



2 0 1 3

Saratoga Springs, NY - USA 8 - 11 July 2013

Twenty-First International Conference on
Case-Based Reasoning
(ICCBR 2013)

Workshop Proceedings

Michael W. Floyd and Jonathan Rubin (Editors)

**Proceedings of the ICCBR 2013
Workshops**

Michael W. Floyd and Jonathan Rubin (Editors)

Preface

We are pleased to present the workshop proceedings from the Twenty-First International Conference on Case-Based Reasoning (ICCBR 2013) held in Saratoga Springs, USA. Over the years, ICCBR workshops have provided an informal setting where participants have the opportunity to discuss specific focus topics in an atmosphere that fosters the active exchange of ideas. This year's proceedings includes papers from three workshops:

- *Case-based Reasoning in Social Web Applications* is a new workshop that looks to highlight how CBR can be used to support and make use of online social interactions. With the growing number of social applications that exist, the workshop aims to bring together a variety of researchers to exchange information and ideas.
- *Case-Based Reasoning in the Health Sciences* is a continuation of a series of successful workshops that cover a variety of areas related to CBR in the health sciences. This includes identifying opportunities for CBR, showcasing applications, and presenting important research results.
- *EXPPORT: EXperience reuse: Provenance, Process-ORientation and Traces* is also a new workshop and aims to bring together researchers that have been focusing on the areas of provenance, process-oriented CBR and traces. Although these areas are often studied on their own, the organizers look to highlight the close relationship and encourage a transfer of ideas.

These proceedings also include the research summaries of students who participated in the Fifth ICCBR Doctoral Consortium. This event allows students to present their doctoral research, interact with senior CBR researchers who serve as mentors, and receive valuable feedback on future research goals and directions. We conclude the proceedings with a paper that motivates the need for reproducible CBR research and proposes a method to evaluate the reproducibility of ICCBR 2013 papers.

Many people contributed to the success of the ICCBR 2013 workshops. We would like to thank the workshop and doctoral consortium organizers who put in a significant effort to organize their events, solicit submissions, coordinate the peer review process, and select high-quality submissions for publication and presentation. Additionally, we would like to thank the authors for their submissions and the various program committee members for reviewing those submissions. Without the time and effort of these people, the ICCBR 2013 workshops would not have been possible.

The ICCBR 2013 conference chairs, Sarah-Jane Delany and Santiago Ontañón, also deserve our thanks for the innumerable things they did to make the conference happen. Additionally, we wish to thank the local chair William Cheetham and Aisha Yousuf for their valuable work coordinating the arrangements in Saratoga Springs. We hope everyone enjoys the ICCBR 2013 workshops!

Michael W. Floyd and Jonathan Rubin

May 2013

Table of Contents

Case-based Reasoning in Social Web Applications

Preface	1
<i>Lara Quijano-Sánchez and Derek Bridge</i>	
Collaboration over the web and experience sharing: what challenges?	5
<i>Amélie Cordier</i>	
The Arguments of the Crowd	6
<i>Enric Plaza</i>	
Mining Experiential Product Cases	7
<i>Ruihai Dong, Markus Schaal, Michael P. O'Mahony and Barry Smyth</i>	
Feedback on group recommendations	17
<i>Lara Quijano-Sánchez and Derek Bridge</i>	
Question Routing in Collaborative Question Answering Systems	27
<i>Nishaanth Shanmughasundaram and Sutanu Chakraborti</i>	

Case-based Reasoning in the Health Sciences

Preface	37
<i>Isabelle Bichindaritz, Cindy Marling and Stefania Montani</i>	
Protein Structure Retrieval Using Preference-Based CBR	40
<i>Amira Abdel-Aziz, Marc Strickert, Thomas Fober and Eyke Hüllermeier</i>	
Model-based Classification of Unstructured Data Sources	50
<i>Kerstin Bach and Klaus-Dieter Althoff</i>	

Medical Literature Mining for Case-based Reasoning in the Biology
of Aging..... 60
Isabelle Bichindaritz

**EXPORT: EXperience reuse: Provenance, Process-ORientation
and Traces**

Preface..... 73
*David Leake, Béatrice Fuchs, Stefania Montani and
Juan A. Recio-García*

Process mining and case-based retrieval for assessing the quality of
medical processes..... 77
*Stefania Montani, Giorgio Leonardi, Silvana Quaglini, Anna Cav-
allini and Giuseppe Micieli*

Anaphora resolution in a pipes-and-filters framework for workflow
extraction..... 87
Pol Schumacher, Mirjam Minor and Eric Schulte-Zurhausen

A Case-based Reasoning Approach to Business Workflow Modelling
Based on Formal Temporal Theory..... 97
Stelios Kapetanakis, Miltos Petridis, Jixin Ma and Brian Knight

An approach for collecting fine-grained use traces in any application
without modifying it..... 107
*Blandine Ginon, Pierre-Antoine Champin and Stéphanie Jean-
Daubias*

Building a Trace-Based System for Real-Time Strategy Game
Traces..... 117
Stefan Wender, Amélie Cordier and Ian Watson

Toward Addressing Noise and Redundancies for Cases Captured
from Traces and Provenance..... 127
David Leake and Joseph Kendall-Morwick

The ICCBR 2013 Doctoral Consortium

Preface	131
<i>Thomas Roth-Berghofer and Rosina Weber</i>	
Preference-Based Case Based Reasoning	134
<i>Amira Abdel-Aziz</i>	
Case-Based Learning of Ontology-Based Goal-Driven Autonomy Knowledge	137
<i>Dustin Dannenhauer</i>	
Recommending Research Profiles for Multidisciplinary Academic Col- laboration	140
<i>Sidath Gunawardena</i>	
Using Ensembles of Adaptations for Case-Based Reasoning ...	143
<i>Vahid Jalali</i>	
Workflow Extraction from Textual Process Descriptions.....	146
<i>Pol Schumacher</i>	
A Case-Based Reasoning Approach to Text Generation	149
<i>Josep Valls-Vargas</i>	
Towards an Artificial Teammate for Supporting and Conducting Ar- guments with Analogies and Cases in Biologically Inspired Design.....	152
<i>Bryan Wiltgen</i>	
Scientific Reproducibility	
A Reproducibility Process for Case-Based Reasoning.....	156
<i>David W. Aha and Odd Erik Gundersen</i>	

Case-based Reasoning in Social Web Applications

Workshop at the
Twenty-first International Conference on
Case-Based Reasoning
(ICCBR 2013)

Saratoga Springs, NY, USA.
July, 2013

Lara Quijano-Sánchez, and Derek Bridge (Eds.)

Co-Chairs

Lara Quijano-Sánchez
Universidad Complutense de Madrid, Spain

Derek Bridge
University College Cork, Ireland

Programme Committee

Ralph Bergmann, University of Trier, Germany
Robin Burke, DePaul University, USA
Amélie Cordier, University Claude Bernard Lyon 1, France
Belén Díaz-Agudo, Complutense University of Madrid, Spain
Jill Freyne, CSIRO, Australia
Conor Hayes, National University of Ireland, Galway, Ireland
Enric Plaza, IIIA-CSIC, Barcelona, Spain
Barry Smyth, University College Dublin, Ireland
David C. Wilson, University of North Carolina, USA

Preface

The Social Web describes how World Wide Web software supports and fosters social interaction. These social interactions form the basis of much online activity including online shopping, education, gaming and social networking. Today hundreds of millions of Internet users regularly visit thousands of social websites to stay connected with their friends, discover new friends, and to share user-created content, such as photos, videos, reviews and commentary. The Social Web is quickly reinventing itself, moving beyond simple web applications that connect individuals to become an entirely new way of life.

The ability to harness and reuse these online experiences has a tremendous potential. One of the main communities that can take advantage of these online experiences is the CBR community, which has been devoted to exploring different aspects of reasoning from experiences for more than twenty years. CBR techniques can exploit the data that people share and store online to recommend new places, events and information.

This workshop on case-based reasoning in social web applications is the first to be held at ICCBR. Within the CBR community, interest has come from the many research groups that have an existing record of research in areas such as recommender systems, personalized search, sentiment analysis, social network analysis, and so on. The motivation for this workshop was to provide a forum for the exchange of information and ideas among CBR researchers working in this area, and to share challenge tasks which can act as benchmarks for comparison among systems in the field.

The workshop program is opened and closed by invited presentations. Amélie Cordier reflects on the lessons learned from her involvement in three applications for supporting web-based collaboration, all of which use some CBR. Enric Plaza looks at the role of CBR and other technologies in social argumentation for participatory political processes.

The programme also includes four refereed papers representing various approaches to CBR in social web applications. The paper by Kerstin Bach & Klaus-Dieter Althoff (which can be found in the section of these proceedings that covers the Workshop on Case-based Reasoning in Health Sciences) deals with text processing to label contributions to an expert forum. The approach consists of classifying unstructured data into topics in the domain of travel medicine. Ruihai Dong, Markus Schaal Michael O'Mahony & Barry Smyth describe a technique for mining user reviews to extract product features not available from ordinary catalog data, where the motivation is the use of user experiences in a case based product recommendation system. The paper from Lara Quijano-Sánchez & Derek Bridge discusses the issue of user feedback in group recommender systems. The issues are explored in part through an illustrative user study experiment. Finally, Nishaanth Shanmugasundaram & Sutanu Chakraborti present a technique for routing questions to users in a collaborative question answering system. They describe various approaches based on relevance and workload.

Overall, these papers represent a good sample of case-based reasoning in social web applications, and we expect the workshop discussions to further clarify and advance work in this area.

We would like to thank everyone who contributed to the success of this workshop, especially Amélie Cordier and Enric Plaza (our invited speakers), the authors, the program committee members, Barry Smyth (who managed the reviewing of our own paper), Michael Floyd and Jonathan Rubin (the Workshop Chairs), and all the organizers of the ICCBR 2013 conference.

Lara Quijano-Sánchez
Derek Bridge

July 2013

Collaboration over the web and experience sharing: what challenges?

Amélie Cordier

Université Lyon 1, CNRS, LIRIS, UMR5205, F-69622, France
amelie.cordier@liris.cnrs.fr,
<http://liris.cnrs.fr/amelie.cordier>

The ongoing development of web technologies, social web tools and semantic web opens many application opportunities. One of these opportunities is the possibility of developing tools allowing users to work together to accomplish specific tasks, solve problems and build knowledge.

During this talk, we will present three examples of applications in which we use tools of the social web to support collaboration between users performing complex tasks. The first application, Wikitaaable, allows users to collaborate to build culinary knowledge which is then used by a case-based reasoning engine. The second application, Wanaclip, allows interactive construction of video clips, and provides users with social recommendations. Interaction traces of previous users are used in order to build contextual recommendations. The third application, Ozalid, is a collaborative tool for correcting and enriching digital documents. The main characteristic of this tool is that user activity is guided and supervised through a dedicated social network.

Throughout the presentation, we will show how case-based reasoning has been used to in the different applications, and will report on successes and failures encountered during the development of these applications. Drawing inspiration from these experiences, we will then discuss open problems and challenges for case-based reasoning in social applications.

The Arguments of the Crowd

Enric Plaza

IIIA-CSIC, Barcelona, Spain
enric@iia.csic.es,
<http://www.iia.csic.es/~enric/>

The wisdom of the crowd, and data mining in the social web, derives its power from Condorcet's Jury Theorem in the pursuit of truth. However, other domains where social groups pursue collective goals cannot be modelled as searching for truth. Such domains as participatory political processes in public or private institutions can better be modelled as a deliberative process where different arguments are proposed, attacked, and ratified. How can we deal with these issues? This talk presents some tentative proposals we are currently studying to address Social Argumentation in the field of participatory political processes.

Mining Experiential Product Cases*

Ruihai Dong, Markus Schaal, Michael P. O’Mahony, and Barry Smyth

CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Ireland

Abstract. Case-based reasoning (CBR) attempts to reuse past *experiences* to solve new problems. CBR ideas are commonplace in recommendation systems, which rely on the similarity between *product queries* and a case base of *product cases*. But, the relationship between CBR and many of these recommenders can be tenuous: the idea that product cases made up of static meta-data type features are *experiential* is a stretch; unless one views the type of case descriptions used by collaborative filtering (user ratings across products) as experiential. Here we explore and evaluate how to automatically generate product cases from user-generated reviews to produce cases that are based on genuine user *experiences* for use in a case-based product recommendation system.

1 Introduction

Consider the 13” *MacBook Pro*. At the time of writing the *product features* listed by Amazon cover technical details such as *screen-size*, *RAM*, and *price*. These are the type of features found in a conventional product recommender. But such features can be few in number – which limits how we assess inter-product similarity at recommendation time – and they can be technical in nature, making it difficult to judge the importance of similarities in any practical sense. However, the *MacBook Pro* has 72 reviews which encode valuable insights into a great many of its other features, from its “*beautiful design*” to its “*high price*”. These capture more detail than a handful of technical features. They also encode the *opinions* of users and, as such, provide an objective basis for comparison.

Can we use such ‘social’ features — features from the collective experiences of users — as the basis for a type of product case, an *experiential product case*? Do such cases represent a viable alternative to more conventional cases made up of catalog features (see [1] for example)? Are such cases rich enough to serve a useful function when it comes to product recommendation? What types of similarity and weighting techniques might we apply to these cases? We will consider these matters in the remainder of this paper as we describe our approach to mining experiential cases and their use in a product recommender system.

We are not the first to consider this kind of approach. For example the work of [8] describes the use of shallow NLP for explicit feature extraction and sentiment

* This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

analysis; see also [3, 4, 10]. The features extracted, and the techniques used to extract them, are similar to those presented here, although in other work they are extracted for the purpose of product description and ranking rather than recommendation. The work of [12] also analyzes the sentiment of comparative and subjective sentences in reviews on a per-feature basis to create an ordering of products, but without considering the recommendation task with respect to a query product. Moreover, our work is related to recent work on textual case-based reasoning [11] and the challenges of harnessing experiential knowledge in many forms from web content as proposed in [7]. In the case of the latter, our work represents a concrete instantiation of such a system, by harnessing product experiences for the purpose of product recommendation.

2 Mining Experiential Product Cases

The aim of this work is to implement a practical technique for converting user-generated product reviews into rich, feature-based, experiential product cases. The features of these cases relate to topics that are discussed by reviewers and the values of these features reflect the aggregate opinions of these reviewers. Our approach is summarised in Figure 1 for a given product P : (1) we use shallow NLP techniques to extract a set of candidate features from $Reviews(P) = \{R_1, \dots, R_K\}$, the reviews of P ; (2) each feature F_i is associated with a sentiment label, L_k , (*positive*, *negative*, or *neutral*) based on the opinion expressed in a review R_j for P ; and (3) these topics and sentiment scores are aggregated at the product level to generate a case of features and aggregate sentiment scores.

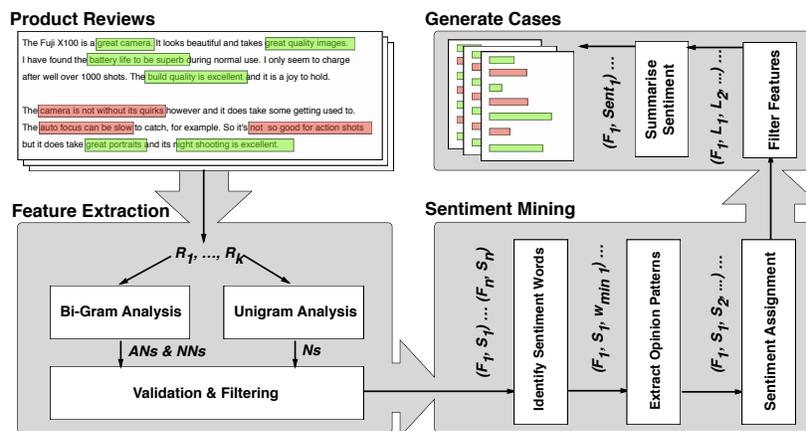


Fig. 1. Extracting experiential product cases from user-generated reviews.

2.1 Extracting Review Features

We consider two types of features — *bi-gram* features and *single-noun* features — and use a combination of shallow NLP and statistical methods, combining ideas from [3, 5] to mine them. For the former we look for bi-grams in reviews which conform to one of two basic part-of-speech patterns: (1) an adjective followed by a noun (*AN*) (e.g. *wide angle*); or (2) a noun followed by a noun (*NN*) (e.g. *video mode*). These feature candidates are filtered to avoid including *AN*'s that are actually opinionated single-noun features; e.g. *great flash* is really a single-noun feature, *flash*. To do this we exclude bi-grams whose adjective is a sentiment word (e.g. *excellent*, *terrible* etc.) in the sentiment lexicon of [4].

For single-noun features we also extract a candidate set, this time of nouns, from the reviews but we validate them by eliminating nouns that are rarely associated with opinionated words as per the work of Hu and Liu [4]. The reason is that such nouns are unlikely to refer to product features. We calculate how frequently each noun co-occurs with a sentiment word in the same sentence (using the sentiment lexicon of [4]), and retain a single-noun only if its frequency is greater than some fixed threshold (in this case 70%).

2.2 Evaluating Feature Sentiment

To calculate sentiment we use a version of the *opinion pattern mining* technique [6] to extract opinions from product reviews. For a feature F_i , and corresponding review sentence S_j in review $R \in \{R_1, \dots, R_K\}$, we determine whether there are any sentiment words in S_j . If there are none then this feature is labeled *neutral*. Otherwise we identify the sentiment word w_{min} which is closest to F_i .

Next we identify the part-of-speech (POS) tags for w_{min} , F_i and any words that occur between w_{min} and F_i . This POS sequence corresponds to an *opinion pattern*. For example, in the case of the bi-gram topic *screen quality* and the review sentence, “...*this tablet has excellent screen quality*...”, then w_{min} is the word “*excellent*” which corresponds to an opinion pattern of *JJ-TOPIC* [6].

After a full pass of all features we compute the frequency of the recorded opinion patterns. A pattern is *valid* if it occurs more than once. For valid patterns we assign sentiment based on the sentiment of w_{min} and subject to whether S_j contains any negation terms within a 4-word-distance of w_{min} . If there are no such negation terms then the sentiment assigned to F_i in S_j is that of the sentiment word in the sentiment lexicon. Otherwise the sentiment is reversed. If an opinion pattern is deemed not to be valid (based on its frequency) then we assign a *neutral* sentiment to each of its occurrences within the review set.

2.3 Generating Experiential Product Cases

For each product P we have a set of features $F(P) = \{F_1, \dots, F_m\}$ mined from $Reviews(P)$, and for each feature $F_i \in F(P)$ we have a set of *positive*, *negative*, or *neutral* sentiment labels (L_1, L_2, \dots) extracted from the particular reviews in $Reviews(P)$ that discuss F_i . Here we only include features in our cases if

they occur in at least 10% of reviews for product P . For these features we calculate an overall sentiment score. The case, $Case(P)$, is then constructed from these scored features as per Equations 1 and 2. Note, $Pos(F_i, P)$, $Neg(F_i, P)$, and $Neutral(F_i, P)$ denote the number of times that feature F_i is associated with positive, negative and neutral sentiment in the reviews for product P , respectively.

$$Case(P) = \{(F_i, Sent(F_i, P)) : F_i \in F(P)\} \quad (1)$$

$$Sent(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neutral(F_i, P)} \quad (2)$$

3 Recommending Similar Products

Our scenario for recommending similar products is that there is a query case Q that represents the user’s interests. Q may have been obtained by interrogating the user as to their needs (the features they are looking for) or it may be a product case that they have identified as interesting. Regardless, as with typical approaches to case-based recommendation we will assume Q to be the starting point for a *more-like-this* style of recommendation. We can compare cases using conventional approaches to similarity for a similarity-based retrieval approach to product recommendation. We do this in two stages: (1) the *retrieval stage* identifies a set of candidate cases based on some minimal feature overlap with some query case Q ; and (2) in the *ranking stage* these cases are then ranked for recommendation based on some suitable similarity metric.

3.1 Case Retrieval and k-Comparability

Feature-based approaches to case retrieval usually rely on shared features between the query and candidate cases. This is usually straightforward because in most CBR scenarios there is a stable feature set underpinning case descriptions. However, in this work our cases do not have fixed features and so it is not possible to guarantee feature overlap. This might be problematic if it leads to cases being retrieved that have limited overlap. We define *k-comparable* (see Equations 3 and Equation 4) as a retrieval constraint to ensure some minimal set of shared features between cases from the case base CB at retrieval time.

$$k\text{-comparable}(C', C'') \iff |F(C') \cap F(C'')| \geq k \quad (3)$$

$$Retrieve_k(Q) = \{C_p \in CB : k\text{-comparable}(Q, C_p)\} \quad (4)$$

3.2 Similarity Ranking

Once we have a set of *k-comparable* cases we can rank them by their similarity to the query product case and return the top n as recommendations. For example, Equations 5 and 6 show standard versions of *Jaccard* and *Cosine* similarity metrics; we refer to these as J and C in our evaluation. Note, $F(Q)$ and $F(C_p)$ refer

to the features of Q and C_p , respectively, while $Sent(F_i, Q)$ and $Sent(F_i, C_p)$ refer to the sentiment value of F_i in Q and C_p , respectively. We assume that missing features have a zero sentiment for the purpose of the *Cosine* metric.

$$Sim_J(Q, C_p) = \frac{|F(Q) \cap F(C_p)|}{|F(Q) \cup F(C_p)|} \quad (5)$$

$$Sim_C(Q, C_p) = \frac{\sum_{F_i \in F(Q) \cup F(C_p)} Sent(F_i, Q) \times Sent(F_i, C_p)}{\sqrt{\sum_{F_i \in F(Q)} Sent(F_i, Q)^2} \times \sqrt{\sum_{F_i \in F(C_p)} Sent(F_i, C_p)^2}} \quad (6)$$

We also produce weighted versions of the above by computing the weight of a (query or case) feature F_i by the fraction of reviews containing it for a given product P (Equation 7). In these locally weighted versions of *Jaccard* (wJ) and *Cosine* (wC) we need to deal with query and case features that are not shared. For example, for features that are unique to the query we use the weights of these features from the query reviews whereas, for product case features, shared and unique, we use weights based on the product case’s reviews.

$$w(F_i, P) = \frac{|\{R \in Reviews(P) : F_i \in R\}|}{|Reviews(P)|} \quad (7)$$

All of this provides a straightforward approach to product recommendation: given a target product as a query Q , recommend the n most similar cases.

4 Evaluation

Can we extract useful case descriptions that provide for a rich set of features? Further, are these case descriptions suitable in a product recommendation setting? We now consider both questions as part of a multi-domain evaluation.

4.1 Datasets

The data for this experiment was extracted from Amazon.com during October 2012. We focused on 4 product categories: *GPS Devices*, *Laptops*, *Printers*, and *Tablets*. In fact, we have analysed 6 different product categories in total with similar results, but for reasons of space we consider only the 4 mentioned above. We focus on products with at least 10 reviews and there was no manual editing of features. Table 1 summarises our datasets and the results of case extraction.

4.2 Feature Extraction Results

We can see from Table 1 that our mining technique is finding many features for different product types. Figure 2 shows more detailed histograms of feature counts. For example, *Laptop* cases (Figure 2(b)) contain a wide range of features,

Category	#Reviews	#Prod.	#Prod. (≥ 10 reviews)	#Features: Mean (Std. Dev.)
GPS	12,115	192	119	24.32 (10.82)
Laptops	12,431	785	314	28.60 (15.21)
Printers	24,369	336	233	16.89 (7.58)
Tablets	17,936	291	166	26.15 (10.48)

Table 1. A summary of product data and case bases.

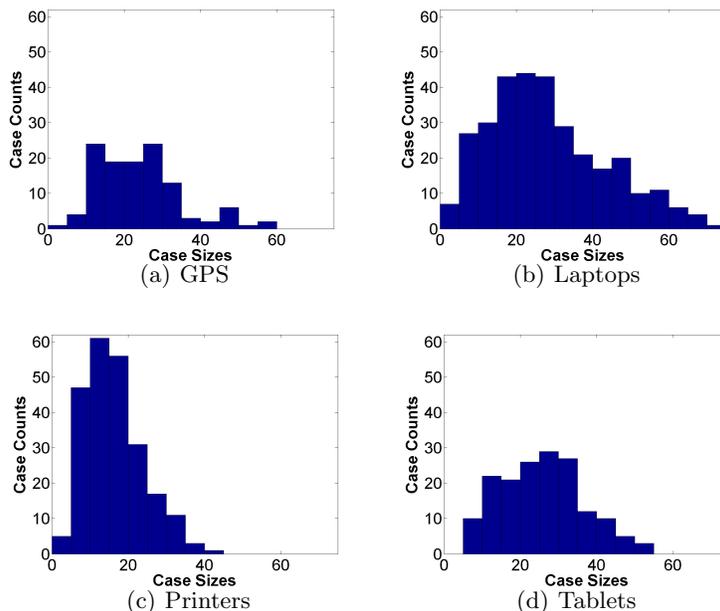


Fig. 2. Feature histograms by case base.

from small cases with very limited features sets of less than 10 to cases with as many as 70 features; the majority of cases contain somewhere between 15 and 30 features. In contrast *Printer* cases (Figure 2(c)) reflect a much narrower distribution; most have 10–20 features and very few have more than 30 features.

Thus we can expect feature-rich cases from our reviews. But this is of limited use unless these features are shared with other cases. Few shared features limits our ability to compare cases; it is akin to the *sparsity problem* in collaborative filtering systems [9]. In [2] we explore these overlap characteristics by examining the average size of the k -comparable sets for different levels of k to find a high level of feature sharing across all case bases. For example, at $k = 15$ we find the mean number of k -comparable cases to be 35% of a case base. With so many cases sharing at least k features, even for large values of k , we must be extracting features that are frequently recurring in reviews.

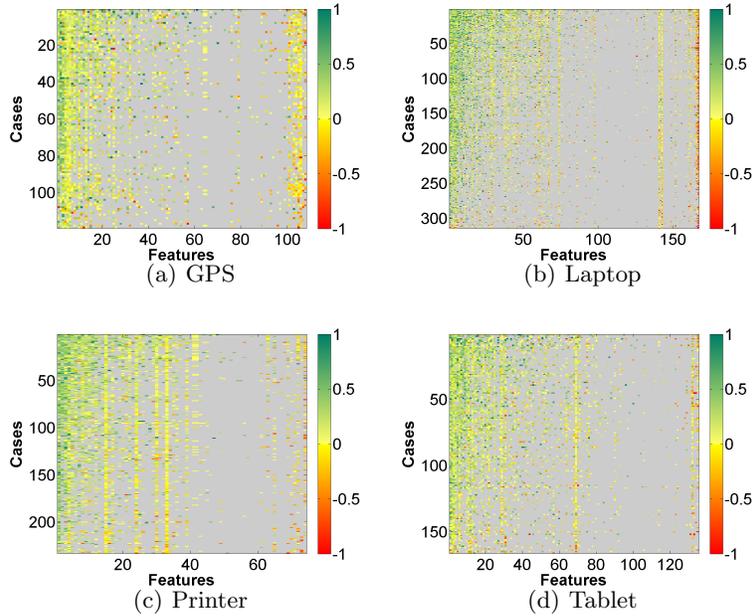


Fig. 3. Sentiment heatmaps by case base; the authors acknowledge the limitation that these are best viewed in colour.

4.3 Sentiment Maps

Figure 3 shows sentiment heatmaps for each of the 4 case bases. Rows correspond to product cases and columns to features. The sentiment of a particular feature is indicated by colour, from red (strong negative sentiment) to green (strong positive); missing features are shown in grey. In this instance both the feature columns and the product rows have been sorted by aggregate sentiment.

There are a number of observations to make. First, because of the ordering of the features we can clearly see that features with relatively high (leftmost) and low (rightmost) sentiment scores also tend to elicit the most opinions from reviewers; the leftmost and rightmost regions of the heatmaps are the most densely populated. By and large there is a strong review bias towards positive or neutral review opinions; there are far more green and yellow cells than red. Some features are almost universally liked or disliked. For example, for *Laptops*, *price*, *screen* and *battery life* all attract positive sentiment. In contrast, features such as *wifi* and *fan noise* are among the most universally disliked *Laptop* features. Across all 4 product domains, *price* features highly, suggesting perhaps that modern consumer electronics pricing models are an excellent fit to consumer needs, at least currently.

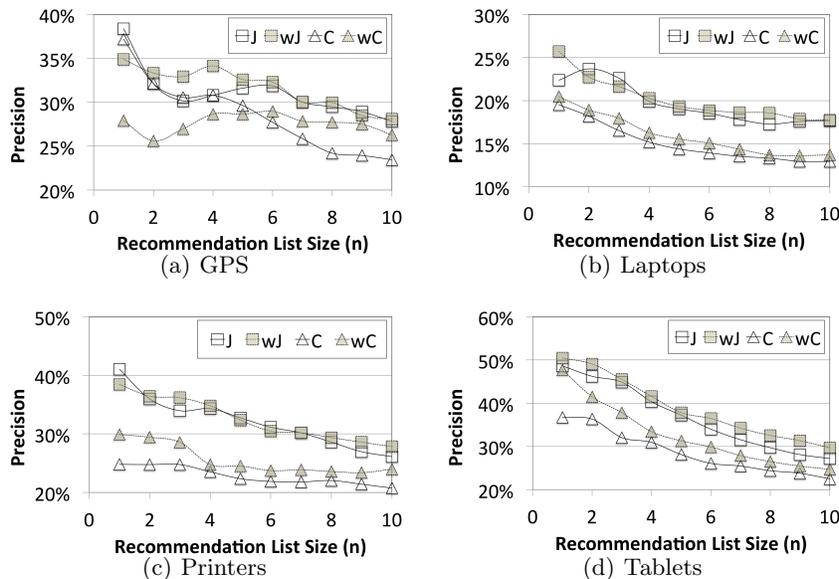


Fig. 4. Precision results for different recommendation list sizes by case base.

4.4 Recommendation Precision

In this section we compare the performance of our 4 retrieval strategies across the product domains in a standard *leave-one-out* recommendation test within each case base. Briefly, each case is selected in turn as a *target query* and we retrieve the top n most similar cases from the remaining cases as recommendations; in this test we focus on a *k-comparable* level of 15 (that is $k = 15$). To evaluate the quality of these n recommendations, we use Amazon’s own recommendations for each target query as the ground-truth, calculating the percentage of recommendations that match Amazon’s for a simple precision-like metric.

The above approach, however, does not provide an ideal ground-truth. Amazon’s own recommendations often include products of different types (e.g. a digital camera may cause a memory card to be recommended) because they are based on purchase patterns rather than any strong sense of product similarity. Moreover, even without this limitation, do Amazon’s own recommendations represent a good objective test? Is it a good thing to have higher precision, for example, or should we look at something like the average review score of recommended products as an alternative ground-truth? We will return to this presently but for now let us use this precision metric as a useful starting point.

The precision results are shown in Figure 4 as graphs of average precision versus recommendation list size (n). In each case we can see reasonably high-levels of precision (up to 50% and often greater than 25% at higher values of n) despite our misgivings about the makeup and origins of Amazon’s own recommendations as a ground truth. It is also clear that as n increases precision

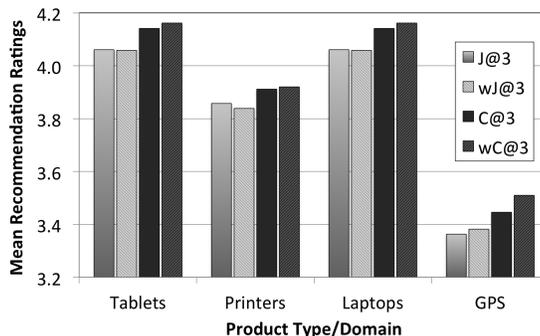


Fig. 5. The average Amazon ratings of overlapping recommended product cases.

falls off, which suggests that our retrieval approaches are tending to rank those products that Amazon is also recommending more highly.

Interestingly, the simpler *Jaccard* based similarity metrics, which ignore sentiment, provide higher precision than the sentiment-oriented *Cosine* metrics, particularly in the *Laptop*, *Printer*, and *Tablet* domains. And while weighting tends to improve the precision of *Cosine*, it offers little or no benefit for *Jaccard*.

4.5 Recommendation Ratings

As an alternative ground-truth let us consider the average ratings of the products being recommended by Amazon *and* our recommender. Summary results for top-3 recommendations ($n = 3$) are presented in Figure 5. Although the relative differences are small (products tend to be highly rated in Amazon) there is a consistent benefit for the *Cosine* based metrics and *wC* in particular. For example, we can see that, on average, *Jaccard* based Tablet recommendations have a rating of about 4.05 whereas the corresponding *wC* recommendations have an average rating of 4.16. By this ground-truth then we can see the potential for *Cosine* approaches to deliver superior recommendations to *Jaccard*.

Obviously there is much that remains to be done for a complete evaluation of this type of technique. Further studies can be found in [2] but the positive results so far, preliminary as they may be, highlight the promising potential for future work in the direction of experiential case mining and recommendation.

