clinical effectiveness \( K_a \) of a clinical action \( a_j \).](image)

According to their pharmacodynamics, pharmacological actions in \( A \) are classified as serums, vasoactive-vasodilators, interventions, epinephrine-bolus, diuretics, hydrocortisone, and antibiotics. They show different time behavior models [16,17]. For example, serums have an immediate effect \((t_0 = 0)\) that lasts while they are supplied \((t_2)\) and then vanish. Their \( t \) values depend on the sort of shock considered, and not on the signs affected. This implies different effectiveness functions for different shocks. Vasoactive-Vasodilators drugs have also an immediate effect \((t_0 = 0)\) that soon reaches its maximal value \((t_1 \approx 5 \text{ min})\) and they last while they’re provided, and then vanish very fast \((t_2-t_2 \approx 5 \text{ min})\). Interventions such as thrombolytic, insertion of intra-aortic balloon, pericardiocentesis, or reperfusion have a time of application \((t_0 > 0)\) and a permanent effect \((t_1 -> \infty)\) that makes the signs affected to converge to the normality parameters \( N_p \) of the patient. Epinephrine-bolus shows a different effect depending on the shock: its application has an instant effect that shortly grows to maximal \((t_0 = 0, t_1 \approx 5 \text{ min})\), then for anaphylactic shock it lasts forever \((t_1 -> \infty)\), but not for the rest of shocks \((t_2 \approx 10 \text{ min}, t_3 \approx 15 \text{ min})\). For antibiotics the effect function does not apply to
our vital signs in S, since the purpose of antibiotics is not to modify the vital signs but to deal with possible infections.

Most of the clinical actions in A have a cumulative behavior (i.e., their effect last when they're cancelled). Consequently, their effect cumulates as function in Fig. 1 shows, when $t_3 \to \infty$. On the contrary, vasoactive-vasodilator drugs have a non-cumulative behavior (i.e., their effect stops immediately after they're cancelled). Consequently, they are atemporal as Fig. 2 shows. Points A, B, C, and D stand for low, medium, high, and maximal dose. These values are provided in Table 1 for vasoactive-vasodilator drugs.

The way effectiveness is calculated for a cumulative and non-cumulative drug is completely different. For cumulative drugs, Fig. 1 directly provides the required effectiveness value. For non-cumulative drugs, Fig. 1 shows the dynamics of relative change, while Fig. 2 and Table 1 show absolute values, and these two need to be combined together to establish the final effectiveness.

For example, an 80 kg patient taking antihistamine (cumulative) drug will have a $K_{\text{antihistamine}}=75\%$ effectiveness (Fig. 1) after a time $t' = \frac{1}{4}(t_0+3t_1)$ or $t' = \frac{1}{4}(t_3+3t_2)$ has passed since the intake time. On the contrary, if this patient is continuously taking dobutamine 20 ml/h (non-cumulative), the factor 66.7 in the last column of Table 1 is used to convert a dose from ml/h to $\mu$g/kg/min given a standard concentration of the drugs in the ICU. In the example, we obtain $20 \times 66.7/80 = 16.7 \mu$g/kg/min. This value is found between the B=15 and the C=30 dobutamine values in Table 1. Then, the $y$ value for $x=16.7$ in the function Fig. 2 is calculated as $16.7/30+1.25 = 1.8$. Then the drug reaches $K_{\text{dobutamine}}=135.5\%$ effectiveness ($=1.8 \times 75\%$); i.e., over 100% effect, if either $t' = \frac{1}{4}(t_0+3t_1)$ or $t' = \frac{1}{4}(t_3+3t_2)$ times have passed since the patient started taking dobutamine.

Table 1. Magnitudes of the effect function of non-cumulative clinical actions dosage, and mg/h to mg/Kg/min conversion factor ($K$).

<table>
<thead>
<tr>
<th>clinical action</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>$K$-convert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dobutamine</td>
<td>7</td>
<td>15</td>
<td>30</td>
<td>60</td>
<td>66.7</td>
</tr>
<tr>
<td>Dopamine</td>
<td>5</td>
<td>12</td>
<td>30</td>
<td>70</td>
<td>66.7</td>
</tr>
<tr>
<td>Epinephrine</td>
<td>0.08</td>
<td>0.2</td>
<td>0.4</td>
<td>3</td>
<td>0.7</td>
</tr>
<tr>
<td>Norepinephrine</td>
<td>0.12</td>
<td>0.6</td>
<td>1.8</td>
<td>3</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Fig. 2. Effectiveness function of non-cumulative clinical actions dosage.
3 The Simulation Cycle

Our simulator defines a cycle with the user in which, at time $t$, the user observes the state of the patient $S^{(0)}$ described in terms of the current values of all the vital signs in $S$, and decides a treatment $A^{(0)}$ which is introduced in the simulator. This treatment contains the actions and dosages for a subset of the possible clinical actions in $A$. At this point, the simulator is able to compute the continuous evolution of the patient till the user decides a new interaction with the patient at time $t'>t$. The process is repeated now with the patient in state $S^{(1)}$.

The way our model computes the continuous evolution of a shock patient $p$ in state $S^{(0)}$ receiving a treatment $A^{(0)}$ after $t'$ minutes, follows a six step process:

1. **Obtain clinical actions effectiveness after a time $t'$**: The $K_{a_j}$ functions (Fig. 1) corresponding to each $(a_j, t_j)$ in $A^{(0)}$ is calculated $%_j = K_{a_j}(t+t'-st_j)-K_{a_j}(t-st_j)$, with $%_j = 0$ if $a_j \not\in A^{(0)}$. If $K_{a_j}(t+t'-st_j) = 0$ for some $a_j$ with $(st_j + t_0) < (t + t')$, then $(a_j, st_j)$ is removed from $A^{(0)}$ because the effect of $a_j$ has passed.

2. **Apply the evolution rules**: The rule-based system described in 2.2 is applied to the current condition $S^{(0)}$ of the patient $p$ to obtain maximal variations $\Delta_i$ of the signs $s_i$ caused by the clinical actions $a_j$ such that $(a_j, st_j)$ is in $A^{(0)}$. Values $\Delta_i$ of the $a_j$ being interventions are calculated respect to the normality parameters $N_j$ of $p$.

3. **Calculate time-dependent effects of clinical actions on each vital sign**: This is achieved multiplying $\Delta_i$ and $%_j$ to obtain the time-dependent values $\Delta_i^{(t+t')}$. 

4. **Personalize effects with the patient sensitivities**: The patient sensitivities $s_{ij}$ in $S_p$ to the clinical actions $a_j$ in $A^{(0)}$ are used to personalize the time-dependent values $\Delta_i^{(t+t')}$. That is to say, the values $P_{ij} = sens_{ij} \cdot \Delta_i^{(t+t')}$ are calculated.

5. **Accumulate treatment effects on the vital sign of the patient including null treatment effects**: The tendency functions $t_l(t)$ in $T_p$ are used to calculate the value $v_i^{(t+t')} = v_i^{(0)} + P_{ij} + t_l(t+t')$ for all the vital signs $s_i$ in $S$. Notice that if $T^{(0)} = \emptyset$ (null treatment), $v_i^{(t+t')} = v_i^{(0)} + t_l(t+t')$ and the natural tendency of the patient applies.

6. **Update patient state**: The new state of $p$ after $t'$ time is $S^{(t+t')} = \{(s_i, v_i^{(t+t')})\}_{0 \leq i \leq n_s}$.

4 Short Discussion and Future Actions

An improved conceptual model to simulate realistic patient evolutions when affected of one among seven shocks has been proposed, based on a former implementation [12], and to solve some previous limitations. The model received initial partial analysis with regard to fidelity and validity [7], but much work remains before it could be incorporated in a real hospital education program as in [12]. Next actions will be to conclude validation, to refine implementation and testing, and finally to apply it at the Emergency Department of the University Clinical Hospital in Barcelona. This works

---

5 $A^{(0)}$ contains values of the form $(a_j, st_j)$, with $st_j$ being the time the action $a_j$ started. These include the active actions in the current treatment, but also past finished actions not being part of the current treatment but whose effects persist at time $t$. 

5 References

A Data- and Expert-driven Decision Support Framework for Helping Patients Adhere to Therapy: Psychobehavioral Targets and Associated Interventions

Szymon Wilk1,2, Dymphna O’Sullivan3, Martin Michalowski4, Silvia Bonaccio2, Wojtek Michalowski2, Mor Peleg5, Marc Carrier6

1 Poznan University of Technology, Poznan, Poland
2 University of Ottawa, Ottawa, Canada
3 City, University of London, London, United Kingdom
4 MET Research Group, Ottawa, Canada
5 University of Haifa, Haifa, Israel
6 The Ottawa Hospital Research Institute, Ottawa, Canada

Abstract. Patient adherence to therapy is one of the key determinants of treatment success, while low levels of adherence potentially lead to the worsening of health outcomes and to increased health care costs. The automatic identification of psychobehavioral targets, i.e., patterns in patients’ psychological characteristics and behaviors that positively or negatively affect their adherence level, should help physicians develop therapies that influence adherence and improve health outcomes. These targets can also be used to develop psychobehavioral interventions, i.e., plans of actions to modify patients’ behavior and psychological stance, that maintain or improve adherence level by motivating patients to avoid negative psychobehavioral targets or to achieve positive ones. In this work, we propose a theoretical decision support framework that helps patients better adhere to therapy by automatically identifying psychobehavioral targets and selecting corresponding psychobehavioral interventions. We use data-driven techniques to discover these targets from patient data. Specifically, we apply dominance-based rough set theory to induce decision rules from data and then automatically extract targets from these rules. Once these targets are identified, we apply expert knowledge to select the most appropriate generic psychobehavioral interventions and customize them to patient characteristics. We illustrate the proposed framework using a case study of patients with atrial fibrillation who follow oral anticoagulation therapy. We describe the psychobehavioral targets identified from data and present associated psychobehavioral interventions.

Topics: Knowledge extraction from health care databases and medical records; Patient empowerment in health care
Submission category: regular paper
1 Introduction

One of the most significant barriers to effective medical treatment is patients’ failure to follow advice and recommendations from their healthcare provider. A Cochrane review by Haynes et al [1] concluded that interventions for enhancing therapy adherence can have a far greater impact on clinical outcomes than improvement in therapy. Non-adherence comes at a significant cost – the Institute for Healthcare Informatics estimated that between $100B and $300B of avoidable health care costs have been attributed to patients’ non-adherence in the US annually [2]. Rates of adherence vary according to the type of therapy. According to the World Health Organization, adherence to long-term therapies in the general population is around 50% in developed countries and lower in developing ones. Moreover, adherence to short-term therapy has been estimated as 70-80%, however complying with lifestyle changes is very low and reported to be 20-30% [3].

Interventions that involve patient education and behavior modification are commonly used with the aim of improving adherence. A systematic review of strategies to improve adherence to self-administered medications for chronic diseases in the US examined the effectiveness of interventions and concluded that case management and patient education with behavioral support improved medication adherence for more than one condition [4].

In this paper, we propose a theoretical decision support framework for the development of psychobehavioral interventions to help patients with their adherence to treatment. The framework involves two phases. The first phase is data driven and it is aimed at identifying psychobehavioral targets, defined as patterns in patient’s psychological characteristics and behaviors that affect adherence. Here we apply a dominance-based rough set approach (DRSA) [5] to induce decision rules from patient data. Obtained rules capture the relationship between a patient’s sociodemographic, psychological and behavioral characteristic [6] and adherence level, and from these rules we identify the psychobehavioral targets. In the second phase, we propose to use expert knowledge to select psychobehavioral interventions, i.e., systematic plans of actions that affect patients’ behaviors and psychological stance, that are appropriate for the targets and to customize them for a given patient. For example, a psychobehavioral target that implies limiting smoking to at most a light level may require an intervention that involves education (customized to the patient's level of health literacy) about negative consequences of moderate and heavy smoking and goal setting exercises along with self-monitoring (operationalized as reporting of smoking behavior).

We illustrate the proposed framework in the context of patients with atrial fibrillation (AF) whose adherence to oral anticoagulation treatment we aim to improve. We present the psychobehavioral targets and explain how they were identified. We also describe possible interventions that correspond to the targets.
2 Methods

2.1 Foundations of DRSA

In this section we introduce those DRSA notions and concepts that are relevant for our framework – a detailed presentation of DRSA can be found in [5]. DRSA is a data analysis and knowledge discovery technique aimed at analyzing imperfect (e.g., inconsistent or incomplete) data. It assumes analyzed objects (e.g., patients, customers) are categorized into ordered classes, typically from worst to best, and they are described with features (with symbolic and numerical domains) whose values may also be ordered (such features are referred to as criteria). Instead of individual classes, DRSA considers their unions, e.g., a set of classes that are at least or at most as good as a given class. Such unions appear in consequences of decision rules; therefore, these rules are referred to as at least or at most decision rules.

DRSA allows decision rules to be viewed from two perspectives – classification- and intervention-oriented [7]. In the classification-oriented perspective, the premise of a rule captures selected features of an object, and the consequence predicts an outcome (a union of classes) associated with these features. The intervention-oriented perspective assumes there are some interventions that may be used to change the object’s features and that these changes affect the object’s classification. Then, the premise of a rule defines the target for intervention and its consequence indicates an outcome resulting from achieving the target. Both perspectives adopt a different interpretation of decision rules, however, they do not impose any special requirements on algorithms used for their induction. Thus, the same algorithm may be used to obtain both types of rules.

Formally, let us assume $r$ is a rule $\phi \rightarrow \psi$ derived from a set of objects $U$, where $\phi$ is a premise (a conjunction of conditions on selected features) and $\psi$ is a consequence (a union of decision classes) decision. A rule $r$ is also characterized by its confidence (or certainty) denoted as $\text{conf}(r, U)$, which is a conditional probability $P(\psi|\phi)$ established in $U$.

Within the intervention-oriented perspective, $\phi$ indicates an intervention target and $\psi$ is a resulting classification given as a union of classes. If $\phi$ is achieved by objects that currently do not match it and that do not belong to $\psi$, then their classification will change and become consistent with $\psi$. A success rate associated with achieving a target is approximated by the confidence of $r$.

An intervention target indicated by the premise of rule $r$ can be evaluated using its impact, defined as the ratio of objects affected by achieving the target. Intuitively, impact indicates how “strong” a target is – the higher the impact, the more objects are affected by the target. Impact is established in the context of $r$ and using the set $U'$ (possibly different than $U$, however, both sets should be homogeneous) and formally it is defined as:

$$\delta(r) = \frac{|m[\neg\phi, \neg\psi'] \cap m(\neg\psi, \neg\psi')|}{|U'|} \times \text{conf}(r, U),$$

(1)
where $m(y, X)$ is a set of objects from $X$ that satisfy the condition $y$ (note that in the above formula negated conditions are considered).

There are, however, certain situations where it is reasonable to limit the set of objects that achieve the target by introducing *applicability profiles* acting as additional conditions imposed on these objects. Formally, an applicability profile $\phi_i$ is defined analogously to $\phi$ as a conjunction of conditions on selected features. Then, the impact of the psychobehavioral target is redefined as:

$$\delta(r) = \frac{|\bigcup_{i=1}^v m(\phi_i, u')| m(\neg \phi_i, u') m(\neg \phi_i')|}{|u'|} \times \text{conf}(r, U),$$

(2)

where $v$ is the number of considered applicability profiles.

There are two types of intervention targets – positive and negative. Positive targets are associated with changes that improve class assignment (i.e., an object gets assigned to a better class) and they are given as premises of *at least* decision rules. On the contrary, negative targets correspond to changes that result in deteriorated class assignment (i.e., to a worse class) and they are given as premises of *at most* decision rules.

### 2.2 Identification of Psychobehavioral Targets

The intervention-oriented perspective in DRSA described in Section 2.1 needs to be adapted to the problem of adherence support. We can no longer assume that an intervention changes any possible characteristic of a patient (i.e., value of any feature). Patients are characterized by sociodemographic, psychological and behavioral features, and only the features from the last two groups can be modified by an intervention – for example it is possible to affect a patient’s smoking habit, while it is impossible to change a patient’s age. In the subsequent text, we refer to features from the last two groups as to *psychobehavioral* features and distinguish them from *sociodemographic* ones.

Given the distinction between sociodemographic and psychobehavioral features we expand the formal representation of a rule $r$ to $\phi_{pb} \land \phi_{sd} \rightarrow \psi$, where $\phi_{pb}$ is a conjunction of conditions on psychobehavioral features, $\phi_{sd}$ is a conjunction of conditions on sociodemographic features, and $\psi$ is an at least or at most union of adherence levels. Moreover, $\phi_{pb}$ defines a *psychobehavioral intervention target* (psychobehavioral target in short), and $\phi_{sd}$ defines a *sociodemographic context* (or a sociodemographic characteristic, e.g., age $> 55$ years and at least middle socioeconomic status).

For induction we use the VCDOMLEM algorithm [8] that produces a minimal set of rules, i.e., a smallest set of rules that capture all patterns in a data set. It is possible to obtain rules with an empty sociodemographic context or psychobehavioral target. Rules with an empty $\phi_{pb}$ are of no use (they indicate no psychobehavioral target) and are discarded. In order to minimize the number of such rules we perform preliminary feature selection. For this purpose we identify so-called *reducts* [5] – minimal subsets of features that have the same discriminatory power as the entire set of features. If there are multiple reducts, then we select the one that contains the largest number of psychobehavioral features. Finally, we expand the selected reduct with the remaining features.
psychobehavioral features, if any. While this expansion may make the obtained set of features not a minimal one, it allows us to identify more psychobehavioral targets. At this point it is still possible to obtain rules with an empty $\phi_{pb}$, however, their number is smaller than when considering the entire feature set.

Splitting the premise of a rule into $\phi_{pb}$ and $\phi_{sd}$ calls for a revised measure to evaluate the impact of the former. As previously stated, we evaluate the target in the context of a rule $r$ that contains it and of the set $U'$. The updated definition is the following:

$$\delta(r) = \frac{|m(\phi_{sd} \cup \neg \phi_{pb}, U') \cap m(\neg \phi_{sd}, U')|}{|U'|} \times \text{conf}(r, U).$$  \hspace{1cm} (3)

Comparing definition (3) with (2) we see that the sociodemographic context $\phi_{sd}$ acts as an applicability profile that limits the impact of the psychobehavioral target $\phi_{pb}$ to those patients who match $\phi_{sd}$.

We also distinguish between positive and negative psychobehavioral targets. A positive target is identified by an at least rule and calls for a psychobehavioral intervention that will help a patient achieve this target, while a negative target is identified by an at most rule and requires an intervention that will prevent a patient from achieving the target. This leads to the following layman’s interpretation of at least and at most rules, respectively:

- if the intervention associated with the positive target is successfully applied and the patient manages to achieve this target, then his/her adherence level will improve,
- if the intervention associated with the negative target is successfully applied and the patient manages to avoid the target, then his/her adherence will not deteriorate.

Here we need to observe that it is possible that an induced rule captures correlation, not causation, between a psychobehavioral target and an adherence level. This will manifest as not improving adherence after the positive target has been attained, or not maintaining adherence after the negative target has been avoided. To account for such undesired situation we plan to add a feedback loop to monitor changes in patients’ adherence, identify rules with poor causality and discard those rules.