5 Conclusions

Our aim in this work has been to automatically extract feature-rich product cases from the type of user-generated reviews and sentiment-laden opinions that are commonplace on sites like Amazon. The resulting experiential cases are feature rich and the extracted features are shared among many cases within each product case base. We also described how this approach can be used in a recommendation setting by starting with some standard *Jaccard* and *Cosine* similarity

metrics. Of course this is simply a starting point for this research. For example, prioritising cases for recommendation just on the basis that they are *similar* to the query misses the opportunity to retrieve cases that are not only similar to the query case, but also *better* in terms of their feature sentiment. In fact this is the approach that we have tried and tested in [2], which is a companion paper to this work. Moreover there are many further opportunities to consider such as the combination of extracted features and catalog features plus, perhaps, retaining the full text of reviews, in order to consider more sophisticated ensemble approaches to product representation and recommendation.

References

1. Derek G. Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320, 2005.
2. Ruihai Dong, Markus Schaal, Michael P. OMahony, Kevin McCarthy, and Barry Smyth. Opinionated Product Recommendation. In *Proceedings of International Conference on Case-Based Reasoning, ICCBR '13*, New York, USA, 2013. Springer-Verlag.
3. Mingqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, pages 168–177, New York, NY, USA, 2004. ACM.
4. Mingqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. *Science*, 4:755–760, 2004.
5. J. Justeson and S. Katz. Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, pages 9–27, 1995.
6. Samaneh Moghaddam and Martin Ester. Opinion Digger: An Unsupervised Opinion Miner from Unstructured Product Reviews. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pages 1825–1828, New York, NY, USA, 2010. ACM.
7. Enric Plaza. Semantics and Experience in the Future Web. In *Advances in Case-Based Reasoning*, pages 44–58. Springer, 2008.
8. Ana-Maria Popescu and Oren Etzioni. Extracting Product Features and Opinions from Reviews. In *Natural Language Processing and Text Mining*, pages 9–28. Springer, 2007.
9. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.
10. Christopher Scaffidi, Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng, and Chun Jin. Red Opal: Product-Feature Scoring from Reviews. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 182–191. ACM, 2007.
11. Rosina O Weber, Kevin D Ashley, and Stefanie Brüninghaus. Textual case-based reasoning. *Knowledge Engineering Review*, 20(3):255–260, 2005.
12. Kunpeng Zhang, Ramanathan Narayanan, and Alok Choudhary. Voice of the Customers: Mining Online Customer Reviews for Product Feature-based Ranking. In *Proceedings of the 3rd Workshop on Online Social Networks, WOSN '10*, Berkeley, CA, USA, 2010.

Feedback on group recommendations

Lara Quijano-Sánchez¹ and Derek Bridge²

¹ Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain

² Department of Computer Science, University College Cork, Ireland
`lara.quijano@fdi.ucm.es`, `d.bridge@cs.ucc.ie`

Abstract. This is a discussion paper on the subject of group recommender systems. In the recent past, we have built such a recommender system, *HappyMovie*, and we have used variants of it in a number of experiments. In the light of our experience, we look at the the kind of feedback users might give to a group recommender, informed also by new results from a survey that we conducted. We conclude with ideas for the development of the next generation of group recommender systems.

1 Introduction

Recommender Systems use inferred preferences to suggest to their users items that the users might like to consume. Group Recommender Systems do the same, but they recommend items to a group of users, where the group intends to consume the items together.

Case-Based Reasoning (CBR) has a long history of contributing to recommender systems [2]. Most simply, we can build a case-based recommender system where the cases represent the items (e.g. products) and the CBR application recommends cases that are similar to the user’s partially-described preferences. More interestingly, the cases in the case base can instead describe the experience of consuming recommended products [12].

We have built a group recommender system for movies. We have also built a variant of our group recommender that uses CBR in the way described at the end of the previous paragraph. We briefly describe our group recommender and this case-based variant in Section 2.

In the course of developing these recommender systems, we have uncovered a number of perspectives on the kind of feedback that group recommender systems might seek, which we present in Section 3. To make this more concrete, we ran a group recommender system experiment with real users and administered a questionnaire to the participants. We describe the experiment and the results of the questionnaire in Sections 4 and 5. We conclude in Section 6 with ideas for the development of the next generation of group recommender systems.

2 Group Recommender Systems

Commonly, group recommender systems aggregate predicted ratings for group members [4]. First, a single-person recommender system predicts each group

member’s rating for each candidate item. This might be done, as it is in our *HappyMovie* group recommender system, using a standard user-based, nearest-neighbours collaborative filtering approach. Next, the recommender aggregates the ratings, e.g. by taking their maximum or their average. Finally, it recommends the candidate items that have the highest aggregated predicted ratings.

There are many possible variations on this common approach. Our *HappyMovie* system, for example, applies a function to each predicted rating *before* aggregation [11]:

- On registration with *HappyMovie*, users take a personality test whose results are converted into a personality score between 0 and 1, where 0 means a cooperative person and 1 means a selfish person [15]. A user’s predicted rating will count for more in the aggregation if her personality score is higher than that of the other group members.
- After registration, the strength of connection (‘trust’) between pairs of users is mined from social network data. A person’s predicted ratings are pulled towards the opinions of the other group members to a degree based on their strength of connection [3].

In [13], we presented a variant of *HappyMovie* that uses CBR: its aggregation of predicted ratings is a lazy and local generalization of the behaviours captured by the neighbouring cases in the case base. First, it uses a user-based, nearest-neighbours collaborative filtering approach to predict each group member’s rating for each candidate item. Next, it retrieves cases, i.e. past group recommendation events, that involve groups that are similar to the active group. Case retrieval uses a user-user similarity measure, and, as a by-product, it aligns each member of the active group with a member of the group in the case. The similarity measure compares group members on their age, gender, personality and ratings and the degrees of trust between members of each group. Then, it reuses each case that is retrieved: the contributions that each group member made in choosing the selected item are transferred to the corresponding member of the active group. This is done by scoring the new candidate items by their item-item similarity to the selected item. In this way, the retrieved cases act as implicit models of group decision-making, which are transferred to the decision-making in the active group. Finally, it recommends the candidate items that have obtained the highest scores.

3 Feedback to Group Recommender Systems

Suppose we have a group recommender; for concreteness, suppose it recommends movies. Consider the scenario where the recommender recommends a movie to a group, the group accept the recommendation, they see the movie together, and some or all of the group members come back and provide explicit feedback in the form of ratings. What sort of feedback should the recommender solicit?

3.1 Actual ratings

Like conventional recommender systems, most group recommender systems ask each user how much she likes the movie, e.g. as a star-rating on a five point scale. User-movie ratings are the most important (and often the only) form of *training data* for collaborative recommender systems. The additional training data may improve single-user predictions. And, since most group recommender systems work by aggregating single-user predictions, this in turn may improve group recommendations. The assumption is that the better the predictions, the better the recommendations.

3.2 User satisfaction with the recommendation

But, even if prediction accuracy is high, it does not follow that recommendation quality will be high. That also depends on how successful the aggregation is. For example, if a user watches a recommended movie in a group and later gives it a low rating, this does not mean that the group recommender has done a poor job. It may even be that the group recommender predicted that this user would give a low rating. But the movie was recommended nonetheless, as it was judged to be the one that best reconciled the different tastes of the group members: sometimes people have to lose out if the recommender is to reach a decision at all; sometimes people lose out to group members who have special priority such as children or members with disabilities; sometimes the preferences of a user who was favoured on a previous occasion may, in the interests of fairness, be weighted lower on a subsequent occasion [14].

So there is a separate dimension that can be measured: user satisfaction with the recommendation. For example, a user who dislikes the movie (gives it a low rating) may nevertheless be satisfied with the recommendation, especially if she appreciates that it has been necessary to balance conflicting interests. Her satisfaction might be all the greater if she has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom she has stronger connections (so-called contagion and conformity effects [9]). A father who takes his children to the cinema provides one such example: if his children like the recommendation, his own satisfaction with the recommendation may increase.

Additionally, *expectations* can influence satisfaction [9], even in single-user recommenders, and these can be influenced to some extent through explanations (e.g. “None of this week’s movies is a good match to your preferences. The one I’m recommending is the best of a poor crop.”). This may be even more important in group recommenders where the trade-offs that have been made can be explained.

3.3 The group experience

But there is yet another dimension to group movie-going which goes beyond both whether each member liked the movie (their rating) and their satisfaction with

the recommendation. There is what we might call the *experience as a whole* (or just *the experience* for short).³ Although the movie might be one that a group member would not choose for herself, she may still have had an enjoyable time. She may not have liked the movie; she may not have been satisfied with the recommendation (e.g. in the way that it traded-off her preferences against those of other members of the group), but watching it with her friends was still fun. Indeed, it might even be the case that the majority of the group thought a movie was terrible but they may still have enjoyed watching the movie with these friends, e.g. perhaps its awfulness provoked hilarity or heated discussion. The father watching a movie with his children may have had a great time, and this is distinct from, although not wholly uncorrelated with, his movie rating and his satisfaction with the way the recommendation traded-off group interests. The same is true of most consumption done in groups, e.g. dining out together, making excursions together, and so on —the quality of the experience is not necessarily related to what each user thought of the item, nor the user’s satisfaction with the recommendation.

It is also possible that different members of the group may evaluate the group experience in different ways. For example, the heated debate that ensued from a controversial movie may be perceived by one group member to have been exhilarating but perceived by another to have been uncomfortable. On the whole, however, we probably expect some agreement about the group experience due to the contagion and conformity effects mentioned earlier [9].

4 *HappyMovie* Experiment

In an effort to explore these issues further, we ran an experiment with real users. Sixty students from a masters-level Artificial Intelligence course participated. They were between 20 and 26 years’ old. Twenty-three were female (38.3%); thirty-seven were male (61.6%). Individually, each student completed a Personality Survey, which used TKI’s Alternative Movie Metaphor [15]: for each of five different dimensions of personality, we showed the student two well-known movie characters whose personalities oppose each other along that dimension; the student selected the member of the pair with which she most identified. The result is a numeric score in $[0, 1]$. In essence, a value of zero is a very cooperative person and a value of one is a very selfish person. Each student also completed a Preferences Survey: we asked them to rate 70 well-known movies using a five-point rating scale. *HappyMovie* uses these ratings for its collaborative filtering. Finally, the strength of connection (‘trust’) between pairs of users was mined from Facebook interactions.

³ We are not referring here to the user experience that comes from engaging with the software [5]; we are referring to the experience of consuming (in our case, in a group) the recommended items.

We formed 20 groups, each comprising three students.⁴ Each group used *HappyMovie* to create a group event —an outing to the cinema together; they received three movie recommendations from *HappyMovie* —the three that the recommender decided were best for the group, from a listing of current movies; and they agreed on one of the recommended movies —the one that their group would go to see. We asked them to imagine going to the cinema to watch that movie with the members of their group.

Then, individually and independently they answered a questionnaire of eight questions.⁵ The first seven questions were about the movie that they had selected:

1. Give your personal rating for this movie (0 for a movie you really disliked, up to 5 for a movie you really liked).
2. Give the rating that you think your friend 1 in the group will give to this movie (0 if you think s/he really disliked it, up to 5 if you think s/he really liked it).
3. Give the rating that you think your friend 2 in the group will give to this movie.
4. Evaluate the enjoyability of your experience of watching this movie with your group (0 for a really bad experience, up to 5 for a good experience — where you had a great time together).
5. Evaluate the enjoyability of the experience that you think your friend 1 in the group will have by watching this movie with your group.
6. Evaluate the enjoyability of the experience that you think your friend 2 in the group will have by watching this movie with your group.
7. Out of the listing of current movies, do you think that this would have been your choice if you had to go to the movies together in reality — without using *HappyMovie* (0 for ‘No, we would have never chosen this movie’, up to 5 ‘Yes, we would have definitely chosen this movie’).

The eighth question asked a more general question about recommendations:

8. When you go to the movies with a group of friends, what do you value most about a recommendation? Order the options by importance (most important first):
 - (a) That the movie was a good movie —in terms of quality.
 - (b) That you personally enjoyed the movie.
 - (c) That you and your friends had a good experience watching the movie.
 - (d) That the recommended movie was the one that you would have chosen as a group.

These relate to the discussion in the previous section in the following way: option (b) is related to movie rating (Section 3.1); option (c) is what we called the group experience (Section 3.3); and option (d) is about user satisfaction with the recommendation (Section 3.2). Option (a) is an ‘objective’ notion of quality.

⁴ Three was the average group size reported by 105 movie-goers in a poll that we conducted [10].

⁵ We ran the experiment with students whose first language was Spanish. The questions that we show here are paraphrases into English of the Spanish questionnaire.

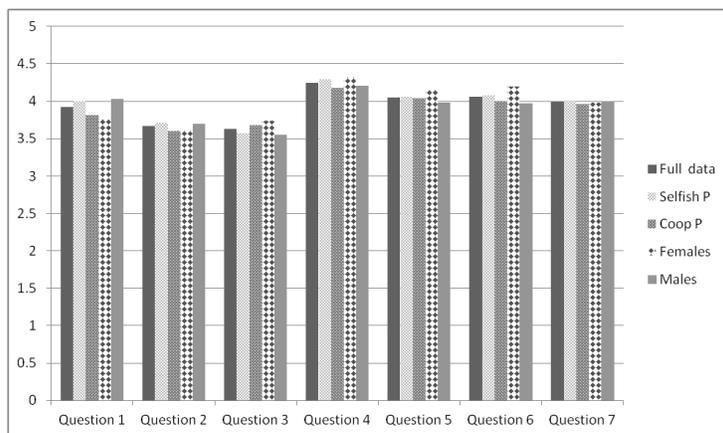


Fig. 1. Average rating by user group of responses to questions 1–7

5 *HappyMovie* Experiment Results

For analysis of the results of the questionnaires, we consider five types of user:

Full data: all sixty users;

Selfish P: the thirty-five users with a more selfish personality, i.e. users whose TKI personality score is no less than 0.6;

Coop P: the twenty-five users with a more cooperative personality, i.e. users whose TKI personality score is less than 0.6;

Females: the twenty-three females; and

Males: the thirty-seven males.

A background observation is that the male students tended to have higher TKI personality values (average 0.68784), implying more selfish personalities, whereas the female students had a lower average TKI personality value (0.46052), implying less selfish personalities.

The results for the first seven questions are in Figure 1. We can conclude:

- On average, these users rate the group experience more highly than they rate the movie (compare Questions 4 and 1), and they think their friends will do the same (Questions 5 & 6 versus 2 & 3).
- On average, these users give higher ratings to the selected movie (Question 1) than they think their friends will give to the movie (Questions 2 and 3). Similarly, their rating of the experience of seeing the movie with these friends (Question 4) is higher than what they think their friends’ ratings of the experience will be (Questions 5 and 6). So they feel that the recommender has favoured them, or that they have ‘won’ in the decision about which movie the group will go to see. This raises the question of whether users tend to rationalise decisions even when the decision goes against them.

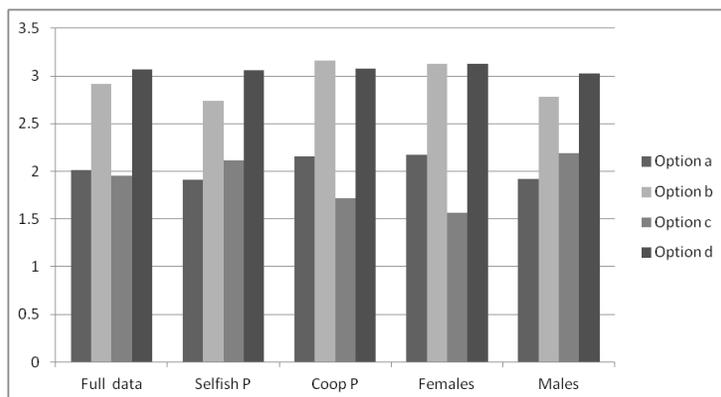


Fig. 2. Average rank by user group of responses to question 8

- The results for users with the more selfish personality values are very similar to the results for male users; and the results for users with the less selfish personality values are very similar to the results for female users. This follows from the background observation we made, that the male students had on average more selfish personalities than the female students.

The results for the eighth question are in Figure 2. In this Figure, if the bar for, e.g., option (a) is shorter than the bar for option (b), then this means that, on average, users gave option (a) greater importance than option (b).

Looking first at the results for the full set of users, we see that on average they ordered the options in decreasing importance as follows: good group experience (option c); good quality movie (option a); high rating (option b); and high satisfaction with the recommendation (option d). From the Figure, we see that the first two options are very close in their average rank. Bear in mind, though, that this experiment has more males than females and hence more users who, on average, are more selfish. A clearer picture emerges when we look at these different types of user separately.

If we look at users with less selfish personalities (and, equally, the female students in this experiment), we see that this ordering is accentuated: the group experience (option c) is more markedly important than the movie quality (option a), and there is more equivocation between options (b) and (d). But for users with more selfish personalities (and, equally, the male students), we see that the ordering of the first two options is reversed: recommending a good quality movie (option a) is more important than recommending a movie that results in a good group experience (option c). It is perhaps no surprise that more selfish users treat the group experience as less important. It is interesting though that movie quality is more important than whether they like the movie (option b) and whether they are satisfied with the recommendation (option d).

Overall, there are two surprises in the results. First, across all users the idea that a recommender does a good job when it recommends the movie that the

users would have gone to see in reality (option d) is always treated as being of low importance. Second, across all users ‘objective’ movie quality is important: perhaps we need to ensure that we recommend items whose expert reviews or population average ratings exceed a minimum quality.

It would be unwise to draw firm conclusions from experiments like this one, particularly because the questions make rather subtle distinctions which the respondents may have misunderstood and the number of respondents is quite low. What we are probably safe to conclude is the importance of the group experience, the importance too of choosing high quality movies, and the sense that, if there is a trade-off to be made, the less selfish people are the ones who can remain satisfied even when the trade-off is at their expense.

6 Discussion

Our investigation has implications for the design of group recommender systems.

A first implication is that group recommender systems need to model, and hence predict, the three dimensions. For each candidate movie, they need to predict how much each user will like the movie; how satisfied the group members will be with the different ways in which their preferences are traded-off; and the group experience. Our experimental results suggest that it may even be important to be able to predict some sort of ‘objective’ movie quality, since this was given high importance by the students in the experiment.

One way a recommender can predict these factors is for us to *design* prediction models. Nearly all work on group recommender systems has taken this approach to the prediction of users’ satisfaction with the recommendation. This is what the different aggregation functions do, including our own social recommender that takes personalities and trust into account (Section 2). But designing such models is difficult. There is a risk that our models are too simplistic, failing to take into account the richness of group dynamics.

A better approach might be to try to *learn* these models, using the feedback that we have been discussing to give us training data. This, after all, is how we predict single-user ratings. Why should we not take the same approach to predictions of recommendation satisfaction and of the group experience? An approach that generalises from training data might be more sensitive to nuances in the ways that groups operate. The case-based variant of our group recommender system (Section 2) works in this way —at least, in a simple-minded form: aggregation is based on ‘replaying’ the decision-making from similar movie-going events. It does not go so far as to predict the group experience.

CBR might be very well-suited to this task. After all, CBR is all about reasoning with experiences [1]. Since groups recur (with small variations) and groups structures (such as a parent and his or her children, or a group of university-age friends) recur, the CBR assumption (similar problems have similar solutions [8]) might apply. A rich case structure can capture multiple aspects of the movie-going event. The problem description part of the case can contain some or all of the following: (a) information about each member of the group —demographic

information, personality information, and information about tastes, e.g. in the form of ratings; (b) information about relationships between group members; (c) the candidate movies, i.e. the ones from which the recommender made its recommendations; (d) predicted ratings for each group member and each candidate movie; and even (e) predictions about the other dimensions (user satisfaction and the group experience). The solution part of the case can contain at least the movie or movies that were recommended and might contain more than this (e.g. the ranking of all the candidate movies).

But to make good recommendations, we cannot simply retain cases of this kind in a case base and replay them. The case may be suboptimal; the movie that the group went to see may not have been the best movie for this group. If we retain it, we will replay it in any future recommendation where it gets retrieved as a neighbour, where it may contribute to suboptimal decisions in the future. We need to store information about how successful each case is. Cases can include a third component (alongside the problem description and the solution), namely the outcome [6]. In a recommender system, the outcome records user feedback —the main subject of this paper. The feedback can be compared with predicted values to give a measure of the (sub)optimality of the case.

But there remains a question of practicality. We suspect that users will be either unwilling or unable to give each of the three kinds of feedback. Furthermore, when current group recommender systems ask their users for a movie rating, it is probable that users do not wholly distinguish between movie ratings (whether they liked the movie), satisfaction with the recommendation (whether the recommender traded-off preferences in a good way) and the group experience. The movie rating they supply is likely to be influenced by the other two factors.⁶

Perhaps if group recommender systems are to ask for only one form of feedback, they should instead ask users for just their rating of the group experience. This is easily understood: “On a scale of 1 to 5 (where 1 means ‘Not at all’ and 5 means ‘A very great deal’), how much did you enjoy watching this movie with your friends?” This by no means solves all the problems we face in building a new generation of group recommender systems. If we ask for only one form of feedback, we then face a *credit assignment problem*: determining how much of their enjoyment (or lack of it) was attributable to various factors, and representing and reasoning with the uncertainty that arises from this credit assignment. Furthermore, in a group recommender, we may have varying degrees of feedback incompleteness: some group members may return to the system and supply a rating; others may not, and this increases uncertainty and introduces bias.

We cannot conclude this paper with a design prescription. But we hope that our reflection on our experience of building a number of group recommender systems, along with some of the insights that come from our experiment, suggest a direction of travel for future work or, at least, will provoke useful discussion.

⁶ Ratings in single-user recommenders also exhibit contextual influences [7]. But, here we are focussing on issues that are specific to, or accentuated in, group recommender systems.

References

1. R. Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Springer, 2002.
2. D. Bridge, M. H. Göker, L. McGinty, and B. Smyth. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320, 2005.
3. J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *4th International Conference on Trust Management*, pages 93–104, 2006.
4. A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 596–627. Springer, 2007.
5. B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Model. User-Adapt. Interact.*, 22(4-5):441–504, 2012.
6. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc., 1993.
7. Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer, 2011.
8. D. B. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. MIT Press, 1996.
9. J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16(3-4):281–319, 2006.
10. L. Quijano-Sánchez, D. Bridge, B. Díaz-Agudo, and J. A. Recio-García. A case-based solution to the cold-start problem in group recommenders. In *20th ICCBR*, pages 342–356. Springer, 2012.
11. L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology*, 4(1), 2013.
12. F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. ITR: A Case-Based Travel Advisory System. In *6th ECCBR*, pages 613–641. Springer, 2002.
13. L. Quijano Sánchez, D. Bridge, B. Díaz-Agudo, and J. A. Recio-García. Case-Based Aggregation of Preferences for Group Recommenders. In *20th ICCBR*, pages 327–341. Springer, 2012.
14. L. Quijano Sánchez, J. A. Recio-García, and B. Díaz-Agudo. User satisfaction in long term group recommendations. In *19th ICCBR*, pages 211–225. Springer, 2011.
15. K.W. Thomas and R.H. Kilmann. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.

Question Routing in Collaborative Question Answering Systems

Nishaanth Shanmughasundaram and Sutanu Chakraborti

Department of Computer Science and Engineering,
Indian Institute of Technology Madras, Chennai- 600036. India.
{nishaant, sutanuc}@cse.iitm.ac.in

Abstract. Collaborative Question Answering systems have revolutionized the art of information seeking for online users in the last decade. Yahoo! Answers, Quora, Stack Overflow are some of the popular sites in this area. Due to proliferation of internet, there has been a huge surge in the number of users and questions in these sites. Hence it is necessary to have an efficient *question routing* mechanism to resolve the questions at a faster pace. We have proposed a case based recommender solution for the problem of question routing. Most of the existing solutions to question routing do not consider routing as a global task. As a result of which a single user may get bogged down with a lot of routed questions. In this paper, we have modeled global question routing as a modified version of the assignment problem.

1 Introduction

Collaborative Question and Answering (CQA) services provide a convenient way for online users to share and exchange information and knowledge, which is highly valuable for information seeking. Examples of Collaborative Question Answering services include Yahoo! Answers, WikiAnswers, Stack Exchange, as well as more social network based services such as Quora, Aardvark and Facebook Questions. Web search engines have come a long way in the last few decades, yet most of the users' needs still remains unaddressed. The main reasons for such a bottleneck are poor understanding of the intent behind the query and absence of content relevant to the query.

2 Question Routing in CQA

2.1 Problem Definition

Let the users in the CQA system be denoted by $U = u_1, u_2, \dots, u_L$ and q be an incoming question.

We intend to solve the following problem:

Question Routing : Recommend a list of expertised users to answer q .

2.2 Motivation

Due to proliferation of internet and diverse needs to users, most of the CQA systems have millions of questions in their archive. Tom Chao Zhou et al [1] have observed that only 12% of the questions are resolved in Yahoo! Answers. Also, of those resolved questions, only 20% of the questions were resolved within two days. These findings shows us that only a small fraction of the questions get resolved *quickly*. Similar observations can also be found in other research works [2,3].

The problem with questions not getting resolved quickly is that the users may leave the CQA system and look for the required information from other sources online. Even the most enthusiastic answerers of the system will get discouraged when they see a lot of irrelevant questions in their feed. Therefore, an efficient question routing mechanism is necessary for any CQA site.

3 Related work

Dror et al [4] built a multi-channel recommender system to route the incoming questions. Tom Chao Zhou et al [1] solved the problem of question routing as a classification task. Both these approaches fail to take *upvotes* of the answer to measure the level of expertise of a user at a fine grained level. Also both these approaches route questions to users only considering the level of expertise of the user. An immediate problem with this approach is that a single user can get bogged down with a lot of questions.

Mehmet H. Göker et al [5] developed a question routing system called *connection machine* for PricewaterhouseCoopers LLP(PwC). This attempts to consider question routing as a global task by capping the number of questions to be routed for each user. It however lacks the mathematical rigour to make an optimal assignment of questions to users. In this paper, we try address this issue by solving the global question routing as an optimization problem.

4 Our Approach

We model the problem as a case based reasoning task. Here, each **user profile** is a case and the solution is the **corresponding user**. So upon the arrival of a new question, it is matched with the profiles of all the users. Top users are retrieved and the question is routed to these users. The problem can also be viewed as a classical information retrieval task. Here, the profile of each user can treated as the vector space representation of the documents (the questions/answers the user has authored in the past). The new incoming question is the query. So, the task is to retrieve the document relevant to the query.

4.1 Evaluation metric

If we recommend the top k users, the *precision* of the recommendation is defined as follows,

$$precision@k = \frac{|R \cap A|}{|R|}$$

where R be the set of recommended users, A be the set of users who actually answered the question. Since the average number of answerers for a question in our dataset is 3, we have set $k = 3$ and have reported $precision@3$. Also for all the experiments, we have used a 10-fold cross-validation to report the precision.

4.2 Experimental details

The data we used for our experiments were obtained from Stack Exchange. Stack Exchange has various sites on myriad of topics like programming questions (stack overflow), english usage, photography, etc. As of now, we have used data from programming questions and english usage domains. Each question posted in the site is associated with *tags* which gives a higher level information about the topic of the question. Examples of tags are C, C++, PHP, grammar.

The domains that we used for this experiment are *Programming Questions* and *English Usage Questions*. We picked the top 500 users of the system based on the reputation points they have obtained. We then obtained all the questions for which these users have posted an answer. We limit this to top 3000 answers for each user. Each question was then processed to remove noise. We extracted both the question *title* and question *body* from the question. We treat both of them separately as they should be given different weights in characterizing the user in the case base. Since we assume the bag of words model to characterize the user, we extracted all the words from the text(question title and body). Then, we removed all the stop words from the text as they don't convey any information about the knowledge of the user. We also *stem* the words in the text. For example, *Walking, Walked, Walks*, all map to the root word *walk*.

We note the frequency of occurrence of each word in the text for each user. Frequency is a *local* measure. This doesn't discriminate between users. So, we also employ a well-known global measure *inverse document frequency*. Inverse document frequency captures the discriminative power of a word between users. It is given by,

$$idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$$

Here $|D|$ is the total number of documents in the corpus and t is the term for which we calculate the *idf*. The metric we use is *tf-idf* combining both term frequency and the inverse document frequency. We store the bag of words for title and body for each user. Apart from the **content** attributes, we also use a coarse level **tags** attribute. Tag attribute is a vector space containing the scores(number of upvotes) obtained by the user in each tag. This gives an idea of his level of expertise in a particular topic. Once this is done, every user is a point in the vector space spanned by the *title*, *body* and *tags* attributes. To retrieve the top users for a new incoming question, we map the question to

the **same vector space**. Now we retrieve the top users based on their similarity with the question. For *local similarity*, we take the *cosine similarity* to measure the similarity between the question and user profiles. We then combine all the local similarities using a weighted linear aggregator to arrive at the *global similarity*. The parameters *title_similarity_weight*, *body_similarity_weight* and *tags_similarity_weight* range from 0 to 1. We discretized the range into the following values { 0.2, 0.4, 0.6, 0.8, 1 }. This gives rise to 125 possible combinations. Each of the possible combinations was used and the results were measured using a 10-fold cross-validation. The optimal weights obtained from the experiment were *title_similarity_weight* = 0.6, *body_similarity_weight* = 0.2, *tags_similarity_weight* = 0.4. We can observe that the title is more important in capturing the essence of a question, hence gets a larger weight. Table 1 gives the precision measure for the two domains. We observed a lower accuracy in the programming domain because of the presence of a large amount of source code in the data. We need better means of representing the source code by using explicit semantic information (background knowledge) to improve accuracy in this domain. Hence, we stuck to english usage domain for the remainder of our experiments.

We improved our solution further by implementing latent semantic indexing on top of the CBR system that we built. Latent semantic indexing is factor analysis in the form of singular value decomposition to achieve dimensionality reduction [6]. Table 2 shows the variation of precision over the number of latent dimensions.

Domain of the data	Precision@3
Programming questions	0.37
English usage	0.49

Table 1: Variation of precision with respect to the domain of the data

Number of latent dimensions	Precision@3
50	0.523
100	0.528
150	0.540
200	0.531

Table 2: Variation of precision with respect to number of latent dimensions

4.3 Limitations of the initial approach

Votes for every answer is not used at a fine grained level ; rather a coarse tag level approach is used. We need to get an exact measurement of how good the user is at a fine grained level. For example, a user may have a lot of upvotes in

the topic *C++* but most of the upvotes may have come from questions related to *pointers* which is not captured by our model.

4.4 Improved solution

The basic intuition of this solution is to build a model for each user, which when triggered with a question, predicts the **number of upvotes** he will get. To solve this problem we build a regressor for each user. The features are the *title* and *body* attributes of a question and the output variable is the *number of upvotes*. We train this model with all the questions the user has answered in the past along with the upvotes received for that question. For an incoming question, we get the regressor scores for each user and rank them accordingly. We have also studied variation of the precision with respect to change in the learning algorithms for regression. We tried a 3 layered network ANN, linear SVM (with $C = 1$) and K nearest neighbours method (K=3) with distance weighting. Table 3 gives the precision these algorithms achieved.

Algorithm	Precision@3
Artificial neural network	0.57
Support vector machine	0.49
K-nearest neighbours	0.45

Table 3: Variation of precision with respect to different regression algorithms

5 Global Question Routing problem

5.1 Motivation

As discussed in Section 3, questions cannot be routed just based on the expertise level of the user. This may lead to a single user getting bogged down with a lot of questions. Also all users in the system are not equally available to answer questions. For simplicity let us take an organization with a hierarchy, the users at the top echelon of the organization should not be overloaded with a lot of routed questions as their time is too valuable for the organization. Hence, we propose a simplified first cut solution to the global question routing task by optimizing the following two criteria

- providing high quality answers to the seekers by routing questions to the most relevant users.
- decreasing the load of routed questions on the *costly* users of the organization.

5.2 Problem Definition

Let $U = u_1, u_2, \dots, u_n$ be the set of users in the CQA system. We propose to solve the following problem:

Global question routing : Given a question set $Q = q_1, q_2, \dots, q_d$ in CQA, route each question in Q to an appropriate user optimizing a global metric

5.3 Assumptions

Let us assume that question routing happens at fixed time intervals (say every night). Let the CQA system consist of a set of users and unresolved questions in that particular period, denoted by $U = u_1, u_2, \dots, u_n$ and $Q = q_1, q_2, \dots, q_m$ respectively. Also each user has a limit on the number of questions that can be routed to him, denoted by C . We define a relevance matrix $\rho_{n \times m}$, where ρ_{ij} denotes the relevance of a user u_i to the question q_j . The relevance matrix can be obtained from the techniques presented in Section 4. All the entries in this matrix are normalized between 0 (most irrelevant) and 1 (most relevant) inclusive. We also define a cost vector $\alpha_{1 \times n}$, where α_i denotes the importance of the user u_i to the organization. The idea is to minimize overloading of very important users of the system as their time is very valuable to the organization. Again these entries are normalized between 0 to 1 inclusive.

5.4 Optimization problem

Let X_{ij} be 1 if the i^{th} user is assigned to answer question j .

We solve the following optimization problem,

$$\text{Maximize } \sum_{i=1}^n \sum_{j=1}^m X_{ij} \rho_{ij} - \sum_{i=1}^n \sum_{j=1}^m X_{ij} \alpha_i$$

Subject to

$$\forall i=1, \dots, n \sum_{j=1}^m X_{ij} \leq C$$

$$\forall j=1, \dots, m \sum_{i=1}^n X_{ij} = 1$$

5.5 Solution

The above optimization is a 0-1 integer linear programming problem. Since this is NP-hard and the search space is exponential, we solved a relaxed linear programming version of the above problem. We added the following two constraints, $X_{ij} \geq 0$ and $X_{ij} \leq 1$ and solved the linear programming problem. In the process, we may get a real valued solution for X due to relaxation. We converted the real valued solution to binary solution by using a greedy approach as follows,

- For every question take the maximum value and assign the question (value of 1) to that user and assign 0 to the rest.
- If the user has already reached the limit C on the number of routed questions, assign it to the next best user for that question.

5.6 Working of the algorithm on a synthetic set

We constructed a synthetic set to evaluate our algorithm. The synthetic set was constructed as follows :

- We assumed a tree structure for the hierarchy of the organization.
- The α vector was allowed to linearly decay at the rate of 0.1 for every level in the hierarchy.
 - The top employee (say CEO of the company) has the highest value of 1 and those who report to him have a value of 0.9 and so on.
- We constructed the ρ matrix as follows,
 - We split the relevance values into 2 categories , low relevance (0 - 0.5) , high relevance (0.5 - 1).
 - For a user with $\alpha_i = c$ the user has a high relevance for fraction c of the total questions (chosen randomly) and low relevance for the remaining questions.
 - This is a fair assumption because a user with a diverse expertise in many topics is also likely to be more important to the organization.

We will illustrate the working of our algorithm on a dataset with 7 users , 15 open questions and a cap of 3 questions per user.

The alpha vector of the dataset is as follows :

$$\alpha = \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ [1, & 0.9, & 0.9, & 0.8, & 0.8, & 0.8, & 0.8] \end{matrix} \text{ cost}$$

The relevance matrix of the dataset where ρ_{ij} gives the *relevance* (fuzzy value) of user i to question j is as follows:

$$\rho = \begin{matrix} & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12} & q_{13} & q_{14} & q_{15} \\ \left[\begin{array}{cccccccccccccc} 0.66 & 0.90 & 0.70 & 0.60 & 0.81 & 0.62 & 0.55 & 0.95 & 0.82 & 0.57 & 0.62 & 0.90 & 0.81 & 0.76 & 0.57 \\ 0.84 & 0.79 & 1.0 & 0.78 & 0.64 & 0.65 & 0.95 & 0.74 & 0.81 & 0.70 & 0.95 & 0.80 & 0.78 & 0.99 & 0.73 \\ 0.64 & 0.82 & 0.57 & 0.52 & 0.59 & 0.90 & 0.40 & 0.94 & 0.57 & 0.84 & 0.51 & 0.78 & 0.57 & 0.50 & 0.62 \\ 0.10 & 0.03 & 0.65 & 0.75 & 0.18 & 0.84 & 0.36 & 0.15 & 0.36 & 0.47 & 0.88 & 0.90 & 0.91 & 0.86 & 0.58 \\ 0.65 & 0.46 & 0.54 & 0.29 & 0.55 & 0.53 & 0.14 & 0.99 & 0.69 & 0.85 & 0.47 & 0.72 & 0.81 & 0.88 & 0.76 \\ 0.55 & 0.86 & 0.50 & 0.36 & 0.96 & 0.57 & 0.55 & 0.14 & 0.47 & 0.51 & 0.86 & 0.60 & 0.73 & 0.63 & 0.80 \\ 0.72 & 0.64 & 0.94 & 0.58 & 0.56 & 0.18 & 0.27 & 0.44 & 0.63 & 0.77 & 0.84 & 0.55 & 0.76 & 0.79 & 0.87 \end{array} \right. \end{matrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{matrix}$$

The assignment matrix ($X_{7 \times 15}$) where X_{ij} is 1 if i^{th} question is assigned to the j^{th} user is presented below :

$$X = \begin{matrix} & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12} & q_{13} & q_{14} & q_{15} \\ \left[\begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right. \end{matrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{matrix}$$

One of the serious limitations with the above greedy approach is that the questions that are *assigned the last* may get routed to users irrelevant to the question. We improvised the above approach with an heuristic addition. The intuition of this heuristic is to find the penalties of not assigning the most relevant user to each question and the most relevant question to each user. The penalty for each question is given by the difference in relevance of the top 2 most relevant users for that question. The penalty for each user is given by the difference in relevance of his top 2 relevant questions. After estimating all the penalties, the one with the higher penalty (can be a question or an user) determines the question and the user to whom it should be routed. This is repeated till all the questions are assigned.

Let us illustrate the difference between the heuristic and greedy approach using an example. For simplicity, let us assume a system with 2 users, 4 unresolved questions and a cap of 2 questions per user. Following are the other parameters of the experiment,

$$\alpha = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} q_1 & q_2 & q_3 & q_4 \end{matrix} \\ \begin{matrix} 1 & 1 \end{matrix} \end{matrix} cost$$

$$\rho = \begin{matrix} & q_1 & q_2 & q_3 & q_4 \\ \begin{matrix} u_1 \\ u_2 \end{matrix} \\ \begin{matrix} 0.5 & 0.5 & 0.6 & 0.2 \\ 0.5 & 0.5 & 0.4 & 0.8 \end{matrix} \end{matrix}$$

The assignments produced by the greedy and the heuristic algorithms are,

$$X_g = \begin{matrix} & q_1 & q_2 & q_3 & q_4 \\ \begin{matrix} u_1 \\ u_2 \end{matrix} \\ \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{matrix} \end{matrix}, X_h = \begin{matrix} & q_1 & q_2 & q_3 & q_4 \\ \begin{matrix} u_1 \\ u_2 \end{matrix} \\ \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix} \end{matrix}$$

As we can see above, the greedy algorithm is sub-optimal producing a total relevance of 2.2 (0.5 + 0.5 + 0.4 + 0.8) compared to heuristic model's total relevance of 2.4 (0.5 + 0.5 + 0.6 + 0.8). Heuristic model does a better job than the greedy approach because it tries to allocate *most relevant users to each question* and *most relevant questions to each user*. This symmetry is lost in the greedy approach where it only tries to allocate most relevant users to each question sequentially. The pseudocode for the heuristic algorithm is as follows :

Algorithm 1 Algorithm to map a real valued *assignment* matrix into a binary valued matrix

Let m and n be the set of users and unresolved questions respectively. Let $X_{n \times m}$ be the real valued solution obtained from solving the relaxed version of the optimization problem. We define $T_{n \times m}$, where $T_{ij} = 1$ if the question i is routed to user j .

Initialize all the entries of T to 0.

for $i = 1 \rightarrow n$ **do**

 // Let us first estimate the row (question) penalties.

$max_row_penalty = -1$

$max_row_index = \{-1, -1\}$

for $j = 1 \rightarrow n$ **do**

if all the entries of row j in T are zero **then**

 Let $penalty$ be the difference between the top 2 values (both the corresponding users should not have reached the limit) of row j in X .

if $penalty > max_row_penalty$ **then**

$max_row_penalty = penalty$

 Let a be the column in which the maximum value falls for row j in X .

$max_row_index = \{j, a\}$

end if

end if

end for

 // Now let us estimate the column (user) penalties

$max_col_penalty = -1$

$max_col_index = \{-1, -1\}$

for $j = 1 \rightarrow m$ **do**

if $\sum_{k=1}^n T_{kj} \leq C$ **then**

 Let $penalty$ be the difference between the top 2 (both the corresponding questions must be unassigned) values of column j in X .

if $penalty > max_col_penalty$ **then**

$max_col_penalty = penalty$

 Let a be the row in which the maximum value falls for column j in X .

$max_col_index = \{a, j\}$

end if

end if

end for

if $max_row_penalty > max_col_penalty$ **then**

 Assign 1 to the entry corresponding to max_row_index in T .

else

 Assign 1 to the entry corresponding to max_col_index in T .

end if

end for

return T

6 Contributions and Future work

In this paper we proposed a case based reasoning solution to the question routing problem and tested it on two domains. We observed that the precision in the

programming questions domain is lesser due to a lot of source code in the content. In case based reasoning parlance, this means the *alignment* is very poor in the data. The precision can only be improved through better means of representation of source code with the help of domain knowledge from wikipedia, javadocs, etc. Hence, we stuck to english usage domain for the remaining experiments.

We also observed an improvement in the precision when latent semantic indexing was performed. We also proposed a novel way of incorporating the upvotes at a finer level and built a regression model to rank the users. We have further reported the variation of precision with respect to various regression algorithms.

This paper also addressed the problem of global question routing. We modeled the global question routing problem as a modified assignment problem and solved the relaxed linear programming version of it. We decoded the linear programming solution to a binary valued solution using both greedy and heuristic approaches. We observed that the heuristic approach performs better than the greedy approach.

In the global question routing problem, we have assumed that the dispatch of questions to users happen at fixed time intervals. We made this assumption to efficiently route the questions by making a global choice. This increases the time the asker has to wait to get an answer. One interesting line of work from here is to solve the online version of the assignment problem by forecasting the distribution of topics of incoming questions.

References

1. Tom Chao Zhou , Michael R. Lyu , Irwin King. A Classification-based Approach to Question Routing in Community Question Answering. *In WWW,2012*
2. B. Li and I. King. Routing questions to appropriate answerers in community question answering services. *In Proceeding of the ACM 19th conference on Information and Knowledge Management, pages 1585-1588, 2010*
3. L. Yang, S. Bao, Q. Lin, X. Wu, D. Han, Z. Su, and Y. Yu. Analyzing and predicting not-answered questions in community-based question answering services. *In Proceedings of the 25th AAAI Conference on Artificial Intelligence, 2011*
4. G. Dror, Y. Koren, Y. Maarek, and I. Szpektor. I want to answer, who has a question? Yahoo! answers recommender system. *In Proceedings of KDD, 2011*
5. Mehmet H. Göker, Cynthia Thompson, Simo Arajrvi, Kevin Hua. The PwC Connection Machine: An Adaptive Expertise Provider *In Eight European Conference on Case-Based Reasoning, ECCBR06*
6. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, Richard A. Harshman. Indexing by Latent Semantic Analysis *JASIS 41(6): 391-407 (1990)*

Case-based Reasoning in the Health Sciences

Workshop at the
Twenty-First International Conference on
Case-Based Reasoning
(ICCBR 2013)

Saratoga Springs, U.S.A.
July 2013

Isabelle Bichindaritz, Cindy Marling and Stefania
Montani (Eds.)

Chairs

Isabelle Bichindaritz
State University of New York at Oswego, USA

Cindy Marling
Ohio University, USA

Stefania Montani
University of Piemonte Orientale, Italy

Programme Committee

Klaus-Dieter Althoff, University of Hildesheim, Germany
Peter Funk, Mälardalen University, Sweden
Jean Lieber, Loria, University of Nancy, France
Beatriz López, University of Girona, Spain
Stefan Pantazi, Conestoga College Institute of Technology, Canada
Petra Perner, Institute of Computer Vision and Applied Computer Sciences,
Germany
Luigi Portinale, University of Piemonte Orientale, Italy
Olga Vorobieva, I. M. Sechenov Institute of Evolutionary Physiology and
Biochemistry, Russia

Preface

The research community working on health sciences applications of case-based reasoning (CBR) has been very active recently, as evidenced by special issues of premier AI journals, as well as by books of edited collections on the topic. In particular, a special section on case-based reasoning in the health sciences is currently in press in *Expert Systems With Applications (ESWA)*.

The community meets again at the International Conference on Case-based Reasoning (ICCBR) this year to share ideas and system descriptions collected in the proceedings of this workshop. This event is the ninth in a series of successful workshops, co-located with different ICCBR/ECCBR conferences. The first eight were held at ICCBR-03, in Trondheim, Norway, at ECCBR-04, in Madrid, Spain, at ICCBR-05, in Chicago, USA, at ECCBR-06 in Olüdeniz, Turkey, at ICCBR-07 in Belfast, Ulster, at ECCBR-08 in Trier, Germany, at ICCBR-09 in Seattle, USA, and at ICCBR-12 in Lyon, France.

Three papers have been selected this year for presentation during ICCBR workshops and inclusion in the Workshops Proceedings. They deal with protein structure retrieval using preference-based CBR [Abdel-Aziz et al.], model-based classification of unstructured data sources [Bach and Althoff], and medical literature mining for case-based reasoning in the biology of aging [Bichindaritz]. They feature advanced trends of CBR integration with social networking, text and multimedia search, and retrieval of complex structures as exemplified in bioinformatics. They exemplify how CBR helps advance the search and reuse of social media and multimedia data.

These papers report on the research and experience of seven authors working in four different countries on a wide range of problems and projects, and illustrate some of the major trends of current research in the area. Overall, they represent an excellent sample of the most recent advances of CBR in the health sciences, and promise very interesting discussions and interaction between the major contributors in this niche of CBR research.

May 25 2013

Isabelle Bichindaritz

Cindy Marling

Stefania Montani

Protein Structure Retrieval Using Preference-Based CBR

Amira Abdel-Aziz, Marc Strickert, Thomas Fober, Eyke Hüllermeier

Department of Mathematics and Computer Science
Marburg University, Germany
{amira, strickert, thomas, eyke}@mathematik.uni-marburg.de

Abstract. Structural databases storing information about geometrical and physicochemical properties of proteins are becoming increasingly important in the field of bioinformatics, where they complement sequence databases in a reasonable way. Structural information is especially important for applications in computational chemistry and pharmacy, such as drug design. A functionality commonly offered by a structural database is *similarity retrieval*: Given a novel protein structure with unknown function, one is interested in finding similar proteins stored in the database—the known function of the latter may then provide an indication of the function of the query protein. In this paper, we make use of the recently developed methodology of *preference-based CBR* to support similarity retrieval in a protein structure database called CavBase. The efficacy of our approach is shown by means of an experimental study.

1 Introduction

Structural bioinformatics has gained increasing attention in the past years. With the steady improvement of structure prediction methods, the inference of protein function based on structure information becomes more and more important. Owing to the commonly accepted paradigm stating that similar protein function is mirrored by similar structure, the comparison of protein structures is a central task in this regard. More specifically, a functionality commonly offered by structural databases is *similarity retrieval*: Given a novel protein structure with unknown function, one is interested in finding similar proteins stored in the database. The known function of the retrieved proteins may then provide an indication of the function of the query structure.

In this paper, we exploit the recently developed methodology of *preference-based CBR* for supporting similarity retrieval in a protein structure database called CavBase. Preference-based CBR [1, 2] is conceived as a case-based reasoning methodology in which problem solving experience is mainly represented in the form of contextualized preferences, namely preferences for candidate solutions in the context of a target problem to be solved.

The remainder of the paper is organized as follows. In the next section, we give an introduction to CavBase and provide some background on the application domain, notably on protein structures and protein binding sites. The

preference-based CBR methodology is outlined in Section 3, and the formalization of protein structure retrieval within this methodology is explained in Section 4. Experimental results are presented in Section 5, prior to concluding the paper in Section 6.

2 The CavBase Database

CavBase [3] is a database storing information about protein structures or, more specifically, protein binding sites. Somewhat simplified, a protein binding site or binding pocket can be thought of as a cavity on the surface of a protein in which important physicochemical reactions and interactions with other biomolecules are taking place, such as the binding of a small molecule (ligand) or the formation of a complex with another protein. Thus, properties of a binding site, both geometrical and physicochemical, are essential for the functionality of a protein. Moreover, binding sites are important targets for drug development.

CavBase supports the automated detection, extraction, and storing of protein cavities (putative binding sites) from experimentally determined protein structures (available through the Protein Data Base, PDB). The database currently contains 248,686 such cavities that have been extracted from 61,516 publicly available protein structures using the LIGSITE algorithm [4].

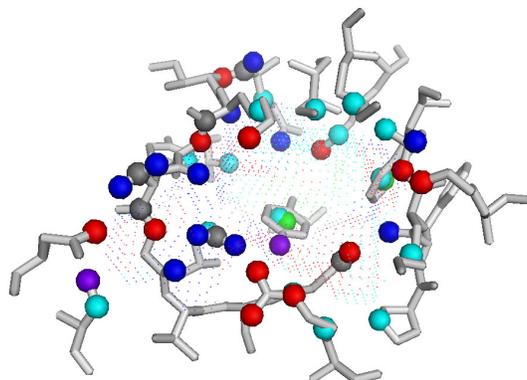


Fig. 1. CavBase representation of a protein binding site. Amino acids are shown in light grey. Pseudocenters are depicted as spheres (donor = red, acceptor = blue, donor/acceptor = purple, pi = grey, aromatic = green, aliphatic = cyan). Dots represent a surface approximation.

2.1 Pseudocenter Representation of Protein Binding Sites

The geometrical arrangement of a binding pocket and its physicochemical properties are represented by predefined *pseudocenters*—spatial points that represent

the geometric center of a particular property. The type and the spatial position of the centers depend on the amino acids that border the binding pocket and expose their functional groups. They are derived from the protein structure using a set of predefined rules [3]. Currently, CavBase distinguishes between seven types of pseudocenters: hydrogen-bond donor, hydrogen-bond acceptor, mixed donor/acceptor, aromatic, aliphatic, metal groups and pi centers.

Pseudocenters can be regarded as a compressed representation of areas on the cavity surface where certain protein-ligand interactions are experienced. Consequently, a set of pseudocenters can be seen as an approximate description of a protein binding site in terms of its most important characteristics, namely its geometry and physicochemical properties; see Figure 1 for an illustration.

2.2 The CavBase Similarity

CavBase also offers the computation of a degree of similarity between two cavities. To this end, the pseudocenter representation of a cavity is turned into a graph representation, in which nodes correspond to pseudocenters (labeled with the respective type) and edges are weighted by the Euclidean distance between the centers. In a first step, the graph representations of the two cavities to be compared are matched by finding their largest common subgraph. This match is then used to superimpose the two structures, and based on this superposition, the final degree of similarity is determined in terms of the overlap of surface patches with similar physicochemical properties. Obviously, the computation of the CavBase similarity is a computationally complex problem (recall that the largest common subgraph problem is already NP-hard).

3 Preference-Based CBR

Just like conventional case-based reasoning, preference-based CBR proceeds from a problem solving setting formalized by a problem space \mathbb{X} and a solution space \mathbb{Y} . However, experiences of the form “solution \mathbf{y} (optimally) solves problem \mathbf{x} ”, as commonly used in conventional CBR, are now replaced by weaker information of the form “ \mathbf{y} is better (more preferred) than \mathbf{z} as a solution for \mathbf{x} ”, that is, by a preference between two solutions *contextualized* by a problem \mathbf{x} . More specifically, the basic “chunk of information” we consider is symbolized in the form $\mathbf{y} \succeq_{\mathbf{x}} \mathbf{z}$ and suggests that, for the problem \mathbf{x} , the solution \mathbf{y} is supposedly at least as good as \mathbf{z} . Correspondingly, the basic regularity assumption underlying CBR, suggesting that similar problems tend to have similar solutions, is turned into a preference-based version: *Similar problems are likely to induce similar preferences over solutions.*

In the following, we assume the problem space \mathbb{X} to be equipped with a similarity measure $S_X : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_+$; thus, for any pair of problems $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$, their similarity is denoted by $S_X(\mathbf{x}, \mathbf{x}')$. Likewise, we assume the solution space \mathbb{Y} to be equipped with a similarity measure S_Y .

CBR as Preference-guided Search

In [2], we have proposed a combination of preference-based CBR and (heuristic) search, namely a formalization of case-based problem solving as a search process that is guided by preference information collected in previous problem solving episodes. This approach appears to be especially suitable for applications characterized by two important properties:

- *The evaluation of candidate solutions is expensive.* Therefore, only relatively few candidates can be tried in a problem solving episode before a selection is made.
- *The quality of candidate solutions is difficult to quantify.* Therefore, instead of asking for numerical utility degrees, we make a much weaker assumption: Feedback is only provided in the form of pairwise comparisons, informing about which of two candidate solutions is preferred. Formally, we assume the existence of an “oracle” which, given a problem \mathbf{x}_0 and two solutions \mathbf{y} and \mathbf{z} as input, returns a preference $\mathbf{y} \succ \mathbf{z}$ or $\mathbf{z} \succ \mathbf{y}$ as output.

We assume the solution space \mathbb{Y} to be equipped with a topology that is defined through a *neighborhood structure*: For each $\mathbf{y} \in \mathbb{Y}$, we denote by $\mathcal{N}(\mathbf{y}) \subseteq \mathbb{Y}$ the neighborhood of this candidate solution.

Our case base **CB** stores problems \mathbf{x}_i together with a set of preferences $\mathcal{P}(\mathbf{x}_i)$ that have been observed for these problems. Thus, each $\mathcal{P}(\mathbf{x}_i)$ is a set of preferences of the form $\mathbf{y} \succ_{\mathbf{x}_i} \mathbf{z}$, which are collected while searching for a good solution to \mathbf{x}_i .

We conceive preference-based CBR as an iterative process in which problems are solved one by one. In each problem solving episode, a good solution for a new query problem is sought, and new experiences in the form of preferences are collected. In what follows, we give a high-level description of a single problem solving episode:

- (i) Given a new query problem \mathbf{x}_0 , the K nearest neighbors $\mathbf{x}_1, \dots, \mathbf{x}_K$ of this problem (i.e., those with largest similarity in the sense of S_X) are retrieved from the case base **CB**, together with their preference information $\mathcal{P}(\mathbf{x}_1), \dots, \mathcal{P}(\mathbf{x}_K)$.
- (ii) This information is collected in a single set of preferences \mathcal{P} , which is considered representative for the problem \mathbf{x}_0 and used to guide the search process.
- (iii) The search for a solution starts with an initial candidate $\mathbf{y}^* \in \mathbb{Y}$, which is determined by means of *case-based inference* (CBI) on \mathcal{P} , and iterates L times. Restricting the number of iterations by an upper bound L reflects our assumption that an evaluation of a candidate solution is costly.
- (iv) In each iteration, a new candidate \mathbf{y}^{query} is determined, again based on CBI, and given as a query to the oracle, i.e., the oracle is asked to compare \mathbf{y}^{query} with the current best solution \mathbf{y}^* . The preference reported by the oracle is memorized by adding it to the preference set $\mathcal{P}_0 = \mathcal{P}(\mathbf{x}_0)$ associated with \mathbf{x}_0 , as well as to the set \mathcal{P} of preferences used for guiding the search process. Moreover, the better solution is retained as the current best candidate.

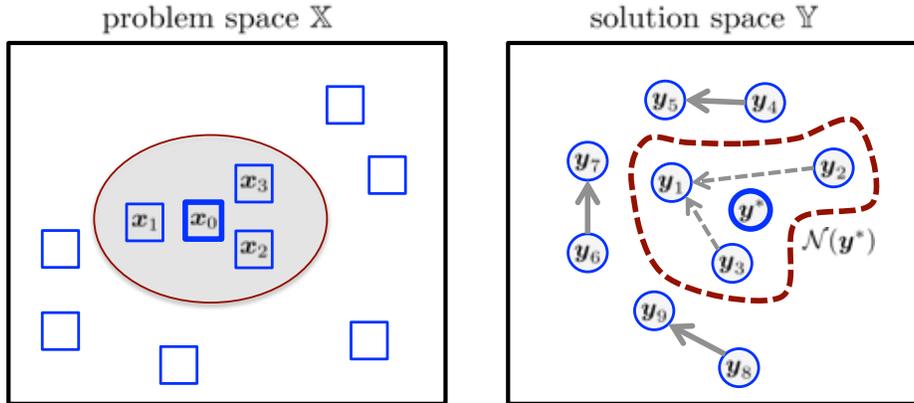


Fig. 2. Left: Query problem \mathbf{x}_0 and its nearest neighbors. Right: Current best solution \mathbf{y}^* and its local neighborhood $\mathcal{N}(\mathbf{y}^*) = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. The arrows indicate preferences observed for the neighbor problems $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, always pointing to the more preferred solution. Given these preferences, \mathbf{y}_1 is likely to be selected by CBI as a promising next candidate (to be compared with \mathbf{y}^* by the oracle), because the (hypothetical) preferences $\mathbf{y}_1 \succ_{\mathbf{x}_0} \mathbf{y}_2$ and $\mathbf{y}_1 \succ_{\mathbf{x}_0} \mathbf{y}_3$ (indicated as dashed arrows) are in good agreement with the observed preferences (they are “pointing in similar directions”).

- (v) When the search stops, the current best solution \mathbf{y}^* is returned, and the case $(\mathbf{x}_0, \mathcal{P}_0)$ is added to the case base.

The preference-based guidance of the search process is realized in steps (iii) and (iv), which make use of the case-based inference (CBI) method introduced in [1]. Based on a model of discrete choice and a statistical estimation technique, CBI essentially answers the following question: Given a set of observed preferences on solutions, considered representative for a problem \mathbf{x}_0 , which among the candidate solutions is likely to be the most preferred one? In the above search process, CBI is used to find a good initial solution and, moreover, to find the most promising candidate among the neighborhood of the current solution \mathbf{y}^* , based on the preferences collected in the problem solving episode so far. By providing information about which of these candidates will most likely constitute a good solution for \mathbf{x}_0 , it (hopefully) points the search into the most promising direction; see Figure 2 for an illustration.

We provide a more formal description of the preference-based CBR procedure outlined above in Algorithm 1, but otherwise refer to [2] for a detailed discussion of this approach.

4 Similarity Retrieval Using Preference-Based CBR

Similarity retrieval in CavBase can be realized by means of the CBR procedure described in the previous section. Indeed, this application perfectly meets our assumptions (cf. Section 3):

Algorithm 1 CBR-Pref Search(K, L, J)

1: $\mathbb{X}_0 \leftarrow$ list of problems to be solved \triangleright a subset of \mathbf{X}
2: $Q \leftarrow []$ \triangleright empty list of performance degrees
3: $\mathbf{CB} \leftarrow \emptyset$ \triangleright initialize empty case base
4: **while** \mathbb{X}_0 not empty **do**
5: $\mathbf{x}_0 \leftarrow$ pop first element from \mathbb{X}_0 \triangleright new problem to be solved
6: $\{\mathbf{x}_1, \dots, \mathbf{x}_K\} \leftarrow$ nearest neighbors of \mathbf{x}_0 in \mathbf{CB} (according to Δ_X)
7: $\{\mathcal{P}(\mathbf{x}_1), \dots, \mathcal{P}(\mathbf{x}_K)\} \leftarrow$ preferences associated with nearest neighbors
8: $\mathcal{P} \leftarrow \mathcal{P}(\mathbf{x}_1) \cup \mathcal{P}(\mathbf{x}_2) \cup \dots \cup \mathcal{P}(\mathbf{x}_K)$ \triangleright combine neighbor preferences
9: $\mathbf{y}^* \leftarrow$ CBI(\mathcal{P}, \mathbb{Y}) \triangleright select an initial candidate solution
10: $\mathbb{Y}^{vis} \leftarrow \{\mathbf{y}^*\}$ \triangleright candidates already visited
11: $\mathcal{P}_0 \leftarrow \emptyset$ \triangleright initialize new preferences
12: **for** $i = 1$ to L **do**
13: $\mathcal{P}^{nn} = \{\mathbf{y}^{(j)} \succ \mathbf{z}^{(j)}\}_{j=1}^J \leftarrow J$ preferences in $\mathcal{P} \cup \mathcal{P}_0$ closest to \mathbf{y}^*
14: $\mathbb{Y}^{nn} \leftarrow$ neighborhood $\mathcal{N}(\mathbf{y}^*)$ of \mathbf{y}^* in $\mathbb{Y} \setminus \mathbb{Y}^{vis}$
15: $\mathbf{y}^{query} \leftarrow$ CBI($\mathcal{P}^{nn}, \mathbb{Y}^{nn}$) \triangleright find next candidate
16: $[\mathbf{y} \succ \mathbf{z}] \leftarrow$ Oracle($\mathbf{x}_0, \mathbf{y}^{query}, \mathbf{y}^*$) \triangleright check if new candidate is better
17: $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \{\mathbf{y} \succ \mathbf{z}\}$ \triangleright memorize preference
18: $\mathbf{y}^* \leftarrow \mathbf{y}$ \triangleright adopt the current best solution
19: $\mathbb{Y}^{vis} \leftarrow \mathbb{Y}^{vis} \cup \{\mathbf{y}^{query}\}$
20: **end for**
21: $q \leftarrow$ performance of solution \mathbf{y}^* for problem \mathbf{x}_0
22: $Q \leftarrow [Q, q]$ \triangleright store the performance
23: $\mathbf{CB} \leftarrow \mathbf{CB} \cup \{(\mathbf{x}_0, \mathcal{P}_0)\}$ \triangleright memorize new experience
24: **end while**
25: return list Q of performance degrees

- As explained in Section 2.2, the computation of the CavBase similarity is computationally expensive. In practice, candidate solutions may even be additionally verified by a human expert, who visually inspects a superposition of two protein structures on a computer screen. Needless to say, although this expert will be able to compare two superpositions in a qualitative way, she will not be able to provide numerical degrees of similarity. Besides, she will only be willing to check a limited number of candidates.
- In general, the optimality of a solution \mathbf{y}^* , i.e., the result of a similarity retrieval for a query \mathbf{x}_0 , cannot be assured in this application. First, a proof of optimality would indeed require an exhaustive search of the complete CavBase, which, for the reasons already mentioned, is in general not feasible. But even then, optimality might be difficult to maintain, given that the CavBase is continuously growing by adding new structures. Therefore, a representation of experiences in terms of problem/solution pairs $(\mathbf{x}_0, \mathbf{y}^*)$, as commonly used in conventional CBR, is indeed questionable. As opposed to this, contextualized pairwise preferences as used in our approach are valid pieces of knowledge.

Similarity retrieval in CavBase is realized as a specific instance of preference-based CBR based on the following specifications:

- The problem space \mathbb{X} and the solution space \mathbb{Y} are both given by the “space” of protein binding sites (all potential ones and those stored in CavBase, respectively).
- The similarity measure S_Y is given by the CavBase similarity (cf. Section 2.2), and a query to the “oracle” is realized by computing these similarities (thus, given \mathbf{x}_0 and two solutions \mathbf{y}^* and \mathbf{y}^{cand} , the oracle returns $\mathbf{y}^* \succ \mathbf{y}^{cand}$ if $S_Y(\mathbf{x}_0, \mathbf{y}^*) > S_Y(\mathbf{x}_0, \mathbf{y}^{cand})$ and $\mathbf{y}^{cand} \succ \mathbf{y}^*$ if $S_Y(\mathbf{x}_0, \mathbf{y}^*) < S_Y(\mathbf{x}_0, \mathbf{y}^{cand})$). As mentioned above, the oracle may in principle also be a human expert; however, for our experimental study (see below), this was of course not possible.
- An interesting aspect is the neighborhood structure on the solution space \mathbb{Y} , which is needed to search this space in a local way. To define this structure, we took advantage of a recent large scale study, namely an all-versus-all comparison of the CavBase in terms of a similarity measure that can be computed much more efficiently than the original CavBase measure [5]. Considering this similarity measure, which is called SEGA [6], as a “proxy” of the CavBase measure, we define the neighborhood $\mathcal{N}(\mathbf{y})$ of a binding site $\mathbf{y} \in \mathbb{Y}$ by the subset of those 10 other binding sites to which it is most similar in terms of the SEGA score.

5 Experimental Study

In our experimental study, Algorithm 1 was run for a random selection \mathbb{X}_0 of 100 query structures from the CavBase. The number of nearest neighbors in the problem space (parameter K in Algorithm 1) was set to 3, the number of queries

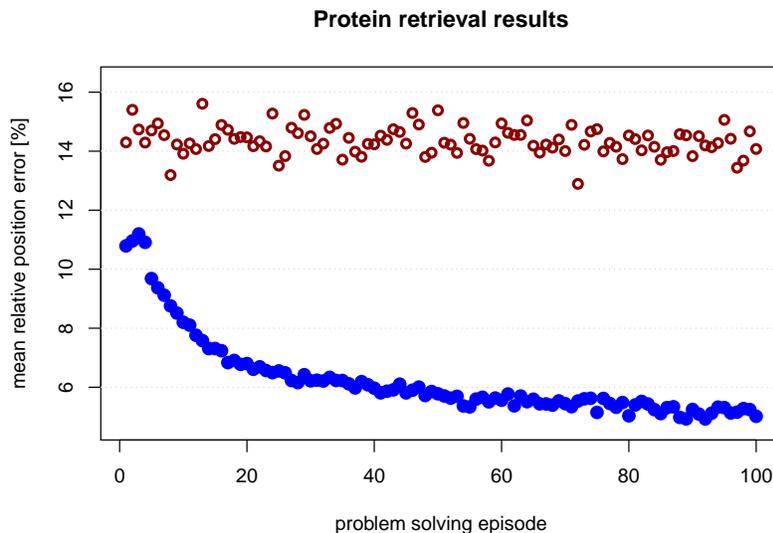


Fig. 3. Performance curves for preference-based CBR and random search: relative position error as a function of the problem solving episode.

to the oracle (parameter L) to 8, and the number of local preferences (parameter J) to 24. As a performance measure, we determined the relative position of the returned solution \mathbf{y}^* in the list of all structures sorted in decreasing order of similarity to the query \mathbf{x}_0 (for example, a value less than 10 means that \mathbf{y}^* is in the top 10% of the list). To stabilize the results, we repeated the whole process 2000 times and averaged the performance measures.