2.3 Selection of Psychobehavioral Interventions

Psychobehavioral interventions combine patient education and behavioral modification. The latter makes use of a series of actions that rely on behavior change techniques catalogued in [9]. Many of these techniques employ behavioral economics principles [10] such as: establishing commitment, offering rewards for adherence, and leveraging social influence.

Patient education is focused on providing information about behavior-health links, educating on consequences of certain behaviors, and providing instructions about proper behavior. In addition to explaining the disease manifestation and its prognosis and management, patient education needs to emphasize the key role that the patient plays in successful therapy through her/his engagement and adherence to care. Educa-
tion should be customized to the type of a target that the given intervention is associated with. For a positive target, it should demonstrate how a patient’s state will deteriorate if he/she fails to change the behavior (the principle of loss aversion). In the case of a negative target, it should show the benefits of maintaining the current good behavior.

Patient behavior modification needs to be focused on engaging in goal setting, providing feedback on goal attainment, and providing encouragement for positive behavior. Critical in behavior modification is monitoring whether a patient is adhering to treatment as well as working towards his/her set goals. Self-reporting (more generally self-monitoring) can be used for this purpose and may even increase adherence to therapy [11]. Accuracy of reporting can be increased by another behavioral economics strategy called priming for honesty [10] that requires a patient to sign his/her adherence report before submitting it. Adherence and engagement with therapy can be monitored by measuring changes in the patient’s level of self-determination to embrace a desired behavior [12] (the more autonomous motivation, the better the engagement and outcomes), as well as changes in his/her perceived competence and self-efficacy for treatment adherence [13].

We use the Transtheoretical Model (TTM) [14] as a theoretical cadre for psychobehavioral interventions. Drawing from many psychotherapy theories, TTM provides a comprehensive understanding of behavior change and allows classifying patients according to where they are in their readiness to change. The TTM proposes sequential stages of change, from pre-contemplation (the patient has yet to recognize the need to change), contemplation (the patient considers change), preparation (the patient plans to take action toward change), action (the patient begins to take actions that induce change), and maintenance (the patient has succeeded in changing behavior and focuses on maintaining the new behaviors in his/her lifestyle). The TTM acknowledges that some patients relapse (the patient returns to the pre-change behaviors). The current stage in the TTM is established through a patient’s answers to a short set of questions [15]. It should drive the customization of actions constituting a behavioral intervention, given that what will work for a patient for example, at the action stage, may not work for a patient at the pre-contemplation stage.

3 Case Study: Management of Atrial Fibrillation

In this section, we present a clinical case study where we apply our proposed framework to help atrial fibrillation (Af) patients with their adherence to oral anticoagulation therapy. We start by presenting the patient data and providing insights into the feature selection process. Then we describe the psychobehavioral targets identified from rules induced from these data. Finally, we describe interventions associated with the targets.
3.1 Analyzed Data and Selected Features

In this case study, we used 12 vignettes describing patients with AF. We prepared them based on systematic reviews concerned with the use of anticoagulation in AF management and using expert knowledge. The vignettes were vetted and revised by the hematologist on our team who ensured they represented typical situations encountered in his clinical practice. Although real data would be preferred, after consulting with the clinicians we are convinced that psychobehavioral targets discovered from vignettes are realistic. Each vignette is described by 10 features and associated with one of three outcomes of adherence level classes corresponding to poor, average, and good adherence levels. Ideally, all sociodemographic, psychological and behavioral features indicated in [6] should be collected, however, in practice they are usually not explicitly recorded and therefore we have focused on those features that could be reconstructed from information available in patient records.

Among all the features we have one behavioral feature describing tobacco smoking or alcohol intake, one psychological feature describing patient’s willingness to be in charge of his/her health and seven sociodemographic features (i.e. age, socioeconomic status or employment status). There is also one feature that describes history of adherence (i.e., a pattern of past behaviors) – it is not referenced in [6] and in our analysis we considered it as a sociodemographic one.

In the feature selection step (see Section 2.2 for details) we identified a subset of three features (one of the reducts) that impact the adherence level: history of adherence, tobacco smoking or alcohol intake, and willingness to be in charge. The first feature is sociodemographic and it cannot be affected by any intervention, while the second and third are psychobehavioral features and can be changed. To further validate this selection we applied the UTA (UTilities Additives) method [16] to construct a value function (or score) for adherence. The results obtained from UTA analysis confirmed our chosen subset of features.

Vignettes with descriptions limited to the selected features are given in Table 1 (in the subsequent text we will use shorter names for the features presented in this table).

In this table, as indicated by the numbers in parenthesis, we show the ordering of values for features and the ordering of adherence levels (decision classes).

Values of some of the features in Table 1 aggregate several precise values, i.e., none or moderate for adherence history or none or light for smoking or alcohol. This aggregation captures domain knowledge implying that from the perspective of the adherence level, lack of adherence history is equally important as having moderate history, and light smoking or alcohol intake is equally important as none.

---

1 Due to the page limit, we are not able to present the full feature list of vignettes in the paper but it is available as an on-line appendix at http://www.cs.put.poznan.pl/swilk/kr4hc2017/vignettes.pdf
### Table 1. Vignettes considered in the study

<table>
<thead>
<tr>
<th>Vignette</th>
<th>Adherence history</th>
<th>Smoking or alcohol</th>
<th>In charge</th>
<th>Adherence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>(3) good</td>
<td>(2) moderate</td>
<td>(2) yes</td>
<td>(2) moderate</td>
</tr>
<tr>
<td>v2</td>
<td>(2) none or moderate</td>
<td>(1) none or light</td>
<td>(2) yes</td>
<td>(3) good</td>
</tr>
<tr>
<td>v3</td>
<td>(2) none or moderate</td>
<td>(1) none or light</td>
<td>(1) no</td>
<td>(2) moderate</td>
</tr>
<tr>
<td>v4</td>
<td>(1) poor</td>
<td>(1) none or light</td>
<td>(2) yes</td>
<td>(1) poor</td>
</tr>
<tr>
<td>v5</td>
<td>(2) none or moderate</td>
<td>(3) heavy</td>
<td>(1) no</td>
<td>(1) poor</td>
</tr>
<tr>
<td>v6</td>
<td>(1) poor</td>
<td>(2) moderate</td>
<td>(1) no</td>
<td>(1) poor</td>
</tr>
<tr>
<td>v7</td>
<td>(3) good</td>
<td>(1) none or light</td>
<td>(2) yes</td>
<td>(3) good</td>
</tr>
<tr>
<td>v8</td>
<td>(2) none or moderate</td>
<td>(1) none or light</td>
<td>(1) no</td>
<td>(2) moderate</td>
</tr>
<tr>
<td>v9</td>
<td>(2) none or moderate</td>
<td>(1) none or light</td>
<td>(1) no</td>
<td>(2) moderate</td>
</tr>
<tr>
<td>v10</td>
<td>(3) good</td>
<td>(2) moderate</td>
<td>(2) yes</td>
<td>(2) moderate</td>
</tr>
<tr>
<td>v11</td>
<td>(2) none or moderate</td>
<td>(1) none or light</td>
<td>(2) yes</td>
<td>(3) good</td>
</tr>
<tr>
<td>v12</td>
<td>(1) poor</td>
<td>(3) heavy</td>
<td>(1) no</td>
<td>(1) poor</td>
</tr>
</tbody>
</table>

### 3.2 Identified Psychobehavioral Targets

Given the limited size of our data set we used the leaving-one-out (LOO) scheme to induce decision rules and to evaluate targets. We iterated over 12 vignettes – in each iteration 11 vignettes formed a derivation set used to induce rules with targets, and one vignette was used to evaluate their impact. The number of induced rules was constant across iterations – each time we obtained 7 rules. Finally, the impact of specific targets was averaged over those iterations in which they were identified.

In Table 2 we present the most frequent decision rules that were induced following the LOO scheme. There are three *at least* rules (r1-r3) and four *at most* rules (r4-r7). The confidence of all these rules is equal to 1.0. Rules r4-r7 were obtained in all LOO iterations, and rules r1-r3 in all but one, where they were replaced by rules with simplified conditions. Two rules – r2 and r4 (both marked with gray background in Table 2) – contain an empty psychobehavioral target and they were discarded from further evaluation.

### Table 2. Decision rules prevalent in the LOO scheme

<table>
<thead>
<tr>
<th>Sociodemographic context</th>
<th>Psychobehavioral target</th>
<th>Adherence history</th>
<th>Smoking or alcohol</th>
<th>In charge</th>
<th>Adherence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&gt;= none or moderate</td>
<td>&lt;= none or light</td>
<td>&gt;= yes</td>
<td>&gt;= good</td>
<td></td>
</tr>
<tr>
<td>r2</td>
<td>&gt;= good</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r3</td>
<td>&gt;= none or moderate</td>
<td>&lt;= none or light</td>
<td></td>
<td>&gt;= moderate</td>
<td></td>
</tr>
<tr>
<td>r4</td>
<td>&lt;= poor</td>
<td></td>
<td>&lt;= poor</td>
<td>&lt;= poor</td>
<td></td>
</tr>
<tr>
<td>r5</td>
<td>&gt;= heavy</td>
<td>&lt;= no</td>
<td>&lt;= moderate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r6</td>
<td>&gt;= moderate</td>
<td>&lt;= moderate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 presents the average impact of psychobehavioral targets established from specific rules. For each target, we identify vignettes that are impacted by this target. The two targets with the largest impact are those included in the premises of rules r1 and r5 and they drive the selection of psychobehavioral strategies. The target indicat-
ed in r1 is a positive target. It states that making a patient be in charge of his/her health and limit drinking or smoking to light or none should improve her/his adherence level to good. This target impacts 6 vignettes from Table 3. The target indicated in r5 is a negative target. It states that increasing drinking or smoking intake to heavy can deteriorate a patient’s adherence level to poor and it impacts 8 vignettes.

There are 3 vignettes that are not impacted by any psychobehavioral target – v4, v6 and v12. Vignettes v6 and v12 do not match the sociodemographic context specified in rule r1. The intervention associated with the target from r1 could still be applied to those patients; however, there are no rules that would explicitly predict success in such a case. Moreover, v4 would have been handled by r2, however, this rule was discarded, thus according to currently considered features, no psychobehavioral target can be specified for this patient.

### 3.3 Selected Psychobehavioral Interventions

During the analysis described above, we identified and selected two psychobehavioral targets that are used for intervention strategies. The positive target from rule r1 calls for an intervention that aims at limiting (or stopping) drinking or smoking and making a patient be in charge of his/her health. This intervention would result in improving the adherence level to good. On the other hand, the negative target from rule r5 calls for an intervention that keeps the smoking or drinking to at most the moderate level. It should maintain adherence at the moderate or good level.

Below we provide more detailed insights into possible interventions that are suggested by our framework for these two targets – the final selection is limited to these suggestions. For each intervention, we describe its components (see Section 2.3): patient education, and behavioral change and self-reporting. These descriptions are formulated initially on a high level and will be turned into actionable prescriptions when the selected interventions are implemented.

<table>
<thead>
<tr>
<th>Table 3. Characteristics of identified psychobehavioral targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule/target</td>
</tr>
<tr>
<td>Average impact [%]</td>
</tr>
<tr>
<td>Impacted vignettes</td>
</tr>
<tr>
<td>v1</td>
</tr>
<tr>
<td>v2</td>
</tr>
<tr>
<td>v3</td>
</tr>
<tr>
<td>v4</td>
</tr>
<tr>
<td>v5</td>
</tr>
<tr>
<td>v6</td>
</tr>
<tr>
<td>v7</td>
</tr>
<tr>
<td>v8</td>
</tr>
<tr>
<td>v9</td>
</tr>
<tr>
<td>v10</td>
</tr>
<tr>
<td>v11</td>
</tr>
<tr>
<td>v12</td>
</tr>
</tbody>
</table>
The intervention for the first target (from rule r1) focuses on limiting smoking and drinking and on improving patient’s willingness to be in charge of his/her health through therapy. Patient education should focus then on links between heavy/moderate smoking or drinking and AF health outcomes, and on the consequences of not being in charge of one’s health for adherence. Teaching materials should highlight such issues as increased risk of stroke resulting from not following prescribed anticoagulation therapy. They should also provide instructions that help the patient increase self-efficacy and autonomy in taking medications (e.g., by giving a choice when to take them within a prescribed regimen) and limiting alcohol or tobacco consumption. Finally, this material should also help the patient see and understand how the proper management of AF can improve his/her life.

The behavioral change component should start with identifying barriers to change (e.g., a habit of smoking after a meal). Then, it should support setting moderately challenging but specific goals (e.g., keeping the number of cigarettes to below 5 per day). To increase goal commitment, public announcement of patient’s goals should be used. As part of self-reporting, a patient should be prompted to report on these goals at meaningful intervals (if the goal is daily, then daily prompt). Moreover, the feedback on goal attainment should be provided to encourage the patient when goals are met and to educate him/her on consequences if otherwise. This feedback can be enhanced by using peer pressure, for example realized as “hall of fame” leaderboards. Prompts and feedbacks should be relatively frequent to provide an accurate insight into patient’s current behavior.

The intervention for the second target (from rule r5) focuses on keeping the patient on the right track and on maintaining his/her drinking or smoking at a reasonable (i.e., lower than heavy) level throughout his/her therapy. The teaching component can be simplified to demonstrate positive consequences of the current good behavior so it is further reinforced. Behavior modification and self-reporting should also include setting goals and prompting for reporting on their attainment. However, the frequency of prompts should be lower than those associated with r1 so that the patient, who meets these goals does not get irritated by too frequent prompts (e.g., an occasional light smoker required to report on smoking habits daily).

Finally, both interventions should be expanded with two types of prompts: real-time and daily ones. The former should be triggered when it is time to take medications and the latter should be triggered to collect a daily “check in” on therapy adherence, symptoms and events that might contribute to increased risk for bleeding.

4 Related Work

The complexity of adherence comes from an interplay of many factors related to the patient, therapy, disease, social and economic status, and healthcare system [3]. Much research has focused on detecting medication adherence and non-adherence using machine learning, see for example [17]. However, less attention has been paid to discovering psychobehavioral targets strongly correlated with adherence or non-adherence. Son et al. [18] applied support vector machines to analyze self-reported
questionnaire data about medication adherence in heart failure patients. They found that gender, education, monthly income, daily frequency of medication, medication knowledge, and Mini-Mental Status Examination were important predictors of adherence. Nordmann et al. [19] identified non-compliant glaucoma patients using a Bayesian network that operated on an eye-drop satisfaction questionnaire. They found that age, self-declared compliance, and patient satisfaction with the patient-physician relationship were directly associated with adherence.

When psychobehavioral targets are identified, interventions that focus on targets can be deployed with the aim to improve or maintain adherence. Interventions that have been used in healthcare rely on behavioral economic principles (see Section 2.2) [10]. For example, social influence using a buddy system was successfully used for smoking cessation [20]. The same study also employed peer pressure in the form of feedback on the patient’s performance and the mean performance of her/his peers, and publishing leader boards. The principle of voluntary commitment by investing a sum of money that would be returned to patients when they have met their goal was used in a smoking cessation program and was proven to be effective [10]. Moreover, weekly lotteries were used to increase the motivation for weight loss and to encourage stroke victims to take their warfarin medication [10].

Adherence can also be increased by reminder systems [21]. These systems can be combined with active choice principles by explaining to patients what they would lose by not receiving reminders – such an approach was effective for promoting vaccinations against the flu [22]. Moreover, efficacy of reminder systems can be optimized by combining them with alternative intervention strategies. For example, Farris et al. [23] used reinforcement learning to automatically adapt and tailor SMS communication to hypertensive patients over time and as patients’ statuses and circumstances changed. Another useful tool for increasing adherence and measuring it is self-reporting. For example, the successful promotion of physical activity behavior was reported by [24] using techniques such as mobile journaling.

5 Conclusions

In this paper, we present a framework for helping a patient’s adherence to therapy by identifying psychobehavioral targets that drive the identification of interventions. We describe how a DRSA is applied to automatically derive such targets from patient data and to evaluate their impact. These targets drive the selection of psychobehavioral interventions aimed at maintaining or improving adherence. We illustrate our framework with an AF management case study, and show how using this framework allows us to derive psychobehavioral targets and associate these targets with psychobehavioral interventions.

The presented case study shows that DRSA is effective at both reducing a patient’s features to a minimal set that sufficiently discriminates between adherence classes, and at providing actionable psychobehavioral targets for interventions in the context of AF. Our research helps define the connection between the targets and psychobehavioral interventions. Moreover, the use of DRSA in combination with psycho-
behavioral interventions is applicable to patient management for other diseases as the presented approach does not make disease-specific assumptions.

We are currently working on integrating the proposed decision support framework with the architecture of the motivational patient assistant (MPA) system. The MPA system is meant to enhance the functionality of the MobiGuide system [25] with support for therapy adherence. We are also preparing a version of the MPA specialized for AF (AF-MPA) and we plan to evaluate it in a pilot study involving physicians (cardiologists and hematologists) and patients with AF. The patient evaluators will be asked to assess the AF-MPA on whether the system helps them feel more autonomous and competent in managing AF, whether the proposed interventions increase their quality of life, and whether the interventions help patients adhere to therapy. The physician evaluators will assess whether the AF-MPA helps their patients along the same dimensions, but also whether the interventions are medically sound, given their expertise with AF and the patient population. Such an evaluation will allow us to validate and improve the proposed interventions and help us achieve our ultimate goal of providing a comprehensive decision support system with “end-to-end” support from the primary care physician's office to the patient’s home [26].

In the longer term we plan to use the proposed framework for other diseases, including hypertension and chronic kidney disease, to demonstrate its ability to identify interventions beyond those for AF.

Acknowledgement. The first author wishes to acknowledge partial support by the Polish National Science Center under Grant No. DEC-2013/11/B/ST6/00963 and by the Institute of Computing Science Statutory Funds.

References

26. Michalowski, M., Michalowski, W., O'Sullivan, D., Wilk, S., Carrier, M. AFGuide system to support personalized management of atrial fibrillation. in 2017 Joint Workshop on Health Intelligence (W3PHIAI 2017).
A general framework for the distributed management of exceptions and comorbidities

A. Bottrighi\textsuperscript{1}, L. Piovesan\textsuperscript{1}, P. Terenziani\textsuperscript{1}

\textsuperscript{1} Computer Science Institute, DISIT, Univ. Piemonte Orientale, Alessandria, Italy
{alessio.bottrighi, luca.piovesan, paolo.terenziani}@uniupo.it

Abstract. Many different computer-assisted management systems for Computer Interpretable Guidelines (CIGs) have been developed. While CIGs propose evidence-based treatments of “typical” patients, exceptions may arise, as well the need to cope with comorbidities. The treatment of such phenomena has attracted a lot of attention in the recent literature. However, in this paper, we propose the first approach which supports the integrated treatment of both exceptions and comorbidities. To achieve such a goal, we propose a modular architecture supporting the concurrent execution of multiple guidelines.