Figure 3 shows the performance curve (i.e., performance as a function of the problem solving episode) thus obtained. As a baseline, we also show the performance of a search strategy in which the preference-guided selection of the initial solution in line 9 and the next candidate solution in line 15 of Algorithm 1 are replaced by a random selection (i.e., an element from \mathbb{Y}^{nn} is selected uniformly at random). Although this is a very simple strategy, it is suitable to isolate the effect of guiding the search behavior on the basis of preference information.

As can be seen, our preference-based CBR approach shows a clear trend toward improvement from episode to episode, thanks to the accumulation and exploitation of problem solving experience. As expected, such an improvement is not visible for the random variant of the search algorithm. More specifically, while the random strategy remains at a constant (relative) position error of about 15%, this error is quickly reduced from around 12% to around 5% in our CBR search strategy.

6 Conclusion

Similarity retrieval and related problems such as case-based recommendation are tackled by conventional CBR approaches since quite a while (e.g., [7]). The standard approach essentially consists of presenting a case as a query and retrieving those objects from a database that are most similar to this case in terms of a given similarity measure.

The approach to similarity retrieval presented in this paper, which is based on the novel methodology of preference-based CBR, differs from conventional case-based retrieval in several respects. The key idea of our method is to exploit previous problem solving experience, which is stored in the form of contextualized preferences, in order to guide the search for the target solution (i.e., the objects most similar to the query). Roughly speaking, these preferences are used to steer the search process into the right direction.

The results presented in this paper clearly show the potential of preference-based CBR for similarity retrieval. Nevertheless, there are of course various extensions of our framework that still need to be addressed in future work. For example, since the number of preferences collected in the course of time may become rather large, effective methods for case base maintenance ought to be developed. Apart from that, our approach is of course not limited to applications in the bioinformatics domain. Therefore, we also plan to explore applications in other fields.

Acknowledgments

This work has been supported by the German Research Foundation (DFG) and the Marburg Research Center for Synthetic Microbiology (Synmikro). Experiments with the CavBase has been done in collaboration with the research group of Gerhard Klebe, Department of Pharmacy, University of Marburg.

References

1. E. Hüllermeier and P. Schlegel. Preference-based CBR: First steps toward a methodological framework. In A. Ram and N. Wiratunga, editors, *Proceedings ICCBR-2011, 19th International Conference on Case-Based Reasoning*, pages 77–91. Springer-Verlag, 2011.
2. A. Abdel-Aziz, W. Cheng, M. Strickert, and E. Hüllermeier. Preference-based CBR: A search-based problem solving framework. Submitted to ICCBR-2013.
3. S. Schmitt, D. Kuhn, and G. Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology*, 323(2):387–406, 2002.
4. M. Hendlich, F. Rippmann, and G. Barnickel. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15:359–363, 1997.
5. M. Leinweber, L. Baumgärtner, M. Mernberger, T. Fober, E. Hüllermeier, G. Klebe, and B. Freisleben. GPU-based cloud computing for comparing the structure of protein binding sites. In *IEEE Conference on Digital Ecosystem Technologies—Complex Environment Engineering*, Campione d’Italia, Italy, 2012.

6. M. Mernberger, G. Klebe, and E. Hüllermeier. SEGA: Semi-global graph alignment for structure-based protein comparison. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5):1330–1343, 2011.
7. Barry Smyth. Case-based recommendation. In *The Adaptive Web 2007*, pages 342–376, 2007.

Model-based Classification of Unstructured Data Sources

Kerstin Bach¹ and Klaus-Dieter Althoff²

¹ Verdande Technology AS
Stiklestadveien 1, 7041 Trondheim, Norway
kbach@verdandetechnology.com

² University of Hildesheim, Institute of Computer Science
Competence Center Case-Based Reasoning
German Research Center for Artificial Intelligence (DFKI) GmbH
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
klaus-dieter.althoff@dfki.de

Abstract. In this paper we present an approach that uses knowledge provided in Case-Based Reasoning (CBR) systems for the classification of unknown and unstructured textual data. In the course of developing distributed CBR systems, heterogeneous knowledge sources are mined for populating knowledge containers of various CBR systems. We present how available knowledge, especially the kind of knowledge stored in the vocabulary knowledge container, can be applied for identifying relevant experiences and distributing them among various CBR systems. The work presented is part of the SEASALT architecture that provides a framework for developing distributed, agent-based CBR systems. We focus on the implementation of the knowledge mining task within SEASALT and apply the approach within a travel medicine application domain. Our underlying data source is a user forum, in which various travel medicine topics are discussed, and we show that our approach outperforms the C4.5 and SVM classifiers in terms of accuracy and efficiency in identifying relevant forum entries to create cases from.

Key words: Case-Based Reasoning, Knowledge Mining, Knowledge Containers, Distributed Case-Based Reasoning

1 Introduction

In application domains where heterogeneous data sources contain relevant experiences for Case-Based Reasoning (CBR) systems we are faced with the challenge of identifying, extracting and formalizing such experiences in order to provide them on request. CBR has been proven to provide experiences, however, there is often significant manual effort necessary to collect experiences. In this work, we assume that experiences are cases in a CBR system, which originate in a web forum where users discuss travel medicine topics. These topics usually cover

among others the target region along with disease, medicament, activity and/or environmental information. Our goal in the work presented is the identification of experiences to be included as cases in a distributed CBR system.

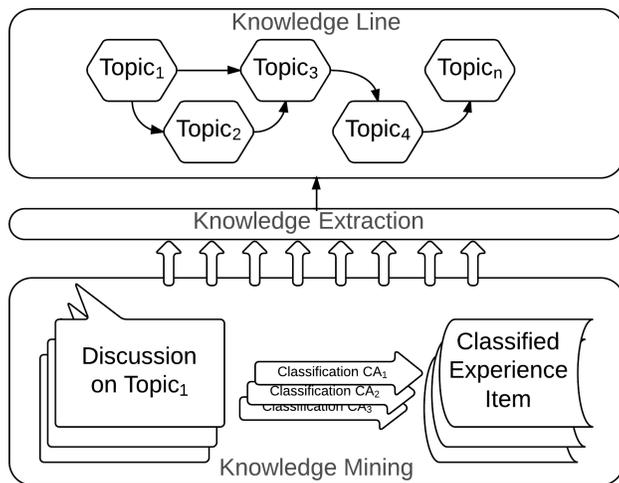


Fig. 1. Basic Knowledge Mining Approach in SEASALT

The Knowledge Mining component described in this paper is part of the SEASALT architecture [2]. SEASALT provides a general framework for creating a distributed knowledge-based system supporting the (semi-)automated identification, extraction and application of knowledge. Within SEASALT, we assume that unstructured text provided by users is available and should be populated into CBR systems. Therefore, we have created a methodology for the identification of distinctive topics that form a so called Knowledge Line [2]. A Knowledge Line describes a set of CBR-based agents, where each agent covers a topic and a solution is assembled by the partial solutions received from those agents (Topic Agents). Region, hospital, activity, person, disease, medicament, and potential risk are the topics of the travel medical application docQuery we will use as our running example in this paper. So, docQuery is a multi-agent system consisting of seven CBR systems as topic agents. The case base specific vocabulary is obtained from each agent’s vocabulary knowledge container [16] and we will make use of the terms that have been modeled in the course of developing each agent. The work presented in this paper can be seen as a pre-processing step in which we are identifying relevant experience that are targeted to become cases eventually. Within the docQuery system, we have human Knowledge Engineers that build the cases as they are ensuring the quality of the cases. However, within SEASALT we are aiming at more and more supporting and automating this case

building task. Mining raw data and identifying relevant information is therefore an initial step.

Within the architecture we expect to have one or more collector agents for each topic agent that monitor the user forum and trigger the knowledge extraction. When and how to trigger is the key task of the model-based classification presented in this paper. The remaining of this paper is structured as follows: In Section 2 we introduce the idea of Model-Based Knowledge Mining while Section 3 describes the supervised classification based on the knowledge models derived from CBR vocabulary as well as the SVM and C4.5 classifiers, which are a part of our agent ensemble. The following section compares the classification quality of these three agents in a real-life application in which forum discussions are classified. Section 5 discusses related approaches and the final section summarizes the findings and gives an outlook on future work.

2 Model-Based Knowledge Mining

The software agents, so called collector agents, require access to knowledge models that have been created during the implementation of the CBR systems as well as being a result of the Knowledge Extraction process [4]. Since we are mostly focusing on CBR systems, knowledge is represented as vocabulary (or gazetteers), cases, similarity measures, and optional transformation rules. The main knowledge containers we are using are the vocabulary for the gazetteer agent and the cases for learning the underlying models. Furthermore, we have included stop word lists for removing terms with less information.

CBR-Driven Vocabulary Within a SEASALT implementation, we create multiple, heterogeneous CBR systems, where each system has an individual case representation and vocabulary to cover the relevant cases. For example, the diseases case representation differs from an activity's case structure. We assume that the relevant vocabularies contain only those terms that are topic specific and characterize a particular domain. We will use this assumption to build software agents for each topic in order to extract relevant forum entries. In the remaining part of this section, we also assume that the CBR systems we created are using the myCBR tool. myCBR's SDK allows accessing the vocabulary per concept and attribute description [3]. We are able to receive all relevant terms, well organized and easy to distribute to the according agents. We decided to have one Gazetteer agent for each topic. The major task of the set of collector agents is identifying entries and organizing them in categories. Alongside the Gazetteer agent, we have also implemented C 4.5 and SVM agents, which use the keywords for learning the required models.

Stop Word Specific Vocabulary Before we can start the classification, we have to normalize the given texts, which in particular means removing stop words, based on stop word lists from the knowledge representation. We use both

German and English stop words since those are the languages we are currently dealing with, as well as HTML stop words. HTML stop words list is a manually created list of HTML tags occurring in the given data bases of forum entries. Also other frequently used terms in mailing lists should be removed in this preparation step. Stop word lists for the German and English language were retrieved from the *Wortschatzportal* of Leipzig University [14], from where they are available as plain text lists³.

Knowledge Sources The instantiation we are currently focusing on, a web forum, is based on a mySQL server and therewith we can easily access the raw data inserted by forum users. The forum is restricted to experienced travelers, so we can assume they are experts in their domain. For that reason we will later on call this forum expert forum. Further on, we used the mySQL data base to store meta information, which has been automatically extracted, along with the manual and automatic classification for each forum entry. This enables us later to carry out various tests on the quality of the classification. The population of these parts will be described later on in this section. First we will introduce and characterize each type of agent.

Collector Agent Types For the collection and classification of forum posts we have three types of agents: Gazetteer agents, C4.5 agents and SVM agents. Since we aim to create modular and learning systems, we will furthermore have a supervisor agent that organizes each input of the basic classification agents and a third type, called learning or apprentice agent that monitors the actions of the Knowledge Engineer in order to provide feedback for the classifiers – or at least recognize if one of the collector agents fails permanently.

3 Supervised Classification in SEASALT

The SEASALT Knowledge Mining agents are realized based on the JADE framework [5] by first implementing the agent platform and then initializing the collector agents. The supervisor and Apprentice agents will be started a certain time after all collector agents are set up. The agent platform connects the software agents to the source data and the user interface, which can also start the agent platform.

For the startup of the supervisor agent, all collector agents are registered at the supervisor agent in order to receive data and monitor the actions carried out by the Knowledge Engineer on the forum entries or keyword lists.

Pre-processing of Forum Entries Before entries can be classified, they have to be normalized to reduce noise. Since we are dealing with natural language in the social web we decided for a case insensitivity approach and substitute all

³ <http://wortschatz.uni-leipzig.de/html/wliste.html>

upper case letters by lower case letters as well as non-standardized characters are either removed or substituted. Finally multiple spaces are reduced to single spaces.

During the pre-processing of data to prepare the classification, we split longer texts into single sentences. From a longer forum discussion, we will receive a sentence as follows:

[...] On the way to your hotel we already used the repellent to avoid mosquito bites. [...]

Later on each term will be handled as one element in an array, while each array contains a whole forum post by one user. Afterwards we carry out a first Named Entity detection for multi-word terms such as *Hepatitis A* or *Parkinson's Disease*, which should not be split up because this will cause a major loss of information. The example will then be represented as follows:

[On] [the] [way] [to] [your] [hotel] [we] [already] [used] [the] [repellent]
[to] [avoid] [**mosquito bites**]

Next, all stop words are removed and we have a resulting array containing potentially relevant terms. The example will then be represented as follows:

[way] [hotel] [used] [repellent] [avoid] [mosquito bites]

Then we look up and tag each term with the topic class it belongs to. We then take for each keyword found n words before and behind and store them as our classification data. Later on, we will use this kind of term template to identify other, unknown terms describing the same or similar content. For $n = 3$ we will store the following data set

way, hotel, used, <keyword>**repellent**< /keyword>, avoid, mosquito bites

with the association that this information entity serves the medication agent, which contains prevention information. Since we are working on a sentences base, we will not include terms from the next sentence.

Based in these entries we will train the intelligent classifiers. In Section 4, we are evaluating how many terms should be included to have an appropriate term template.

The overall goal is to collect experiences based on their description with which they are presented to others. For each topic or category, we are training the classifiers to recognize terms which are not included in our keyword list. This observation somehow creates a context in which keywords are used. This approach combines the boolean classification by the C4.5 and SVM agents with a probabilistic model, because we are trying, like Hidden Markov Model (HMM), to use surrounding information to derive classification for unknown terms. In comparison to HMM [9], which is based on probabilistic models, we use the C4.5 and SVM models. This approach can be compared to [8].

We perform this classification for each topic individually in order to receive independent classifications of the source data. This might lead to multiple classifications, which can be resolved by the Knowledge Engineer or confidence values. Currently we rely on the Knowledge Engineer in this regard. These steps are managed by the supervisor and Apprentice agent.

4 Experimental Evaluation

The evaluation of the knowledge mining has been carried out with two different data sets. The first one has been created manually from a Knowledge Engineer while the second model has been created semi-automatically. The creation of the semi-automatic model has been described in [4]. Each agent has been used in combination with the automatically and manually obtained knowledge models.

The goal of the evaluation is to find out which of the three implemented collector agents performs best in the given domain as well as how the two knowledge models work within our Knowledge Mining approach.

The data set has been created using the expert web forum with 700 entries in German. For training of the SVM and C4.5 agents we have used 200 entries and for the evaluation we took 500 test entries. From previous tests we learned by experience that taking into account a certain number of surrounding words, i.e., five words before and after a keyword, returns the best results [1], because 7 words were usually too many and sentence delimiters shortened the sequence, while 3 words did not produce stable results. Further on, we decided to use the standard SVM and C4.5 classifiers as they are available in WEKA.

For the evaluation we always used the complete Knowledge Models for the Gazetteer Agent and the SVM and C4.5 has been trained once. The variable factor is the noise in the incoming data in terms of stop words, which reduce the density of keywords (see Section 3). In the course of the evaluation we have used the three different kinds of stop word lists: stop word lists containing 100, 1,000 and 10,000 terms. In each run we collected the suggested classifications until 500 entries have been reached.

Figure 2 shows the F1 measures for the Knowledge Mining process using the Gazetteer agents for diseases, regions and medications. The complete results, which have been used to determine the F1 measure, can be found [2]. The F1 measures in this figure show the results for both models and there is a clear tendency that the Gazetteer agent performs much better than the SVM and C4.5 agent, respectively. As expected the model manually created by the Knowledge Engineer (first set of charts in Figure 2) is more reliable in the classification of new entries than the automatically created model.

Since the vocabulary of each CBR agent contains the terms relevant for representing the cases, the accuracy of the Gazetteer Agent is very high. The performance of the SVM and C4.5 agents turn out to be on the same level, while the SVM performs slightly better.

Overall, our experiments show that knowledge available in the vocabulary can be successfully used to classify unknown data during the pre-processing of

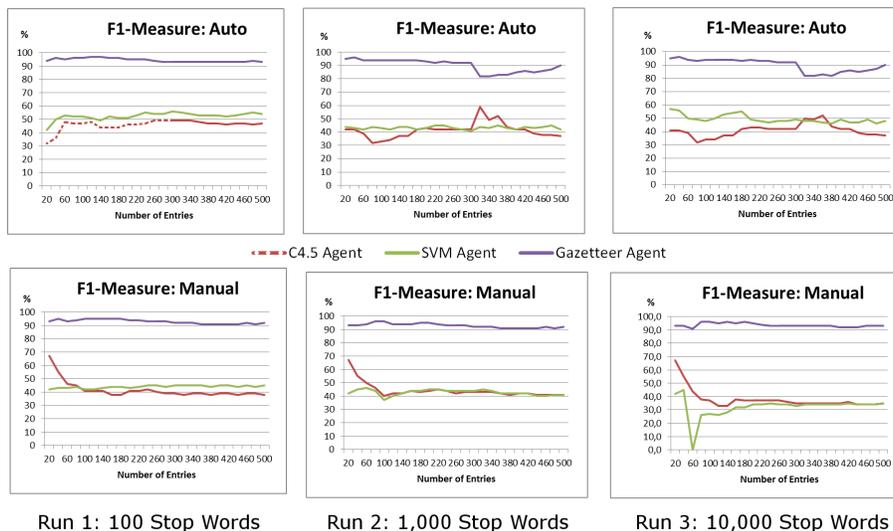


Fig. 2. F 1 Measures for the Diseases, Regions and Medication Agents

WWW resources in order to populate cases. However, this only enables a good classification, while the capturing of cases will be a different challenge.

5 Related Work

A related approach has been presented by Garcia and Wiratunga [15] in the context of Textual Case-Based Reasoning, where an unsupervised approach of learning taxonomies from web sources was introduced. However, our work can still be seen as a pre-processing step for the distributed CBR-driven multi-agent system, while their approach is directly applied within the CBR system without any human interaction. Similarly, Roth-Berghofer et.al. [17] used the vocabulary knowledge container to automatically index cases. We have taken this approach into account, and further developed these ideas away from the required rather static case structure to highly flexible and distributable case representations. Further, Zhang and Lesser [19] also address an hierarchical organisation of agents for distributed content sharing. However, their motivation is improving the performance of the computation, while our approach focuses on specialisation of tasks and content-based clustering of topics.

An alternative to the implemented knowledge mining approach could be making use of SMILA, an architecture specialized for the search in unstructured information sources. SMILA has been developed as middleware platform within the Theseus program - mainly for the application scenario ORDO⁴. SMILA is based on the OSGi framework [18,11].

⁴ <http://www.theseus-programm.de/anwendungsszenarien/ordo/default.aspx>

The architecture is divided in two parts: pre-processing and the search engine itself. Since SMILA heavily uses OSGi's service components it contains various individually configurable modules [10]. The *Pre-Processing* uses agents or services for crawling and processing unstructured information in order to build an index that can be searched afterwards. The main contribution of SMILA is the provision of an open middleware that has to be further developed.

The development in SEASALT and SMILA was carried out in parallel and at an early stage the middleware did not meet our expectations regarding a very strict indexing and searching focus rather than a high variability of information and knowledge processing. Today, after SMILA is an active project within the eclipse foundation an integration of our modules is more feasible and SEASALT could benefit from SMILA's performance when dealing with big data. Since the main focus is searching the provided processes are tailored in this way, however as shown in [7], SMILA can also be used in various ways such as for dealing with more structured sources and carrying out more sophisticated tasks like providing adaptation capabilities.

Further on, rather than including knowledge models from myCBR, also Protégé [13,6] would be an option if just ontologies are to be included. We have worked with both, but eventually decided for myCBR since we are focusing on CBR-driven applications. Ontologies modeled with the *Protégé-Frames Editor* are also accessible from our tool [1].

6 Conclusion and Outlook

The work presented in this paper targets at reusing the vocabulary knowledge container for classifying new entries whether they fit in the topic of existing CBR systems. The approach has been implemented as SEASALT instance [2]. SEASALT as well as the introduced Knowledge Mining approach have been applied in the real-life application docQuery and the data used for the evaluation of our work was obtained from an expert forum in travel medicine. The experiments show that the pre-processing and selection of web-data can be based on the knowledge created in CBR systems as the gazetteer agents, which are based on various CBR system's vocabularies, outperform standard Machine Learning approaches. Moreover, the effort of creating the Gazetteer agents is very low, since they directly use the knowledge models provided by myCBR. In contrast, training data for the SVM and C4.5 classifiers has to be created before the classifier can be applied.

The Knowledge Mining approach presented in this paper offers a new, pragmatic perspective for constructing WebCBR systems [12] with a positive cost-benefit relationship. Moreover, the compatibility to SMILA can be used to create more parallel knowledge mining approaches which will enable a more effective creation of CBR systems capturing cases from web resources. Also, up to now, we only use the plain keywords rather than the complete taxonomies for the classification. A direction we will investigate further is the development of case-based

classifiers, which can be directly derived from each CBR agent in the Knowledge Line.

Acknowledgement We would like to thank our students Kirsten Skibbe, Manuel Ahlgrim, and Alena Rudz for their contributions to the work presented in this paper.

References

1. Ahlgrim, M.: Developing software agents for experience classification from web forums (Analyse von Webcommunities und Extraktion von Wissen aus Communitydaten für Case-Based Reasoning Systeme). University of Hildesheim, Master thesis (December 2010)
2. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013)
3. Bach, K., Althoff, K.D.: Developing Case-Based Reasoning Applications Using myCBR 3. In: Watson, I., Agudo, B.D. (eds.) Case-based Reasoning in Research and Development, Proceedings of the 20th International Conference on Case-Based Reasoning (ICCBR-12). pp. 17–31. LNAI 6880, Springer (September 2012)
4. Bach, K., Sauer, C.S., Althoff, K.D.: Deriving case base vocabulary from web community data. In: Marling, C. (ed.) ICCBR-2010 Workshop Proc.: Workshop on Reasoning From Experiences On The Web. pp. 111–120 (2010)
5. Bellifemine, F., Caire, G., Trucco, T., Rimassa, G.: JADE Programmer’s Guide. CSELT S.p.A., TILab S.p.A., Telecom Italia S.p.A., o.O. (2010)
6. Gennari, J.H., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubzy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of protg: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies* 58, 89 – 123 (2002), http://bmir.stanford.edu/file_asset/index.php/52/BMIR-2002-0943.pdf
7. Hanft, A., Schäfer, O., Althoff, K.D.: Integration of drools into an osgi-based bpm-platform for cbr. In: Agudo, B.D., Cordier, A. (eds.) ICCBR-2011 Workshop Proceedings: Process-Oriented CBR (2011)
8. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural svms. *Machine Learning* 77(1), 27–59 (2009)
9. Marsland, S.: *Machine Learning - An Algorithmic Perspective*. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, CRC Press, Taylor and Francis Group, Boca Raton, Florida, USA (2009)
10. Nieland, D., Stiglic, J.: Open services gateway initiative (osgi) to drive development of gateway standard for homes, soho and remote locations. Palo Alto: Open Services Gateway initiative, Palo Alto, Californien, USA (November 1999), <http://www.osgi.org/News/19991122EN>
11. Novakovic, I.: Smila/architecture overview. Ottawa: Eclipse Foundation, Inc., Ottawa, Ontario, Canada and Empolis GmbH (Januar 2010), <http://wiki.eclipse.org/SMILA>
12. Plaza, E.: Semantics and experience in the future web. In: ECCBR ’08: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. pp. 44–58. Springer-Verlag, Berlin, Heidelberg (2008)
13. Protégé: what is protégé? Stanford : Stanford University School of Medicine, Stanford Center for Biomedical Informatics Research (2012), <http://protege.stanford.edu/overview/index.html>

14. Quasthoff, U., Richter, M.: Projekt deutscher wortschatz. Babylonia (2005)
15. Recio-Garcia, J.A., Wiratunga, N.: Taxonomic semantic indexing for textual case-based reasoning. In: Proceedings of the 18th international conference on Case-Based Reasoning Research and Development. pp. 302–316. ICCBR'10, Springer-Verlag, Berlin, Heidelberg (2010)
16. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S. (eds.) Case-Based Reasoning Technology – From Foundations to Applications. LNAI 1400, Springer-Verlag, Berlin (1998)
17. Roth-Berghofer, T., Adrian, B., Dengel, A.: Case acquisition from text: Ontology-based information extraction with scoobie for mycbr. In: Bichindaritz, I., Montani, S. (eds.) Case-Based Reasoning. Research and Development, Lecture Notes in Computer Science, vol. 6176, pp. 451–464. Springer Berlin Heidelberg (2010)
18. Schütz, T.: D11.1.1.b: Concept and design of the integration framework. Bundesministerium für Wirtschaft und Technologie and Theseus-Ordo and Empolis GmbH (2008)
19. Zhang, H., Lesser, V.: A Dynamically Formed Hierarchical Agent Organization for a Distributed Content Sharing System . In: Proceedings of the International Conference on Intelligent Agent Technology (IAT 2004). pp. 169–175. IEEE Computer Society, Beijing (2004), <http://mas.cs.umass.edu/paper/373>

Medical Literature Mining for Case-based Reasoning in the Biology of Aging

Isabelle Bichindaritz
State University of New York
Oswego, New York 13126, USA
ibichind@oswego.edu

Abstract. Scientific literature has been quickly expanding as the availability of articles in electronic form has increased rapidly. For the scientific researcher and the practitioner alike, keeping track with the advancement of the research is an on-going challenge, and for the most part, the mass of experience recorded in the scientific literature is largely untapped. In particular, novice scientists, non researchers, and students would benefit from a system proposing recommendations for the problems they are interested in resolving. This article presents the first stages of the Digital Knowledge Finder design, a case-based reasoning system to manage experience from the scientific literature. One of the main functionality of the system is to enable both to represent the experience in a declarative and searchable form, and to reason from it through reuse – the latter being a consequence of the former. This article focuses on research findings mining and results from an aging literature dataset.

1. Introduction

Intelligent programs regularly top the news headlines, often in the form of a robot or a game – such as recently DeepQA, the Jeopardy game playing agent from IBM. The most famous accomplishment of intelligent systems has been the defeat of the chess world champion Gary Kasparov in 1997 by IBM's Deep Blue, after four decades of intensive research focused on mastering the chess game [1] as the holy grail of computational intelligence. This was indeed a major milestone on the evolutionary tree of computational beings. However for the computational intelligence specialist, the significant progress in the academic literature presents challenges. One main question we are facing in science, and particularly in computer science, is: how do we deal with the rapid rate of expansion in our field? A researcher in data mining for examples can be both pleased with and concerned by the greater number of papers to read and review each year. Is it not time to apply what we know, these sophisticated intelligent beings we design, to our own field?

This article presents the first steps in developing the Digital Knowledge Finder which is no less than a computational knowledge discoverer, a partner for the human scientist whether in education, in research, or in industry - the help he or she needs to answer the particular questions our life may depend upon. The Digital Knowledge

Finder is a case-based reasoner conceived for the reuse of the scientific experiences represented in the scientific literature, where each article constitutes a case.

This article is organized as follows. The second section presents the high-level architecture of the system and its major components, and the case-based reasoning component is explained in the next section. The fourth section highlights the research finding process, and the fifth section evaluation matters. The conclusion discusses a broader perspective on the system.

2. High-Level Architecture

The Digital Knowledge Finder, closely interacting with a human data miner, is centered around a memory, serving as a knowledge repository, interacting with a reasoner capable of solving tasks relevant to a data miner. Therefore, at the highest level, the two main components are a reasoner and a memory. Aspects dealing with the integration between the components will be presented in a separate paper. A set of knowledge acquisition components allows for editing knowledge to add to the memory by a human user, and for mining for knowledge automatically for adding to the memory with minimal human intervention.

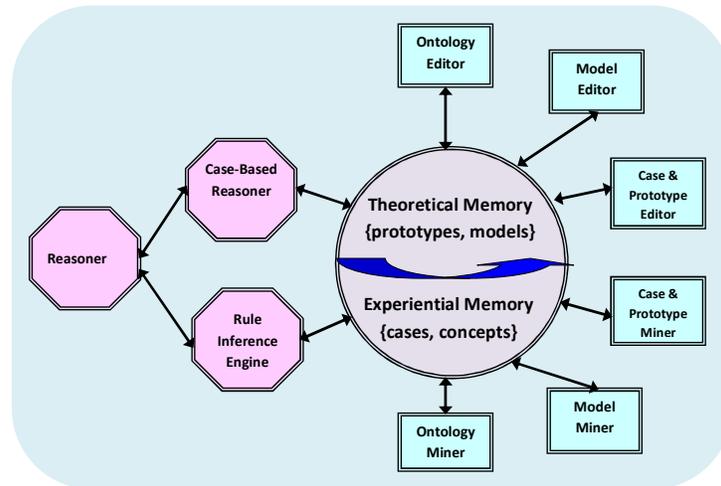


Figure 1. High-level architecture of the system, where octagons represent the reasoning components, rectangles the other components, and the circle represents the knowledge base.

Figure 1 provides a schematic overview of the system’s architecture, combining the following components:

- The *Memory* combines a declarative component and a procedural component (see Figure 2). The declarative component is truly the semantic memory of the system, where knowledge is represented in an explicit – or declarative manner. Declarative knowledge can be theoretical knowledge (prototypes and models) or experiential (cases and concepts). Theoretical knowledge is knowledge having

lost all links to the data and/or experiences from which it was devised. Such knowledge may derive from an expert or from other sources such as books, the Internet, and so forth. It is mostly also cut from the context in which it was learned or synthesized. Experiential knowledge represents knowledge having kept some links to the individual elements, as well as the context, in which it has been [2]. The declarative knowledge comprises in particular an ontology of the scientific domain.

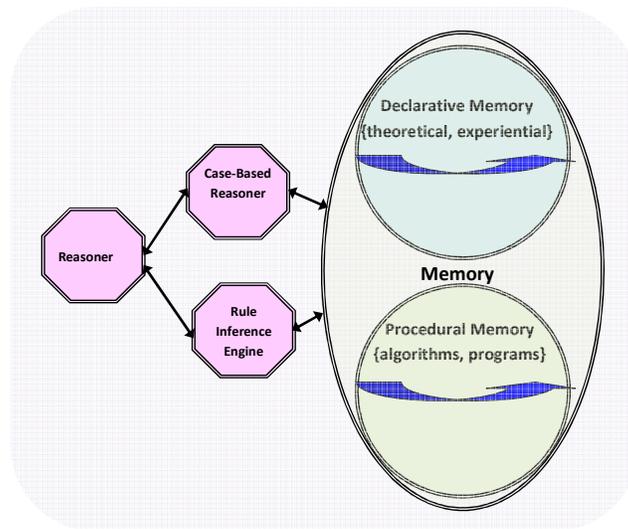


Figure 2. Architecture showing the declarative and the procedural memory in interaction with the reasoner.

- The *Reasoner* comprises two main components: a rule inference engine [3] - to solve problems involving mainly theoretical knowledge, and a case-based reasoner [4] to solve problems dealing with the reuse of experiential knowledge.
- *Knowledge acquisition components* aim at acquiring knowledge from experts or other sources, such as existing software.
 - a. Ontology editor to add/update the terminology and ontology represented in OWL.
 - b. Model editor to add/update models. The knowledge representation formalism is that of conceptual graphs.
 - c. Case & prototype editor to add/update cases generally in the form of prototypical cases.
- *Data mining components* aim at mining for knowledge directly from scientific literature or from the Web through text mining algorithms inspired by the works of Swanson [5, 6] and Fuller [11].
 - a. Ontology miner to mine for building blocks of the terminology and ontology.

- b. Model miner to mine for models in the form of conceptual graphs.
- c. Case & prototype miner to mine for cases from the scientific literature – a case represents an article most of the time.

3. Case-based Reasoning Component

The Digital Knowledge Finder reasoning process relies essentially on a case-based reasoning component. The vision aims at a reasoning process that takes a problem from the user as input, asks a minimum set of questions to the user, selects the best combination of scientific experiments to solve it, and shows them to the user, eventually after adapting them. The agent will provide explanations about the choice of the methods and the interpretation of the results. The reasoner is being developed in steps, the first ones being presented in this section for the case-based reasoning component.