Keywords: computer interpretable guideline (CIG), concurrent execution of multiple CIGs, CIG-exception handler, system architecture

1 Introduction

Clinical Practice Guidelines (CPGs) represent the current understanding of the best clinical practice. CPGs are gaining a major role to improve the quality and to reduce the cost of health care. The ICT technology can further enhance the impact of CPGs. Many different systems have been developed to manage Computer Interpretable clinical practice Guidelines (CIGs for short). A comparison of Asbru, EON, GLIF, Guide, PROforma, PRODIGY can be found in [1]. [2] extends it to consider also GLARE and GPROVE. [3] is a recent survey of the state-of-the-art.

One of the main goals of CPGs and CIGs is to capture medical evidence and to put it into practice. However, from one side, evidence is essentially a form of statistical knowledge, and it is used to capture the generalities of classes of patients, rather than the peculiarities of a specific patient. From the other side, demanding to expert committees the elicitation of all possible executions of a CPG on any possible specific patient in any possible clinical condition is an infeasible task. Thus, several conditions are usually implicitly assumed by experts building a CPG: (i) ideal patients, i.e., patients that have only the disease considered in the CPG (thus excluding the concurrent application of more than one CIG), and (ii) “statistically relevant” patients not presenting rare peculiarities/side-effects; (iii) ideal context of execution, so that all necessary resources are available. However, when a specific physician applies a given
CIG to a specific patient, unexpected conditions may show up. Such situations are unexpected, and, as such, cannot be specified a priori in the CIGs. However, especially in case of unexpected life threatening problems, the physician must start soon to cope with the new problems (possibly suspending or ending the “standard” execution of the current CIG, or concurrently with it). Such problems have been usually indicated with the term “exceptions” within the CIG community, since they are exceptions with respect to the “standard” execution of a CIG.

Another challenging problem that might involve deviations from the “standard” execution of a CIG is the treatment of comorbid patients. The problem is that, by definition, CIGs address specific clinical circumstances (i.e., specific pathologies), and, unfortunately, in comorbid patients the treatments of single pathologies may dangerously interact with each other. Also, the approach of proposing an ad-hoc “combined” CIG to cope with each possible comorbidity does not scale up [4]. For these reasons, new methodologies have been recently introduced to manage multiple CIGs on comorbid patients (see, e.g., the survey in [5]).

1.1 Related works

Within the CIG community, several frameworks have been already proposed to cope with “exceptions” (see, e.g., [6–11]). In most of such approaches, the “standard” executor of a CIG is extended with some mechanism to trigger exceptions (on the basis of the patient’s data) and to activate their treatment, synchronizing such a treatment with the execution of the current CIG. Different mechanisms of synchronization have been proposed.

On the other hand, a range of different technical solutions have been proposed to cope with comorbidities, spanning from the use of constraint logic programming [4] to answer set programming [12], from rules [13] to agents [14]. Notably, some of such approaches focus on the automatic generation of a unique “merged” CIG avoiding the undesired CIG interactions (consider, e.g., [13, 14]). In such a way, a “standard” CIG executor can be used to enact the “merged” CIG. However, in the clinical practice, (1) there are usually different ways to manage interactions, and physicians want and must (for ethical reasons) be the protagonists of such a decision. And, more importantly for the current work, (2) though in clinical practice the interactions between CIGs are managed, physicians do not look at the solution as a single “merged” CIG: they still look at the treatment of a comorbid patient as the concurrent execution of multiple CIGs.

1.2 Goals and original contributions

Until now, within the CIG literature, exceptions and comorbidities have been treated as separate phenomena\(^1\), so that current approaches cope either with exceptions, or with comorbidities. This is a clear limitation of the state of the art,\(^2\)

---

\(^1\) This choice is, in our opinion, quite surprising, since there does not seem to be a clear cut between the two phenomena. Just as one prototypical example, in [6] heart failure is considered as an “exception” for a patient treated with a CIG for trauma. But, when a patient with a trauma manifests a heart failure, she becomes a comorbid patient, and attention must be paid to avoid dangerous interactions between the treatment (CIG) for the trauma and the treatment (CIG) for the heart failure.

---

68
since both phenomena may co-occur on specific patients. In this paper, we first propose a CIG approach facing both phenomena, thus overcoming such a relevant limitation of the current literature. Additionally, unlike [13, 14], we aim at maintaining the distinction between CIGs, even in the case of comorbid patients.

Our goal is to propose the first general framework that copes with both exceptions and comorbidities. The starting point of our approach is that the management of patients affected by multiple problems requires

(1) A support for the concurrent and distributed (i.e., carried on by different agents) execution of CIGs, and to synchronize them. In the case of comorbid patients, it will support the execution of one CIG for each one of the patient’s diseases; in the case of an “exception”, it will support the execution of the original CIG plus the plan (which can be formalized as a CIG) to manage the exception.

(2) A support for detecting the possible interactions between such concurrent CIGs, and for managing them (avoiding dangerous interactions)

(3) A support for detecting new patient’s problems (i.e., changes in the status of the patient that require new treatments – thus, new CIGs).

In the rest of the paper, we propose the first CIG framework providing all such supports, in an integrated way. While (1)-(3) are the main goals of our approach, we also take into account a further, more “technical” goal. The specialized literature proposes several “consolidated” execution engines, to execute a CIG for a specific patient. Many of such approaches (including GLARE and its recent extension META-GLARE [15, 16]) have achieved good results, providing physicians with friendly environments to execute CIGs. We think that it would really be a pity to waste such an amount of good work, building from scratch a new concurrent execution engine. Thus, an additional main goal of our approach is that of devising a modular approach for the concurrent execution, in which the execution engine of a CIG in isolation is maintained, and it is extended and integrated in a general framework supporting synchronization and concurrency. Notably, although we are building our framework on top of META-GLARE (in the sense that each “Exec” module -see Fig.3 below- is an instantiation of META-GLARE execution module), our methodology is general and can be adapted for similar CIG systems (such as, e.g., [7, 17]).

Furthermore, notice that the framework we have developed also supports the fact that multiple healthcare agents may be involved in the execution of each single CIG for a specific patient. We cope with multiple agents using the methodology we have already published in [18] and generalizing it to the context of multiple CIGs. For the sake of brevity, such a topic is no further discussed within this paper.

Notably, the main contribution of this paper is the definition of (the architecture of) a system-independent framework for the distributed management of exceptions and comorbidities. We are currently implementing a prototype of the framework on top of META-GLARE and of our module for managing the interactions [19].

2 A general view of the behavior of our framework

While the architecture of our framework is proposed in Section 3, here we informally discuss the basic data/knowledge sources (ovals in Fig. 1) managed by our framework, and its general behavior (see Fig. 1). First, though in the literature (see,
e.g., [6, 11]) different types of exceptions have been identified, for the sake of brevity in this paper we focus only on the most “common” ones, i.e., on the exceptions arising because of unexpected changes in the status of the patient, requiring a (new) treatment (i.e., with CIG-independent patient-exceptions, in the terminology in [6]).

Second, it is important to clarify that there is a main “practical” difference between the management of exceptions and the one of interactions (e.g., the interactions that may arise between actions of different CIGs operating on the same patient). Exceptions cannot be avoided: the status of the patient has already changed, and an exception arises because of such a change (i.e., it is triggered by the new status of the patient). Moreover, of course, the exception must be managed, usually by starting a new treatment (which, indeed, may be represented by a CIG) for it.

In our approach, the treatment of exceptions is modeled by a KB (called “Exception KB”; see Fig. 1) consisting of triggering rules of the form <Condition, Manag>, where “Condition” indicates a Boolean condition (called “triggering” conditions) on the status of the patient, and “Manag” represents the actions to cope with such a condition (representable as a new CIG) plus constraints about how such new actions have to be synchronized with (the execution of) the current CIG(s). Notably, in our approach, the “Exception KB” is static, in the sense that the rules it contains are patient-independent, and are permanently stored.

On the contrary, undesired interactions should be detected a-priori and avoided (through some management operation). In our previous work we have provided a framework (see the Interaction Analysis module in Fig. 1; rectangles represent computational modules) for (i) supporting physicians in the focusing on specific parts of the CIGs² [20], (ii) automatically detecting (based on CIG-independent ontological knowledge) the interactions between the focused actions [21], as well as a suite of management options (derived from the medical literature) to manage them (e.g., avoiding them by delaying some actions, or managing them through local modifications of the involved CIGs [22]; see the Management Definition module in Fig. 1). Although the analysis of interaction should be performed a priori, generally the management option chosen by physicians has not to be enacted soon. Indeed, it had to be enacted only if and when, during the execution of the CIGs, the conditions identifying the onset of the interactions arise³. As a consequence, the treatment of interactions may be modeled by a KB (called “Interaction KB”; see Fig. 1) containing <Condition, Manag> pairs. However, differently from the rules for exceptions discussed above, here:

(i) triggering conditions have as input the status of execution of the CIGs

² CIGs may consist of hundreds of actions and/or alternative paths. An extensive check of all interactions could provide a combinatorial number of cases, most of which are not interesting for the patient at hand. Physician-driven focusing is an essential step to avoid an unnecessary combinatorial explosion of the computation and of the number of the identified interactions.

³ As an example, a possible undesired interaction between the actions Act1 in CIG1 and Act2 in CIG2 can be detected and physician can choose to manage it via the substitution of Act2 with a set of actions achieving the goal of Act2, but non-interacting with Act1. However, such a substitution must be performed only in case the execution of the two CIGs enforces the execution of both Act1 and Act2 (at times such that their effects may overlap in time). Indeed, if in CIGs a path of actions not including Act1 has been selected for execution, there is no need to substitute Act2.
(ii) “Manag” indicates the operations to implement the management option chosen to cope with an interaction (e.g., operations to locally modify a CIG).

The “Interaction KB” is dynamic: the system dynamically add rules into it whenever a new interaction has been detected and a management for it has been chosen, and it deletes such rules when they are not useful any more. Indeed, there would be no reason to permanently store such rules, since they are specific to the execution of a set of CIGs for a specific patient.

In Fig. 1, we show that a unique manager, the “Trigger Manager” can uniformly operate on both types of rules. Fig. 1 also shows that the output of the execution of an “Exception” rule may be the activation of a new CIG, while the result of the execution of an “Interaction” rule may usually be a modification of some CIGs. It also shows that, when multiple CIGs are active, the Interaction Analysis module may be used to analyse possible interactions, and the Management Definition module can support the management of such interactions. The output of the Management Definition module is a new <Condition, Manag> rule, dynamically stored in the Interaction KB. While Fig. 1 graphically illustrates the behavior of our framework, in Section 3 we discuss the architecture we have identified to achieve it, and considering the fact that the management of exceptions and interactions may involve forms of synchronizations between the different CIGs (possible synchronizations between CIGs have been omitted, for the sake of clarity, from Fig. 1).

![Figure 1. Graphical representation of our treatment of exceptions and interactions.](image)

2.1 Case study

In this section, we present a “synthesized” case study, which has been created with the help of some physicians of Azienda Ospedaliera “San Giovanni Battista” in Turin in order to be able to exemplify the main features of our approach. We consider a comorbid patient, who is treated for Peptic Ulcer (PU) and for deep Venous Thrombosis (VT) and has a heart failure (i.e., an “exception” arises) during the execution of these CIGs. The two diseases are managed by two specific CIGs (the upper part of Fig. 2 shows simplified parts of the CIGs). Besides the CIGs, additional medical knowledge is available, including the trigger for exceptions. In our example, among them, we consider the exception for heart failure (notably, in this context, heart failure can be considered an exception: it is not statistically recurrent in PU and VT, thus its treatment is not contained into the original CIGs).
In our example, the CIGs for PU and VT are executed concurrently by two different physicians: Physician\textsubscript{1} manages PU, and Physician\textsubscript{2} manages VT. We consider a sample working section articulated as follows. **Step 1.** We suppose that Physician\textsubscript{1}, at a certain point of execution of PU and VT (e.g., at the beginning), decides to analyze the possible interactions between the two CIGs. To do so, Physician\textsubscript{1} exploits the Interaction Manager module (see Fig. 1) focusing on relevant parts of PU and of VT, and the Interaction Manager module detects an interaction between warfarin therapy (WT) in VT and amoxicillin therapy (AT) in PU. **Step 2.** Physician\textsubscript{1} chooses to manage such an interaction replacing the action WT with an alternative plan, having the same goal. For instance, the new therapeutic plan may be the combination of acetylsalicylic acid (AA) therapy and omeprazole (OT) therapy. As we will see in the Section 4, the Interaction Manager module creates a trigger rule to implement such a management (if when required). **Step 3.** Physician\textsubscript{1} and Physician\textsubscript{2} goes on with the independent executions of the CIGs. We suppose that in PU “PU start”, “H.Pylori test”, “H.Pev” with exit “positive” has been executed; in the meanwhile, in VT “VT start”, “intervention decision”, with exit “pharm”, and “AntiD” with exit “a” has been executed. **Step 4.** At this point, the chosen management of the interaction is required, and is executed (i.e. the trigger rule created at step 2 above is executed, thus modifying the CIG as shown in the lower part of Fig. 2). **Step 5.** The execution continues on the modified CIGs. **Step 6.** To exemplify all the main features of our approach, we further suppose that at this point the patient has a heart failure, and we show how our framework supports its treatment.

![Diagram](image)

**Figure 2.** Part of PU and VT original CIGs (above) and the updated version of VT after the managing of the interaction (below).

3 Architecture for the concurrent execution of multiple CIGs

3.1 The architecture

Our approach is based on the client-server model. This choice is motivated by the need (i) to support a distributed execution of the patient treatments, since different
CIGs can be managed by different physicians (i.e. each physician needs a client to manage her CIGs) and (ii) to have a global vision of patient treatments, and to “synchronize” them (such a vision will be stored and managed in the server). Notably, from an abstract point of view our approach can be described as an agent based system (i.e. each module can be seen as an agent).

For sake of simplicity, in this paper we assume that all the CIGs are related to the same patient. The extension to cope with more than one patient is obvious. We propose a server model (see Fig. 3) composed by the following modules:

- a “General Manager” (in the middle of the “Server” in Fig.3): it maintains the global vision of the patient and of her treatments (i.e. global data structures). It interacts with the other modules to update such a vision and to synchronize them (the functionalities of such a module are described in more detail in subsection 3.2);
- the “Executor Modules” (“Exec CIG1”, “Exec CIG2”, and “Exec CIG3”, in the left part of the “Server” in Fig.3; notably, there is one “Exec” module for each CIG under execution for the patient). Each Executor manages the execution of a CIG for a specific patient [15] (see subsection 3.3);
- the “Interaction Manager” (top right part of the “Server” in Fig.3): it supports the study of the interactions between CIGs and defines how they should be managed (see subsection 3.4);
- the “Trigger Manager” (bottom right part of the “Server” in Fig.3): it manages the triggers in KBs (see subsection 3.5).

Notably, the architecture of our framework is open. It is possible to add the new modules to provide new facilities, by specifying their communication API (i.e. how they communicate with the other modules, the patient’s DB and the client).

The client provides physicians with a GUI to support the execution of one or more CIGs (e.g. in Fig. 3 Client allows to manage the execution of CIG1 and CIG2, while Client supports the execution of CIG3). Each client sends/receives messages to/from the executor module to manage the execution of the CIGs. Moreover, physicians can activate the interaction module to study possible interactions between two or more CIGs (e.g. in Fig. 3, Client activates it).

3.2 The General Manager module

The General Manager is the core of the system, since it supports the concurrent execution of CIGs on a given patient. To achieve such a goal, it manages the interplay between the other modules in the server by (i) sending/receiving messages, and (ii) maintaining two data structures to provide a “global vision” of the execution of the CIGs: (i) the graph of CIG dependencies and (ii) the yellow pages of CIGs. Such data structures work as a shared memory, where all the modules have the read permissions, while the General Manager has also the write permission.
The graph of CIG dependencies has two components: nodes and arcs. Each node represents one CIG under execution (for the given patient). The arcs represent the dependencies between such CIGs. An arc starting from a node A and ending into a node B means that B must be suspended by the execution of A. Thus, the graph represents the synchronization between CIGs: CIGs without entering arcs are active, while CIGs reached by an arc are temporarily suspended. We provide a set of primitives to update the graph: creation/deletion of a node, creation/deletion of an arc.

The yellow pages of CIGs store all the instances of CIGs currently in execution. The updates to the data structures are triggered by messages sent by the other modules in the server. A message represents a list of instructions expressed using the primitives described above. The General Manager manages messages as transactions, i.e., units of work performed in an atomic way. It performs all the updates required and then it notifies such updates to the modules to maintain the synchronization. The General Manager manages the message of updates using a FIFO policy.

3.3 The Executor module

The Executor module manages the execution of a CIG instance for a specific patient. In our approach, there is an instance of Executor for each CIG under execution. The Executor of a CIG can be active or suspended depending on the current state (i.e., active/suspended) of the CIG represented in the graph of dependencies. Each instance of Executor reads (i) the instance of the CIG that it has to execute from the yellow pages, and (ii) the patient data from the Patient DB. The Executor interacts with a specific client to execute the current actions in the CIG. In case the CIG is terminated, the Executor sends a message to the General Manager, to remove the node (representing the CIG) from the graph, and remove the CIG from the yellow pages. Specifically, we use the executor of META-GLARE [16], but our
methodology is mostly system-independent, and it can be adapted for use any CIG executor (such as, e.g., PROforma [7] or Asbru [17]).

3.4 The Interaction Manager module

The Interaction Manager module supports the detection and the definition of management for CIG interactions. It is composed by two modules (see Fig. 3): the Interaction Analysis and the Management Definition. The Interaction Analysis module (see [21]) operates in two steps. First, it provides physicians with a navigation tool (operating at the different abstraction levels supported by the given CIGs) supporting the choice of a specific part (called “focus”) of the CIGs, the part currently of interest for the treatment of the current patient. Second, it provides a knowledge-based tool that automatically detects all the possible interactions between the actions in the “focus”. Moreover, this module has been recently extended with a set of facilities to temporally analyze interactions [(19)], distinguishing among temporally certain, possible or impossible interactions and performing hypothetical reasoning. Once detected an interaction, the Management Definition module [22] supports physicians in the selection of a management, choosing among different modalities (i.e., the management options in [22]). Notably, in our approach, managements are not applied immediately to CIGs, but through the creation of dynamic trigger rules (see the discussion in Section 2). The triggers have the form <Condition, Manag> where “Condition” indicates a Boolean condition on the execution of specific CIG action(s) or decision result(s), and “Manag” represents the actions to cope with such a situation. These actions can be described using a subset of primitives to operate on the global data structures (see Section 3.2). Such trigger rules are automatically generated by the by a specific component of the Management Definition module (the “Trigger Generator”, not detailed in Fig.3 for the sake of brevity and clarity). The “Trigger Generator” takes as input from the other modules the detected interacting actions, and the management options chosen to manage such an interaction, plus additional parameters. The Trigger Generator consists of a set of parametric procedures, one for each management option, to automatically generate a trigger, on the basis of the input parameters. Then, the trigger is sent as a message to the Trigger Manager.