3.1 Memorized case representation.

A case represents a scientific experience as described in a scientific article. A similar representation is used to represent prototypical cases [8]. A memorized case may comprise a variety of elements, some of which may be entered as text, others as concepts from the ontology, and yet others may be left unknown. However the major elements are the research question(s), the research finding(s), and the research design, with its material and its methods components. Therefore a memorized case is represented as a triple (see Eq. (1)):

$$RQ \xrightarrow{RM} RF \quad (1)$$

where

- RQ represents one or more research question(s) of hypothesis(es) (for example: “Does caloric restriction in older people decrease life expectancy?”). These also encompass the broader goals of the research.
- RF represents a set of research findings or results (for example: “Caloric restriction in older people *does* decrease life expectancy”).
- RM represents the research method(s) and involves material and methods. The research method differs in computer science and in medical science or experimental science. In computer science, a lot of the method describes systems architecture or design questions (for example the algorithm used or designed for the article study) while in an experimental science domain the research method focuses mostly on evaluation matters (description of the population used, sampling strategy, data collection and analysis for example). These aspects are also used in computer science however as part of the evaluation methodology. Therefore, if we generalize between these domains, we obtain a research methods description based on two elements:
 - *The design process* (for example the knowledge representation paradigm or the algorithm designed).
 - *The evaluation process* (for example the dataset used).

Although both of these components are generally described in a computer science paper, in a medical research paper the evaluation process constitutes the core of the research methods.

Additional components of a case representation include the following:

- Title of the study.
- Authors and affiliations.
- Background and motivation.
- Keywords
- Research topic (for example: biology of aging – this is more global than the precise research question addressed).
- Constraints and limitations.
- References (title, authors, affiliations...).
- Link(s) to the article online.
- Financial, equipment, and human support.
- Future directions.
- Other / comments (novel algorithm...).

3.2 Case-based reasoning functionality.

The system can be used in two modes: the information retrieval / interrogation mode and the problem-solving mode:

- *Information search functionality.* The system can be interrogated about the knowledge it has learned, namely provide statistical information about, for example in the data mining domain, the number of algorithms memorized, their evolution over time, the main research questions addressed, the main research findings, and so forth. The potential interest of the system as a scientific literature tracking and monitoring system are endless. Case-based reasoning contributes to this aspect through similarity-based retrieval.
- *Problem-solving functionality.* In the problem-solving mode, the system will comprise more advanced algorithms for *adapting* and *combining* memorized experiences. The agent will select, from the goal and characteristics of the scientific study, the optimal subset of pertinent articles, based on the retrieved cases. The system will retrieve the most similar memorized cases, based on the data available, and either provide them to the user as a ranked list, or adapt the most similar to propose an experimental set-up for the new case to solve. For example, in a data mining study, the system will propose the best algorithms to carry on the study as well as evaluate the time and space requirements, contrast eligible algorithms through their advantages and drawbacks, and find optimal algorithm parameters values [9]. The agent will also select the best test strategy, between cross validation, training set, or independent test set. The agent will communicate its choices to the user, and always leave the user control over the process, so that it can either function autonomously or be directed by the user. The agent traces its reasoning process so that the user is notified in real time of all actions undertaken as well as the rationale for performing these. Graphical communication is preferred whenever possible – in particular for the results.

4. Research Finding Miner

The first building block of the system is the ability to mine research articles for their research findings. Indeed this is the core of a scientific paper, which will determine the reader's interest in the paper. We have designed a research finding component and evaluated it in the domain of aging literature. The system comprises a ConcepMiner and a RelationshipMiner component.

4.1 Concept Miner

The ConcepMiner system [10] presented serves as the basis for RelationshipMiner, while expanding it to incorporate semantic naming of relationships. While ConcepMiner could process only figure and table legends, RelationshipMiner can be run to specifically process figure and table legends, document parts, or full documents.

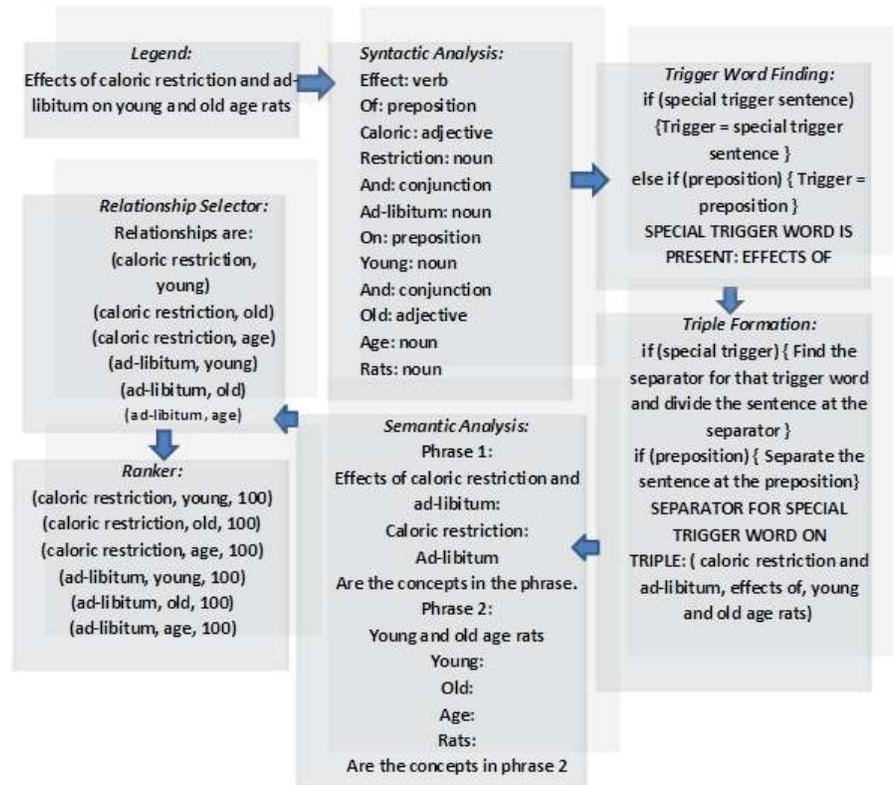


Fig. 3. ConceptMiner system process flow.

ConceptMiner was initially developed for the Telemakus system [11], which consists of a set of domain documents (original focus was the biology of aging), a

conceptual schema to represent the main components of each document, and a set of tools to query, visualize, maintain, and map the set of documents through their concepts and research findings [11]. For that purpose, this system mines and maps research findings from research literature. At present, knowledge extraction resorts to systems with both manual and automated components. A key area of current work is to move towards automating the research concept identification process, through data mining [11]. This is exactly why ConceptMiner was developed.

Concept mining involves processing articles already stored in a domain-specific database (DSDB). These articles actually did not comprise the full text of the original articles, only the tables and figures descriptions, referred to as *legends*, which are considered the most probable placeholders for research findings. It has been established by Telemakus project team that the most interesting information about research literature is usually found in legends [11].

ConceptMiner process flow is illustrated in Fig.3. The system processes through several steps, the main ones being syntactic analysis, semantic analysis, and concept mapping and association.

Given an article or a set of articles, the system starts by extracting all legends already stored in the database, processes each legend by identifying interesting relationships, filters relationships, ranks those relationships based on a number of parameters, and finally writes the resulting relationships to an XML file for later use. For comparison purposes, precision and recall are also computed by the system on a per-article basis. Details of the steps involved (see Fig. 3), namely syntactic analysis, semantic analysis, and concept mapping and association, are described in [10].

UMLS Project

The “Unified Medical Language System” (UMLS) from the National Library of Medicine (NLM) [12], a specialized ontology in biomedicine, provides standardized concepts for the creation of a controlled domain vocabulary. The UMLS provides a very powerful resource for rapidly creating a robust scientific thesaurus in support of precision searching. Further, the semantic type descriptors for each concept and semantic network offer some interesting opportunities for intelligent searching and mapping of concepts representing research findings, and their relationships.

4.2 Relationship Miner

RelationshipMiner system improves ConceptMiner by keeping the names of the relationships mined, and not only the concepts. For instance, the list of candidate relationships provided in the previous example results in *<effects of, caloric restriction, young age rats>*, *<effects of, caloric restriction, old age rats>*, *<effects of, ad-libitum, young age rats>* , and *<effects of, ad-libitum, old age rats>*, by keeping the “effects of” relationship name.

RelationshipMiner resorts to the UMLS also for this task of mining for relationship names. First, the project team has created a list of potential trigger words for relationships. This list is long, and comprises of course the relationship names from

the UMLS (see Fig. 4), but many others as well, such as synonyms, and variations. *MMTx* semantic analyzer [13], augmented by a domain dependent thesaurus including additional relationships, maps all these relationship names into their preferred form in the UMLS, called a canonical form. Canonical forms are the 54 relationship types in the UMLS semantic network.

isa associated_with physically_related_to part_of consists_of contains connected_to interconnects branch_of tributary_of ingredient_of spatially_related_to location_of adjacent_to surrounds traverses functionally_related_to affects manages treats disrupts complicates interacts_with prevents brings_about produces causes	[associated_with] (continued) [functionally_related_to] (continued) performs carries_out exhibits practices occurs_in process_of users manifestation_of indicates result_of temporally_related_to co-occurs_with precedes conceptually_related_to evaluation_of degree_of analyzes assesses_effect_of measurement_of measures diagnoses property_of derivative_of developmental_form_of method_of conceptual_part_of issue_in
--	--

Fig. 4. Extract of UMLS relationships (from NLM's UMLS project [29]).

More generally, RelationshipMiner mines for triples $\langle relationship-1,2, concept-1, concept-2 \rangle$ from a document. It also attaches a condition to a triple when it finds it to represent the information that IF a condition occurs, THEN an action or test is undertaken. This can be represented as $\langle relationship-1,2, concept-1, concept-2 \rangle IF$

$\langle \text{relationship-3,4, concept-3, concept-4} \rangle$. An example can be $\langle \text{startTreatment, Patient, PrednisoneAndCyclosporineTherapy} \rangle$ IF $\langle \text{property_of, ImmunosuppressantAgentNOS, absent} \rangle$. This structure is called a triple pair.

The RelationshipMiner involves two knowledge bases, UMLS database, and domain specific database (DSDB), which in particular stores the pre-processed documents that will serve as the input to the system. Within DSDB, the domain specific thesaurus represents the standardized vocabulary of the domain. Concept mining involves processing articles already stored in domain-specific database (DSDB). These articles comprise the full text of the original articles, parsed in several parts, such as title, summary, section part, figure and table legends, and so forth.

The RelationshipMiner follows these steps:

- Receive as input from ConceptMiner triples of the form $\langle \text{relationship-1,2, concept-1, concept-2} \rangle$.
- Map relationships to their canonical form in the UMLS.
- Detect patterns between the triples from one sentence, such as a “*property_of*” relationship in one triple, which signal the description of the state of objects, and other triples connected by expressions indicating a causal or sequential interaction, such as “*if ... then ... else ...*”, or their variants.
- Group corresponding triples into pairs of triples, in the form of $\langle \text{relationship-1,2, concept-1, concept-2} \rangle$ IF $\langle \text{relationship-3,4, concept-3, concept-4} \rangle$,
such as
 $\langle \text{startTreatment, Patient, PrednisoneAndCyclosporineTherapy} \rangle$
IF $\langle \text{property_of, ImmunosuppressantAgentNOS, absent} \rangle$.
- Produce as output triples organized in a semantic network through their association with other triples in pairs of triples.

In the Digital Knowledge Finder, we are mostly interested in the triples $\langle \text{relationship-3,4, concept-3, concept-4} \rangle$. A set of such triples is extracted from the figure and table legends, the text surrounding where they are referenced and discussed, and other key parts of the articles. This set is stored as the research findings of the articles.

5. Evaluation

This system was first evaluated with regard to its indexing feature for information search purposes. The success of the system is determined by how it affects the recall and precision ratios of the concept mining system. Previous results showed an average recall of 81% and precision of 50% for partial match for ConceptMiner [10]. Precision is the ratio of matching relations to the total number of relations identified. Recall is the ratio of matching relations to the total number of relations identified by the manual process.

The precision and recall are calculated in two ways: partial matching and total matching. In partial matching strategy, if the system extracted relationship (muscle mass – caloric restriction) and the manual results provided relationship (muscle mass increase – caloric restriction), and then this relationship is considered a match. In total matching, the relationship should be present in the manual results exactly matching both concepts. For RelationshipMiner, partial recall increases to 82%, and partial precision to 75%, which is a significant improvement.

The system is evaluated for 30 random articles. The average values of recall and precision for these 30 documents are shown in Table 1. It shows that the average values of precision and recall are much higher when partial matches of the concepts are also considered as a match. The reason for considering partial matching is that, there can be some implied knowledge that is used by the domain expert during the manual process, but that kind of knowledge is either not available to this system or hard to automate.

Table 1. Precision and recall ratios

Number of Documents	Total Recall	Total Precision	Partial Recall	Partial Precision
ConceptMiner	53%	35%	81%	50%
RelationshipMiner	63%	51%	82%	75%

The interpretation of why the precision in particular is significantly increased is that the system is able to better determine which pairs of concepts correspond to research findings, versus to background knowledge or other information. Human indexers were specifically trained at retaining from the documents their research findings, as the most interesting information for researchers to get from the articles. This was a notable limitation of ConceptMiner to not be able to discriminate enough between research findings and other types of information from the research articles, and one of the motivation to add the semantic relationships types dimension to the text mining process. One of the main issues to solve in data mining is to be able to discriminate among the knowledge learnt which is important and novel. In a system such as the concept miner, many more pairs of concepts are generated by the automatic process than by the human experts – even when limiting to mine figure and table legends. Therefore, a ranking system permits, with different criteria such as repetition and location in the document among others, to rank the pairs of concepts as being more or less important. The improvement to this ranking is in RelationshipMiner that the type of relationship is an essential criterion for assessing the importance of a relationship learned.

Research findings have been identified here as their relationship types being within the groupings of “*functionally_related_to*”, “*temporally_related_to*”, and some of the “*conceptually_related_to*” (see figure 4). Exclusion of semantic types such as

“*physically_related_to*” and “*spatially_related_to*” has proved to be a major advance in this system. Further tests are under way to refine more which relationship types are the most pertinent to keep. This analysis is not straight forward since the human indexers did not record the semantic types of the relationships, but only that there was a relationship between for example “caloric restriction” and “aging”, without further precision. Therefore it is by testing the level of recall and precision when adding or removing certain types of relationships that it is possible to learn which ones should be kept in priority.

Although the results of 82% in recall and 75% in precision are not perfect, in terms of information retrieval they are quite acceptable [14] – 60% precision is a minimum success threshold. Moreover, the system proposes a new functionality in terms of learning named relationships, which is a difficult task that few systems have been tackling.

6. Conclusion

In its current development state, the Digital Knowledge Finder system is capable of mining for research findings in scientific literature with good precision and recall. Precision is the ratio of matching relations to the total number of relations identified. Recall is the ratio of matching relations to the total number of relations identified by the manual process. The research finding mining process achieves 82% partial recall and 75% partial precision, which are satisfactory results. The next step in the system development is to mine for research questions and other parts of a case. We plan for a supervised mining process allowing experts and authors to update the automatic mining process. We will also focus on the reuse of this information to answer knowledge retrieval tasks of users such as: which are the most pertinent articles for my research? How is research evolving in particular research topics? Which could be interesting research questions to explore in a particular domain? Even with incomplete cases as they are now in the system, focusing on research findings, the Digital Knowledge Finder is capable of providing some answers to these questions.

References

1. Russell S., Norvig P.: Artificial Intelligence: A Modern Approach, 2nd edition. Prentice Hall (2003)
2. Bareiss, E.R., Porter, B.W., Wier, C.C.: Protos: an exemplar-based learning apprentice. In Proceedings of the Fourth International Workshop on Machine Learning, pp. 12–23. Morgan Kaufmann, Los Altos, CA, USA (1987)
3. Sandia National Laboratories.: Jess: the Rule Engine for the Java Platform. <http://www.jessrules.com/jess/index.shtml> (accessed: 15 June 2007) (2007)
4. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System. AI Communications 7(1). 39-59 (1994)
5. Swanson, D.R.: Information discovery from complementary literatures: Categorizing viruses as potential weapons. Journal of the American Society for Information Science Vol. 52(10), 797-812 (2001)

6. Swanson, D.R., Smalheiser, N.R.: An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence* Vol.9, 183-203 (1997)
7. Witten, I., Frank, R. *Data mining: Practical Machine Learning Tools and Techniques*, Second Edition. Morgan Kaufman Series in Data Management Systems. Elsevier Inc., San Francisco (2005)
8. Schmidt, R., Waligora, T., Vorobieva, O. Prototypes for Medical Case-Based Applications. In *Industrial conference on Data Mining. Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York (2008)
9. Montani, S.: Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. *Applied Intelligence*, 275-285 (2007)
10. Bichindaritz I., Akineni S.: Concept Mining from Biomedical Literature. In: Perner, P., Imiya, A. (eds.): *Proceedings of MLDM 05. Lecture Notes in Artificial Intelligence*, Vol. 3587. Springer-Verlag, Berlin, Heidelberg, New York (2005) 682-691
11. Fuller, S., Revere, D., Bugni, P., Martin, G.M.: A knowledgebase system to enhance scientific discovery: Telemakus. *Biomed Digit Libr.* Sep 21;1(1):2 (2004)
12. National Library of Medicine: The Unified Medical Language System. <http://umls.nlm.nih.gov> [Last access: 2005-04-01] (2005)
13. National Library of Medicine: MetaMap Transfer (MMTx), <http://mmtx.nlm.nih.gov> [Last access: 2005-04-01] (2005)
14. Schäfer U., Uszkoreit H., Federman C., Marek T., Zhang Y.: Extracting and Querying Relations in Scientific Papers. In: Dengel A. et al. (eds.): *KI 2008: Advances in Artificial Intelligence. Lecture Notes in Computer Science*, Vol. 5243. Springer-Verlag, Berlin, Heidelberg 127-134 (2008)

**EXPPORT: EXperience reuse:
Provenance, Process-ORientation and
Traces**

Workshop at the
Twenty-First International Conference on
Case-Based Reasoning
(ICCBR 2013)

Saratoga Springs, U.S.A.
July 9, 2013

David Leake, Béatrice Fuchs, Stefania Montani, and
Juan A. Recio-García, (Eds.)

Chairs

David Leake
Indiana University, U.S.A.

Béatrice Fuchs
LIRIS, University Jean Moulin Lyon 3, France

Stefania Montani
University of Piemonte Orientale, Italy

Juan A. Recio-García
Complutense University of Madrid, Spain

Programme Committee

Pierre-Antoine Champin, University Claude Bernard Lyon 1, France
Amélie Cordier, University Claude Bernard Lyon 1, France
Pedro A. González-Calero, Complutense University of Madrid, Spain
Joseph Kendall-Morwick, Indiana University, U.S.A.
Mirjam Minor, Frankfurt University, Germany
Hector Muñoz-Avila, Lehigh University, U.S.A.
Thomas Roth-Berghofer, University of West London, UK
Barbara Weber, University of Innsbruck, Austria

Invited Reviewer

Daniel Garijo, Universidad Politécnica de Madrid, Spain

Preface

The workshop “EXPPORT: EXperience reuse: Provenance, Process-Oriented and Traces,” held at ICCBR-13, aimed to provide an opportunity for exchange of new results and ideas about case-based reasoning for processes, traces, and workflows. Provenance, process-oriented CBR, and traces are all strong areas of current interest in the CBR community, and are closely related. Process-oriented CBR focuses largely on workflows, which define sequences of actions for use and reuse. Provenance compiles the results of such sequences, providing a resource for generating workflow cases and for analyzing how action sequences may be refined. Finally, traces capture the results of action sequences generated on the fly, to generate useful cases from execution information. All these areas have been the subject of successful workshops in their own right at previous case-based reasoning conferences, with workshops on process-oriented case-based reasoning (PO-CBR), trace-based reasoning (TRUE), and provenance-aware in case-based reasoning (PA-CBR). EXPPORT brought these research communities together for the first time, providing a forum for the discussion of trends, research issues and practical experiences in all the areas illustrated above.

The EXPPORT program included six papers, reflecting a range of perspectives from researchers addressing issues in the workshop’s three target areas. In “Process mining and case-based retrieval for assessing the quality of medical processes,” Montani et al. describe how process mining and case based retrieval can be used to help understand and redesign health case processes to improve their quality and efficiency. The paper “A pipes-and-filters framework for the extraction of workflow cases from text” by Schumacher, Minor, and Schulte-Zurhausen presents a mixed textual CBR and process-oriented CBR approach to aid workflow extraction, focusing on a case study of anaphora extraction. In “A Case Based Reasoning Approach to Business Workflow Modelling Based on Formal Temporal Theory,” Kapetanakis, Petridis, and Knight enhance the monitoring of business processes by means of formal temporal theory used to represent processes and workflows. This representation enables a better explanation of both the similarity and relevance of retrieved cases.

“Collecting fine-grained use traces in any application without modifying it,” by Ginon, Champin and Jean-Daubias addresses the trace capture problem, presenting a trace collector which it can monitor Windows target-applications to collect the user’s traces. The paper “Building a Trace-Based System for Real-Time Strategy Game Traces,” by Wender, Cordier and Watson present a conception of a visualization and transformation tool for the real-time strategy game StarCraft, aimed at improving the learning of a machine learning agent, and illustrate how the tool can help to better understand player behavior traces. Finally, the paper “Toward Addressing Noise and Redundancies for Cases Captured from Traces and Provenance,” by Kendall-Morwick and Leake, identifies new issues raised by extracting cases from traces or provenance information, and proposes new research directions to address them.

We would like to thank everyone who contributed to the success of this workshop, especially the authors, the program committee members and invited reviewer, and the organizers of the ICCBR 2013 conference.

July 2013
David Leake

Béatrice Fuchs

Stefania Montani

Juan A. Recio-García

Process mining and case-based retrieval for assessing the quality of medical processes

S. Montani (1), G. Leonardi (1,2), S. Quaglini (2),
A. Cavallini (3), G. Micieli (3)

(1) DISIT, Computer Science Institute, Università del Piemonte Orientale,
Alessandria, Italy

(2) Dipartimento di Informatica e Sistemistica, Università di Pavia, Italy

(3) IRCCS Fondazione “C. Mondino”, Pavia, Italy - on behalf of the Stroke Unit
Network (SUN) collaborating centers

Abstract. In a competitive healthcare market, hospitals have to focus on ways to deliver high quality care while at the same time reducing costs. To accomplish this goal, hospital managers need a thorough understanding of the actual processes. Process mining can be used to extract process related information (e.g., process models) from data. This process information can be exploited to understand and redesign processes to become efficient high quality processes. Process analysis and redesign can take advantage of Case Based Reasoning techniques.

In this paper, we present a framework that applies *process mining* and *case retrieval* techniques, relying on a novel distance measure, to stroke management processes. Specifically, the goal of the framework is the one of analyzing the quality of stroke management processes, in order to verify whether different patient categories are differently treated, and whether hospitals of different levels (defined by the absence/presence of specific resources) actually implement different processes (as they auto-declare). Some first experimental results are presented and discussed.

1 Introduction

Healthcare institutions are increasingly facing pressure to reduce costs, while at the same time improving the quality of care. In order to reach such a goal, healthcare administrators and expert physicians need to evaluate the services the institution provides. Service evaluation requires to analyze medical processes, which are often automated and logged by means of the workflow technology.

Process analysis (PA) covers functions of simulation and diagnosis of processes. While simulation can support performance issues evaluation, diagnosis can highlight e.g., similarities, differences, and adaptation/redesign needs. Indeed, the existence of different patients categories, or of local resource constraints, can make differences between process instances necessary, and process adaptation compulsory (even when the medical process implements a well-accepted clinical guideline). Proper PA techniques are strongly needed when a given process model does not exist, e.g., because a full clinical guideline has

not been provided, and only some recommendations are implemented. In this case, *process mining* techniques [4] can be exploited, to extract process related information (e.g., process models) from log data. It is worth noting, however, that the mined process can also be compared to the existing guideline (if any), e.g., to check conformance, or to understand the required level of adaptation to local constraints. Thus, the mined process information can always be used to understand, adapt and redesign processes to become efficient high quality processes.

The *agile workflow* technology [17] is the technical solution which has been invoked to deal with process adaptation/redesign. In order to provide an effective and quick adaptation support, many agile workflow systems share the idea of recalling and reusing concrete *examples of changes* adopted in the past. To this end, *Case Based Reasoning (CBR)* [1] has been proposed as a natural methodological solution [9, 16, 11, 12, 7]. In particular, the *case retrieval* step has been extensively studied in PA applications, since the nature of processes can make distance calculation and retrieval optimization non-trivial [13, 14, 2, 8].

In this paper, we propose a framework for medical process analysis and adaptation, which relies on **process mining** and **case retrieval** techniques.

Specifically, our goal is the one of analyzing the quality of stroke management processes, in order to verify: (i) whether different patient categories are differently treated (as expected), and (ii) whether hospitals of different levels (defined by the absence/presence of specific resources for stroke management) actually implement different processes (as they auto-declare).

First, our system extracts process models from a database of real world process logs. In particular, we learn different models for every patient category, and/or for every hospital. Given one of the models as an input, we then retrieve and order the most similar models we have learned. An examination of the distance among the models, to be conducted by a medical expert, can provide information about the quality of the processes, by verifying and quantifying issues (i) and (ii) above. To this end, we have introduced a proper *distance definition*, that extends previous literature contributions [5, 3, 2] by considering the available information, learned through process mining. In this paper, we will focus on issue (ii). Technical details of our approach and experimental results are discussed in the next sections.

2 Methods

2.1 Process Mining and the ProM tool

Process mining describes a family of a-posteriori analysis techniques exploiting the information recorded in logs, to extract process related information (e.g., process models). Typically, these approaches assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well defined step in the process) and is related to a particular case (i.e., a process instance). Furthermore, some mining techniques use additional information such as the timestamp of the event, or data elements recorded with the event.

Traditionally, process mining has been focusing on discovery, i.e., deriving process models and execution properties from enactment logs. It is important to mention that there is no a-priori model, but, based on process logs, some model, e.g., a Petri net, is constructed. However, process mining is not limited to process models (i.e., control flow), and recent process mining techniques are more and more focusing on other perspectives, e.g., the organization perspective, the performance perspective or the data perspective. Moreover, as well stated in [6], process mining also supports conformance analysis and process enhancement.

We are relying on the process mining tool called ProM, extensively described in [15]. ProM is a platform independent, open source framework which supports a wide variety of process mining and data mining techniques, and can be extended by adding new functionalities in the form of plug-ins.

In particular, we are exploiting the Heuristic miner [18] plug-in for mining the process models, and a performance analysis plug-in to project information of the mined process on places and transitions in a Petri Net. Different kinds of performance indicators can be obtained for the discovered Petri Net. Moreover, once such a Petri Net is available, simulations with different parameters can be run to see what the consequences are after the removal of a bottleneck, e.g., change in throughput time. For instance, the Petri Net can provide average/variance of the total flow time or the time spent between activities.

2.2 Distance definition for case retrieval

In order to retrieve process models and order them on the basis of their distance with respect to a given query model, we have introduced a distance definition that extends previous literature contributions [5, 3, 2] by properly considering the available information, learned through process mining.

In particular, since mined process models are represented in the form of graphs (where nodes represent activities and edges provide information about the control flow), we define a distance based on the notion of graph edit distance [3]. Such a notion calculates the minimal cost of transforming one graph into another by applying insertions/deletions and substitutions of nodes, and insertions/deletions of edges.

As in [5], we provide a normalized version of the approach in [3], and as in [5, 2], we calculate a *mapping* between the two graphs to be compared, so that edit operations only refer to mapped nodes (and to the edges connecting them).

Moreover, with respect to all the previous approaches, we introduce two novel contributions:

1. we calculate the cost of node substitution f_{subn} (see Definition 2 below) by applying **taxonomic distance** [14, 13] (see Definition 1), and not string edit distance on node names as in [5]. Indeed, we organize the various activities executable in our domain in a taxonomy, where activities of the same type (e.g., Computer Assisted Tomography (CAT) *with* or *without* contrast) are connected as close relatives. The use of this definition allows us to explicitly take into account this form of domain knowledge, since the distance between

two activities is set to the normalized number of arcs on the path between the two activities themselves in the taxonomy (see Definition 1);

2. we add a cost contributions related to edge substitution ($fsube$ in Definition 2 below), that incorporates information learned through process mining, namely (i) the percentage of patients that have followed a given edge, and (ii) the reliability of a given edge, i.e., of the control flow relationship between two activities. Both items (i) and (ii) are outputs of Heuristic miner (see below for definitions).

Formally, the following definitions apply:

Definition 1: Taxonomic Distance.

Let α and β be two activities in the taxonomy t , and let γ be the closest common ancestor of α and β . The *Taxonomic Distance* $dt(\alpha, \beta)$ between α and β is defined as:

$$dt(\alpha, \beta) = \frac{N_1 + N_2}{N_1 + N_2 + 2 * N_3}$$

where N_1 is the number of arcs in the path from α and γ in t , N_2 is the number of arcs in the path from β and γ , and N_3 is the number of arcs in the path from the taxonomy root and γ .

Definition 2: Extended Graph Edit Distance. Let $G1 = (N1, E1)$ and $G2 = (N2, E2)$ be two graphs, where Ei and Ni represent the sets of edges and nodes of graph Gi . Let M be a partial injective mapping [5] that maps nodes in $N1$ to nodes in $N2$ and let $subn$, $sube$, $skipn$ and $skipe$ be the sets of substituted nodes, substituted edges, inserted or deleted nodes and inserted or deleted edges with respect to M . In particular, a substituted edge connects a pair of substituted nodes in M . The fraction of inserted or deleted nodes, denoted $fskipn$, the fraction of inserted or deleted edges, denoted $skipe$, and the average distance of substituted nodes, denoted $fsubn$, are defined as follows:

$$fskipn = \frac{|skipn|}{|N1| + |N2|}$$

$$fskipe = \frac{|skipe|}{|E1| + |E2|}$$

$$fsubn = \frac{2 * \sum_{n,m \in M} dt(n, m)}{|subn|}$$

where n and m are two mapped nodes in M .

The average distance of substituted edges $fsube$ is defined as follows:

$$f_{sube} = \frac{\sum_{(n1,n2),(m1,m2) \in M} (|rel(e1) - rel(e2)| + |pat(e1) - pat(e2)|)}{|sube|}$$

where edge $e1$ (connecting node $n1$ to node $m1$) and edge $e2$ (connecting node $n2$ to node $m2$) are two substituted edges in M , $rel(ei)$ is the reliability $\in [0, 1]$ of edge ei , and $pat(ei)$ is the percentage of patients that crossed edge ei .

In particular, the reliability of a relationship (e.g., activity a follows activity b) is not only influenced by the number of occurrences of this pattern in the logs, but is also (negatively) determined by the number of occurrences of the opposite pattern (b follows a). Specifically, the reliability of the edge ei assessing that activity a directly follows activity b in sequence (i.e., ei is an arc from b to a) is calculated as [18]:

$$rel(ei) = \frac{|a > b| - |b > a|}{|a > b| + |b > a| + 1}$$

where $|a > b|$ is the number of traces in which activity a directly follows activity b , and $|b > a|$ is the number of traces in which activity b directly follows activity a (a negative reliability value means that we must conclude that the opposite pattern holds, i.e., activity b follows activity a).

On the other hand, $pat(ei)$ is calculated as:

$$pat(ei) = \frac{|a > b| * 100}{ALLTRACE}$$

where $ALLTRACE$ is the total number of available traces (i.e., of patients) in the database.

The extended graph edit distance induced by the mapping M is:

$$ext_{edit} = \frac{wskipn * fskipn + wskipe * fskipe + wsubn * fsubn + wsube * fsube}{wskipn + wskipe + wsubn + wsube}$$

where $wsubn$, $wsube$, $wskipn$ and $wskipe$ are proper weights $\in [0, 1]$.

The extended graph edit distance of two graphs is the minimal possible distance induced by a mapping between these graphs. To find the mapping that leads to the minimal distance we resort to the greedy algorithm described in [5].

3 Experimental results

In clinical practice, no support is available to physicians/administrators to verify whether hospitals of different levels actually implement different processes when caring a specific pathology (see issue (ii) described in the Introduction). In [10], process mining was relied upon to provide physicians with a graphical view of the

mined processes. A visual inspection of those figures can be a first help towards the fulfillment of the tasks related to issue (ii). However, mined processes can be huge and very complex (see figure 1 for an example), so that an automated comparison among them, like the one we are providing in this framework, can truly be an added value for quality evaluation.

In the rest of this section, we discuss our experimental results, related to issue (ii). In particular, we wished to test whether the level of 37 hospitals located in the Lombardia Region (Northern Italy) could be verified (or corrected) through our framework, when referring to stroke care. Hospital levels (i.e., 1, 2, 3) have to be defined in Lombardia Region according to the available human and instrumental resources. Every hospital auto-declares its own level. Specifically, we mined the stroke management processes implemented in all 37 hospitals. We then chose one level-2 hospital process as a query, and we retrieved and ordered the mined processes of the 36 others (21 of which were declared as level-2 hospitals as well). We wished to test whether the most similar processes were in the same category as the query process. We performed retrieval and ordering both resorting to the distance defined in [5], and to the novel one introduced in section 2.2. Results are reported in figure 2.