For example in section 4 we show the trigger created by the Interaction Manager module to manage the interaction between WT and AT (see section 2.1).

3.5 The Trigger Manager

The Trigger Manager module manages the triggers. To achieve such a goal, it has (i) to check whether the triggers stored in the KBs fire and then to notify that the management had to be applied and (ii) to maintain up-to-date the Interaction KB, since it is a dynamic KB (see details in Section 2).

To cope with (i), the Trigger Manager evaluates whether a rule in the KBs had been executed. The form of rules is <Condition, Manag> (see Section 2) and the Trigger Manager checks whether Condition is true or not (i.e. the patient status retrieved in the Patient DB or the execution status of the CIGs retrieved in the yellow pages satisfy Condition). If Condition is true, the Trigger Manager sends a message to the General Manager. Such a message contains Manag (i.e. the set of instructions to cope with the situation described in Condition).
To cope with (ii), the Trigger Manager adds a trigger to the Interaction KB, when it receives a message from the Interaction Manager module. Each message contains a trigger that has to be added. The Trigger Manager manages the messages using a FIFO policy.

The triggers in the Interaction KB are not permanent, since they are context and patient dependent. Thus, the Interaction Manager removes a trigger: (i) when it is used (in the case that it is not reusable, e.g. in the case it is applied to a repeatable part of the CIGs, it is removed when the repetitions of such a part is ended) or (ii) when one of the CIGs in its Condition ends.

4 Our system in action: managing the case study

We describe how our framework works on the case study described in subsection 2.1. The patient is affected by both Peptic Ulcer (PU) and deep Venous Thrombosis (VT) and two CIGs are executed to treat such diseases. Two physicians are involved: Physician1, managing PU, and Physician2, managing VT. In our framework, each physician interacts with the system via a client: (1) Physician1 uses ClientPU to execute the CIG PU via the executor instance ExecutorPU, and (2) Physician2 uses ClientVT, to execute the CIG VT via the executor instance ExecutorVT. Suppose that both physicians are managing the first action in the CIG (but this is not restrictive at all). In such a context, the graph of dependencies contains two independent nodes (one for PU and one for VT), while the yellow pages contain the current instances of the CIGs. The Interaction KB is empty and Exception KB contains the triggers to manage the exceptions. In our example, among the others, the trigger TR-HF (i.e. the trigger for heart failure) is stored in the Exception KB:

\begin{verbatim}
TR-HF:
    (1) <(Heart Failure = TRUE),
    (2) (ADD_NODE HF-PLAN TO GRAPH;
    (3) ADD HF-PLAN TO YELLOW PAGES;
    (4) ADD_ARC from HF-PLAN to VT;
    (5) ADD_ARC from HF-PLAN to PU;)
\end{verbatim}

In TR-HF, the Condition (line 1) captures that the patient has a heart failure, the Manag (lines 2-5) describes the instructions that must be executed to manage it. In short, (2)-(5) encode the commands to activate a new CIG “HF-PLAN” suspending the execution of VT and PU.

To analyze the possible interactions between the two CIGs, Physician1 (through ClientPU) calls the Interaction Manager module and selects the relevant part of CIGs that the module has to analyse (i.e., the “focus”). The Interaction Manager identifies all the interaction between the actions in the “focus” via the Interaction Analysis module. In this specific example, the Interaction Manager module finds an interaction between warfarin therapy (WT) and amoxicillin therapy (AT). Such an interaction increases the anticoagulant effect of warfarin and raises the risk of bleedings. As described in subsection 2.1, Physician1 decides to apply the replanning management option [22], substituting WT with an alternative new plan. Such a new plan is automatically generated by the Management Definition module (as described in [23]).

76
In our example, the new therapeutic plan is the combination of acetylsalicylic acid (AA) therapy and omeprazole (OT) therapy.

Then the Trigger Generator is invoked. It takes as input the PU and VT CIGs, the management option chosen by Physician: (i.e., the replanning option) and the new alternative plan, and (automatically) produces as output the trigger rule TR-WTAT described below. The Condition part of TR-WTAT represents the conditions under which the interaction can occur. In particular, in our example, AT and WT interacts in case (line 1): (i) the decision AntiD has been taken, having as result to execute the path “a” which contains WT, and (ii) either the decision HPev has been taken with result “positive” (i.e. choosing the path containing AT), or HPev has not been already executed (in this last situation, the system prevents the management, avoiding the cases in which decision HPev is taken with “positive” result only after WT has been executed, impeding the application of the management). Notably, such a condition is automatically built by the Trigger Generator, through a navigation throughout the PU and VT CIGs. The Manag part of TR-WTAT is automatically built by the Trigger Generator on the basis of the management option chosen by Physician and the new alternative plan. Specifically, the Manag part of TR-WTAT prescribes to (line 2) remove WT, and (lines 3-4) to add AA, OT and (lines 5-6) the corresponding arcs in the CIG VT (the result of the execution of TR-WTAT is shown in Fig. 2).

**TR-WTAT:**

1. \(<(\text{Exec(AntiD)})=\text{a} \ \text{AND} \ \ (\text{Exec(HPev)})=\text{positive} \ \text{OR} \ \text{NOT} \ \ \text{Exec(HPev)})>\)
2. (remove action WT in VT)
3. ADD ACTION AA to VT;
4. ADD ACTION OT to VT;
5. ADD ARC in VT from AntiD to AA;
6. ADD ARC in VT from AA to OT;)

Then, the Interaction Manager module sends a message containing the TR-WTAT rule to the Trigger Manager.

As a consequence, the Trigger Manager adds it to the Interaction KB (see Fig. 2). Then, the two physicians can independently go on with the execution of the CIGs.

For instance, suppose that Physician: (through Client\(v\)) has executed the actions “PU start”, “H.Pylori test”, and “HPev”, which results positive; in the meanwhile, Physician: (through client Client\(v\)) has executed “VT start”, “intervention decision”, with exit “pharm”, and “AntiD” with exit “a”. This situation triggers TR-WTAT (i.e. Condition in TR-WTAT is satisfied). Thus, the Trigger Manager sends a message to the General Manager containing the instruction to manage such an interaction (i.e. the Manag component in TR-WTAT, i.e. lines 2-6) and removes TR-WTAT from the Interaction KB, since it is not reusable during the patient treatment.

Then, the General Manager executes as a unique transaction the instructions in the message, updating the global vision. In our example, the instance of VT in the yellow pages is updated by replacing WT with the alternative plan (see lines 2-6 in TR-WTAT), as shown in the lower part of Fig. 2. Thus, the General Manager notifies to Executor\(v\) that the instance of VT in the yellow pages has been updated. As consequence, Executor\(v\) sends a message to Client\(v\) to refresh the visualization of VT, and let Physician: go on with the execution of the updated CIG.
Moreover, let us suppose that, during the execution of such CIGs, the patient has a heart failure. As a consequence TR-HF is triggered by the Trigger Manager. Then the Trigger Manager sends a message to General Manager with the instructions to manage the heart failure (lines 2-5 in TR-HF). The General Manager executes these instructions. The first two instructions (lines 2-3 in TR-HF) generate (both in the graph of CIG dependencies and in the yellow pages of CIGs) the node corresponding to the CIG to treat heart failure. As a result of such a generation, our framework supports the search for a physician accepting the responsibility of executing the new CIG (following the approach in [18]), and generates a new instance of Executor module to manage the Heart Failure CIG. The selected physician can manage the execution of the CIG trough a client. In case s/he is already executing a CIG for the specific patient, the Heart Failure CIG is added to its client, otherwise a new client is initialized for her/him. Moreover, the interpretation of lines 4-5 in TR-HF adds two (suspension) arcs in the graph of CIG dependencies, then the General Manager notifies the suspension to ExecutorVT and to ExecutorMU. Consequently, the two executors notify the suspension to the corresponding clients.

5 Conclusions

Traditional CIG execution engines provide physicians with consolidated support for the execution of a single CIG on a single patient. However, the treatment of “exceptions” and of comorbid patients demands for more extended supports. In this paper, we provide the first homogeneous framework for the management of both “exceptions” and interactions. Our approach is modular, in that it adds a further layer building upon “traditional” execution engines for a single CIG. Though our framework is being built on top of META-GLARE, our methodology is general, and can be adapted for similar CIG systems (such as, e.g., PROforma [7] or Asbru [17]).

We are currently implementing our approach using Java. As soon as the implementation will be completed, we plan to develop an extensive experimentation of our framework, especially in the context of comorbidity treatment.

Acknowledgments

This research is original and has a financial support of the Università del Piemonte Orientale, and of INdAM.

References

A Label Taxonomy to Support Collaborative Formalization of Computer-interpretable Guidelines and Classification of Modeling Issues

Arturo González-Ferrer¹, María Ángel Valcárcel¹, Henar González-Luengo¹, Enea Parimbelli²

¹Fundación de Investigación Biomédica, Hospital Clínico San Carlos; Instituto de Investigación Sanitaria (IdISSC) San Carlos, Madrid 28040, Spain
²Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Italy

Abstract. Modeling Computer-interpretable Clinical Guidelines (CIGs) entails the need to involve a heterogeneous group of professionals in the tasks required, mainly knowledge engineers and physicians, but possibly others (e.g. nurses, patient associations). The workflow process followed by the team is diverse, depending on who raise an issue, demand a requirement, introduce expert criteria deviating from the guideline, solve a bug, detect a duplicate, etc. Follow-up of this process is one of the challenges, as responsibilities are fuzzy and traditional communication via skype meetings or e-mail sometimes is not efficient enough. In this position paper we describe a preliminary label taxonomy to support classification of modeling issues and to facilitate collaborative formalization of CIGs. The taxonomy was designed during a project where a PROForma CIG model for hyponatremia diagnosis and classification was formalized. Beyond tracking and organizing issues, the use of this taxonomy might facilitate analysis of modeling events and time efforts.

Keywords: computer-interpretable guidelines, facilitating knowledge-acquisition of healthcare

1 Introduction

Computer-interpretable Clinical Guidelines (CIGs) [1] are software-executable models that use Clinical Practice Guidelines (CPGs) as knowledge source. These CIGs are developed by knowledge engineering teams and require the use of software tools to define a formal specification of the model using one of the available CIG languages (e.g. ASBRITE, GLIF, GLARE, PROForma, etc.) [1]. This model is saved in the chosen format and later interpreted by the corresponding execution engine, enabling the real application of the model by care professionals and/or patients. The MobiGuide Patient Guidance System [2,3] is an excellent example of both the complexity that the development entails and the usefulness for patient follow-up.
The correctness of CIG models is critical to guarantee patient safety and ethic-legal requirements [4], since the model will eventually serve to provide evidence-based recommendations to physicians. Knowledge Engineers (KE) involved in the modeling process pursue achieving this correctness through many iterations of modeling, verification and validation steps that test the expected behavior of the system. Beyond this traditional way of CIG development, Shalom et al. developed a nine-step methodology for incremental collaborative markup of the CIG model and evaluating completeness and correctness [5], Goud et al. proposed concurrent development of CPG and CIG [6], while Tso et al. [7] recently extended the 13 steps that Shiffman et al. identified [8] to translate CPG knowledge for use in clinical decision support. Whatever methodology followed by the KE team during the modeling stage, but also during verification and validation, numerous meetings with the physician experts need to be carried out to cope with a number of situations (many of them associated to the decision categories described in [6],[8]): raise and solve questions, meet and clarify requirements that are not properly specified in the text of the CPG (e.g. thresholds for a lab result value involved in a decision condition), define expert criteria for the local settings, etc. Exposed to this problem during a recent project [9] for developing a CIG for diagnosis and classification of hyponatremia (low serum sodium), we designed a fifteen label taxonomy that permits to classify, possibly using multiple labels simultaneously, the situations raised during the modeling process. The rest of the paper describes this taxonomy, explore some statistics that we observed in our case and discuss the practical utility of such process.

Table 1. The CIG DRS label taxonomy. The third column indicates how much the label was used in our project. Note that an issue can be assigned multiple labels. The cell background color group the type of label (defect, stage, role).

<table>
<thead>
<tr>
<th>Label</th>
<th>Meaning</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug</td>
<td>A serious wrong behavior in the application (e.g. formulae calculation, clinically incorrect information, the model does not reflects the care process well, etc.)</td>
<td>11.1</td>
</tr>
<tr>
<td>Correction</td>
<td>This label is used when correcting a bug, closing the issue</td>
<td>6.4</td>
</tr>
<tr>
<td>Duplicate</td>
<td>If KE recall any problem previously recorded with same meaning, the issue is closed and marked with this label.</td>
<td>1.9</td>
</tr>
<tr>
<td>Enhancement Model</td>
<td>Used when an improvement is proposed by any KE or user over the current model. It is different of a bug because some subjacent reason exists to consider it an improvement (e.g. more appropriate, better informed, extra decision rules improving basic behavior, etc.), not a wrong behavior to be fixed.</td>
<td>26.9</td>
</tr>
<tr>
<td>Enhancement UI</td>
<td>Any KE or user finds out that the user interface (UI) should be more appropriate that the current one (e.g. presentation of data enquiries can be personalized in PROForma, messages may need some rephrasing, etc.)</td>
<td>9.3</td>
</tr>
<tr>
<td>Expert Criteria</td>
<td>It is used to identify aspects that were required by experts but not reflected in the original CPG. It can later be used to retrospectively</td>
<td>12.0</td>
</tr>
</tbody>
</table>
The CIG defect-role-stage label taxonomy

The design of the defect-role-stage (DRS) taxonomy was carried out considering three aspects: the nature of the defect, the role that was raising the issue and the stage during the development process. The former was directed to classify the type of problem, while the second and the later were directed to help revisiting the unsolved issues, identifying and prioritizing them visually and efficiently. We tried to find a balanced design on the granularity of our taxonomy, in order to keep relatively low number of labels for the sake of usability.
The concept of defect [10] in our case is different from typical software development: it is not uniquely related to the resulting CIG model, instead it is linked to the whole modeling process. For example, the “expert criteria” label could be interpreted not as a defect: if there was a lack of knowledge expressed in the CPG that is needed (e.g. a lab threshold), then that is considered a defect, i.e. it is a fail that impedes to properly build the model. Similarly, if the CPG does not reflect the need of the user, that needs to deviate from it to adapt the resulting product to its requirement, there is some failure during knowledge acquisition that needs to be dealt with.

Table 1 shows the labels. Note that they were designed for our project but different labels could prove more useful in other projects. For example, a CIG using temporal patterns in ASBRU [11] or GLARE [12] would probably add a “Time-related” label instead of “weight change”, which is more useful in PROForma projects. Regarding stage-related labels, as mentioned in the Discussion section, we need to decide if these label will stay in the base taxonomy, given they could be classified using milestones.

3 Use in our project

We developed a CIG for diagnosis and classification of hyponatremia using PROForma [13]. The model included 4 plans, 11 enquiries, 60 source data (variables), 10 data type definitions, 5 actions, 4 decision nodes, 19 associated candidates, 8 precondition formulas, 7 post-condition formulas and 218 arguments: 117 for arguments, 87 against and 16 neutrals. During the development we used the DRS taxonomy to classify more than a hundred issues in the issue tracking system of a Github repository created to collaborate. Github’s issue tracking system allows to replicate the taxonomy as colored labels (see Figure 1), whose assignment to an issue can be modified during its lifetime (i.e. remove an assigned label or assign a new label to the issue). Clicking on a label automatically filters all the issues labeled with it in the repository. This is useful for KEs to organize work and to make corrections according to the type of problem (e.g. revising all the "help wanted" items in the face-to-face meeting with the expert).

![Figure 1. Using colored taxonomy labels on project’s issue tracking system](image-url)
The three aspects reflected by our taxonomy, defect, roles and stage, were informative for the KEs and helpful when filtering and prioritizing solutions to make progress (e.g. the "expert demands" label is useful to put in first place physician requirements). Regarding the patterns of use, Table 1 shows the individual percentage in our 108 reported issues. The heatmap in Figure 2 shows a picture of how the labels were combined, assigning a ranged color that is darker when the labels were combined more often. Note that each cell contains the number of times the labels were combined pairwise. Cells with the same name in axis x and y count issues that were tagged only with that label. To explore combinations of three or more labels together, other graphical representation or analysis would be needed. For example, the “duplicate” label was only used two times, but it was combined with four labels, in groups of three: once it was combined with “enhancement model” and “final review”, while in other case appear combined with “expert demands” and “weight change”. These statistics were obtained with a R script1 that connects to the Github API, easing analysis of the taxonomy use in other future projects. Figure 2 shows a high number of the following situations: demands from experts to enhance the model, the UI, inclusion of expert criteria and the modification of weights, help wanted from KEs either in comprehension of the guideline (KE team question) or to improve the model. The “wontfix” label had zero occurrences, but some issues are still open without decision, so we foresee some will be tagged with that category.

<table>
<thead>
<tr>
<th></th>
<th>weight change</th>
<th>expert criteria</th>
<th>enhancement model</th>
<th>KE team question</th>
<th>final review</th>
<th>help wanted</th>
<th>expert demands</th>
<th>validation error</th>
<th>correction</th>
<th>bug</th>
<th>duplicate</th>
<th>invalid or discarded</th>
<th>enhancement UI</th>
<th>lack of information</th>
<th>documentation</th>
<th>enhancement model</th>
<th>help wanted</th>
<th>final review</th>
<th>KE team question</th>
<th>enhancement model</th>
<th>expert criteria</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>7</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

`Figure 2. Heatmap showing when labels were combined pairwise.`

1 R script available at: https://gist.github.com/arturogf/39ae3ca833ce8793ba9b577e692616825
4 Discussion and Future Work

We would like to catch the attention of the workshop audience, demonstrating that this type of methods may serve to improve our knowledge about how do we carry out CIG modeling, what are the archetypical problems occurring, where should we enforce teaching for new KEs approaching CIG modeling or how well do we solve issues in the model as the development life-cycle progress. While the taxonomy might be incomplete or inaccurate to reflect any possible pattern of use, we think that the idea might be further developed in collaboration with other members of the CIG community. Further discussion is needed about what labels should stay in a base taxonomy, which are optional depending on the CIG formalism or if the labels reflecting a stage of the process (e.g. final review or validation error), should be removed and treated as “milestones”. It is important to realize that the heatmap representation is just a possible way to show information from the process, but of course there are other interesting pieces of information in the repository accessible through Github's API. Thus, there is further need to identify in future work which information would be relevant to extract from the issue tracking system (e.g. temporal analysis of labels’ evolution through time, bug-closing times and which type of issue usually get priority when solving, etc.). Similarly, it would be interesting identifying information that could be useful at development time, beyond post-analysis (e.g. detecting requirements that are not being accomplished).

Besides extraction of post-development knowledge in our proof of concept project, the taxonomy served to organize the modeling process and to make the development agile. Our future plans include a validation of the taxonomy selecting a subset of issues – possibly extracted from multiple projects- described in natural text that will be tagged separately by several KEs, then calculating the agreement among the participants using appropriate inter-rater reliability measures (e.g. Krippendorf’s alpha).