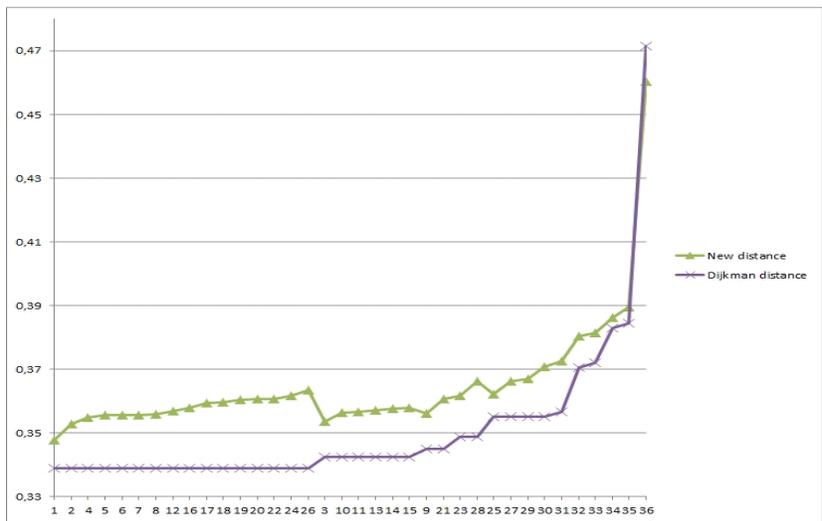


Fig. 2. Retrieval and ordering of 36 mined processes, implemented in 36 different hospitals in the Lombardia region, with respect to the selected query process (on the x-axis: process number; on the y-axis: distance value from the query). Results are shown in two different framework settings: when relying on the metric in [5] (Dijkman distance), and when relying on the metric defined in section 2.2. Processes are order on the basis of the Dijkman distance.

First, we can observe that our distance is able to discriminate among every single mined processes, while the one in [5] only identifies some groups, composed by several processes, whose distance from the query process does not change (see horizontal segments in figure 2). We believe that the finer distinction we could obtain is due to the use of taxonomic distance, and of edge information, which are disregarded by [5]¹. This additional information can be very significant from a medical viewpoint. For instance, hospitals 2 and 20 are not distinguishable according to [5], but in hospital 20 more than 70% of the patients undergo ECG immediately after CAT, while in hospital 2 this occurs for only 10% of the patients. Almost all patients undergo these tests in the two hospitals indeed, but within different control flow patterns. In hospital 20 there seems to be a behavioral rule pushing for the pattern CAT *immediately followed by* ECG, while in the other hospital this direct sequential pattern does not exist. This is an edge-related information extracted by Heuristic miner, and properly used by our metric for providing its finer ordering.

As for the declared hospital levels, we considered the 22 closest processes (i.e., hospitals) with respect to the query. This number was chosen because it is the sum of the number of processes in the two closest groups when resorting to [5] (16 processes belong to the first group, 6 to the second), and with [5] it is not possible to further refine the ordering among these examples. If the auto-declared level of these examples was correct (and confirmed by the mined processes), we should find 21 level-2 hospitals in this set. However, this did not happen. When resorting to [5], we found only 13 level-2 hospitals in these nearest neighbors. Of them, only 9 were listed in the closest 16 (i.e., the first group). When exploiting our distance, we still found 13 level-2 hospitals in the first 22, but 11 of them were in the first 16. Our results were thus closer to the expected ones.

We analyzed the situation of the remaining 8 level-2 hospitals, that were not found in the nearest neighbors. Very interestingly, 7 of these missing examples are the very same when resorting to the two different metrics. Indeed, the visual examination of the graphs highlights important differences with respect to the query hospital. For example, one of them does not perform the thrombolysis treatment, even if typical of level-2 stroke units (see figure 1). We have to say that some local conditions (e.g., specific resources availability) may have recently changed, altering the real level of some hospitals with respect to the originally declared one. This conclusion thus supports the quality of the implemented metrics, and of our novel contribution in particular.

As a final consideration, we can quickly comment on 4 cases, that were differently ordered by the two metrics. According to the auto-declared levels, our ordering is closer to reality in 3 of them (no. 9, 22 and 24), while in the fourth case (no. 26) our metric overestimates the distance between the hospital and

¹ Distance values are of course not identical (the two distance definitions are different); our distance usually provide slightly larger values, but this is not a significant information per se. On the other hand, the ability to better discriminate among single processes is interesting, and potentially very useful in our application domain.

the query. Despite the overall positive outcome, this motivates further improvements, like the ones we will discuss in section 4.

4 Discussion, conclusions and future work

This work showed that process mining and case retrieval techniques can be applied successfully to clinical data to gain a better understanding of different medical processes adopted by different hospitals. It is interesting to analyze the differences, to establish whether they concern only the scheduling of the various tasks or also the tasks themselves. In this way, not only different practices may be discovered that are used to treat similar patients, but also unexpected behavior may be highlighted.

In this paper we have shown some first experimental results. More tests are obviously needed, including leave-one-out style experiments and comparisons with other metrics, and are planned for the next months.

In the future we also wish to extend our contribution, by including the treatment of time in *fsube* (see Definition 2 in section 2.2). Indeed, by projecting the mined process on a Petri Net (see section 2.1), we can obtain information about delays between activities, possible overlaps and synchronizations. We would like to explicitly compare this information between mapped processes. We believe that, since in emergency medicine the role of time is clearly central, this enhancement could represent a relevant added value in our framework, and make it even more reliable and useful in practice.

5 Acknowledgments

This research is partially supported by the GINSENG Project, Compagnia di San Paolo. Patients data are property of SUN; as such, they are not publicly available.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. *AI Communications*, 7:39–59, 1994.
2. R. Bergmann and Y. Gil. Retrieval of semantic workflows with knowledge intensive similarity measures. In A. Ram and N. Wiratunga, editors, *Proc. International Conference on Case-Based Reasoning (ICCBR) 2011, Lecture Notes in Artificial Intelligence 6880*. Springer-Verlag, Berlin, 2011.
3. H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689694, 1997.
4. W. Van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters. Workflow mining: a survey of issues and approaches. *Data and Knowledge Engineering*, 47:237–267, 2003.
5. R. Dijkman, M. Dumas, and R. Garca-Banuelos. Graph matching algorithms for business process model similarity search. In *Proc. International Conference on Business Process Management*, pages 48–63, 2009.

6. [http : //www.win.tue.nl/ieetfpm](http://www.win.tue.nl/ieetfpm). IEEE Taskforce on Process Mining: Process Mining Manifesto.
7. S. Kapetanakis, M. Petridis, B. Knight, J. Ma, and L. Bacon. A case based reasoning approach for the monitoring of business workflows. In I. Bichindaritz and S. Montani, editors, *Proc. International Conference on Case Based Reasoning (IC-CBR)*, pages 390–405. Springer, Berlin, 2010.
8. J. Kendall-Morwick and D. Leake. On tuning two-phase retrieval for structured cases. In L. Lamontagne and J. A. Recio-García, editors, *Proc. ICCBR 2012 Workshops*, pages 25–334, 2012.
9. Z. Luo, A. Sheth, K. Kochut, and J. Miller. Exception handling in workflow systems. *Applied Intelligence*, 13:125–147, 2000.
10. R. Mans, H. Schonenberg, G. Leonardi, S. Panzarasa, A. Cavallini, S. Quaglini, and W. Van der Aalst. Aprocess mining techniques: an application to stroke care. In *Proc. Medical Informatics Europe (MIE)*, pages 573–578, 2008.
11. M. Minor, A. Tartakovski, D. Schmalen, and R. Bergmann. Agile workflow technology and case-based change reuse for long-term processes. *International Journal of Intelligent Information Technologies*, 4(1):80–98, 2008.
12. S. Montani. Prototype-based management of business process exception cases. *Applied Intelligence*, 33:278–290, 2010.
13. S. Montani and G. Leonardi. Retrieval and clustering for supporting business process adjustment and analysis. *Information Systems*, DOI: <http://dx.doi.org/10.1016/j.is.2012.11.006>.
14. S. Montani and G. Leonardi. Retrieval and clustering for business process monitoring: results and improvements. In B. Diaz-Agudo and I. Watson, editors, *Proc. International Conference on Case-Based Reasoning (ICCBR) 2012, Lecture Notes in Artificial Intelligence 7466*, page 269283. Springer-Verlag, Berlin, 2012.
15. B. van Dongen, A. Alves De Medeiros, H. Verbeek, A. Weijters, and W. Van der Aalst. The proM framework: a new era in process mining tool support. In G. Ciardo and P. Darondeau, editors, *Knowledge Mangement and its Integrative Elements*, pages 444–454. Springer, 2005.
16. B. Weber, M. Reichert, and W. Wild. Case-based maintenance for CCBR-based process evolution. In T. Roth-Berghofer, M. Goker, and H. Altay Guvenir, editors, *Proc. European Conference on Case Based Reasoning (ECCBR) 2006, LNAI 4106*, pages 106–120. Springer, Berlin, 2006.
17. B. Weber and W. Wild. Towards the agile management of business processes. In K. D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer, editors, *Professional knowledge management WM 2005, LNCS 3782*, pages 409–419, Washington DC, 2005. Springer, Berlin.
18. A. Weijters, W. Van der Aalst, and A. Alves de Medeiros. *Process Mining with the Heuristic Miner Algorithm, BETA Working Paper Series, WP 166*. Eindhoven University of Technology, Eindhoven, 2006.

Anaphora resolution in a pipes-and-filters framework for workflow extraction

Pol Schumacher, Mirjam Minor, and Eric Schulte-Zurhausen

Goethe Universität Frankfurt - Institut für Informatik
D-60325 Frankfurt am Main, Germany
[schumacher|minor|eschulte]@cs.uni-frankfurt.de

Abstract. A central issue in Textual Case-Based Reasoning (TCBR) is the creation of structured case representations from text. Usually these systems apply a Bag-of-Words indexing approach, only few use more advanced methods. We aim at combining TCBR and the recently emerged process oriented Case-Based Reasoning (POCBR). Therefore we developed a workflow extraction framework which allows deriving a formal representation based on workflows (wf) from textual process description. The framework is based on a pipes-and-filters architecture and uses NLP tools to perform information extraction steps. Besides these standard tasks, our framework is able to incrementally create wf models by analysing and modifying the wf models. In detail we present the step of anaphora resolution. Anaphora resolution is a part of the creation of the data-flow, which shows the flow of the data-objects through the wfs. We performed an evaluation of the data-flow for 37 workflows.

1 Introduction

The creation of structured case representations from text has been a central research issue in Textual Case-Based Reasoning (TCBR) for several years [1]. Many Bag-of-Words indexing approaches have been developed based on information extraction methods [2, 3]. A couple of TCBR approaches extract information beyond Bag-of-Words in order to facilitate adaptation of text [4] and more advanced similarity measures for case retrieval[5]. Recently, the extraction of workflows (wfs) for structured case representations has been introduced in research on process-oriented Case-Based Reasoning (POCBR) systems [6, 7]. Traditionally, *workflows* are "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [8]. In a broader notion, wfs describe any flow of activities not necessarily in the sense of a classical business process that involves several participants. This more general view on wfs is used during the paper. A wf consists of a set of *activities* combined with *control-flow structures* like sequences, parallel or alternative branches, and loops. In addition, activities consume *resources* and create certain *products*, which both can be physical matter (such as cooking ingredients or a physical business card) or information. Samples for wfs are the step-by-step creation of a

business card, the processing of an order from a customer or a cooking procedure for preparing a meal.

The extraction of wfs from textual descriptions requires Natural Language Processing (NLP) [9]. NLP tool-chains apply a set of transformations to a stream of text data called *filters*. However, standard NLP tool-chains [10,11] are not adequate for wf extraction since they do not cover wf-specific filters. Further, the adaptation of a standard NLP tool-chain to a new text corpus is quite laborious. Wf extraction requires a certain degree of flexibility. As an example, the source for the extraction might be a Web page with semi-structured content like cooking recipes. If a new Web page with a slightly different structure of the content is integrated, some of the extraction filters should be exchanged or omitted. We identified a research gap addressing a flexible, partly domain-specific tool-chain dedicated to wf extraction. This paper presents on a novel filter-and-pipe framework for the extraction of wfs from textual process descriptions. This paper is organized as follows. In the next section we describe the framework and a short description of the application in the cooking domain. The subsequent section introduces the data-flow creation and anaphora resolution (see section 3) approaches followed by an evaluation and the discussion of the results. The paper ends with the related work, a short conclusion and an outlook on our future work.

2 Workflow extraction framework

Systems which process natural language need to be flexible and extensible. While there are a lot of systems for generic NLP tasks, there is none for wfs. Therefore we developed a framework which should support the flexibility that is needed for such an application. The framework is based on a pipes-and-filters architecture.

2.1 Workflow representation

The target of wf extraction is a formal representation of the wf in a wf description language. Our wf description language is block oriented and based on the data-model of the CAKE framework¹, this enables us to use the built-in functionality of the framework for Case-Based Reasoning [12]. A wf consists of the control-flow and the data-flow. The control-flow describes the order in which activities are executed. An activity processes resources like information or ingredients. The simplest form of a control-flow is a sequence of activities. A sequence can contain an XOR-, AND-, or LOOP-blocks. These building blocks cannot be interleaved but they can be nested. In addition an activity has a set of semantic descriptors, resources and products. A semantic descriptor is for example the name of the task or additional information which describes "how" a task should be performed, e.g. "for 10 minutes". Resources contain a set of semantic information, these describe additional information about the resources, e.g. amounts or if a resource should be preprocessed like "chopped".

¹ Collaborative Agile Knowledge Engine

2.2 Information extraction software

The framework uses the information extraction software SUNDANCE (Sentence UNDERstanding ANd Concept Extraction) developed by Ellen Riloff [13]. SUNDANCE performs the usual NLP task like tokenization or part of speech tagging but we use SUNDANCE because it has good balance between coverage and robustness.

The SUNDANCE parser assigns syntactic roles (subject, direct object, and indirect object) to text snippets based on a heuristic. Then the SUNDANCE information extraction engine tries to fill a case frame as follows. Each case frame specifies a trigger phrase. If the trigger phrase is detected in a sentence the according case frame is activated. This means that the activation functions of the frame try to match the specified linguistic patterns with the syntactic roles. A slot specifies a syntactic role whose content is extracted from the text.

2.3 Extraction pipeline

The framework is based on a pipes-and-filters [14] architecture. Such an application is a sequence of filters which are connected by pipes. A filter is a self-contained element which performs a data transformation step on the data-stream. The pipes channel the data-stream from the output of a filter to the input of the subsequent filter. A data-stream is sent through this pipeline and each filter is applied to the stream.

At the beginning of the pipeline, the case initially consists of the textual process description. While the case passes through the pipeline it is enriched with additional structure. At the end of the pipeline we have a complete case consisting of the textual process description and the formal wf representation.

Our framework extends the original pipes-and-filters architecture. We allow two different types of filters. The first one, the so called *local filters* operate with a focus on one case. The second one, the *window filters* collect a part of the case-stream (e.g. 5000 cases) and operate on that. The model of a window filter is necessary, because the framework processes a stream of cases which is potentially infinite. The intention is to employ statistical methods for a larger number of textual process descriptions. The statistical approaches benefit from the pipes and filter principle because we employ them on processed data. This intermediate data is the result of the preceding steps of the extraction pipeline. It contains less noise and has more structure than the raw input data.

Figure 1 shows a sample pipeline for the cooking domain. The different filters are created manually the details are described in [6].

3 Data-flow creation and anaphora resolution

In this section we introduce our method which is used to resolve anaphoras in a cooking wf. An anaphora is a linguistic entity which indicates a referential tie to some other entity in the same text [15]. The anaphora resolution is necessary

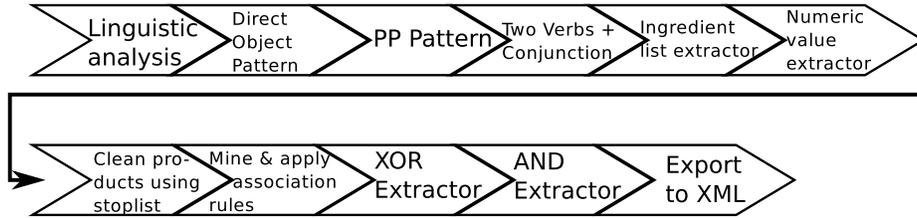


Fig. 1: Overview of the extraction pipeline for the cooking domain.

to complete the data-flow. Several approaches exist to perform the anaphora resolution. Our anaphora resolution approach is based on frequent sequential pattern. We chose this approach because it does not need a complex ontology. We are going to describe the mining and the application of those patterns.

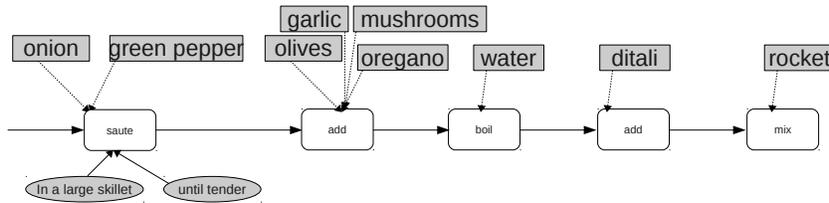


Fig. 2: Sample workflow for cooking rocket pasta with vegetable.

Listing 1.1: Sample transactions for workflow in Fig. 2.

WorkflowId, TransactionTime, { Items }	0.025: <item=butter><item=dough>
0.0, { Ditali, rocket, water, herbs, onion, mushrooms, garlic, oregano, green pepper }	0.024: <item=butter><item=mixture>
0.1, { onion, green pepper }	0.024: <item=flour><item=batter>
0.2, { garlic, mushrooms, olives, oregano }	0.021: <item=eggs><item=batter>
0.3, { water }	0.021: <item=flour><item=dough>
0.4, { ditali }	0.018: <item=yeast><item=dough>
0.5, { rocket }	0.016: <item=baking powder><item=batter>
	0.016: <item=butter><item=batter>
	0.015: <item=garlic><item=mixture>
	0.015: <item=vanilla><item=batter>

Listing 1.2: Top 10 pattern.

3.1 Mining sequential pattern

We use a method that was presented by Agrawal [16] to mine sequential patterns in a sequence of transactions. Listing 1.1 displays the transactions which are created from the wf in Fig. 2. It assumes that the wf has the id 0. The items at the transaction at time t are taken from the corresponding ingredient list of

the original recipe. For the details of the sequential pattern mining algorithm we refer to the original paper [16]. The algorithm delivers a set of sequential patterns (see Listing 1.2) with a corresponding support value. The minimum support value which we use is 0.005 . This value is domain dependent and needs to be tuned for a specific domain.

3.2 Creation of data-flow

A sequential pattern can be seen as an association rule. The left side of the rules is the first item-set and the right side of the rule the second item-set of a sequential pattern. A sequential pattern like e.g. $\langle\langle groundbeef, tomato \rangle, \langle pastasauce \rangle\rangle$ would result in an association rule $\{ground\ beef, tomatoes\} \Rightarrow \{pasta\ sauce\}$. Two observations about anaphoras in cooking wf can be made. Firstly anaphoras are not enumerated in the resource list. Secondly resources of an anaphora are used before the anaphora or are included in the resource list.

The first observation enables us to delete a lot of unnecessary rules. We can delete all rules whose right side contains an ingredient.

The creation of the data-flow is a two phase procedure. The first phase is related to the extraction of tasks. Activities are usually extracted with a set of resources related to them. These resources are used as products. The second phase is the anaphora resolution and the creation of products. We implemented three different approaches which we are going to introduce.

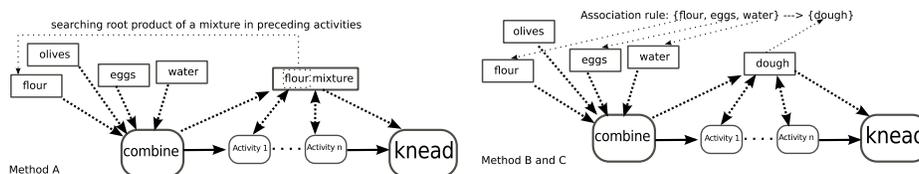


Fig. 3: Illustration of M. A and M. B.

Method A (M. A, see Fig. 3) is based on the observation, that a lot of anaphoras contain the token "mixture". This approach searches for resources which contain this token. If such a resource is found, we copy the resource and delete the token "mixture". Now the root resource is produced. In the next step we scan the resources of the preceding activities for this root resource. For those activities we check if they contain multiple resources. In that case we create a sole product *root-product + "mixture"* for that activity. At the end we complete the data-flow in the way that products are copied as resources to the next activity. For Fig. 3 the resource "flour mixture" is found at the activity "knead". After deleting the token "mixture" we get the resource "flour". We search for the resource flour in the preceding activities and we find it at the activity "combine"

and the activity uses four resources. We assume now, that the activity combine produces the product "*flour mixture*"

Method B (M. B, see Fig. 3) includes M. A. In addition it iterates over all activities, starting at the first. For each resource of the activity, it is checked if there is an association rule with a matching right side. If such a rule is found, it is looked if one of the previous activities has a resource matching the left side of the rule. We assume then that the anaphora enters the wf at the activity where the left side of the rule is found. Therefore we add the right side of the rule (the anaphora) as sole product of that activity. In the case that multiple rules are found, the one with the best support value is chosen. In Fig. 3 the search for a right side of a rule results with the resource "*dough*". After a match, the algorithm searches the preceding activities for the resources of the left side of the rule "*flour, eggs, water*". These resources are linked to the activity "*combine*". Although the activity has more resources than the left side of the rules enumerates (the resource "*olives*" is not included in the rule) the sole product "*dough*" is created for the activity "*combine*".

Method C (M. C) includes M. B. A domain specific list of items is added. This list in order to filter out items which were incorrectly extracted as resource. In the cooking domain this list is used to filter out cooking ware.

4 Evaluation

This section describes our hypotheses and our data-flow evaluation approach. The performance of the different methods was measured using the standard evaluation functions Precision, Recall and F-measure [17]. We tested three hypotheses. The first one is that the data-flow created by M. A has the best precision in comparison to M. B and M. C. The second one states that the data-flow created by M. B has a better recall than M. A. The last one declares that the data-flow created by M. C has the highest F-measure in comparison to M. B and M. C.

4.1 Experimental Setup

The experiment was performed on a set of 37 recipes. These recipes were selected randomly from a set of 36 898 recipes which were crawled from a cooking community website². A human expert modelled the data-flow for the recipes in the test set. This served as the golden standard for the evaluation. As the evaluation aimed at the data-flow our expert got the control-flow which was automatically extracted as framework for the golden standard wf. This approach eliminated the paraphrasing and granularity problem of the control-flow. The expert was allowed to use all resources and products that she thought should be in the data-flow, even if they were not mentioned in the text. We got a semantically correct data-flow. The only constraint was, that the expert was not allowed to

² www.allrecipes.com

Recipe	P_A	P_B	P_C	R_A	R_B	R_C	F_A	F_B	F_C
Mexican Egg Bake	0.55	0.55	0.55	0.67	0.67	0.67	0.61	0.61	0.61
Classic Thumbprint Cookies	0.58	0.37	0.37	0.32	0.29	0.29	0.41	0.33	0.33
Cranberry Glazed Roast Pork	0.48	0.46	0.52	0.31	0.31	0.33	0.38	0.37	0.41

Table 1: Results of the evaluation for selected cases.

use synonyms for products which were mentioned in the original recipe texts. If a product was mentioned in the text, this term must be used in the data-flow. This restriction should reduce the paraphrasing problem for the data-flow. We adapted them to our scenario. Every activity in the golden standard wf had per definition a corresponding activity in the evaluated wf.

Let T and T' be sets of activities of the wfs W and W' . W' is the golden standard wf. Each activity $t_i \in T$ has a corresponding activity $t'_i \in T'$ which are equal except for the resource and product sets. Let I_i and I'_i be resource sets and O_i and O'_i product sets for the activities t_i and t'_i . The precision for an activity $t_i \in T$ is defined as: $precision(t_i) = \frac{|I_i \cap I'_i| + |O_i \cap O'_i|}{|I_i| + |O_i|}$. The recall for a activity is defined as: $recall(t_i) = \frac{|I_i \cap I'_i| + |O_i \cap O'_i|}{|I'_i| + |O'_i|}$

This leads to the evaluation functions for a wf: $precision(W) = \frac{1}{|T|} \sum_{i=1}^{|T|} precision(t_i)$
 $recall(W) = \frac{1}{|T|} \sum_{i=1}^{|T|} recall(t_i)$

The F_1 measure is defined as: $F_1(W) = 2 \frac{precision(W) * recall(W)}{precision(W) + recall(W)}$

4.2 Results

Table 1 and Table 2 show that the results for the three methods are very close. The best average precision is achieved by M. A. Table 1 shows that only for 7 cases the application of additional filter (M. B) reduces the precision compared to M. A. The best average recall is performed by M. C. We are going to interpret some of the results and examine the case of the "Mexican Egg Bake" recipe for which the results are equal for the three methods. This can happen, when no matching association rule is found and when no cookware item is filtered out by the stop-list. For the sample of the "Classic Thumbprint Cookies" recipe a wrong association rule is chosen, therefore the precision and the recall is lower for M. A & B as it is for M. A. A very interesting case is the one of "Cranberry Glazed Roast Pork". There we see a drop in precision from M. A to M. B. There we see first a wrong rule is used in M. B which is corrected by M. C, which ends in a higher precision for M. C. The values of the recall might raise the question, how is it possible that a filter step produces a higher recall. Due to the fact that the filter step is preceding the association rule mining step in the pipeline, we can get a different set of association rules. If the rule mining step would precede the filter step, then indeed the results would show a higher precision but a recall which cannot be higher than before the filtering.

	P	R	F_1
M. A	0.5127	0.3034	0.3812
M. B	0.4828	0.3124	0.3793
M. C	0.4892	0.3130	0.3817

Table 2: Summary of the average results for the three methods for precision (P), recall (R) and F_1 -Measure (F_1).

5 Discussion

The results show that the data-flow which we create has room for improvement. This evaluation is more a formative evaluation, we want to identify the promising approaches which are useful for the future work. The results of the previous section indicate that the benefit of the statistical anaphora resolution is low compared to the result which is achieved with the simple approach of M. A. Although we try to build an evaluation approach which reduces the paraphrasing problem for the control-flow, it still remains for data-flow. The measurement approach is pessimistic because it uses only lexical comparison to decide if a product is relevant or not. For example *broth* and *soup* would be rated as a mismatch even if they are semantically very close or sometimes even equal. Therefore in reality the results of M. B and M. C should be better than the measurements of the evaluation. Our intention is to develop an anaphora resolution method which does not need any ontology or other special domain knowledge. The evaluation has shown, that our approach cannot fully distinguish between an anaphora and a cooking tool. Therefore we need a list of cooking tools to differentiate between a cooking tool and an anaphora but we don't need a complex ontology. The evaluation has shown, that the application of a filter for cooking tools produced better results.

6 Related work

We are going to present related work of different research areas. The area of statistical anaphora resolution had been approached by computer linguists. Gasperin and Briscoe [18] showed a statistical approach for anaphora resolution in biomedical texts. They built a system which was based on the naive-Bayes classifier. Markert et al. [19] showed an approach that was based on the number of results of a search engine query. The queries were built with all possible antecedents for that anaphora and the anaphora themselves they were embedded in sample phrases e.g. "*fruits such as apples*". The TellMe [20] system allowed the user to define procedures through utterances in natural language which were interpreted by the system and transformed to formal wf steps. In comparison to our system, the process of the TellMe system was interactive; the user might get feedback from the system and could specify his input.

Friedrich et al. [21] developed a system to extract a formal wf representation of a textual process description. To avoid noise, a manual preprocessing step

was applied. Our approach is capable to process textual content as it is. The work of Dufour-Lussier et al. [7] is very similar to ours. They extracted a tree representation of recipes from text. In contrast to our method, the focus was on food components which were transformed by cooking activities to other food components. The TAAABLE ontology was applied to resolve references in the text by matching sets of food components, e.g. “blueberry” and “raspberry” with “fruits”. They presented in [7] an evaluation for the data-flow. Their system delivered very good result. However the test set was not representative and they were only counting the first occurrence of a product within a recipe.

7 Conclusion and future work

In this paper we presented a stream based framework for wf extraction. During the development of the different methods for anaphora resolution we experimented and tried a lot of ideas. The framework supported this work by its flexibility. A similar effect can be expected during the development of new applications for other domains. The framework allowed analysing and incrementally building wfs. We presented a wf extraction application for the domain of cooking. We presented three different anaphora resolution approaches. Two approaches used association rules which were created during an analysis of the case-base of wfs. The evaluation was performed with wfs which had a semantically correct data-flow created by a human expert. In the future we are going to develop an extraction application for a different domain.

8 Acknowledgements

This work was funded by the German Research Foundation, project number BE 1373/3-1.

References

1. Weber, R.O., Ashley, K.D., Brninghaus, S.: Textual case-based reasoning. *Knowledge Engineering Review* **20**(3) (2005) 255-260
2. Lenz, M., Hbner, A., Kunze, M.: Textual CBR. In Lenz, M., Bartsch-Sprl, B., Burkhard, H.D., Wess, S., eds.: *Case-Based Reasoning Technology From Foundations to Applications*. LNAI 1400, Berlin, Springer (1998) 115–137 The original publication is available at www.springerlink.com.
3. Wiratunga, N., Koychev, I., Massie, S.: Feature selection and generalisation for retrieval of textual cases. In Funk, P., Calero, P.A.G., eds.: *Advances in Case-Based Reasoning*. Number 3155 in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (January 2004) 806–820
4. Lamontagne, L., Lapalme, G.: Textual reuse for email response. In Funk, P., Calero, P.A.G., eds.: *Advances in Case-Based Reasoning*. Number 3155 in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (January 2004) 242–256

5. Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Event extraction for reasoning with text. In Agudo, B.D., Watson, I., eds.: *Case-Based Reasoning Research and Development*. Number 7466 in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (January 2012) 384–398
6. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: *Workshop Proceedings: WWW'12*, Lyon, France (2012)
7. Dufour-Lussier, V., Le Ber, F., Lieber, J., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems*
8. {Workflow Management Coalition}: *Workflow management coalition glossary & terminology*. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf (1999) last access 05-23-2007.
9. Jurafsky, D., Martin, J., Kehler, A., Vander Linden, K., Ward, N.: *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Volume 163. MIT Press (2000)
10. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. (2002) 168175
11. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. (2003) 423430
12. Minor, M., Schmalen, D., Bergmann, R.: XML-based representation of agile workflows. In Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P., eds.: *Multikonferenz Wirtschaftsinformatik 2008*, GITO-Verlag Berlin (2008) 439–440
13. Riloff, E., Phillips, W.: An introduction to the sundance and autoslog systems. Technical report, Technical Report UUCS-04-015, School of Computing, University of Utah (2004)
14. Zhu, H.: *Software Design Methodology: From Principles to Architectural Styles*. Butterworth-Heinemann (March 2005)
15. Tognini-Bonelli, E.: *Corpus Linguistics at Work*. John Benjamins Publishing (2001)
16. Agrawal, R., Srikant, R.: Mining sequential patterns. In: *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. (1995) 314
17. Kowalski, G.: *Information system evaluation*. In: *Information Retrieval Architecture and Algorithms*. Springer US (January 2011) 253–281
18. Gasperin, C., Briscoe, T.: Statistical anaphora resolution in biomedical texts. In: *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1. COLING '08*, Stroudsburg, PA, USA, Association for Computational Linguistics (2008) 257264
19. Markert, K., Modjeska, N., Nissim, M.: Using the web for nominal anaphora resolution. In: *EACL Workshop on the Computational Treatment of Anaphora*. (2003) 3946
20. Gil, Y., Ratnakar, V., Fritz, C.: Tellme: Learning procedures from tutorial instruction. In: *Proceedings of the 15th international conference on Intelligent user interfaces*. (2011) 227236
21. Friedrich, F., Mendling, J., Puhmann, F.: Process model generation from natural language text. In: *Advanced Information Systems Engineering*. (2011) 482496

A Case-based Reasoning Approach to Business Workflow Modelling Based on Formal Temporal Theory

Stelios Kapetanakis¹, Miltos Petridis¹, Jixin Ma² and Brian Knight²

¹ School of Computing, Engineering and Mathematics, University of Brighton, Moulsecoomb Campus, Lewes road, Brighton BN2 4GJ, UK
{s.kapetanakis, m.petridis}@brighton.ac.uk

² School of Computing and Mathematical Sciences, University of Greenwich, Maritime Greenwich Campus, Park Row, Greenwich, London SE10 9LS
{j.ma, b.knight}@gre.ac.uk

Abstract. The intelligent monitoring of the execution of business processes is important in ensuring their effectiveness and smooth operation. The case base which can be used to provide relevant information usually derives from the execution traces of the business processes and workflows. A CBR system can retrieve and reuse useful knowledge that can allow the effective monitoring of business workflows. However, it is difficult to relate such information to the abstract definition and generalisation of the overall business process and workflows. This paper investigates the use of a formal temporal theory that allows a CBR system to relate episodic event trace knowledge to the formal definition of workflows. This allows for better explanation of both the similarity and relevance of retrieved cases providing necessary context to the retrieved knowledge. The paper presents the formal temporal theory and foundation for workflows and shows how these can be used within a case-based system. The graph based case representation and similarity measures are presented and the approach is evaluated using two real world application domains.