Acknowledgments

This work has been possible due to partial funding provided by Otsuka Pharmaceutical Spain. The research was also supported by the "Plan Estatal de I+D 2013-2016", the Instituto de Salud Carlos III (ISCIII) under ITEMAS network, expedient number PT13/0006/0003 and the regional development program (FEDER).

References

Performance Analysis of Markov Chains-based Models in Process Mining: A Case Study

Roberto Gatta[1], Jacopo Lenkowicz[1], Mauro Vallati[2], Eric Rojas[3], Andrea Damiani[1], Carlos Fernandez-Llatas[4], Lucia Sacchi[6], Berardino De Bari[5], Arianna Dagliati[7], and Vincenzo Valentini[1]

1 Università Cattolica del Sacro Cuore, Italy roberto.gatta.bs@gmail.com  
2 University of Huddersfield, United Kingdom  
3 Universidad Pontificia Católica de Chile, Chile  
4 ITACA-Universitat Politecnica de Valencia, Spain  
5 Centre Hospitalier Régional Universitaire de Besançon, France  
6 Università degli Studi di Pavia, Italy  
7 IRCCS ICS Maugeri Pavia, Italy

Abstract. In the last decade, Process Mining has become a significant field to help healthcare process experts understand and gain relevant insights about the processes they execute. One of the most challenging question in Process Mining typically is: how good are the discovered models? Previous studies have suggested approaches for comparing the (few) available discovery algorithms and measure their quality. However, a general and clear comparison framework is missing, and but none of the analyzed algorithms exploits Markov Chains-based Models. In this paper, we discuss about quality and performance of discovered models. This is done by focusing on a case study, where the pMiner tool is used for generating Markov Chains-based models, on a set of real clinical guidelines and workflows.

Keywords: Process Mining, Markov Model, Models Evaluation

1 Introduction

In the past decade, Process Mining has become a significant field to help healthcare process experts understand and gain relevant insight about the processes they execute [10]. Process Mining is a relatively young research area that provides, through data from Hospital Information Systems (HIS) and Electronic Health Records (EHR), detailed models and information about the performed processes. The data extracted from the HIS/EHR are used to build an event log which includes the relevant information (activities completed, time stamps, case ids and the resources that performed those activities), to be used to discover the executed process model, allowing the detailed visualizations of the processes.

In this discipline, one of the most pressing question arising when a process model is discovered, is: how good is it?

For the aims of hospitals involved in this case study, Markov Models has been indicated by medical experts as a possible formalism for sharing results between
Markov Models in Process Mining: A Case Study

physicians and computer scientists. For this reason, we analyzed the (few) available algorithms that are able to discover processes in the healthcare domain, and that provide a way for measuring the quality of the generated models. However, none of the included and analyzed algorithms are Markov Chains-based Models based [2, 5, 9]. Actually, with the exception of the qualitative previously cited indicators, a shared framework to quantitatively evaluate performances in Process Discovery in healthcare is still missing.

The main aim of this work is to evaluate the usefulness of Markov Chain-based models for representing discovered models, and to propose a way for assessing the performance of models generated using such Markov-based approach. This is done by focusing on a case study, where the pMiner tool is used for generating Markov Chain-based models, according to real clinical guidelines and workflows. It should be noted that an extensive comparison of all the existing formalisms and techniques for discovering and representing processes mined in the healthcare context is beyond the scope of this paper.

2 Background

There are three main areas subsumed by process mining: process discovery, conformance checking, and enhancement. Buijs et al. [2] explain how automatic process discovery allows process models to be extracted from an event log; how conformance checking allows monitoring deviations by comparing a given model with the event log; and how enhancement allows extending or improving an existing process model using information about the actual process recorded in the event log.

Discovered models can be represented using different formalisms, including Markov chains [6], Petri networks [7], and BPMN [12]. These formalisms include specifications to model the processes and are included in a small amount of specific tools that apply the process mining techniques, such as PROM [11], DISCO [4], and PALIA [3].

Four metrics have been introduced in order to measure the quality of a discovered model: fitness, simplicity, generalization, and precision [10]. Fitness corresponds to how the discovered model can replay an event log, this can be computed by replaying all the traces in the event log in the model [10]. Simplicity indicates how simple a model can be for a human. Generalization can be defined as the capability of a model to deal with previously unseen behaviors and/or logs. Finally, precision gives a measure of how a model does not allow for too many additional (and not strictly related) cases.

2.1 Why R and Markov Models

In our investigations in Process Mining in medicine, we normally have an important constraint regarding the choice of the language and the platform. This is due to the fact that our multidisciplinary team also includes physicians who
cannot invest too much time in coping with complex formalisms or programming languages. Such constrain, in our case, was related to the use of R (which is widely used in our Department) and the adoption of models which can be easily understood from our physicians. In several internal investigations on the use of Process Mining, we decided to exploit FOMM/SOMM because the involved physicians expressed the opinion that such models are easier to understand than other models (e.g., Petri Networks or Fuzzy Miner). Because of that, and given the fact that the performance of SOMM and FOMM for Process Mining in Healthcare have not been yet extensively investigated, we focused our attention on them.

3 Methodology

We considered First Order Markov Models (FOMM) and Second Order Markov Models (SOMM), with different probability thresholds to accept arcs among nodes: a FOMM with threshold 0.2 is a FOMM where, in the transition matrix, probabilities below 0.2 are set to zero, and the remaining probabilities are renormalized accordingly, in order to have sum 1 on each row of the transition matrix. In terms of notation, we will write \( FOMM(i) \) and \( SOMM(i) \) to indicate, respectively, a FOMM and a SOMM built with a threshold \( i \).

In order to catch the idea of fitness, generalization and precision, we tested how Markov Models are able to detect good and bad instance processes.

To estimate the simplicity of a model, we considered the total number of arcs, which is a reasonable measure of complexity from a human point of view.

3.1 Dataset

In order to test fitting and generalization, we built a ground truth for comparison using a set of known models and we used them to generate the event Logs. By this way we avoided to use real world data which can be –particularly in medicine– extremely noisy in terms of missing values and wrong data entry. Because of we are interested in testing how FOMM and SOMM are able to stay close to the real process model which generates a set of process instances in this analysis we consider synthetic data generated according to real-world workflows. This allow us to generate noise-free data, or data with a specified element of disturbance, and to have a ground truth for comparison As it is pivotal to generate data as close as possible to real-world data, we only adopted real clinical workflows or real clinical guidelines.

To represent CG and workflows, we used the Pseudo Workflow Language (PWL), which is an internal language of \( pMineR \) specifically designed to build and represent workflows, for implementing four different sets:

- **Test1.** A workflow representing the different steps of patient care in a Radiotherapy Dept.
- **Test2.** An internal protocol, we adopted to treat patients affected by rectal cancers.
3.2 Data Generation

We generated synthetic data according to the following approach: (I) we implemented the workflow or guideline and we generated a set of 1,000 valid process instances (training set); we also generated a set (testing set) of 500 valid process instances and 500 invalid process instances. The invalid process instances were generated by corrupting valid process instances in a subtle way, by adding one non-legal consequent event to the process instances. (II) Given the built process instances, we trained 30 SOMMs and 30 FOMMs with threshold values ranging from 0 to 0.6, with a step of 0.2.

We then used pMineR as a case study for testing the effectiveness of FOMMs and SOMMS in recognizing correct and incorrect sequences of events, and assessing the complexity of generated models.

4 Experimental Results

The experimental analysis has been designed for investigating two main aspects: the ability of models in discriminating between valid and invalid process instances and the complexity, from a human perspective, of the models.

Results presented in Figure 1 show the performance of FOMM and SOMM models in recognizing valid and invalid process instances. In terms of accuracy, FOMM and SOMM models tend to perform similarly, but SOMMs usually allow to achieve better performance when low threshold values are exploited. Similarly, SOMMs tend to provide better sensitivity and F1 score when very low threshold levels are used. According to the presented results, SOMMs usually exhibit better performance on the considered data sets.

From a complexity perspective, SOMMs outnumbered FOMMs in terms of arcs and nodes: for this reason FOMMs have better performances. In addition, even if increasing the threshold reduce complexity of the graph, should also be noted that it increases the number of ‘0’ in the transition matrix, inducing the model to recognize a restricted set of process instances. For this reason the best trade-off is in adopting a soft threshold, which can help in reducing arcs which represent noise and preserve the expressiveness of the network.

5 Discussion

Process Mining is an emerging discipline, and subsumes three important areas: process discovery, conformance checking, and enhancement. Process discovery, in particular, is the step where an algorithm tries to build a model able to fit
Fig. 1. FOMM (on the left) and SOMM (on the right) performance on Test1 (a, b), Test2 (c, d), Test3 (e, f), and Test4 (g, h) in terms of accuracy, sensitivity, specificity, and F1 score. X-axis indicates the different threshold levels exploited, while Y-axis indicates the performance of the considered metrics.
given real world data in the form of a process. In this step, two elements are of pivotal importance: the language (formalism) used to represent the model of the process, and the algorithm exploited to build such model. Given the fact that Process Mining is a relatively recent field; few tools dealing with such task are currently available. Moreover, there is a lack of frameworks and methods for comparing and evaluating different approaches.

In this paper we proposed an experimental case report, applying FOMM and SOMM to data generated by considering four real-world workflows, for mining the underlying processes.

In particular, we used pMineR to implement four different workflows (based on three real clinical guidelines) and we used them to generate the sets to train FOMM/SOMM models, with different threshold. Then, for each workflow, we generated a testing set composed by a set of sequences compatible with the workflow and a set of sequences not compatible due to little changes in the sequences.

Afterward, we evaluated the ability of FOMM/SOMM models in recognizing which sequence of events were compatible with the workflow and which were not. We focused our attention to accuracy performance, which allow us to summarize the behavior of the FOMM/SOMM models in identify both true positive and true negative on the entire testing set.

Our analysis highlighted that: (i) SOMMs have generally better performances than FOMMs, but FOMMs are simpler and easier to represent. It is therefore important to select the best approach according to the expected use; (ii) high threshold values have a detrimental impact on overall performance. We empirically observed that “soft” threshold values (i.e., around 0.02) usually achieve better performance.

Another remarkable detrimental effect associated to high threshold is the fact that the suppression of the arcs can lead to a not connected graph, making some events not reachable from the model. Fortunately, that pitfall is easy to avoid by a preliminary check of the set of reachable nodes.

For our issues, even if FOMM models are intuitively weak due to the poor memory they seemed to have a quite good ability in understanding the general rules behind a clinical guideline, ranging from accuracy, for reasonable thresholds, from 0.8 to 1.0.

Future work includes the evaluation of different approaches for discovering Process Mining models, and the formalization of an extensive and robust framework for comparing tools. Intuitively, it may be the case that approaches based on techniques such as Petri Networks, BPM, and Fuzzy Miner, can have remarkable performance. Specifically, we are interested in investigating metrics that allows to compare all the steps of the Process Mining task.

References

A Self-Enforcing Network for the Analysis of Fall Cases in a University Hospital

Christina Klüver\textsuperscript{1} and Christian Dahlmann\textsuperscript{2}

\textsuperscript{1}University of Duisburg-Essen, COBASC Research Group, Faculty of Economy and Business Administration
\textsuperscript{2} Department of Nursing Development and Research, University Hospital Essen

Abstract. Accidental falls are a relevant safety problem in hospital inpatients. Falls can prolong hospital stays and are potentially life-threatening. Further, next to physical injuries emotional harm (e.g. subsequent fear of falling) can occur, increasing the risk for other falls. Great efforts have been taken to prevent falls at the University Hospital Essen, Germany (UK Essen), resulting in a comparatively low rate of falls of less than 2 per 1.000 inpatient days. To examine the remaining falls a Self-Enforcing Network (SEN) is used to cluster and to analyze the complex medical data. In order to achieve this, a cue validity factor is defined, which steers the clustering of the data according to specific research interests allowing closer investigations of the selected data within a cluster. We investigated 143.741 clinical inpatient records of the UK Essen (patients discharged 2013 - 2015). For these cases a fall is recorded for the number of 2.596 inpatients; 11 cases with rather severe consequences of the falls are selected and examined in more details. The preliminary results show that main and secondary medical diagnoses of the patients should be examined more closely in order to assess possible correlations between the diseases and the fall consequences.

Keywords: Self-Enforcing Network, self-organized learning neural networks, medical data, severity levels in fall cases.

1 Introduction

The analysis of fall risk factors and the development of prevention strategies in hospitals is one of the important research fields in medical care. As a result, the investigations in fall detection and developed prevention strategies have in general a positive effect (Hester et al., 2016). The UK Essen has undertaken intensive effort to reduce the rate of accidental falls over the last years, e.g. routinely performing assessments of risk for falls, measures to prevent falls with a special focus on patients at risk, structured digital documentation of each occurring fall (including those without causing harm to a patient) and the implementation of clinical experts for analyzing fall events. All measures are compliant with the recommendations of the “Deutsches Netzwerk für
Qualitätsentwicklung in der Pflege” (German Network for Quality Development in Nursing).

As a result, the rate of accidental falls (counting all falls, whether with or without patient injury) has been comparatively low over the last years with a mean of 2 per 1,000 inpatient days (0.2%). Though this rate is considerably low, activities are ongoing to further improve patient safety.

To examine the remaining falls a Self-Enforcing Network (SEN) is used to analyze the clinical data. The intention was to check if SEN is a suited algorithm for the analysis of routine clinic data and to get insights about possible commonalities in the fall cases. SEN can be used without predefining any parameters, which need specific knowledge about the method, as is the case with e.g. k-means. In addition the features or attributes in the data can be switched off allowing an analysis according to specific interests; if e.g. the severity of the fall consequences are in the focus of the analysis then only this attribute is activated and the clusters are automatically generated according to the severity levels; the selection of one cluster enables detailed analysis with another SEN with the advantage of the reduced numbers of cases.

All used clinical data are from the existing reports/reporting systems and are anonymized by removing all personal information. Demographic data like age and gender were kept. Further, inpatient treatment numbers and person-ID-numbers derived from the Hospital information system (HIS) were pseudonymized. This was done in a way, which still allowed the authors to assign multiple inpatient stays of the same person to the correct person.

In this article we first describe the data used for the analysis and subsequently SEN is presented in the main components for this analysis; in the next section main results are discussed.

2 Clinical data

Typically, medical data have not only a big volume; they are also very complex, containing different types of numerical data and text components, which have to be pre-prepared for a suited algorithm. Using clinical routine data from the University Hospital in Essen means to have all information about a patient; in total 143,741 (DRG-cases) data are at disposal from the year 2013 to 2015. Additional information as recorded fall cases, according visible consequences, the action, the place where the fall has occurred, and the number of total diagnoses means to have millions of available information.

Each record has a different number of columns containing numerical values and text descriptions; the latter have not been taken into account. The column containing a diagnosis according to ICD-10\(^1\) has numerical values and character combinations, which are transformed in only numerical (A = 10, ..., Z = 35; e.g. the ICD-10 code “A18.2“ for *Tuberculous peripheral lymphadenopathy* is coded as 1018.2). The

---

\(^1\) International Statistical Classification of Diseases and Related Health Problems, 10th Revision - Australian Modification.
encoding of these categorical data enables the analysis of similarities between main, secondary diagnosis, and the fall cases, preserving the diagnosis.

The German ICD-10 contains 22 chapters of diseases, 261 groups of diseases, 2,037 three-digit-numbers categories, and 12,161 four digits for sub-categories. In consequence the possible diagnostic specification is already very heterogeneous, having about 77,000 codes only for the attribute “diagnosis”.

The fall cases have additional information as e.g. the severity level 0 – 2, with e.g. level 2 meaning rather severe consequences, falling day, time, place, action, and visible consequences, coded as 0 (no visible consequences), 1 (bruises), 2 (pain), 3 (abrasions) to 4 (fracture) and additional information about the body parts.

In total a record of 2,596 fall cases during the period 2013 – 2015 (severity level 0 = 1,930; level 1 = 630; level 2 = 36) are numbered consecutively to ensure the data protection and analyzed in more details.

The collected and pre-prepared data are automatically imported in the so-called semantical matrix of SEN meaning that the real data are inserted into the network, starting the training process.

The records contain a huge amount of information, therefore we selected for the analysis of the 2,596 data with fall records following attributes for SEN (Table 2):

Table 2: The used case information for the analysis

<table>
<thead>
<tr>
<th>Attributes (case information)</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnosis: A00 – Z50</td>
<td>100 (A00) – 3550 (Z50)</td>
</tr>
<tr>
<td>Age</td>
<td>0 (premature) - 99</td>
</tr>
<tr>
<td>Location</td>
<td>1 = common room; 2 = shower; 3 = corridor; 4 = hospital area; 5 = wash room; 6 = patient room; 7 = toilette; 8 = else</td>
</tr>
<tr>
<td>Activity</td>
<td>1 = when standing up; 2 = when sitting down; 3 = when walking; 4 = else; -1 = unknown</td>
</tr>
<tr>
<td>Fixation (excerpt)</td>
<td>3 = bedrail one-sided; 4 = bedrail double-sided</td>
</tr>
<tr>
<td>Visible consequences</td>
<td>100 = bruises; 200 = pain; 300 = abrasions; 400 = fracture</td>
</tr>
<tr>
<td>State of consciousness</td>
<td>0 = conscious; 1 = awake; 2 = gloomy; 3 = slowed down; 4 = unconscious</td>
</tr>
</tbody>
</table>

3 The Self-Enforcing Network

The SEN is a self-organized learning neural network, developed by the Research Group "Computer Based Analysis of Social Complexity" (CoBASC); the tool is continuously enhanced with new components and features based on different user requirements (cf. e.g. Klüver, 2016a; Klüver, 2016b). Because SEN offers several functionalities, only the ones used for this study are shortly presented.

SEN orders or classifies data sets, i.e. objects with certain attributes. The data are represented in a “semantical matrix”. The rows of the matrix represent the objects \( a \) and the columns the according attributes \( a \); the values of the matrix \( w_{ab} \) are the “degree of affiliation” of the attributes to the objects. In this case the values of the
semantical matrix are the real medical data imported from csv-files, using the min-
max normalization, accordingly adjusted for the SEN with the interval [-1.0 – 1.0].

SEN starts by analyzing the values of the semantical matrix and by transforming
the values into the weight matrix of the network. The weight matrix, hence, is
generated from the semantical matrix and not at random.

The learning rule of a SEN that changes the values of the weight matrix is:

\[
w(t+1) = w(t) + \Delta w, \hspace{1cm} (1)
\]

\[
\Delta w = c \times w_{ao},
\]

where \(c\) is a constant usually defined as \(0 \leq c \leq 1\). It has the same function as the learning
rate in standard neural networks.

For real data the “cue validity factor” (cvf) (cf. Klüver, 2016a) becomes important.
The cue validity is a measure how important certain attributes are for membership in a
given category (Rosch and Mervis, 1975; Klüver and Klüver, 2011); it also allows to steer
the building of clusters by setting the cvf = 0 for the attributes, which should be
excluded from the analysis and cvf = 1.0 for the selected ones.

Then Equation (1) becomes

\[
\Delta w = c \times w_{ao} \times cvf_{a}.
\]

(2)

As in each neural network the dynamics of a SEN is generated by so-called
activation functions. For this study we have used the linear function:

\[
a_{j} = \sum w_{ij} \times a_{i},
\]

where \(a_{j}\) is the activation value of the receiving neuron \(j\), \(a_{i}\) are the activation values of the
sending neurons \(i\), and \(w_{ij}\) as usual are the according weight values.

The topology of SEN can be understood as a two-layered network with a feed-
forward topology by considering the attributes (case information) as input neurons
and the objects as according output neurons (case ID).

After the learning process is finished, a user can insert a so-called input vector
meaning new medical data with the same attributes name.

The tool offers different visualizations for the results; for the analysis of fall cases
we use the so-called “map visualization”, representing the approximated similarity
between the objects (for details cf. Klüver, 2016b):

If the map visualization of the trained network allows the detection of clusters,
polygon features select the objects within one cluster. The marked objects can be
exported to a csv-file and imported by another SEN or transformed into “input-

4 Results

Because the falls with rather severe consequences (level 2) are of special interest,
only these data are selected. For the 36 cases the main and 5 secondary diagnoses are
inserted into the semantical matrix to analyze if there are possible correlations
between the main and secondary diagnoses.
In this case the cvf is 0.2 for the main (MD) and the secondary diagnoses (SD); this value is suited to have a kind of differentiation in the MD and SD. To steer the clusters primary to the visible consequences the cvf is in addition 1.0 for this attribute, and 0.0 for all the others. The result is shown in Fig. 1:

![Fig 1. Result of SEN](image)

SEN clusters the data according to the visible consequences. The data marked with a polygon on the lower part of Fig. 1 contain fall cases with a fracture as consequence of the fall. Most important is the building of sub-clusters within the clusters because of the different diagnoses. The outlier with the ID 2561 e.g. has some distance to the cluster because the patient has only one diagnosis; the similarity is in this case due to the main diagnosis and fracture as consequence of the fall. There is an indication that some patients have commonalities in the MD and SD (overlapped data in the clusters).

To evaluate this assumption the analysis of fall cases with a fracture as fall consequence are analyzed in more detail (Table 2):

Table 2: The main (MD) and five secondary diagnoses (SD) for 11 fall cases with a fracture and additional information about the patients, as the phase of the falling (column 10) and the total stay in hospital (column 11). For 6 cases no information about previous falling is available and only in one case a previous falling is recorded (last column).

<table>
<thead>
<tr>
<th>MD</th>
<th>SD-1</th>
<th>SD-2</th>
<th>SD-3</th>
<th>SD-4</th>
<th>SD-5</th>
<th>age</th>
<th>location</th>
<th>activity</th>
<th>duration of fall</th>
<th>visible co-occurrence</th>
<th>Gencord</th>
</tr>
</thead>
<tbody>
<tr>
<td>C06.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
<tr>
<td>C08.0</td>
<td>D9.11</td>
<td>R55</td>
<td>SQ2.3</td>
<td>SQ0.2</td>
<td>J96.0</td>
<td>43</td>
<td>walk</td>
<td>0</td>
<td>02</td>
<td>fracture</td>
<td>2 F N</td>
</tr>
</tbody>
</table>

The diseases have the following classification: C00 – D48: Neoplasms; D50 – D89: Diseases of the blood and blood-forming organs and certain disorders involving
the immune mechanism; E00 – E90: Endocrine, nutritional and metabolic diseases; G00 – G99: Diseases of the nervous system; I00 – I99: Diseases of the circulatory system; J00 – J99: Diseases of the respiratory system; K00 – K93: Diseases of the digestive system; R00 – R99: Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified; S00 – T98: Injury, poisoning and certain other consequences of external causes.

This classification of the diagnoses according to ICD-10 corresponds to the result found by Hester et al. (2016) for the primary discharge diagnoses, but in our study there is an additional indication that the combinations of these diseases should be additionally taken into consideration.

4.1 Comparison of the trained fall records with randomly selected data

The next result (Fig. 2) supports the assumption that the combination between main and second diagnoses are of relevance: the 36 cases with severity level 2 are chosen as objects, that have beside the IDs the severity level and the visible consequences merely as information in the object name; only the main and secondary diagnoses remain as attributes in the semantical matrix.

As input vector 500 cases with similar main diagnosis as the 36 cases are randomly selected from the whole data set and inserted in SEN, containing again only the main and 5 secondary diagnoses of the case ID.

Fig 2. Result of SEN according to the main and secondary diagnoses.

Most of the patients that have a fall record with the severity level 2 are clustered visible together (red symbols). The cases inserted as input (blue symbols) are classified according to their similarities to the trained fall cases with a severity level 2.
From the 135 IDs classified near to the cluster containing the fall cases with a severe consequence only 21 have no fall record; the others have records of severity level 0 or 1. The remaining 365 IDs are clustered in different sub-clusters.

It can in addition be observed that there are several clusters having commonalities in more than the main diagnoses.

5 Final remarks and further work

In this study a preliminary investigation with SEN shows some promising results. Due to the described fall cases with severe consequences one should e.g. consider additional assistive devices in the patient rooms, since most of them were falling trying to stand up or to walk in the room. The cases, which are placed nearer to the cluster containing patients with a fall record of level 2, should be analyzed in more detail by considering more than only the primary discharge diagnosis.

The cue validity factor enables to cluster the data according to a current issue reducing in consequence the large number of data. If only the attribute “severity level” has a cvf of e.g. 1.0 than the according data are clustered visible together and the other data can be excluded from the further analysis.

The goal is to predict possible fall cases with severe consequences by means of the specific combination of the diagnoses; the obtained result gives us the opportunity to check, if the meanwhile collected data for the year 2016 can be used to predict future according fall cases by taking into account the combination between the main and secondary diagnoses. To evaluate this prediction with SEN we will insert the new clinical data as input vectors, containing the main and secondary diseases as shown in Fig. 2, without the information about the severity level and visible consequences – or any other information about the cases – to verify if the data are classified accordingly.

6 References

Towards automatic clinical operation encoding (CHOP) with convolutional neural networks

Yihan Deng\textsuperscript{1}, André Sander\textsuperscript{2}, Lukas Faulstich\textsuperscript{2}, and Kerstin Denecke\textsuperscript{1}

\textsuperscript{1} Bern University of Applied Sciences, Biel/Bienne, Switzerland
yihan.deng@bfh.ch, kerstin.denecke@bfh.ch

\textsuperscript{2} ID Information and Documentation GmbH, Berlin, Germany
a.sander@id-berlin.de, l.Faulstich@id-berlin.de

Abstract. Automatic encoding can increase the interoperability and efficacy of the clinical cooperation. Classification systems such as ICD-10 for diagnoses or the Swiss Operation Classification System (CHOP) for procedure classification in the clinical treatment are essential for clinical management and information exchange. Traditionally, classification codes are assigned manually or by systems that rely upon concept-based and rule-based classification methods for code generation. Such methods can reach their limit easily due to the restricted coverage of handcrafted rules or of the vocabulary in underlying terminological system. The conventional machine learning approaches depend normally on a manually annotated training set with selected features. However, it is quite laborious to obtain a good labeled data set and its generation can easily be influenced by accumulative errors generated by human factors. To overcome this, we will present our preliminary design and processing pipeline for the CHOP classification task realized with the convolutional neural network for query matching. The results are evaluated based on query logs recorded in an existing encoding retrieval system. The advantages and limitations of the proposed method are discussed.

Keywords: Automatic Encoding, Classification, Machine Learning, Deep Learning, Information Retrieval

1 Introduction

In order to claim costs to the health insurance and for clinical documentation purposes, it is necessary and even legally required to encode diagnoses and procedures by classification codes from relevant classification systems. In Switzerland, these are ICD-10-GM for diagnoses and the Schweizerische Operationsklassifikation (CHOP)\textsuperscript{3} for clinical and surgical treatments. In everyday clinical practice, many documents are given in the form of a free-text. Beyond, physicians want to use free-text queries when searching for an appropriate classification code. For example in Fig. 1 the user query is \textit{Chirurgie Fluoroskopie} (surgery fluoroscopy). As it can be seen in Figure 1, for this query multiple classification codes could

\textsuperscript{3} https://www.bfs.admin.ch/bfsstatic/dam/assets/483959/master
be assigned since the category "computer assisted surgery fluoroscopy" which is on of the six sub-classes under "computer assisted surgery". Extraction services and automatic encoding start at this point and offer the possibility to automatically extract specific information from texts and query and support in classifying them according to standard medical terminologies. The underlying methodological approach is often rule-based [1]. Hence, comprehensive rules are specified beforehand, which is often time-consuming and also difficult to achieve completeness and correctness. Methods of machine learning aim at learning rules automatically and deriving intrinsic models consisted in large data sets. Although manually chosen features are proven to be effective for some specific classification tasks, the biases caused by features selection at each processing step can not be avoided. Convolutional neural networks employ all possible combinations by predefined filtering size and select the most representative features by pooling process (max pooling or average pooling). The convolution and pooling process can largely reduce the deviation caused by manual feature selection. Our aim is applying and evaluating convolutional methods for determining relevance between query text and CHOP category text. More specifically, we are assessing which kind of neural network and network configuration can yield the best matching accuracy. We are focusing on the Swiss CHOP classification, which consists of 4000 classes distributed in 16 different categories. However, the approach is adaptable to other classification systems. Further, we will study the performance and efficiency that can be achieved by applying convolutional neural network to this task in comparison to traditional machine learning approaches. This specific task is characterized by query-document pairs and the corresponding relevance labels. The network model will be trained and deployed to predict the relevance of the query and document.

Fig. 1. CHOP category and query scenario
2 Related work

Deep learning has already sparked extensive attention in the fields of semantic matching and automatic text classification (e.g. for sentiment classification [7], or web query matching [2]). Kim [3] has proposed a single layer convolutional neural network (CNN) with one convolution and pooling layer for sentence classification. The approach has conducted the convolution with different filtering sizes (3,4,5) with 100 feature map for each. Similar to Kim and Shen et al. [2] have approved the feasibility of the application of the CNN to obtain the most representative features to conduct classification and matching from text data. Instead of using a single convolutional and pooling layer, Hu et al. [4] have designed a deeper CNN with four layers (double successive of convolution and pooling) for the matching of natural language sentences. The network enables the matching of sentences at both token and sentence level, whereas the sentence level sequential order has also been considered. More specifically, the sequence order of the tokens in a matched sentence is preserved through the 2D pooling based on the feature generated by low level features from first pooling layer. The iterative convolution and pooling process has improved the accuracy of text matching.

Lu et al. [5] have presented a comprehensive matching strategy for the mapping of short text. The researchers have defined a matching level as inner product in the feature space. The model considering localness and hierarchy in the language has then been proposed. The model employed topic modeling and a layered directed acyclic graph to model the topic hierarchy, so that the matching process could be done in a hierarchical way. The architecture has successfully outperformed other benchmarks by the matching tasks of objects from two different domains.

A few works reported the application of deep learning in the medical domain. Minarro-Gimnez et al. [6] used word2vec to identify pharmaceutical relations between medication and diseases or procedures in medical corpora. However, the word2vec toolkit has only revealed a maximum accuracy of 49.28% which suggests a limited ability of word2vec to capture linguistic regularities on the collected medical corpora compared with other published results. This accuracy is still not suitable for applications requiring high precision as required in the medical domain. Peng et al. [7] introduced DeepMeSH that exploits deep semantic information for MeSH indexing. The Medical Subject Headings thesaurus (MeSH) is standard medical terminology for the indexing of biomedical literature. The word embedding is combined with inverse document frequency. 23,343,329 citations have been downloaded from Pubmed to evaluate the proposed methods. In comparison with the conventional index algorithms (Medical text indexer with an F-Measure of 0.5724 and MeSHLabeler with an F-Measure of 0.6248), the DeepMeSH method has achieved an improvement of 12% and 2%, respectively compared to the existing approaches. According to aforementioned state of the art analysis and the attributes of our training data in our task, convolutional neural network (CNN) with multiple filter sizes could be a suitable technology to solve the matching of CHOP encoding. There are three reasons:
- We are considering the retrieval task of finding relevant CHOP classification texts matching a user query.
- The structure of the query and the CHOP category description are substantially different compared to texts and queries in other tasks. The query and CHOP category description are both short and contain normally only a limited number of biomedical concepts. No complex sentence structure can be found neither in query nor in the text content due to the nominal style. Only the semantic relation between medical concepts between query and document should be recognized.
- Given the limited length of query and text, also the sequence order of terms is considered irrelevant in this tasks.

In this work, we will exploit CNN with a single convolutional pooling layer to solve the CHOP query matching since existing research demonstrated good results with one layer. The feasibility of the approach will be evaluated based on query and result set.

3 Data set

For testing and developing our approach, we are using the search logs generated by the application of an existing retrieval system based on semantic mapping. These search logs contain the user query and the relevant CHOP classification codes. For each query / classification text pair, it has been annotated whether one CHOP category matches the user query. To each CHOP code, a set of textual category description is available. For generating the vectors, query and category texts of CHOP have been mapped to concepts of a medical terminology through terminology server ID MACS® medical semantic network, a software provided by the German company ID Information und Dokumenation im Gesundheitswesen. With ID MACS® it is possible to analyze medical texts and, for instance, to extract structured information on diagnoses and procedures and map them onto a chosen medical terminology [1, 7]. The semantic network incorporates the Wingert Nomenclature, a German derivative of an early version of SNOMED, as a knowledge base [8]. It maps the text onto the medical concepts and depicts the relationship between individual concepts. For each text string, a set of concepts is assigned. From this list of concepts per text / query, a vector is generated. The vector representation forms the basis for our developments. Totally, the training data set consists of positive (7521) and negative (16569) relevant query pairs. The vector is presented in ARFF format. For example, a query A \{2 1, 3 -1, 5 1\} means vector a \(V_a[2] = 1, V_a[3] = -1, V_a[5] = 1, V_a[i] = 0\). The number 1 indicates the existence of this concept in the vector, while the number -1 shows the exclusion of one concept in the vector.

Furthermore, the query and document pairs belong to certain search issue whereas an issue is an internal identifier for a series of similar query document pairs. The successful solving of one search issue requires the matching of the entire inclusive query pairs and also exclusion of all the exclusive pairs. The using of issues have
increased the level of matching and yielded higher requirement on generalization of the learned models.

4 Convolution based classification method

In general, we consider the task of determining the relevance between a category description and a specific query as a classification problem: the text vector for a document has to be classified as relevant (1) or irrelevant (-1) for a query. To realize this task, our processing pipeline comprises two steps: concept embedding and determining the relevance using convolutional neural networks. These steps are presented in more detail in the following.

4.1 Concept embedding

The word embedding methods and word2vec toolkit were developed by Mikolov [9] to generate structure preserving semantic representation. The embedding vector generated by word2vec are trained based on large amount of news text. The modeling of embedding is based on Continuous bag-of-words (CBOW), Skip-gram and negative sampling. However, the queries for our use case contain in most cases only a limited number of medical terms. The pre-trained word2vec representation will not be able to reflect the features and distribution of biomedical concepts used in CHOP encoding. A task-specific concept embedding had therefore been established based on our training set: Firstly, the query and CHOP category text are represented as concept vector pairs; the concepts are obtained from dictionary-based concept recognizer based on the terminology server ID MACS\textsuperscript{R} (see section 3). The advantage of transforming the query and category text into a concept vector is the reduction of dimensions: the vector comprises not each word in the texts and queries, but just the concepts. The other advantage of using medical concepts is the robustness against variance in the text such as misspelling and also the future possibility of extending the vector through the knowledge from the semantic network. For example synonyms are represented by the same concept. In a second step, the embedding is learned from the training set characterized by the concept vector.

4.2 Convolutional neural network for relevance determination

Our architecture exploits a convolutional neural network with one layer of convolution and one layer of max pooling. In the convolutional layer, different filter sizes can be defined to cover the potential semantic scope in the query and category description. The architecture is therefore similar to the CNN employed by Kim and Shen [2]. As input for the training, the vectors of the query and document from the training set are concatenated as one vector. The features are obtained from multiple filters (3,4,5); convolution will be selected through

\footnote{https://github.com/dav/word2vec}
max-pooling and fully connected into one feature vector. The filter size 3,4,5 has been determined based on the length statistics of our training corpus as most of concept vector of query and category have shown a length ranged from 3 to 5. Let \( x_n \) be the k dimension concept vector, which is the n-th concept in one query or CHOP description. These k dimensional vectors represent the embedding of \( x \) in k dimensional space. A query can then be represented as \( Q \), with length j, while the category description is the vector \( D \), with length k. Let \( \| \) be the concatenation operation. The query and category pair

\[
P_i = Q_i \| D_i
\]

will all be padded to the dimension of \( m \), since the padded sentence with same length can facilitate the batching and follow up transformation process in CNN. Hence, a pair of query and retrieved category description is modeled as

\[
P_i = x_1:Q_j\|x_1:Q_k\|x_{m-(j+k)-m} = \{x_1, x_2, \ldots, x_m\}
\]

The convolution process is conducted through the filter with predefined size, which slides through the concatenated concept vector. In our network, a filter of size h generates a window of \( x_{i:i+h} \). The corresponding features are obtained through an activation function \( f \) in form of

\[
\text{Convolutional features} = f(w \cdot x_{i:i+h-1} + b)
\]

For each dimension of concept embedding (size:128) we have assigned one filter to slide through all the possible windows in the concept vector. As it can be seen in figure 2, a feature vector with length of \( n-h+1 \) is generated at each dimension of the embedding. As next, the maximum value in the feature vector is pooled out through the max pooling process. Through this process, the most salient features covered by the single filter size (3,4,5) are selected. In this way, three filtered features with different coverage are generated to create a fully connected feature vector.

### 4.3 Implementation with Tensorflow

The aforementioned architecture is implemented with the Tensorflow framework\(^5\). We have chosen Tensorflow, since it provides comprehensive toolkits for the construction of embedding and neural network. The computational network works as a roadmap of the data processing workflow, whereas the real data will not be loaded into the computing graph. The input will be only defined with sizes and attributes within placeholders. The real value input will be triggered only after the session has been initialized through queue feeding or dictionary feeding, while queue is defined for asynchronous tensor input and dictionary is used to input the data with small amount statically. This deferred form of graph computing can also optimize the resource utilization and scalability during the

\(^5\) https://www.tensorflow.org
Fig. 2. Convolutional neural network for CHOP query matching

processing. For the creation of a convolutional neural network, Tensorflow has provided interfaces like `tf.nn.conv2d` and `tf.nn.max_pool`. The convolution and pooling process can be added and configured intuitively to create our proposed CNN architecture. In this implementation, we have also used the Python APIs from the package from scikit-learn\(^6\) and numpy\(^7\) to implement the benchmark.

### 4.4 Configuration of network parameters and evaluation

As general network configuration for all experiments, rectified linear unit (Relu) has been applied as activation function; filter windows of 3, 4, 5 with 128 dimensional feature maps are applied. The dropout rate has been defined by 0.5 whereas an L2 constraint (\(s\)) of 0.01 has been defined to avoid over-fitting. The batch size is 64 as the size were chosen through incremental experiment using evaluation set. For the evaluation, we have separated the data set into training/dev/test. Training and dev was assigned with 70% of total data amount (training 70%, dev 30% of the first 70%), whereas the test set has assigned with 30% of the total data amount. Due to the internal definition of the retrieval system, the query and document pairs stem from certain search issues. The training, dev stem from similar issues, while the test data are generated from completely different issues. We performed on the one hand an evaluation with the trained network. On the other hand, the accuracy for predicting unknown issues using learned model is determined.