Keywords: Case-based Reasoning, Business Workflows, Temporal Reasoning, Graph Similarity, Explanation

1 Introduction

Business processes are a necessary part of the current economical, societal and organisational world by organising and standardising production paths, actor roles and related industrial tasks. Structural functional hierarchies can be represented from a qualitative and quantitative perspective and a process is usually there to sketch how this is conducted in a detailed way.

Due to the volume in data and the numbers of stakeholders, the need for effective management and standardisation in business processes is imperative. As a result a number of standards exist, covering a wide range of representations in terms of the definition, instrumentation and composition of a business process. The Business Process Modelling Notation (BPMN) developed by the Business Process Management Initiative (BPMI) and Object Management Group (OMG) provide a graphical repre-

sentation standard for workflow-based business processes [1]. The Business Process Definition Meta-model (BPDM) [2] defines a number of concepts that can be used to exchange business process modelling representations among vendors. WS-BPEL, proposed by Organization for the Advancement of Structured Information Standards (OASIS), is an execution language describing the “behaviour of business processes in a standards-based environment” [3]. The XML Process Definition Language (XPDL) provided by the Workflow Management Coalition (WfMC) [4] offers a standardised format for business process definitions exchange among vendors. Additionally the development of a set of complementary standards like UML diagrams, BPEL for Web Services, the EPC and YAWL allow the modelling and representation of processes regarding different perspectives [5]. The emerged standards in accordance with Service Oriented Architecture provide a loose but robust connection between services and their “consumers”. This increases their efficiency and interoperability across systems [6].

With the help of the current standards stakeholders can understand the flow of actions and available decision paths. These are executed when the process is followed and can be instantiated in terms of workflows. Such workflows are being executed and can produce data in terms of temporal logs showing the event sequences followed during a particular execution.

While attempting to monitor a workflow, any past events can be traced back through their logs, a process that can be conducted by a human expert. This can be a trivial task in most of the cases since nowadays there are ways to easily follow and understand a structured text log [7]. However, the rapid data generation in fast changing workflows makes the monitoring more difficult due to: complex log content, overlapping relationships among events, hidden or missed important events [8] due to the nature of its environment and the presence of uncertainty [8]. Processes that deal extensively with human resources can be exposed to the above since they include, combine and apply processes from different hierarchical layers within an institution; making hard to follow / monitor the actual process from the derived sequences of events.

The difficulty to clearly understand a workflow execution instructs the challenge for its intelligent monitoring. Such could indicate systems that provide the intelligence to understand the current state and provide the identified insights to human auditors. However, in order to be able to do so, a formal way of capturing and representing the available temporal data should be adopted which could be accorded to the actual business process definition. In such way monitoring could be applied more widely, ensuring the broad elicitation of knowledge from a system and its execution data.

This paper presents an approach to assist Case-based Reasoning (CBR) in the intelligent monitoring of workflows. The general time theory [9] is being used to define a formal representation of a business process while using its temporal data. The motivation to that is that usually CBR resorts to the formal representation of available execution data in order to identify useful past knowledge and further assist in the monitoring of an investigated case. However, in such approach the actual business process knowledge is being overlooked, losing the opportunity to identify a possible solution to the current problem. Therefore, a hybrid solution is being proposed by having a

formal representation of both the business process and the executed workflow traces in a way that they can be used together to enhance the value of the knowledge that can be derived from them.

The structure of this paper follows as: Section 2 provides the CBR background, the state of the art in terms of the workflow monitoring and the general time theory; Section 3 illustrates the proposed approach in terms of the formal temporal event representation as well as its correspondent business process one, Section 4 presents a simple evaluation and finally Section 5 concludes with the summary of this work and the indicated path for future work.

2 Formal Case Representation for Workflow monitoring

In the process of workflow monitoring past knowledge can be a guide to understand the present. In a case investigation an auditor attempts to understand the meaning of events and their temporal information in accordance to the present workflow state, by retrieving knowledge from past executed traces that seem similar.

In order to understand how similar the cases are, their traces should be compared. For the calculation of similarities among business processes, Dijkman et al. [12] have investigated algorithms focused on tasks and the control flow relationships among them. Petri-nets have also been used in the comparison of process models based on behavioural observations in workflows [13].

CBR [10] seems a natural approach to the above since by retrieving, reusing, revising and retaining past case(s) can identify similarities and additionally present the available context for the investigated case. Specifically in terms of workflows that can be represented as a graph and are subject to structural similarity measures; CBR has been proposed as a possible approach to their reuse and adaptation [11]. Similarity measures for structured representations of cases have additionally been applied to real life applications [14] requiring reuse of past available knowledge as applied to structural case bases [15, 16].

CBR has been shown efficient in the intelligent monitoring of real business workflows [5, 8, 17, 18] where knowledge repositories of past cases were used. Similarity measures were applied on those taking into account the nature of event traces, the conducted actions and the existing temporal relationships. These characteristics were represented in terms of graphs and the similarity measures were applied among them.

For the temporal representation the existing formalisms with points [19, 20], intervals [21] or points and intervals [9] were investigated, with the adoption of the latter due to its advantage in terms of the comprehensive representation of the overall domain [22].

The representation of workflow events in terms of the general theory of time proposed by Ma and Knight [9], the temporal relationships are reduced from the ones proposed by Allen & Hayes [23] to one primitive one, the “meets” relationship. With this approach temporal similarity measures can be defined within the context of CBR systems. Any events / intervals can be represented as graphs based on their temporal relationships. Graph matching techniques can be further applied such as the Maxi-

imum Common Sub-graph [5, 8, 17, 18]. Based on this foundation explanation can be extracted from the temporal instances for explanation purposes [24] even when instance event traces may contain partial or incomplete knowledge.

The representation of workflows as graphs based on the general theory of time [9] allows the formulation of a number of similarity measures between workflow executions. The main one is based on the Maximum Common Sub-graph (MCSG) [14, 15, 16, 17] between two graphs representing workflows.

The approach proposed here allows the formulation of similarity measures between cases even where there is uncertainty about the workflow, such as unknown durations of intervals or events not captured by the system.

Research using the CBR-WIMS System [17,18] has shown that this approach can be used to support the efficient monitoring of workflow executions, as well as to provide relevant explanation and insight into the reasoning and the context of the retrieved knowledge.

However, this approach is limited by the fact that it operates predominantly on the domain of episodic executions of workflows and provides only limited opportunities to relate this to the workflow definition. This can be seen as missing the opportunity to provide additional explanation, as well as consistency checking that could reveal any departure from the formally defined process.

3 From events traces to plans: Towards a formalism for workflow representation

As it has been seen from Section 2, events constitute certain snapshots in time within a workflow's execution. Workflow stakeholders regard event traces as a series of events capturing specific information about a workflow execution. An investigated event trace can be regarded as a special case of a plan. A plan contains several pre-defined events that can be a working analogy of a workflow path. By following the path a certain problem is solved. In any attempt to solve a problem a more abstract approach should be adopted since all the events indicate the lowest level of a process, making it hard to understand the overall context. Furthermore, any manual overrides that take place in a system are not usually compatible with the defined process path(s). Therefore the formal mapping of the events allows the first part of the sought abstraction. Examples with application of the General Time theory [5, 8] have shown that such can be accomplished successfully. However, the focus of problem solving eventually is been transferred to the formal representation of the process in terms of its definition rather than a particular execution. This has to be conducted at a standard's level e.g. BPMN. By doing so when following a sequence of events there can be a direct association, not with past knowledge but with the actual process branch which should be executed.

With the parallel routes of the executed workflow trace and workflow definition, a direct mapping is needed from the one to the other. Based on such, explanation can be derived upon its current status regarding the data provision and the existing constraints. Existing work has been focused on the CBR monitoring and explanation from

trace executions. However, the focus should also include relation to already defined business process paths since a lot of contextual valuable information could be left unused.

In theory we could think of an observer looking at workflow executions over a period of time and be able to “reverse engineer” the executed traces to their workflow origins. However, besides all the available paths, solutions to real problems involve overrides and possible “erroneous” reordering of executions that could totally confuse the observer. Additionally, business processes change and unless an observer is aware of changes occurring to the business process, any changes will introduce confusion and in fact make the business process appear with more permitted execution paths than those actually intended. A formal process definition could assist in providing further knowledge which is important to understand and intelligently reason with the execution, providing a context to the overlooked execution which is not always possible to elicit from the existing traces.

The work presented here still focuses on looking at the current execution traces but while doing so, a formal mapping is defined and maintained between the business process as an action plan and the instantiation of events during a particular execution. In doing such, a temporal model for business processes had to be defined and is applied providing a firmer foundation for business process modelling.

The formal foundation for the definition of business workflows was proposed by Petridis et al.[25]. This defines an event trace as a formal mapping from the model to a specific set of events and their temporal relationship as defined in the general theory of time discussed in Section 2.

Following the defined formalism for business process workflows, actions can be mapped to certain types or event instances that refer to their actual performance at a particular time [25]. An action instance (**ai**) comprises an action (**a**) and a time moment (**t**). Formally an **ai** can be written as **ai = (Name (ai), Time (ai))** where:

- **Name** is a function from a set of action instances (AI) to a set of action names **A**.
- **Time** is a function from the set of AI to the set of time moments **M**.

Action instances are distinct. In order to symbolise that an action instance takes place over a time moment **t** the temporal proposition *Performs* can be used from reified temporal logic [21, 25] e.g. *Performs* (ai, t).

Similar to the action and action instances the definitions of events and event instances can be introduced. Event names refer to explicit types of instantaneous activities. The set of available events can be referred as **E** where the individual events can be $e_1, e_2 \dots$ etc. Event instances equivalently to the action ones can be represented as a pair of an event name and a time point (**e, p**) where $e \in E$ and $p \in P$. The set of event instances can be represented as **EI**. Each event instance can be written as **ei = (Name (ei), Time (ei))**, where **Name** is a function from the set of event instances **EI** to the set of event names **E**. Time is a function from the set of event instances **EI** to the set of time points **P**. Temporal proposition Occurs, e.g. Occurs (ei, p) can be used from reified temporal logic [21, 25] to represent that an event instance ei occurs at a time point p.

The definition of a business process can be given equivalently to the definition of events, actions and their relevance to instances. A business process name is a set of a certain type of the business process, e.g. the general university enquiries process. A business instance is a set of action instances and event instances [22]. Based on the approach of Petridis et al. [25] it is possible to define a temporal model TM for a business process (Pro). The process can be defined as the minimal set of temporal facts about action times, i.e. facts which remain true in each business instance of the business process.

Temporal sequences do not necessarily refer to exact time instances since in a lot of applied domains seems almost impossible to be able to record all available events. Therefore their sets are usually given in terms of incomplete or partial knowledge. Durations could be included but this should not be expected as the norm.

An example follows showing the above. Fig. 1 shows a simplified event log of workflow traces. Additionally to that, reference to events known to a workflow stakeholder is included. The above information if represented in terms of events along with their relationships and durations can be shown as a graph (Fig. 2)

1	Backup operation took place before the 30 th of April 2012
2	Action A was applied on Monday 30 th of April 2012
3	Action B was applied after operation A
4	Action C was applied, 3 days after operation B
5	Action D was applied on Saturday 5 th of May 2012
6	Action A was applied on Monday 7 th of May 2012
7	Action E was applied before operation Z
8	Action F was applied after operation A
9	Action G was applied one day after operation F
10	Action H was applied one day after G
11	Action I was applied after H
12	Action J was applied after I and before Z
13	Action K was applied 1.5 days after G
14	Action L was applied after K
15	Action M was applied after L
16	Action N was applied 3.5 days after M and before Z

Fig. 1. Simplified event log

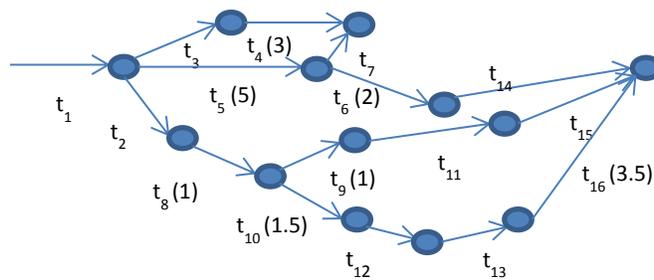


Fig. 2. Event-trace representation following the general theory of time

This approach allows a CBR business workflow monitoring system such as CBR-WIMS [18] to maintain a link between episodic event traces and business workflow definitions to enhance the ability of the system to provide explanation and context to workflow managers.

4. Deployment and Evaluation

In order to demonstrate and evaluate the approach, the CBR-WIMS system was extended to maintain both business workflow definitions as well as the case bases of workflow executions. The system has been developed to maintain the mapping between the two, as well as providing a business process version control system to allow for changes to the business process and corresponding workflow definitions. The system allows users to visualise specific trace event sequences by highlighting the corresponding branch in a BPMN diagram depicting the business workflow definition. The approach proposed here was evaluated through the use of the CBR-WIMS on two real world case studies:

1. The Exam Moderation System (EMS) that coordinates the business workflows among a number of actors for the moderation of exam papers as part of a formal Quality Assurance process within a university.

2. The Box Tracking System (BTS) that monitors the maintenance and movements of boxes containing business artefacts between a warehouse and customer premises using a fleet of delivery vans.

Both of the case studies above have been used to evaluate previous work on business workflows using CBR-WIMS [5, 17]. The evaluation in this instance consisted of repeating the previous experiments [5, 17] and evaluating the additional effect of the approach proposed in this paper and the corresponding new capabilities implemented in CBR-WIMS. The focus of the evaluation was on the accuracy of the achieved monitoring and explanation capabilities of the system.

In all cases, the performance of the system was compared against that of expert business process managers who were asked to identify problems in workflow execution and evaluate the explanation provided by CBR-WIMS.

The key findings of the evaluation can be summarised as:

1. Although the accuracy of the CBR retrieval did not change (this was based on retrieval of previously observed similar sequences of events), the system now picked up a number of anomalies that were related to “shortcuts” or “manual overrides” by operators. In a way, the system now became a hybrid one picking up any non-adherence to the workflow definitions.
2. There was a reduction in false positives based on the version control of business processes. Especially in the EMS system there were a few changes to the business workflows over the years that made some sequences of events appear as anomalies to the CBR system. These were now clearly identified as the version control system showed the different provenance of cases.

3. Constraining the retrieval of cases to cases of the same provenance within the business process, increased the monitoring accuracy. This in fact was shown in previous sets of experiments [5] but there the attribution of series of events to specific stages in the business process were handcrafted requiring considerable extra manual intervention that would have been too onerous for larger test cases.
4. The explanation capabilities of the system were enhanced. This was done both in terms of the ability to explain the relevance of the similarity retrieval process and in terms of the ability to explain the nature of problem and proposed solution to the CBR process.
5. In all cases, the connection of the event traces to the formal business workflow definitions provide the provenance and context of specific sequences of events and reduce the scope for uncertainty in the business workflow monitoring process.

The table below (table 1) summarises the results of the simple evaluation of the explanation capabilities of WIMS-CBR on the EMS case study. The experts replied to the questions using a scale of 1 (disagree) to 5 (strongly agree). The results were averaged over the 20 target cases.

Table 1. Explanation improvement using business process mapping for event traces

	WIMS-CBR no explanation	WIMS-CBR with event trace visualisation	WIMS-CBR with event trace visualisation and business process mapping visualisation
Correct classification is clear	3.2	4.2	4.5
Similarity is obvious to the 3NN	2.8	3.9	4.2
Advice clarity	3.3	4.5	4.8

The table shows that besides WIMS-CBR's ability to visualise sequences of workflow execution events, it can relate them to the business process formal graphical definition and further enhance the explanation provision.

4 Conclusions

This paper presents a formal way to depict the representation of business process definitions, execution(s) and maintain their relationship within the scope of CBR. The adopted mathematical formalism gives a different perspective to the existing approaches since it alleviates the uncertainty of interpreting the same workflow execution in many different ways. With the existing approach a business process execution has a unique mapping to its defined model.

The current enhancement in CBR monitoring of business workflows is a significant step forwards, in unifying the space of business process definition with the episodic approach of case base reuse. With the adopted approach, knowledge and explanation can be extracted from the formal representation of the executed events, their

formal definition of the business workflows and the formal mapping between them. It also allows for changes in business workflows to occur in a way that is now transparent to the CBR system.

In many ways, the enhancement proposed in this paper provides context and provenance to cases in the workflow case base. Additionally, this enhancement allows the development of hybrid CBR systems that can add other forms of reasoning to the episodic CBR system as now reasoning can occur both at the episodic and the formal process definition domain spaces.

Future work, as indicated from the current findings, could possibly use existing workflows and their traces to generate new ones that should present similar functionality. An example to that could be an *approval* process. Based on its existing traces, knowledge can be extracted and used potentially to shape: *a mortgage approval, a production part approval* or generally *a transaction approval*. This will allow a CBR system to extract useful episodic knowledge that can be used between similar business processes. This can also help with the “cold start” problem that can occur when CBR systems do not have enough useful past cases to extract relevant information from the case base.

Equivalently by identifying the *families* of similar workflows we could identify the re-occurrence of problems and be able to re-use the existing knowledge to deal with them or other explanation purposes. The identification of similar event traces among workflows could even lead, in a hypothetical scenario, to the change of its underlying business process; since there could be a case that the knowledge from a similar system could improve the one under investigation. In this case a new potential could be highlighted, where the identification of working workflow patterns could lead to the *dynamic adaptation* and *learning* of an existing workflow.

Further work will involve reasoning associated with the instances of workflows as well as their principal models.

5. References

1. Business Process Management Initiative (BPMI): BPMN 2.0: OMG Specification, January, 2011, <http://www.omg.org/spec/BPMN/2.0/>, accessed May 2012. P. Atkin. Performance maximisation. INMOS Technical Note 17.
2. Business Process Definition MetaModel (BPDM): Business Process Definition MetaModel (BPDM), Version 1.0, 2008, <http://www.omg.org/spec/BPDM/1.0/>, accessed January 2013. IBM: Business process standards, Part 1: An introduction, 2007, http://www.ibm.com/developerworks/web-sphere/library/techarticles/0710_fasbinder/0710_fasbinder.html, accessed May 2012.
3. Workflow Management Coalition (WfMC): XPD L 2.1 Complete Specification (Updated Oct 10, 2008), <http://www.wfmc.org/xpdl.html>, accessed April 2009.
4. Kapetanakis, S., Petridis, M.: An Evaluation of the CBR-WIMS Architecture for the Intelligent Monitoring of Business Workflows using Case-Based Reasoning. In: Lamontagne, L., Recio-Garcia, J.A. (eds.) Proceedings of PO-CBR: Process-oriented Case-Based Reasoning, ICCBR 2012, pp.45-54. (2012)
5. Hill, J. B., Sinur, J., Flint, D., Melenovsky, M. J.: Gartner’s Position on Business Process Management. Gartner, Inc. (2006)
6. Michaelis, J. R., Ding, L., McGuinness, D. L. (2009). Towards the Explanation of Workflows. In: Proceedings of the IJCAI 2009 Workshop on Explanation Aware Computing (ExaCt). Pasadena, CA, US.

7. Kapetanakis, S., Petridis, Ma, J., Bacon, L. (2010). Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning. In Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.): Proceedings of the Fifth International workshop on Explanation-aware Computing ExaCt (ECAI 2010). Lisbon, Portugal.
8. Ma, J., Knight, B.: A General Temporal Theory, the Computer Journal, 37(2), 114-123 (1994).
9. Aamodt A., Plaza E.: Case-based reasoning; Foundational issues, methodological variations, and system approaches. AI Communications, vol. 7, no. 1, pp. 39-59 (1994)
10. Minor, M., Tartakovski, A. and Bergmann, R.: Representation and Structure-Based Similarity Assessment for Agile Workflows, in Weber, R., O. and Richter, M., M. (eds.) CBR Research and Development, Proceedings of the 7th international conference on Case-Based Reasoning, ICCBR 2007, Belfast, NI, UK, August 2007, LNAI 4626, pp 224-238, Springer-Verlag (2007)
11. Dijkman, R. M., Dumas, M., Garcia-Banuelos, L.: Graph matching algorithms for business process model similarity search. In Dayal, U., Eder, J. (eds.), Proc. of the 7th Int. conference on business process management. (LNCS, Vol. 5701, pp. 48-63). Berlin: Springer (2009)
12. van der Aalst, W., Alves de Medeiros, A. K., Weijters, A.: Process Equivalence: Comparing two Process Models Based on Observed Behavior, In Proc. Of BPM 2006, vol 4102 of LNCS, pp 129-144, Springer, (2006)
13. Bunke, H., Messmer, B.T.: Similarity Measures for Structured Representations. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) Topics in Case-Based Reasoning. LNCS, vol. 837, pp. 106-118, Springer, Heidelberg (1994)
14. Mileman, T., Knight, B., Petridis, M., Cowell, D., Ewer, J.: Case-Based Retrieval of 3-D shapes for the design of metal castings in Journal of Intelligent Manufacturing, Kluwer. 13(1): 39-45 (2002)
15. Wolf, M., Petridis, M.: Measuring Similarity of Software Designs using Graph Matching for CBR, In workshop proceedings of AISEW 2008 at ECAI 2008, Patras, Greece (2008)
16. Kapetanakis, S., Petridis, M., Knight, B., Ma, J., Bacon, L. : A Case Based Reasoning Approach for the Monitoring of Business Workflows, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, LNAI (2010)
17. Kapetanakis, S., Petridis, M., Ma, J., Knight, B.: CBR-WIMS, an Intelligent Monitoring Platform for Business Processes. In Petridis, M. (ed.): Proceedings of the 15th UK CBR workshop, pp. 55-63. Cambridge: CMS press (2010)
18. Shoham, Y.: Temporal logics in AI: Semantical and Ontological Considerations, Artificial Intelligence, 33: 89-104 (1987)
19. Shoham, Y.: Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence, MIT Press (1988)
20. Allen, J.: Maintaining Knowledge about Temporal Intervals, Communication of ACM, 26: 832-843 (1983)
21. Kapetanakis, S.: Intelligent Monitoring of Business Processes using Case-based Reasoning, PhD Thesis, University of Greenwich (2012)
22. Allen, J., Hayes, P.: Moments and Points in an Interval-based Temporal-based Logic, Computational Intelligence, 5, 225-238 (1989).
23. Ma, J., Knight, B., Petridis, M.: Deriving Explanations from Partial Temporal Information, in workshop proceedings of ExACT-08 at ECAI 2008, Patras, Greece (2008)
24. Petridis, M., Ma, J., Knight, B.: Temporal model for business process. Intelligent Decision Technologies. 5 (4), pp. 321 – 332 (2011)

An approach for collecting fine-grained use traces in any application without modifying it

Blandine Ginon^{1,2}, Pierre-Antoine Champin^{1,3}
and Stéphanie Jean-Daubias^{1,3}

¹ Université de Lyon, CNRS,

² INSA-Lyon, LIRIS, UMR5205, F-69621, France

³ Université Lyon 1, LIRIS, UMR5205, F-69622, France
{name}. {surname}@liris.cnrs.fr

Abstract. We propose a technique to collect use traces in any existing application, without a need to modify this application. This technique is based on the use of accessibility libraries. We implemented our technique in a collector that uses UIAutomation and JavaAccessibility libraries: it can monitor Windows target-applications to collect the user's traces. The traces are then stored in a trace-base management system in order to be exploited thereafter. We have tested our collector on more than fifty applications in order to evaluate our approach.

Keywords: Use traces, event detection, accessibility.

1 Introduction

There is a growing interest in collecting traces of the interaction of a user with computer applications, for various purposes, like trace analysis [6], trace visualization [2] and trace-based assistance [20]. However, most applications are not designed to collect traces and it would be costly to redevelop them when a need to do so appears. Furthermore, the people that need to collect traces in an application do not always have its source code, if even they wished to modify it.

In the context of the AGATE project (Approach for Genericity in Assistance To complex task) that aims at facilitating the use of complex software, without constraints on this software, we exploit the user's traces to provide personalized assistance. For this reason, we propose a technique to collect use traces in any application, without a need to modify it. This technique is based on the use of accessibility libraries that make possible the subscription to different kinds of events, like the mouse entered on an image or the selection of an item in a combo box, in order to know when those events occur, but also to know on which component of the user-interface they occurred. We implemented this technique in a collector that uses two accessibility libraries that target Windows applications: UIAutomation and JavaAccessibility.

After presenting related work in section 2, we present our approach in section 3, and we describe in section 4 how we implemented it. Section 5 discusses a prelimi-

nary evaluation of our implementation. Finally, we conclude and propose future improvements.

2 Related work

There is an abundant corpus of work on the analysis of logs and traces [5], [12-14], [21]. Historically, log analytics has first been dedicated to focus on the behavior of programs, for debugging or monitoring purposes. Then, the potential of using it to analyze the user's activity has been gradually recognized. Data mining and machine learning techniques have therefore been used for discovering processes in computer-mediated activities [3], [18], and more recently on identifying communities in social networks based on the user's interaction patterns [17].

Statistical and/or synthetic analysis is not the only way to exploit activity traces. Activity traces can also be considered as a repository of individual experiences that can be reused in a similar context, either identically or after an adaptation [4]. This is the underlying assumption of a number of efforts, such as those aiming at providing recommendations to users based on past experiences [8], [11], or monitoring the progression of a student in e-learning applications [16]. With the increasing availability of mobile devices and wearable sensors, practices of tracing various aspects of one's day-to-day life are also developing. Known as lifelogging [15] or quantified self [19], those practices aim at a better self-awareness or recollecting past events.

Depending on their intended tasks, the different approaches cited above require different kinds of events recorded in the respective traces. Except for lifelogging application (which are focused on real-world information acquired via sensors), most approaches rely on relatively high-level events (i.e. run application, open file). It follows that available tools for collecting interaction traces¹ are limited to capturing those high-level events. We believe, however, that some applications, such as personalized user assistance, require more fine-grained traces.

3 A technique to collect use traces

We propose a technique to collect fine-grained use traces in any existing application, that we will call a target-application. It has been stated in the previous section that this is already possible for high-level events. There are also tools to collect individual clicks or keystroke², but the only contextual information they provide is the application in which those events happened. By contrast, we want to be able to associate each traced event with a component of the user-interface of the target-application. For example, knowing on which button or menu item the user clicked is much more informative about his/her activity than only recording the application in which he/she clicked.

¹ For example <http://dev.nepomuk.semanticdesktop.org/> or <http://intersectalliance.com/snareagents>

² For example <https://github.com/gurgeh/selfspy> or <http://www.mykeylogger.com/>

In this work, the traces we consider are sequences of records describing events (event type, time stamp, and other attributes)³. For this purpose, our technique is based on the use of accessibility libraries. Those libraries were initially created to allow accessibility tools (such as screen readers or braille terminals) to get information about the applications, in order to make them more accessible to disabled people. Using these libraries, it is possible to subscribe to different kinds of events, in order to know when those events occur, but also to know on which component of the user-interface they occurred. Indeed, accessibility libraries provide access to the full hierarchy of GUI components available to the user, as illustrated by Fig. 1: the root element represents the screen and its children represent all the open application windows: the calculator, Regards [7], Google Chrome and Paint in the example. What's more, we can see that the desktop (Program Manager) contains four elements: the trash icon, a pdf file, the NetBeans icon and the jar file Regards.jar.

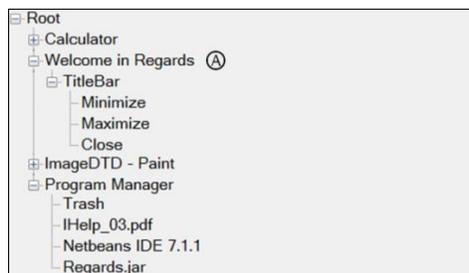


Fig. 1. Component tree detected by accessibility libraries

Thanks to accessibility libraries, we can also access information concerning each component of the user-interface: its type (window, button, check box...), its text, its position, its possible accessibility description... Note that, in the general case, a component has no persistent identifier, yet, when we record an event associated with a component of the user-interface, we need to be able to identify that component in the future uses of the trace. A solution is to characterize a component by its hierarchical position in the component tree and by additional information.

For this purpose, we consider the component hierarchy as an XML tree. Each XML element represents a component with its type, text, and accessibility description (if any). Each component is then characterized by an Xpath [1] containing all the relevant information about the component and its parents. As a simple example, Fig. 2 shows the XML tree describing the user interface of the Windows calculator (Fig. 3). Of all the available information about each component, we only keep their type, text and accessibility description (if any). For example, we can see that not all buttons have a description, but that the button “CE” has one: “Clear entry” (cf. (A) Fig. 2 and (A) Fig. 3). This button can be characterized by the following Xpath:

```

//window[@type="CalcFrame" and @text="Calculator"]/
component[@type="Button" and @text="CE" and @description="Clear entry"]
  
```

³ How these sequences are delimited (per session, per day, per application...) is out of scope, as our approach is neutral to that.

```

<Interface>
  <window type="CalcFrame" text="Calculator">
    <component type="TitleBar" text="Calculator">
      <component type="Item" text="Minimize"/>
      <component type="Item" text="Maximize"/>
      <component type="Item" text="Close"/>
    </component>
    <component type="MenuBar" text="Application">
      <component type="Item" text="View"/>
      <component type="Item" text="Edit"/>
      <component type="Item" text="Help"/>
    </component>
    <component type="Button" text="MC" description="Memory clear"/>
    <component type="Button" text="MR" description="Memory restore"/>
    <component type="Button" text="MS" description="Memory store"/>
    <component type="Button" text="M+" description="Memory add"/>
    <component type="Button" text="M-" description="Memory removal"/>
    <component type="Static" text="12*3+"/>
    <component type="Static" text="4"/>
    <component type="Button" text="" description="Return"/>
    <component type="Button" text="C" description="Clear"/>
    (A) <component type="Button" text="CE" description="Clear entry"/>
    <component type="Button" text="," description="Decimal"/>
    (B) <component type="Button" text="" description="Negation"/>
    <component type="Button" text="/" description="Divide"/>
    <component type="Button" text="*" description="Multiply"/>
    <component type="Button" text="-" description="Substract"/>
    <component type="Button" text="+" description="Add"/>
    <component type="Button" text="" description="Square root"/>
    <component type="Button" text="" description="Percentage"/>
    <component type="Button" text="" description="Reciprocal"/>
    <component type="Button" text="" description="Equal"/>
    <component type="Button" text="7"/>
    <component type="Button" text="4"/>
    <component type="Button" text="1"/>
    <component type="Button" text="0"/>
    <component type="Button" text="8"/>
    <component type="Button" text="5"/>
    <component type="Button" text="2"/>
    <component type="Button" text="9"/>
    <component type="Button" text="6"/>
    <component type="Button" text="3"/>
  </window>
</Interface>

```

Fig. 2. Description of the Windows calculator user interface.

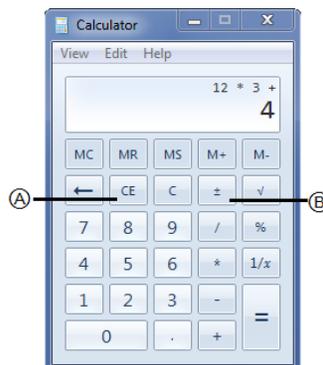


Fig. 3. Screenshot of the Windows calculator.

This description is of course redundant, but redundancy is important because each individual piece of information can be ambiguous in some contexts, as will be illustrated in Section 5. Note that the XML tree does not even need to be stored; it is just a way to formalize the information that is dynamically provided by accessibility libraries, and to justify the use of Xpath to address individual components.

Whenever an event is detected, it will be recorded in the trace with its type, the current time, the user's name and the Xpath characterizing the component on which the event appeared. Depending on the type of event, additional information can also be recorded (see Section 4). Let's come back to the example of the calculator. If the user enters the formula "12*3+4", the collector will detect a series of mouseClicked events on following buttons: "1", "2", "*", "3", "+" and "4". Other kinds of events may also be detected in the meantime (for example, mouseEntered events on the buttons hovered by the pointer during the moves between clicks).

4 Implementation of our technique

We implemented this technique in an operational collector that makes possible the collection of use trace in any existing Windows application, without a need to modify it. Our collector is based on two complementary accessibility libraries: the first one, UIAutomation [10], is aimed at Windows native applications, and the second one, JavaAccessibility [9], is aimed at Java applications.

4.1 Complementarity of accessibility libraries

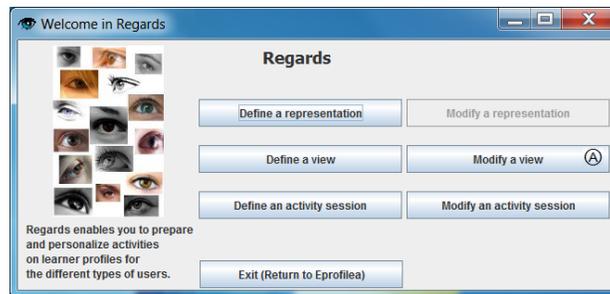


Fig. 4. Welcome screen of the Java application Regards.