---


\(^7\) [http://www.numpy.org](http://www.numpy.org)
At the first level of test, we will only evaluate the matching accuracy between query and document pairs. In a second step, an issue based evaluation will be applied to determine the matching performance in a real world application. More specifically, one issue is solved correctly when, of each inclusion code, at least one of the inclusion items and none of the exclusion items are assigned.

5 Evaluation and Result

Based on the hyper parameters defined in section 4.4, we have tested some additional combinations of parameters. The defined network has equipped with a dropout of 0.5 and L2 with lambda 0.0. The accuracy is the mean average value calculated based on the entire data set (24090 pairs) with the corresponding model configuration. The models have all been trained over 7000 full training iterations. The result has shown that the CNN with Tri-filters (3,4,5) with Ada delta and L2 (lambda:0.01) can achieve the best dev accuracy of 99.9932%. The traditional SGD classifier has only reached 80.41% accuracy and the stochastic gradient descent together with CNN has only reached an accuracy of 89.21%. However, the accuracy on dev set has only shown our model has converged to the training set, the accuracy on the test data has presented the portability of these models on totally new query-document pairs in different issues as it is used in the training set. The accuracy on the test set has shown that the current method has learned some of the distribution from the small training set. The limitation of the current methods will be discussed in the following section.

<table>
<thead>
<tr>
<th>No.</th>
<th>Configuration</th>
<th>Accuracy@DEV</th>
<th>Accuracy@Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SGD classifier</td>
<td>80.41%</td>
<td>20.23%</td>
</tr>
<tr>
<td>2</td>
<td>Logistic regression (two class softmax)</td>
<td>89.21%</td>
<td>18.14%</td>
</tr>
<tr>
<td>5</td>
<td>CNN 3,4,5 Ada L2 0.01</td>
<td>99.993%</td>
<td>67%</td>
</tr>
</tbody>
</table>

Table 1. Classification result based on DEV and Test Set for relevance determination based on different configurations, 3,4,5 means three filters with size of 3,4,5 respectively has been applied

6 Discussion

The current results indicate that the method has not reached the ideal accuracy in real production. As extension we consider the enrichment of the current vector representation. Two enrichment methods can be proposed:

- One possible solution can be, at the step of concept embedding, enriching the current vector with concepts from the knowledge base and keep the original
text sequence. Knowledge from the terminology hierarchy like (Wordnet\textsuperscript{8} or PPDB\textsuperscript{9}) can be used to facilitate the query matching. Similar attempts of embedding with semantic enrichment have been conducted by Yu und Dredze [10] and Celikyilmaz et al. [11]. The type of relations and weights of the relations between concepts have been combined with a general embedding model linearly. The joint models require different data for training; while the word embedding will be trained on a training corpus (get the sequence data). These methods based on semantic embedding has approved to yield between 2\% to 19\% improvement.

– The other possible solution is based on terminology enrichment and auto-encoder [12]. The conceptional hypernyms und hyponyms in the terminology hierarchy can be used to enrich the vector, while the entire vector can be reduced through auto-encoder to a suitable dimension. After that, the vector can be used as input to a logistic regression process. The main difference of this method regarding the aforementioned semantic embedding is that the sequence of the term in vector can be ignored after the hierarchical terminology enrichment.

– In addition to the current query document concatenation, a possible way to increase the effectiveness of the convolutional filter would be the sorting, aligning and interleaving the query sequence and the document sequence to make sure that the filter always sees a subsequence of the query together with the corresponding subsequence of the document. E.g., for Q= (a,c,e) and D= (b,c,d), this would yield $P= ((a, \_), (\_ , b), (c, c), (\_ , d), (e, \_))$ instead of $P= (a, c, e, b, c, d)$. The "\_" indicates all subsequences in one vector. E.g. the "\_" of Q is (a) (b) (c) (a,c) (a,e) (c,e) and (a,c,e).

7 Conclusion

We have evaluated a convolutional neural network with single convolutional and max pooling layer for the task of relevance determination for a query matching in Swiss operation encoding. The query and category text are represented as vectors of biomedical concepts. The model has proved to achieve a high converge on training and a moderate accuracy in real productive test. The efficiency of the training based on the defined network is good due to the simple structure and effective deferred training structure. As next step, the model will be extended with enriched semantic vector representation. A more holistic query matching method based on sequence-less semantic enrichment will be tested.

8 Acknowledgement

This work is supported by ID Information and Documentation GmbH, Berlin, Germany.

\textsuperscript{8} https://wordnet.princeton.edu
\textsuperscript{9} http://www.cis.upenn.edu/ ccb/ppdb/
References


Corpus Annotation for Aspect Based Sentiment Analysis in Medical Domain

Department of Computer Science, Universidad de Jaén
Campus Las Lagunillas, E-23071, Jaén, Spain
{sjzafra, maite, mdmolina, lauren}@ujaen.es

Abstract. This paper introduces a new Spanish corpus of drug reviews annotated for aspect based sentiment analysis. It presents the annotation guidelines, describes the corpus and provides corpus statistics. The resource, named DOS (Drug Opinions in Spanish) corpus, consists of 877 reviews from patients about different aspects of several drugs. A total of 2,230 side effects (aspects) were annotated by three human annotators. It is freely available for the research community. The quality of the annotation process has been assessed by measuring the inter-annotator kappa agreement, which was of 72.25%.

1 Introduction

Sentiment Analysis (SA) is a task of Natural Language Processing (NLP) which focuses on treating the subjectivity in textual information [9]. Although SA is a relatively new area, it has been widely studied in the last years and even some commercial systems are currently working within an appropriated effectiveness. The rapid growth of the information in social media through the Internet and the advances in NLP and Machine Learning Techniques present an opportunity to mine these data in different domains, such as tourism, marketing or the political environment.

Nevertheless, there are some domains where we can find a lot of social media data but very few research projects are using this information. This is the case of the medical domain where there are an important number of research applying NLP techniques but mainly oriented towards managing factual information. However, mining sentiments and opinions in medical social media can be also useful for multiple applications: detection of adverse drug effects and decision making about patient health or pharmacovigilance systems, among others. In this paper we focus on mining reviews and opinions from patients about different aspects of several drugs.

On the other hand, most of the approaches related to SA are oriented towards detecting polarity at document level instead of studying the polarity toward target entities (e.g., restaurants, laptops) and their aspects (e.g., food, service, price, battery). In contrast, Aspect Based Sentiment Analysis (ABSA) [12] is focused on mining opinions from text about specific entities and their aspects. ABSA
identifies the aspects of a given target entity and estimates the sentiment polarity for each aspect mentioned. Of course, this task is much harder than that of simple polarity detection, and the first step to tackling the problem is to prepare a corpus with the entities and aspects annotated with polarity. Thus, our main goal in this paper is the creation of an Aspect Based SA corpus about adverse drug effects as there is no such resource for Spanish. We focus on identifying opinions expressed within drug reviews about specific aspects, especially side effects. We present a new corpus with opinions and reviews from patients about different aspects of several drugs annotated with polarity and intensity by three human annotators. The corpus DOS (Drug Opinions in Spanish) will be useful not only for information extraction in order to detect possible adverse reactions to certain drugs but also to determine the polarity and intensity of the side effects.

2 Related work

Aspect Based Sentiment Analysis (ABSA) [15] is a challenging task that is attracting the attention of the research community mainly due to its complexity. The main goal is to analyse large amounts of unstructured text and extract fine-grained information not included in the user ratings which are available on some review sites. Perhaps the main problem associated with this task is the lack of resources like annotated corpus that can be provided as a base for training the computational systems. One of the initiatives that has contributed to the evolution of the discipline was the SemEval shared task on ABSA organized in 2014 [14], 2015 [13] and 2016 [12]. Several resources were built not only in English but also in other languages like Arabic, Dutch or Chinese. In addition, different domains were also examined, mainly related to tourism (restaurants, hotels, museums...) and digital products (mobile phones, laptops, digital cameras...).

We can also find some interesting papers studying ABSA in other languages different from English. For example, Steinberger, Brychcin and Konkol (2014) [16] present a corpus for ABSA based on data from restaurant reviews in Czech. The work of Jiménez-Zafra et al. (2015) [6] introduces a multilingual dataset in Spanish, English and Italian into the tourism domain about hotel reviews. Al-Smadi et al. (2015) [1] provide a corpus of book reviews in Arabic which has been annotated by humans with aspect terms and their polarities.

Regarding the medical domain, we can find a good survey about medical social media information in [2]. The author introduces a detailed state of the art, and some interesting tools and resources are described.

Mining patient opinions or experiences has been another source of research in the health domain, considering web data such as medical blogs or forums [3]. Research as the study by Greaves et al. (2013) [4] are focused on analysing the subjective information in websites like "Patients like me"1 or "Ask a patient"2. While these papers only manage English reviews, the one of Plaza-del-Arco et

1 https://www.patientslikeme.com/
2 http://www.askapatient.com/

Concerning resources about drugs and their side effects, there are some remarkable papers. For example, the work of Herrero-Zazo et al. (2013) [5] addresses the problem related to the lack of corpora annotated with pharmacological substances and drug-drug interactions. The authors create a manually annotated corpus with around 1000 documents extracted from Medline3 abstracts and text from the DrugBank database4. The Drug-Drug Interactions corpus and the annotation guidelines are freely available for academic research at the following web address: http://labda.inf.uc3m.es/ddicorpus. In [10] the corpus IxaMed-GS is presented. The corpus consists of collections of Electronic Health Records (EHR) annotated with medical entities and events in Spanish. IxaMed-GS includes syntactic and semantic annotations in discharge summaries written by Spanish doctors. The corpus is used for the automatic extraction of adverse drug reaction events using machine learning. Finally, [8] present CADEC, an interesting corpus with patient reviews of different drugs. The corpus is annotated with Adverse Drug Events (ADEs) including concepts related to drugs, adverse effects, symptoms and diseases. In addition, the entities discovered are linked to their corresponding concepts in controlled vocabularies (SNOMED and MedDRA). This is publicly available at https://data.csiro.au.

In this paper, we present a Spanish corpus with information extracted from a social web (http://www.mimedicamento.es) with reviews of drugs and their side effects. Patients express their opinions about the effectiveness and experience taking a specific drug, also including a rating between 1 and 5 stars for several satisfaction items.

3 Annotation guidelines

The goal of the annotation described herein is to identify opinions expressed within drug reviews towards specific aspects, specifically towards side effects. In this section, the annotation guidelines used in the annotation of the Drug Opinions Spanish (DOS) corpus are described, which are based on the respective guidelines of the SemEval-2016 ABSA task 5 [12].

Given a review of a particular drug, the task of the annotator is to identify, at the sentence-level, the following information:

- **Aspect Category.** Identify the entity (aspect category) towards which an opinion is expressed. It can be one of the following:
  - **SIDE_EFFECTS** for opinions expressed on the side effects produced by the drug that is being reviewed. Ej. Review about “Champix (varenicline)”:  

4 http://www.drugbank.ca/
5 http://alt.qcri.org/semeval2016/task5/
“Lo malo de este medicamento es que me ha provocado un dolor de cabeza muy fuerte y no desaparece.” → {SIDE_EFFECTS, “dolor de cabeza”}

“The bad thing about this drug is that it has caused me a very strong headache and it does not go away.” → {SIDE_EFFECTS, “headache”}

• OUT_OF SCOPE SIDE EFFECTS for opinions expressed concerning the side effects produced by other drugs that the reviewer has taken.

Ej. Review about ‘Amoxicilina (amoxicilina)” in which “Denvar (cefixima)” side effects are exposed:

“Antes de amoxicilina tomaba cefixima y me provocaba diarrea y dolor de estómago.” →
→ {OUT_OF SCOPE SIDE EFFECTS, “diarrea”}
→ {OUT_OF_SCOPE SIDE EFFECTS, “dolor de estómago”}

“Before amoxicilina I took cefixima and caused me diarrhea and stomach ache.” →
→ {OUT_OF_SCOPE SIDE EFFECTS, “diarrhea”}
→ {OUT_OF_SCOPE SIDE EFFECTS, “stomach ache”}

– Opinion polarity. Each identified aspect category of a sentence has to be assigned a polarity label from the set \( P = \{ \text{positive, neutral, negative} \} \). The neutral label applies for mildly positive or mildly negative sentiment, thus it does not indicate objectivity.

– Opinion intensity. Each identified aspect category of a sentence has to be assigned an intensity label from the set \( I = \{ \text{high, medium, low} \} \).

– Opinion Target Expression (Aspect Term). Each identified aspect category is representative of an Opinion Target Expression (OTE). An OTE is an explicit reference (mention) of the reviewed entity E (aspect category). This reference can be a verbal form, an adjective, a common noun or a multi-word term, and is uniquely identified by its starting and ending offsets. Below are shown some examples (offsets are omitted):

(a) “Me provoca una horrible migraña.” → {SIDE EFFECTS, “migraña”, negative, high}

“It gives me a horrible migraine” → {SIDE EFFECTS, “migraña”, negative, high}

(b) “Me da tirones musculares en las piernas.” → {SIDE EFFECTS, “tirones musculares en las piernas”, negative, medium}

“It gives me muscle twitches in my legs.” → {SIDE EFFECTS, “muscle twitches in the legs”, negative, medium}

(c) “Estoy encantado con este medicamento, hace que me espabile un poco.”

→ {SIDE_EFFECTS, “espabile”, positive, low}

“I am delighted with this drug, it livens me up a little.” →
→ \{SIDE\_EFFECTS, \textit{“liven up”}, positive, low\}

Considerations to take into account:

- When a sentence contains more than one mention (e.g. nominal and pronominal) of the same side effect only the nominal one should be annotated.

- If an OTE has more than one occurrence in the same sentence, all the occurrences should be tagged.

- When a side effect is only implicitly referred to (e.g. through pronouns) or inferred in a sentence, then it will not be annotated. Only explicit references of side effects should be annotated.

- The annotations should be assigned at the sentence level, taking into account the context of the whole review.

4 Corpus description

The data for the Drug Opinions Spanish\textsuperscript{6} (DOS) corpus was sourced from the medical forum called \textit{mimedicamento}, which is an independent platform for sharing experiences with drugs. It is available in 9 countries (Austria, Belgium, France, Germany, Italy, The Netherlands, Spain, Switzerland and United States). We accessed the Spanish forum\textsuperscript{7} because there are few studies focused on this topic in this language.

The corpus was built by crawling \textit{mimedicamento} forum on March 14, 2017. All the available consumer reviews about the 30 most reviewed drugs were downloaded: Abilify (aripiprazol), Aclasta (acido zoledronico), Amoxicilina (amoxicilina), Ana-franil (clomipramina), Aromasil (exemestano), Atarax (hidroxizina), Atorvastatina (atorvastatina), Ceresat (desogestrel), Champix (varenicilina), Cialis (tada-lafilo), Citalopram (citalopram), Crestor (rosuvastatina), Cymbalta (duloxetina), Doxicrisol (doxicilina), Fluoxetina (fluoxetina), Humira (adalimumab), Keppra (levetiracetam), Levotiroxina (levotiroxina), Lyrica (pregabalina), Metformina (metformina), Mirena (levonorgestrel), Ofloxacino (ofloxacin), Paroxetina (paroxetina), Seroxat (paroxetina), Sertralina (sertralina), Tamoxifeno (tamoxifeno), Tramadol (tramadol), Valdoxan (Agomelatina), Victoza (liraglutida) and Xarelto (rivaroxaban).

Each review contains information about the date in which it was posted, the gender and age of the consumer, the disease and the drug used for it, the textual opinion and a rating for the following satisfaction categories: overall, efficacy, side effects quantity, side effects severity and ease of use. Textual opinions are written by web users instead of health professionals. Due to this fact, they can be

\textsuperscript{6} http://sinai.ujaen.es/dos-2/

\textsuperscript{7} https://www.mimedicamento.es
grammatically incorrect and include spelling mistakes or informal expressions. The opinions are rated on a scale from 1 to 5 stars, where 1 star means that the satisfaction is very low and 5 stars means very high satisfaction. We added an identifier to each review and we annotated each sentence at the aspect-level using the annotation guidelines showed in Section 3. Figure 1 shows an example of a typical drug review.

![Fig. 1. Example of a drug review](image)

The DOS corpus is composed of 877 opinions about the 30 most reviewed drugs (reviews without textual opinion were discarded). Each drug has at most 32 reviews because this web portal only shows the 32 most recent opinions. Table 1 shows the number of reviews per rating. Ratings 1 and 2 are those that present the greater and the fewer number of reviews respectively. The distribution of reviews between the other rating values (3, 4 and 5) is more or less uniform. In Table 2, the distribution of comments can also be seen, but taking into account the satisfaction rating that consumers assigned to the efficacy, side effects quantity, side effects severity and ease of use of the drugs reviewed.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>253</td>
</tr>
<tr>
<td>2</td>
<td>125</td>
</tr>
<tr>
<td>3</td>
<td>163</td>
</tr>
<tr>
<td>4</td>
<td>176</td>
</tr>
<tr>
<td>5</td>
<td>160</td>
</tr>
<tr>
<td>Total</td>
<td>877</td>
</tr>
</tbody>
</table>

Table 1. Overall Ratings
Table 2. Satisfaction ratings

<table>
<thead>
<tr>
<th>StarRating</th>
<th>Efficacy</th>
<th>Sideeffects</th>
<th>Sideeffects</th>
<th>Easeofuse</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>quantity</td>
<td>severity</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>135</td>
<td>206</td>
<td>132</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>94</td>
<td>190</td>
<td>155</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>143</td>
<td>230</td>
<td>252</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>267</td>
<td>125</td>
<td>163</td>
<td>249</td>
</tr>
<tr>
<td>5</td>
<td>238</td>
<td>126</td>
<td>175</td>
<td>517</td>
</tr>
<tr>
<td>Total</td>
<td>877</td>
<td>877</td>
<td>877</td>
<td>877</td>
</tr>
</tbody>
</table>

5 Annotation

In this section, the procedure followed in the annotation of the DOS corpus is described. The corpus was annotated at the aspect-level with the side effects of the consumer drugs and their opinion polarity and opinion intensity about them. Annotation was performed using the brat tool [17], a web-based annotation tool, which was configured appropriately for the needs of the task. Firstly, the corpus was automatically tokenized into sentences using the NLTK Punkt sentence tokenizer for Spanish\(^8\). Secondly, after the creation of the initial annotation guidelines, 71 sentences corresponding to 10 reviews were tagged by the annotators together in order to refine the guidelines. Thirdly, a subset of the corpus was annotated in parallel (179 sentences corresponding to 50 reviews).

Table 3. Kappa inter-annotator agreement

<table>
<thead>
<tr>
<th>Element</th>
<th>%Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side effects</td>
<td>72.25</td>
</tr>
<tr>
<td>Polarity</td>
<td>92.01</td>
</tr>
<tr>
<td>Intensity</td>
<td>67.77</td>
</tr>
</tbody>
</table>

After this, a kappa inter-annotator agreement test between the 3 human annotators was conducted obtaining good results. Table 3 shows the inter-annotator

\(^8\) http://www.nltk.org/_modules/nltk/tokenize/punkt.html
agreement test for each of the following elements: side effects, opinion polarity and opinion intensity. We discussed disagreements and updated the guidelines. Finally, we proceeded to tag the remaining corpus.

6 Corpus Statistics

In this section, statistical data of the DOS corpus are presented. Table 4 shows general statistics related to the number of reviews, words and sentences; and also linguistic features (number and average of nouns, adjectives, verbs and adverbs). The corpus is composed of 877 reviews with a total of 3,784 sentences, with 18.14 being the average number of words per sentence.

Table 4. General statistics

<table>
<thead>
<tr>
<th>DOS corpus</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Reviews</td>
<td>877</td>
</tr>
<tr>
<td>#Sentences</td>
<td>3,784</td>
</tr>
<tr>
<td>#Words</td>
<td>68,627</td>
</tr>
<tr>
<td>#Nouns</td>
<td>17,576 (25.61%)</td>
</tr>
<tr>
<td>#Adjs</td>
<td>3,782 (5.51%)</td>
</tr>
<tr>
<td>#Advs</td>
<td>4,609 (6.72%)</td>
</tr>
<tr>
<td>#Verbs</td>
<td>11,917 (17.36%)</td>
</tr>
<tr>
<td>Avg. sentences per review</td>
<td>4.31</td>
</tr>
<tr>
<td>Avg. words per sentence</td>
<td>18.14</td>
</tr>
<tr>
<td>Avg. nouns per sentence</td>
<td>4.64</td>
</tr>
<tr>
<td>Avg. adjs per sentence</td>
<td>1.00</td>
</tr>
<tr>
<td>Avg. advs per sentence</td>
<td>1.22</td>
</tr>
<tr>
<td>Avg. verbs per sentence</td>
<td>3.15</td>
</tr>
</tbody>
</table>

The 3,784 sentences were manually tagged at aspect-level with the side effects described in them and with an opinion polarity label and an opinion intensity label according to the patients’ experiences. In Table 5, statistics related to side effects annotation can be seen. A total of 2,230 side effects were annotated, out of which 2,127 are related to the drugs that the patients specifically reviewed (SIDE_EFFECTS category) and 103 to those produced by other drugs that the reviewer took (OUT_OF_SCOPE_SIDE_EFFECTS). Most of the side effects are considered negative by the consumers but there are also some positive and neutral opinions about them (negative: 95.02%, positive: 4.40%, neutral: 0.58%).
In relation to the intensity of the side effects, the 29.37% have been annotated with a high level, the 66.64% with a medium level and only the 3.99% with a low level.

Table 5. Side Effects aspects per polarity and intensity

<table>
<thead>
<tr>
<th>Side Effects</th>
<th>Out of Scope Side Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Positive 22</td>
<td>1</td>
</tr>
<tr>
<td>Negative 594</td>
<td>33</td>
</tr>
<tr>
<td>Neutral 5</td>
<td>0</td>
</tr>
<tr>
<td>Total 621</td>
<td>34</td>
</tr>
</tbody>
</table>

7 Conclusions and further work

In this paper, we describe the construction of a corpus, called the DOS corpus, which has been manually annotated with drug side effects as well as their polarity and intensity. The quality of the annotation process has been assessed by a kappa inter-annotator agreement of 72.25% for the identification of side effects and of 92.01% and 67.77% for polarity and intensity classification respectively. To the best of our knowledge, this is the first Spanish annotated corpus for ABSA in the medical domain. We consider the creation of the corpus as the first step towards training a system for automatically extracting adverse drug effects and detecting its polarity.

Due to the fact that textual opinions can be grammatically incorrect and include spelling mistakes or informal expressions, we think that the annotation of the DOS corpus in order to detect efficacy and overall polarity opinion is a great challenge. We are in this process of annotation.

Our next aim is twofold: On the one hand, we plan to automatically normalize the different ways to refer to the same side effect to an overall concept. On the other hand, we intend to develop comparable corpus to those presented in this paper for other languages, such as English and Dutch, in which we have previous experience.

Acknowledgements

This work has been partially supported by a grant from the Ministerio de Educación Cultura y Deporte (MECD - scholarship FPU014/00983), Fondo Europeo
de Desarrollo Regional (FEDER) and REDES project (TIN2015-65136-C2-1-R) from the Spanish Government.

References

Analyzing cancer forum discussions with text mining

Gerard van Oortmerssen\textsuperscript{1}, Stephan Raaijmakers\textsuperscript{2}, Maya Sappelli\textsuperscript{2}, Erik Boertjes\textsuperscript{2}, Suzan Verberne\textsuperscript{3,4}, Nicole Walasek\textsuperscript{3}, Wessel Kraaij\textsuperscript{2,4}

\textsuperscript{1} Tilburg University
\textsuperscript{2} TNO, The Hague
\textsuperscript{3} Radboud University, Nijmegen
\textsuperscript{4} LIACS, Leiden University

Abstract. We present a multilingual, open source system for cancer forum thread analysis, equipped with a biomedical entity tagger and a module for textual summarization. This system allows users to investigate textual co-occurrences of biomedical entities in forum posts, and to browse through summaries of long discussions. It is applied to a number of online cancer patient fora, including a gastro-intestinal cancer forum and a breast cancer forum. We propose that the system can serve as an extra source of information for medical hypothesis formulation, and as a facility for boosting patient empowerment.

1 Introduction

Online patient communities are a potentially valuable source of information for cancer patients. In these communities, patients share detailed information on their disease, treatment, side effects of treatments and coping strategies, as well as their experienced quality of life. The aggregated information from the entire history of discussions can contribute to patient empowerment, but may also inspire clinical hypotheses. This short paper\textsuperscript{5} presents an open source system\textsuperscript{6} for the automated analysis of cancer forum posts supported by text mining.

2 System architecture

Our system has three main components: a pipeline for analyzing entities and relations in forum posts, a summarization module for summarizing discussion threads, and a user interface for explorative search. This section describes the three components in detail. The system currently contains data from three forum communities: the Dutch Breast Cancer community (BVN), the Facebook community \textit{GIST Support International}, and the medical section from the Dutch Viva forum\textsuperscript{7}. In order to display potential relations between elements such as

\textsuperscript{5} This work is supported by an SIDN Fund grant (https://www.sidnfonds.nl).
\textsuperscript{6} https://github.com/patientforumminer/PFM.
Fig. 1. Overview of the individual steps of the entity tagging pipeline.

side effects and treatments, textual entities (names and concepts) are tagged with their respective medical (semantic) categories, for both Dutch and English. As a first step, the system preprocesses forum post threads. Data is lower-cased, URLs and all non-alphanumeric characters are removed except for hyphens and commas. Weights and dosages are extracted from the threads based on regular expressions and directly tagged as such. After this step, all numbers are removed, followed by tokenization and stop word removal. Subsequently, a database lookup is executed for all single terms, using the Unified Medical Language System (UMLS) which is in English. Each Dutch term is first translated into English using a translation dictionary extracted from DBpedia. For every input term, the semantic types are extracted from UMLS and the most frequent one is chosen as the category for the term. If a particular term cannot be matched within the UMLS database, the DBpedia database is queried for that term and the most specific type is extracted. If this lookup also fails, the Medical Subject Headings (MeSH) database is queried and the term is matched to the first topical descriptor for an exact string match with the broader descriptor. The next step is to apply spelling correction to the remaining terms that have not yet been matched in the first look-up step. Only unmatched terms with a low frequency in the corpus (≤ 2) are considered. These either represent true misspellings or rare morphological variants of the unmatched types. For each of these low-frequency terms, the matched entity with the lowest weighted relative edit distance is determined. Character changes at the beginning of the term are prohibited. To increase the number of matched entities we include morphological variances based on lemma matching (using Pattern [2]) for terms with more than 4 characters. Moreover we expand the number of entities using contextual relations. For this purpose, a Word2Vec model [1] was trained on all data. The category of the majority of the 5 closest neighbors was assigned to each unmatched term, if a threshold of 3 was exceeded. Since lemma lookup and Word2Vec expansion are executed simultaneously, the results were merged. In case of disagreement, the result of Word2vec was preferred. As a final step, a selection of categories for the application was made. Moreover, high-frequency terms (such as ‘sleep’ or ‘live’) were excluded from the results, since tagging these terms was not relevant for the application. We evaluated the system in terms of the precision for the identi-

---

9 Parameter settings: model=CBOW, feature dimensionality= 500, window size=3, minimum word count=3, number of cores=3, other parameters are set to default values.
 fig. 2. A screenshot of the web-based interface with in the top-left corner the query box, below it the entity graph, and on the right side the results that are retrieved for the query (in Dutch).

fication and classification of the most frequent 300 matched entities. For the first
task, the average precision across annotators was 0.79 (inter-rater agreement in
terms of Cohen’s $\kappa = 0.77$). The classification task yielded an average weighted
precision per category of 0.74 ($\kappa = 0.67$).

The search module of the system returns posts in the context of a discussion
thread, often consisting of dozens or even hundreds of posts. We automatically
summarize the discussion threads with extractive summarization: showing the
most relevant sentences in the thread while hiding the less relevant sentences
in between them. Input for this summarization is a ranking of the sentences
by their relevance for a summary. For the prediction of relevance, we trained a
linear regression model on human reference summaries created for the English
and Dutch forum data. In the model, we used the number of raters that selected
a sentence as outcome variable (a sentence selected by 4 or 5 raters is more
relevant than a sentence selected by 1 or 2 raters). As independent variables we
used a number of generic sentence features such as the position in the thread,
the sentence length and the similarity with the full thread. We performed a
blind side-by-side comparison of the model’s summaries with human-created
summaries, which showed that our model’s summary was judged as equally good
as or better than the human-created summary [4].

The graphical user interface of the system allows for an iterative search pro-
cess in which the user quickly reaches relevant search results, supported by query
expansion, entity tagging and automatic thread summarization. Figure 2 shows
the system’s GUI. It is divided into two main parts, the left part supporting

---

10 We used 7 raters: 5 non-experts and two experts. More information about the refer-
ence summaries and the summarization module was published in [3].
the querying process, the right part for browsing search results. The user typically starts with entering one or more keywords (upper left), that are expanded with related terms by the system using Word2Vec trained on all patient forum data. The system presents the expansion terms to the user, allowing to decide whether or not to include these terms in a next query. The term network shows terms that occur in the search results and their inter-connection. Two terms are connected when they co-occur frequently in the same context (e.g. message) in the result set. The strength of the relationship is depicted by the thickness of the edge in the network graph. Each node is coloured according to its classification (e.g. medicine, food, symptom), and a node’s size is proportional to the number of occurrences in the results. The network facilitates the discovery of unexpected links between terms (e.g. a food substance mentioned frequently in combination with a symptom). This is typically useful for expert users looking for new medical hypotheses. The right hand side of the GUI shows the search results. These are threads, consisting of the first message in a thread, followed by a list of comments on that message. The opening post of the thread is always shown; sentences of other posts in the thread are only shown if the user prefers to see more detail, governed by the slider on top of the screen. Which sentences are shown first when the slider is moved to the right is decided by the summarization module.

3 Conclusions

We have presented an open source system for the automated analysis and interactive inspection of cancer forum posts. For the purpose of knowledge extraction from patient-generated forum data, our future work will focus on techniques for matching new forum threads with existing ones, and on connecting user-generated content to moderated content, such as curated taxonomies and published medical information. Further, we will address query relevance for summarization, and the thorough evaluation of our system in task-based settings.

References

Design of an Information Extraction Pipeline for German Clinical Texts

Madeleine Kittner¹, Bariya Bajwa², Damian Rieke³, Mario Lamping³, Johannes Starlinger¹ and Ulf Leser¹

¹ Humboldt Universität zu Berlin, Germany, {surname}@informatik.hu-berlin.de
² Universitätsklinikum Tübingen, Germany, Bariya.Bajwa@med.uni-tuebingen.de
³ Charité - Universitätsmedizin Berlin, Germany, {firstname.surname}@charite.de

Abstract. We present a pipeline under development for extracting information of progression and therapy in cancer from German discharge summaries. This pipeline intends to cope with current limitations regarding data, tools and terminologies for processing German medical text. Main components of the pipeline include anonymization of documents, annotation of a gold standard, and combination of tools for information extraction. We support time consuming steps like anonymization and annotation using pre-annotations of relevant entities. In parallel, lacking a gold standard corpus for detailed evaluation, we are testing available terminologies and tools for text processing.

Keywords: clinical information extraction, German medical text

1 Introduction

Clinical documentation is essential for planning, provision, accounting, and review of decisions and treatment orders in patient care. Comprehensive analysis of the information contained in clinical documents is highly desirable, but requires structured and semantically unambiguous data. Most documentation is done as free text in different types of medical reports, and needs clinical NLP systems to extract medical events, entities, and temporal relations. For extraction of medical entities from English corpora, several state-of-the-art tools such as MedLEE, MetaMap and cTakes exist. These are based on dictionary look-up using the Unified Medical Language System (UMLS) which combines several established dictionaries and provides a unique identifier for each concept. Machine-learning approaches have also been applied for recognizing medically relevant entities. Currently, some of the best performing systems for medical information extraction use hidden markov models, conditional random fields, and support vector machines [1].

Processing German medical text is more challenging in terms of data, tools, and terminologies. Currently, no German medical text corpus is available for academic research. A major reason is the lack of a standard definition of protected health information and stringent privacy criteria for working with narrative data. Each clinical NLP project needs to develop its own guidelines, tool chain, and
evaluation corpora which is a costly process. As such, availability of tools to process German medical text is limited. To the best of our knowledge, only the Julie Lab published respective tools, trained on an undisclosed corpus [2]. Available German terminologies also are limited, impeding concept recognition and normalization. Only few German translations are included in UMLS or are provided by the German institute for medical documentation (DIMDI) [3]. However, work on German biomedical text has been done in previous projects such as Khresmoi or Theseus Medico as well as in the current German smart data initiative KDI.

The paper first briefly introduces the purpose of our work and recapitulates main challenges for information extraction (IE) from German medical text. Next, we present the design of an IE pipeline.

2 Challenges

Our work aims to provide structured information about progression, metastasis, and therapy in cancer by extracting relevant medical entities and their temporal relations from discharge summaries in German. As part of the PersOnS project, extracted information is integrated with mutation profiles and literature data. Additionally, providing intuitive access to the collected knowledge through a touch-based analytics tool PersOnS aims to support decision making during molecular tumor board meetings.

Developing an IE pipeline for German medical text is challenging [4]. As shown in Figure 1, several steps are important and require close collaboration with clinical staff: formulating stringent privacy criteria in agreement with data security officers, anonymization of medical documents according to these criteria, and annotation of a gold standard corpus for evaluation. Additionally, only few tools to process German medical text are available and the coverage of German terminologies is low. In the following, we present our ideas on solving these tasks.

3 Design of an IE pipeline

Discharge summaries of patients suffering from melanoma or hepatocellular carcinoma will be collected at the Comprehensive Cancer Centers of University Hospital Charité Berlin and University Hospital Tübingen.

In collaboration with data security officers we developed a stringent anonymization protocol, according to which, the following data is removed: names of patients, doctors, and other persons mentioned as well as any kind of location information (hospital names, addresses, phone numbers etc.). All dates within a document are changed by a random number of days, retaining temporal relations between events. Secondary diseases not linked to the main disease are randomly substituted with another likely secondary disease. Similarly, left and right are swapped for anatomical locations like lobe of the liver. Although this makes no sense from a medical point of view it generates a high level anonymized corpus which still contains the original variety of all medical parameters.
In one hospital, anonymization is performed by clinicians manually, and supported by pre-annotation of private data in the other hospital, see Figure 1. Pre-annotations are obtained using regular expressions and the annotation tool Ellogon\(^4\). Clinicians check whether pre-annotations are correct, false or missing. Both approaches require examination of anonymized documents by a (second) staff member. On our small corpus, we expect both approaches to be similarly expensive in terms of time: Pre-annotations speed up anonymization, but need to be optimized first. On a large corpus, pre-annotation would clearly save time compared to manual anonymization. In terms of accuracy, preliminary analysis indicates that pre-annotation supported anonymization performs better.

Next, a gold standard corpus will be developed by annotating medical entities using UMLS semantic types and identifiers. In particular, we focus on the semantic types Disorders (for diseases, abnormalities or symptoms), Anatomy (for body parts or organs), Procedures (for diagnostic or therapeutic procedures), and Chemicals and Drugs (for medications). We will also annotate dates and temporal expressions like after 24 hours. A detailed annotation guideline is in development. We expect the low coverage of German terminologies to be a major challenge. A single dictionary may not contain the most exact annotation of a term, or even no annotation at all. Therefore, we will include several dictionaries: German vocabularies provided by UMLS and DIMDI as well as English vocabularies included in UMLS.

We will use the annotation tool brat\(^5\) and pre-annotate concepts from the defined semantic types using the prototype IE pipeline described in the following. Clinical staff validates pre-annotations. This leads to significant time savings without sacrificing annotation accuracy [5].

We are developing a prototype IE pipeline including general modules of text processing as shown in Figure 1. Without accurate performance evaluation due to the lack of an annotated corpus we tested available tools using a subset of our

---

4 http://www.ellogon.org/index.php/annotation-tool
5 http://brat.nlplab.org
We developed a simple rule-based sentence splitter as available splitters did not deal well with missing punctuation present in the corpus. We use JCORE tools [2] for tokenization and part-of-speech tagging and the German Snowball Stemmer [6]. Named entity recognition and normalization is based on dictionary lookup and follows a stepwise matching strategy. All possible n-grams ($n < 5$) are compared to the dictionary by i) exact matching, ii) matching stemmed terms, iii) fuzzy matching allowing an edit distance of 1 per token with length > 5 characters using Apache Solr. For negation detection, NegEx will be used which has been applied successfully to German clinical notes [7]. Detailed evaluation of this pipeline will be performed once the gold standard corpus is ready.

4 Discussion and Conclusion

We highlighted difficulties of clinical information extraction for German discharge summaries and presented a design of an IE pipeline under development. Since anonymization and annotation of documents by clinical experts is very expensive, we use pre-annotations of private data and medical entities. We expect annotation and extraction to be challenging as available German vocabularies provide only limited coverage. We intend to overcome this problem by integrating several German and English dictionaries and using cross-lingual matching. Alternatively, generation of synonyms, translation of terms to English or edit distance measures such as Levenshtein or Jaccard could be helpful. Additionally, we will consider application of NLP methods available within GATE or the DKPro Core collection.

Acknowledgment: This work is part of PersOnS (Personalizing Oncology via Semantic Integration of Data) and funded by BMBF.

References

4. Starlinger, J., Kittner, M., Blankenstein, O., Leser, U.: How to improve information extraction from german medical records. it-Information Technology (2016)