UIAutomation detects any application running in Windows. However, for the case of Java applications, UIAutomation is only able to detect the frame of the application. Indeed, the inner components of Java applications are managed by the JVM (Java Virtual Machine) and not by Windows. As an example, we can see that only the frame with the title bar of the Java application Regards is detected (cf. [Ⓐ] Fig. 1). For this reason, we need a second collector for Java applications. We implemented it based on JavaAccessibleBridge and on the JavaAccessibility library, which is the equivalent of UIAutomation for the JVM. JavaAccessibility detects only applications running in the JVM, but contrarily to UIAutomation, it can detect their complete component tree. As an example, an extract of the interface description of the Java application named "Re-

gards” is given in Fig. 5. We can see that the welcome screen of Regards contains a button with a label “Modify a view” (cf. ④ Fig. 5 and ④ Fig. 4). The creator of Regards didn’t associate any accessibility description to this button.

```

<Interface>
  ▼<window text="Welcome in Regards" type="class regard.Main" >
    ▼<component type="JRootPane" >
      <component type="JPanel" />
      ▼<component type="JLayeredPane">
        ▼<component type="JPanel" >
          <component text="Regards" type="JLabel" />
          <component text="Define a representation" type="JButton"/>
          <component text="Modify a representation" type="JButton"/>
          <component text="Define a view" type="JButton" />
          ④ <component text="Modify a view" type="JButton" />
          <component text="Modify an activity session" type="JButton" />
          <component text="Define an activity session" type="JButton" />
          <component text="Regards enables you to prepare and personalize activities
            on learner profiles for the different types of users."
            type="JLabel" />
          <component text="Exit (Return to Eprofiléa)" type="JButton"/>
        </component>
      </component>
    </component>
  </window>
</Interface>

```

Fig. 5. Extract of the interface description file for the Java application Regards.

4.2 Storing and managing traces

Fig. 6 shows the main events that our collector can detect using UIAutomation and JavaAccessibility. For instance, our collector can detect when the end user of the target-application clicks on a button (*mouseClicked*), when he/she moves the mouse pointer over an image (*mouseEntered*), when he/she selects an element in a combo box (*elementSelected*), and when he/she deselects a check box (*propertyChanged*). For some of these events, our collector detects complementary information. For instance, for the event *tooltipOpened*, our collector detects the text of the *tooltip*, and for the event *propertyChanged*, our collector detects the name of the property that changed (like *enabled*, *size*, *itemCount*, *rowCount*, *selected*, *visible*...) and the previous and new value of this property. Our collector can also detect additional information about a component depending on its type (Is the component enabled, selected, checked, collapsed, editable? Has it got the focus? What are its position and dimension, its value, its font and background color? ...).

The traces gathered by our collector are stored in a system called kTBS⁴. kTBS is an open-source implementation of a Trace-Based Management System (TBMS) [4] [20] developed in our team. It is a RESTful service, accessible through the HTTP protocol. Our collector sends the events to record to kTBS through an HTTP-POST request. The resulting traces can then be retrieved through an HTTP-GET request. This makes our collector relatively independent of kTBS; it can store traces in any other TBMS as long as they comply with the same protocol.

⁴ <http://iris.cnrs.fr/sbt-dev/ktbs/>

It is worth mentioning that TBMS are not only meant to store traces: they are also able to compute transformed traces that provide different points of view on the traced activity, at different levels of abstraction. This is why our collector is only focused on low-level events; it relies on the TBMS to provide higher-level traces if needed.

	Events	UIAutomation	JavaAccessibility
About action	Performed	✓	✓
About mouse	Clicked	✓	✓
	Entered	✓	✓
	Moved	✓	✓
	Pressed	✓	✓
	Released	✓	✓
	Dragged	✓	✓
	Exited	✓	✓
About key	Typed	✓	✓
	Released	✓	✓
	Pressed	✓	✓
About focus	Gained	✓	✓
	Lost	✓	✓
	Changed	✓	✓
About menu	Selected	✓	✓
	Deselected	✓	✓
	Opened	x	✓
	Closed	x	✓
About tooltip	Opened	✓	x
	Closed	✓	x
About window	Opened	✓	✓
	Closed	✓	✓
About text	Changed	✓	x
	SelectionChanged	✓	x
About element	Selected	✓	✓
	AddedToSelection	✓	x
	RemovedFromSelection	✓	x
About property	Changed	✓	x

Fig. 6. Main events detected with UIAutomation and JavaAccessibility.

5 Evaluation and Discussion

In order to test our approach, we have successfully used our collector in more than fifty varied applications⁵ (Windows native and Java, created by different developers). The aim of this test was to confirm if our collector can really collect events (see Fig. 6) in these applications; and to verify if the component concerned by each event can be identified in the component tree, with a view to make possible a rich exploitation of the collected traces. During this evaluation, we encountered problems in reliably identifying components. First, notice that if several components of the same type are at the same level in the hierarchy (a very frequent situation), only their text and de-

⁵ List of those applications available at <http://liris.cnrs.fr/blandine.ginon/detection.html>

scription allow us to distinguish them. Note also that some components have no text associated to them, but only an image. In that case, only the accessibility description makes it possible to distinguish the component. This is consistent with the initial purpose of accessibility descriptions: to provide a text for accessibility tools (e.g. screen readers) when the component has no text of its own, or when the text is not descriptive enough.

In the example of the Windows calculator (cf. Fig. 2), several buttons have no label but only an icon, like the buttons “negation” and “square root”. As a consequence, in our description file, we can distinguish these buttons only thanks to the description associated with the buttons. The Windows calculator has been developed with accessibility in mind, but unfortunately, this is not the case for all applications. Indeed, making an application accessible is time-consuming, and many developers prefer spending that time at adding new features to the application.

Accessibility descriptions themselves are not enough if they are not carefully chosen. Consider the case of Regards; the button  Fig. 5 has no label, but only an icon representing an eraser and its description is “erase”, like all the other buttons with an eraser in that window. As a consequence, it is not possible with our technique to distinguish those buttons from each other. Indeed, they will have exactly the same Xpath. This problem would not exist if the creator of the application had provided a more specific description for each button; “erase the view for the activity 'Visualize his profile' ” for instance.

One way to overcome the lack of (good enough) accessibility descriptions is to use additional information to characterize components, especially their position in the window (which is also made available by accessibility libraries). We decided not to resort on that solution for the following reasons. First, the position of components within the window, and relatively to each other, may vary depending on a number of parameters (display settings, font size, window size). Second, setting good accessibility description is what developers should do anyway. So we prefer to encourage good practices, rather than compensate the absence of reliable information (accessibility description) by an alternative that may prove just as unreliable (position).

Another problem we encountered is that some applications are still not well detected by either the techniques of our collector. Those applications seem to manage their components without relying on the Microsoft foundation classes, and so are not correctly detected by UIAutomation. Most of them appear to be using the GTK toolkit⁶, so a solution would be to add a part of our collector dedicated to GTK (as we did for Java).

Finally, another limitation of our approach is that it is based on localized information: texts and descriptions of components vary depending on the language of the interface. It is however technically quite simple to “translate” Xpaths in a language to another language, using a translation table as the one used internally by the application. For open-source software, this is even simpler as those translation tables are usually available in a standard format⁷.

⁶ <http://www.gtk.org/>

⁷ <http://www.gnu.org/software/gettext/manual/gettext.html#PO-Files>

6 Conclusion and perspectives

There is an increasing number of applications that make possible interesting exploitations of use traces. For this reason, it is interesting to collect traces and to store them with a view to exploit them thereafter. We propose a technique to collect fine-grained use traces in existing applications, without a need to modify these applications.

The strengths of our technique are the accuracy of the use traces that it can collect, and its genericity. The more accurate the use traces, the more rich and varied will be the possible exploitation of these traces. Contrarily to existing techniques that collect clicks or keystrokes with very little contextual information, our technique associates each traced event with the concerned component from the user-interface of the target-application. What's more, our technique is not specific to an application but it can collect use traces in any application without a need to redevelop this application, even if they are not designed to collect use traces. On the other hand, as our technique is based on the use of accessibility libraries, it is impeded by the lack of accessibility features of some applications. Indeed, if a developer doesn't make any effort to make her application accessible, useful information can be missing, like the description of images.

In order to demonstrate the feasibility of our technique, we have implemented it in a collector that uses UIAutomation and JavaAccessibility. This collector can monitor Windows native applications and Java applications. We have tested this collector on more than fifty various applications from the simplest, like the Windows calculator, to the most complex, like the IDE NetBeans. These tests have showed the overall efficiency of our technique to collect fine-grained use traces in very varied applications.

We are currently working at the implementation of our technique with other accessibility libraries in order to make possible the collection of use traces in GTK Windows applications, as well as in Linux and Mac OS.

Acknowledgments. The authors would like to thank warmly Amélie Cordier for her participation to this work, in particular for her remarks and advices.

References

1. Berglund, A. *et al.* XML Path Language (XPath) 2.0 (Second Edition). W3C Recommendation. <http://www.w3.org/TR/xpath20/>. (2010).
2. Clauzel, D., Sehaba, K. and Prié, Y. Modelling and visualising traces for reflexivity in synchronous collaborative systems. In International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), Barcelona, Spain. pp. 16-23. IEEE Computer Society Los Alamitos, CA, USA. ISBN 978-0-7695-3858-7. (2009)
3. Cook, J. E., Wolf, A. L. :. Discovering Models of Software Processes from Event-Based Data. In: ACM Transactions on Software Engineering and Methodology, 7(3), 215–249. (1998)
4. Cordier, A., Lefevre, M., Champin, P-A., Georgeon, O., Mille, A. Trace-Based Reasoning: Modeling interaction traces for reasoning on experiences. In: 26th International FLAIRS Conference, St. Pete Beach, Florida, USA. (2013)

5. Dwyer, M. B., Avrunin, G. S., & Corbett, J. C. Patterns in property specifications for finite-state verification. In: Software Engineering, 1999. Proceedings of the 1999 International Conference on (pp. 411-420). IEEE. (1999)
6. Fuchs, B. and Belin, A. Trace-Based Approach for Managing Users Experience. In Workshop TRUE: "Traces for Reusing Users' Experience". In: ICCBR 2012 TRUE and Story Cases Workshop, Luc Lamontagne, Juan A. Recio-Garc ed. Lyon, France. pp. 173-182. (2012)
7. Ginon, B., Jean-Daubias, S.: Models and tools to personalize activities on learners profiles. In: Ed-Media, Lisbon, Portugal (2011)
8. Groza, T.; Handschuh, S., Möller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Zajayeri, M., Reif, G., Gudjonsdottir, R. The NEPOMUK Project -- On the way to the Social Semantic Desktop. In: Proceedings of the Third International Conference on Semantic Technologies (I-SEMANTICS 2007), Graz, Austria. (2007)
9. Harper, S., Khan, G., Stevens, R.: Design Checks for Java Accessibility. In: Accessible Design in the Digital World, Dundee, Scotland (2005)
10. Haverty, R.: New accessibility model for Microsoft Windows and cross platform development. In: ACM SIGACCESS Accessibility and Computing, pp 11-17. (2005)
11. Hug, C., Deneckere, R., Salinesi, C.: Map-TBS: Map process enactment traces and analysis. In: International Conference on Research Challenges in Information Science (RCIS), Valencia : Espagne (2012)
12. LeBlanc, T. J., Mellor-Crummey, J. M., & Fowler, R. J. Analyzing parallel program executions using multiple views. *Journal of Parallel and Distributed Computing*, 9(2), 203-217. (1990)
13. Lee, W., Stolfo, S. J., & Mok, K. W. A data mining framework for building intrusion detection models. In Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on (pp. 120-132). IEEE. (1999)
14. Mansouri-Samani, M., Sloman, M.: GEM: a generalized event monitoring language for distributed systems. In: Distruted Systel Engineering, 4(2). (1997)
15. O'Hara, K., Tuffield, M. M., & Shadbolt, N. Lifelogging: Privacy and empowerment with memories for life. *Identity in the Information Society*, 1(1), 155-172. (2008)
16. Poltrack, J., Hruska, N., Johnson, A., & Haag, J. The Next Generation of SCORM: Innovation for the Global Force. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) (Vol. 2012, No. 1). National Training Systems Association. (2012)
17. Sachan, M., Contractor, D., Faruquie, T. a., Subramaniam, L. V.: Using content and interactions for discovering communities in social networks. In: Proceedings of the 21st international conference on World Wide Web - WWW '12, pp 331-341. (2012)
18. Song, M., Günther, C. W., Van der Aalst, W.: Trace Clustering in Process Mining. In M. Van der Aalst, M. *et al.* (Eds.), *Business Process Management Workshop* (pp. 109–120). Springer Berlin Heidelberg. (2009)
19. Wolf, G., Carmichael, A., & Kelly, K. The quantified self. TED http://www.ted.com/talks/gary_wolf_the_quantified_self.html . (2010)
20. Zarka, R., Champin, P-A. , Cordier, A., Egyed-Zsigmond, E. , Lamontagne, L., Mille., A. TStore: A Web-Based System for Managing, Transforming and Reusing Traces. In: ICCBR 2012 TRUE and Story Cases Workshop, Luc Lamontagne, Juan A. Recio-Garc ed. Lyon, France. pp. 173-182. (2012)
21. Zhang, Y., and Wenke L. Intrusion detection in wireless ad-hoc networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 275-283. ACM. (2000)

Building a Trace-Based System for Real-Time Strategy Game Traces

Stefan Wender¹, Amélie Cordier², and Ian Watson¹

¹ The University of Auckland, Auckland, New Zealand

² Université Lyon 1, LIRIS, UMR5205, F-69622, France

s.wender@cs.auckland.ac.nz, amelie.cordier@liris.cnrs.fr, ian@cs.auckland.ac.nz

Abstract. We describe the conception of a visualization and transformation tool for traces of the real-time strategy (RTS) computer game StarCraft. The development of our tool StarTrace is driven by the domain the traces originate from as well as the observable elements those traces contain. We elaborate on those influences, which also include both the structure of the existing game traces and the requirement to use these traces to improve the performance of a machine learning (ML) agent that attempts to learn to play parts of the game. We then describe the architecture of the browser-based tool and the trace model behind it. The purpose of StarTrace is to eventually improve the learning process of this agent by providing the means to harness the enormous amount of data included in complex RTS games. Finally, an example application showcases how the tool can help to better understand the player behavior stored in game traces.

1 Introduction

RTS games, such as StarCraft, provide a challenging test bed for AI research. They offer a polished environment that includes numerous properties such as: incomplete information, spatial and temporal reasoning as well as learning and opponent modeling that are interesting for AI research [1]. For this reason we chose StarCraft as a testbed for a machine learning (ML) approach that tries to learn how to manage combat units on a tactical level (“micromanagement”)[2]. The tool described in this paper is part of this effort to create an autonomous agent that uses ML techniques such as reinforcement learning (RL), CBR and Trace-based reasoning (TBR) to solve the complex problem of micromanaging units in StarCraft. StarTrace is conceived to facilitate the use of StarCraft traces by making the information contained in those traces more accessible and understandable.

The quality and complexity of StarCraft has made it very popular, which in turn has led to a vast body of StarCraft traces, so-called “replays”, that are readily available online. These replays contain implicit expert knowledge in the form of recorded actions of the players, basically traces of their gameplay. TBR [3] is a paradigm that helps us to make this knowledge explicit and reusable by agents.

There are several incentives for applying TBR to the problem of managing combat units in StarCraft. The large amount of expert knowledge that is available online comes in the form of game replays which have an inherently trace-like structure. This expert knowledge can be extracted from the replays and, with the right tools and transformations, used to improve the control of combat units by the agent.

Furthermore traces can improve the general learning process of an agent trying to learn how to perform tasks inside the game. Currently our agent uses a hybrid ML approach that involves case-based reasoning (CBR) to manage game information. The case representation is based on the environment state and unit attributes. The use of traces of unit attributes instead of attributes from a single point in time can add valuable information and thus improve the learning process in a number of ways [4].

In this paper we describe the conception and development of our tool StarTrace, a browser-based application to visualize and transform traces of RTS game data. The ability to create trace transformations is one of the main features of the tool. Its graphical interface allows easy specification and modification trace transformation criteria. The data that is displayed and modified by StarTrace is either obtained from previously recorded games or by actively monitoring user interactions during gameplay.

2 StarTrace

2.1 Hybrid CBR/TBR and Reinforcement Learning for Micromanagement in RTS Games

The aim in an RTS game such as StarCraft is to manage an economy by collecting resources and buildings, to create combat units and to eventually eliminate all enemies with the help of those units. Choosing and managing a strategy is one of the major AI research problem areas in RTS games. Micromanaging units at a tactical level is another, equally important area.

Presently our agent uses a hybrid CBR/RL approach to learn how to manage those units in combat situations. Cases are based on single units and consist of a case description, the current state of the game environment, as well as a solution, which is a set of all possible actions for the currently active unit. Each action has a value assigned to itself that represents its fitness. Since the number of possible states the game environment can be in is huge, abstraction is needed for the representation. The environment state is abstracted into influence maps to represent the areas of influence for opposing and own units. If we regard these influence maps as simple greyscale pictures with intensity values representing the influence values, the pictures can in turn be converted into histograms to make them usable as an efficient similarity metric. A nearest neighbor (NN) retrieval of cases in the case base is done based on the histograms and furthermore based on other attributes specific to the single unit in question.

2.2 Trace Recording

The extraction of primary, unaltered traces from StarCraft can happen in two ways. On the one hand, game traces can come from game replays, i.e. recorded games that have already been played. StarCraft provides functionality to record games in a standardized way that produces such primary game traces with a common model (M-Traces) regardless of where and when games are recorded.

On the other hand, traces can also be recorded from active gameplay to create M-Traces similar to those created by the game itself. This behavior is comparable to other approaches that build trace-based systems which record user interactions [5]. As a result both actively recorded M-Traces and those stored as replays have an identical structure irregardless of provenance.

Replays are simply a recording of all actions a player performs during a game, inherently traces of user interactions with the game. Figure 1 shows the structure of such a trace. Expert players can perform several hundred of those recorded user interactions per minute. As can be seen in the structure of the replay, the type of actions can vary greatly and involves differing numbers of parameters.

Time	Player	Action	Parameters	Units ID
17728	Ryan[Shield]	Move	(1455,1804),0,228,0	0
17728	Ryan[Shield]	Train	Interceptor/Scarab	
17728	MenSol[Zero]	Move	(1148,2014),0,228,0	0
17730	MenSol[Zero]	Move	(1145,2007),0,228,0	0
17732	Ryan[Shield]	Move	(1409,1853),Hydralisk,228,0	3420
17734	MenSol[Zero]	Attack Move	(1144,2002),0,228,	0
17736	Ryan[Shield]	Move	(1418,1872),Hydralisk,228,0	3410
17736	MenSol[Zero]	Hotkey	Select,3	
17738	MenSol[Zero]	Move	(1173,1950),0,228,0	0
17752	MenSol[Zero]	Select	Hydralisk(x5)	3452,3422,3393,5424,5499
17754	Ryan[Shield]	Shift Select	3475	3475
17756	MenSol[Zero]	Move	(1008,1820),0,228,0	0
17758	MenSol[Zero]	Move	(961,1861),0,228,0	0
17760	MenSol[Zero]	Move	(1024,1897),0,228,0	0
17762	Ryan[Shield]	Move	(1149,1891),Shuttle,228,0	3492

Fig. 1. Contents of a StarCraft Replay

While both actively recorded traces and those read from replays can share the same model, the content of such traces can be slightly different. The reason for this is that game replays are written by the game itself with direct access to game states while active recording is done through an interface on top of the game (Broodwar API, BWAPI) which abstracts from the underlying interactions of the player with the game.

This was one of the reasons why we chose not to work directly with primary traces in our application but with an already transformed version. Another reason is the way game replays are shown when they are examined with the means that the games provides. As their structure already suggests, replays are not simply played like a movie. Instead, the actions stored in them are executed within the game engine (much like in an actual game) and the results are shown. Therefore we can simply read replayed games through the same BWAPI interface that we use for active recording. Transforming a replay like this leads for instance to a “Move” command to a unit at one point in time being translated into a whole

number of attribute changes (x- and y-coordinates, velocity, angle etc.) for that unit over the next few game cycles.

The last and most important reason for performing preliminary trace transformations is the actual result we hope to achieve from using the information stored in these traces: We are aiming for a better performance of our ML agent trying to play parts of the game. While player interactions can certainly provide important information to this end, player actions are always only a reflection on what happens inside the game environment. By looking at those in-game states and omitting actions issued to units, the information on the human interaction with the game is translated into only in-game information. This is however the information that is most relevant for an AI agent learning how to play the game. Because of these considerations we chose to use in-game data. This in-game data is, according to the previous elaborations and also according to the theory behind TBR [3] a trace transformed by the game engine. However, unlike TBR theory, this transformation is not fully deterministic and also not reversible due to the complex nature of the computations inside the game engine. While traces can always be linked to their original replays, at the lowest level, i.e. the level at which we record actions from the game environment, there can be slight differences between transformed traces that were generated from the same replay making it non-deterministic.

A further decision we made, was to not record the entire game state at any one point in time, but only differences between the current and the previous state. The major reason for this choice was the far lower computational requirements while retaining the same amount of information. StarCraft, much like any RTS game, requires real-time actions. However, internally it runs through a set number of game cycles each second. At standard speed there are 24 cycles each second which would require huge amounts of information to be written to the database at any one cycle. If, on the other hand, only differences are recorded, these writing operations become a fraction of the previous amount. Currently for each unit 60 attributes are monitored. Each player can have up to 200 units. This would mean that 24K float values (192KB) per cycle or roughly 4.5 MB per second have to be written to the database. In an average game with a length of 20 minutes this can easily lead to 5GB of data. If we only record differences on the other hand, we can manage to store about 7.5h of gameplay in a 3GB database while retaining all information.

3 Trace Model

The model of the original primary traces gathered from user interactions is defined by the game itself which records all user interactions and saves them as replays of the game. Figure 1 shows an excerpt of a replay displayed in a third-party tool “BWChart” since replays are recorded in binary format.

We defined the trace model based on the BWAPI interface while focusing on “Units” as the important objects. This includes any changeable environment components, e.g. units and buildings. The actual observed elements (obsels)

[6] that make up the core of the trace are attribute changes in the previously mentioned Unit objects.

One example for an observed element is the change of the X value of the position of the monitored unit 33787 by -46 that can be seen in the first line of Figure 4.

Apart from the obsels, initial values of a game are also stored. These values such as map parameters and starting positions are only saved once since they are static, therefore no deltas for these values have to be recorded. These static values are omitted from the database model in Figure 2 for reasons of simplification.

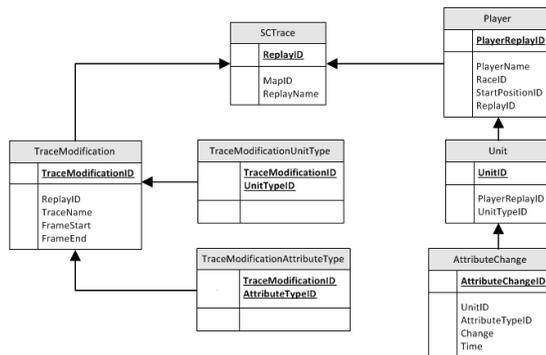


Fig. 2. Simplified Database Model for SC Traces

The model of the trace is similar to other trace models but more heavily focused on StarCraft specific data in specific and the problem domain in general. The structure is common to most RTS games and includes the previously mentioned obsels stored as “AttributeChange” that are linked to initial, unchanging, state of the environment in the “SCTestrace” component. The “Trace Modification” part and its attachments represent the knowledge on trace transformations.

All of the data is stored in a relational database in order to facilitate efficient and easy access.

4 Trace Visualization and Transformation

Some features of StarTrace, like the possibility to display attribute changes over time in diagrams, were planned from the very start. Other features were added as demand arose from designing and implementing the agent that learns the game.

- Enable users to create new traces based on transformations applied to one or more existing traces.
- Provide access to all recorded information through visual means.

- Visualize the recorded traces in an easy to understand way that allows a top-down approach for recognizing patterns and episode signatures.
- Offer filtering opportunities to only display selected parts of one or several traces.

For now, the interface allows the expert to more easily identify salient features of traces, but the goal is ultimately to allow the agent to learn from its traces. This can either be done by improving the learning performance directly (finding and reusing episode signatures during the retrieval phase of cases) or by better understanding the intrinsic agent behavior by visualizing its performance over time.

4.1 Design and Implementation

StarTrace is a browser application based on PHP, HTML, CSS and AJAX. It uses the MySQL database which the traces are recorded to. Figure 3 shows the layout of the main window. StarTrace provides several filter options when searching through the primary traces. Primary traces in context of the tool are traces that have already been transformed from recorded actions to in-game attributes as described in Section 3. Once a primary trace has been selected, further transformations can be applied in order to filter through the trace. Users can select which types of obsels they want to look at by limiting both Unit objects that attributes are displayed for and attributes that are displayed.

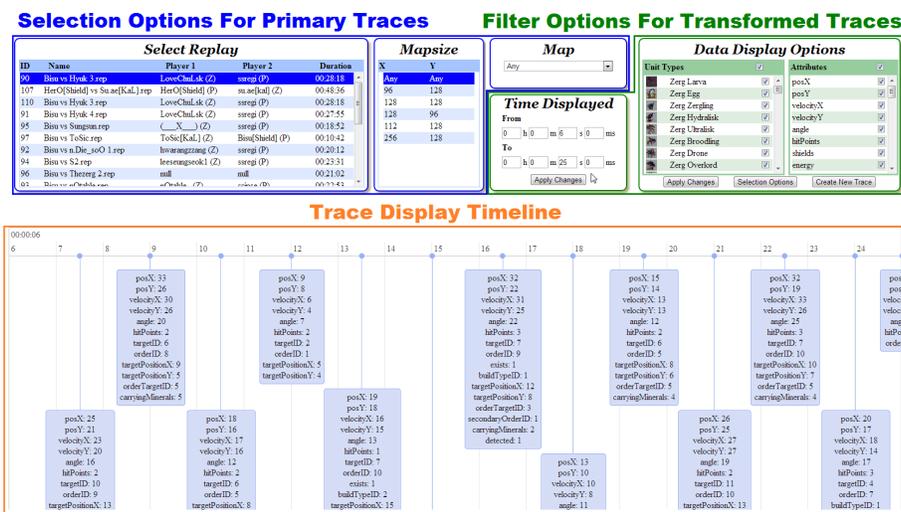


Fig. 3. Overview of the StarTrace Main GUI

The trace display window shows the result of the filter options selected: All changes of chosen attributes from the selected trace in the selected time frame

for chosen unit types. As the time frame can be freely modified between a few milliseconds and several minutes, the sheer amount of displayed information could easily have become confusing. Therefore, the standard view only gives a summary of the numbers and types of attributes that have changed in the selected time frame. Details on attribute differences can be seen when one of the summary elements is selected (Figure 4).

Unit ID	Unit Type	Attribute Type	Attribute Change
33787	Zerg Drone	posX	-46
33787	Zerg Drone	posY	From 0 h 0 m 2 s 143 ms To 0 h 0 m 2 s 143 ms
33788	Zerg Drone	posX	-1

Close Details

Fig. 4. Detailed View of Attribute Differences

A central feature of a TBS is the ability to define new traces based on existing ones. Currently, StarTrace allows the creation of new traces based on the filter criteria that are also available in the main view (Figure 5). Additionally we added a feature that is mostly based on the desire to enable the ML agent to learn from these traces: the ability to not use the differences of attribute values for each time step but to use aggregate values. In terms of unit locations this means for instance that the trace will not contain the change in position for a certain time step but the absolute position at each time step.

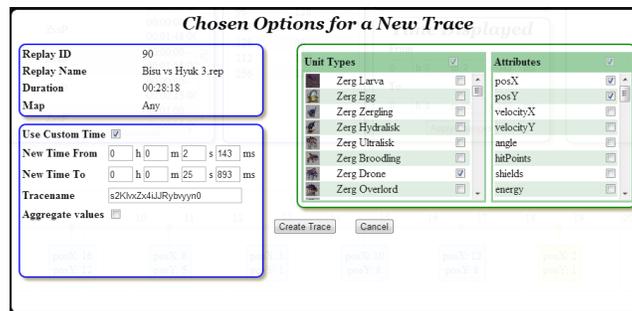


Fig. 5. GUI for Creating a New Trace

Section 4.2 showcases how an aggregate trace displayed in the tool can elaborate agent behavior inside the game and thus eventually enable improvements in the learning performance.

4.2 Using StarTrace to Understand Agent Behavior

Figure 6 shows an excerpt of a diagram display for an example transformed trace. The transformation resulted in an M-trace consisting of aggregate values instead of differences and only two of the 60 original attributes. Furthermore, the transformed trace only contains data for one specific type of unit.

For each individual unit, the development of the values for both attributes hitPoints (green) and groundWeaponCooldown (red) is shown over time.

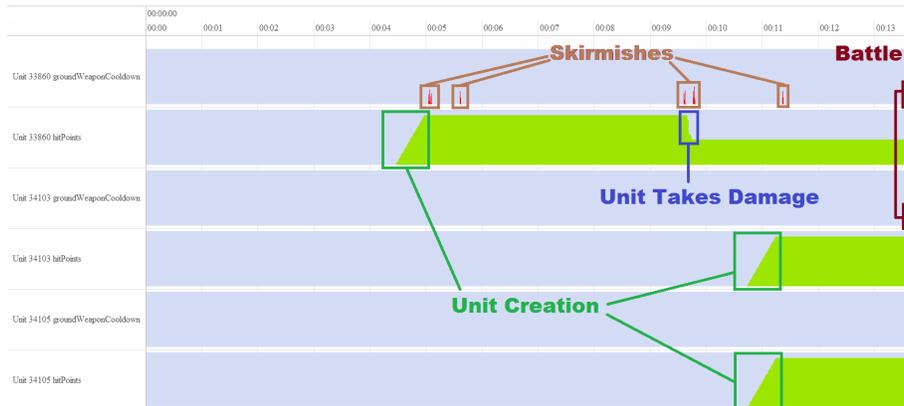


Fig. 6. Diagram of a Trace Transformed to Display Selected Aggregate Values (with Annotations)

The displayed trace can already show certain effects that are important to gameplay. On a strategic level, the rise of the hitPoint variable marks the creation of the respective units, i.e. the point in time when the player felt it necessary to build this type of unit. The short cycles where the groundWeaponCooldown variable first rises (after firing a weapon) and then decreases back to zero marks confrontations. As can be seen from the intensity, at first there is mostly short skirmishes. Towards the end of the displayed period there is a full scale battle in which two of the three displayed units are involved.

Distinguishing between skirmishes and full battles which both require different behavior would allow the AI agent to decide on different micromanagement behavior patterns. This makes sense as these two situations usually have different aims such as scouting/reconnaissance for skirmishes versus elimination of strategic groups of buildings or units for battles. However, elements like movement patterns and targeting strategies can not be learned from the values included in the currently displayed transformed trace. Therefore, after identifying these key events, a next step would now be to go back to the original M-Trace and create another transformation containing observed elements such as changes in x/y positions, velocities and target IDs.

5 Related Work

There are numerous approaches that try to use the intrinsic knowledge stored in StarCraft replays. [7] analyze a large corpus of StarCraft replays in the context of cognitive research. The authors try to find a correlation between actions that are observable in the replays and performing successfully in the games. Their results show that winning games is directly related to the number of actions that a player performs. In [8] the authors build a case base from automatically annotated StarCraft replays. They do this by defining a “Goals-to-win-StarCraft” ontology and automatically breaking up replays into cases by splitting them according to the actions happening.

[4] directly applies TBR to micromanagement in StarCraft in a way that is at least partially similar to what we are planning for our agent. The author creates an ML agent to micromanage combat unit based on CBR. Some traces of unit attributes are used in the case descriptions. However, the development process does not involve the analysis of game replays or a review of recorded agent behavior, attribute selection is only based on expert knowledge.

There are other systems to record, manage and transform traces that work similar to StarTrace. Georgon et al. [9] develop ABSTRACT, a tool to analyze users interactions with a complex technical device such as a car. ABSTRACT visualizes those traces and allows experts to define patterns in the user traces. The aim is to use the combination of expert knowledge and trace representation to define a cognitive model of the user that generates the trace. TStore [10] is a web-based TBMS which handles the storage, transformation, and reuse of modeled traces. Besides providing predefined options for transforming traces, it also offers the ability to define customized transformations based on Finite State Transducers. The authors test the trace recording performance of their environment by collecting user interaction traces from Wanaclip, a video clips composer.

6 Conclusion and Future Work

In this paper we presented StarTrace, a tool to visualize and transform traces of the RTS game StarCraft. The tool enables users to work with M-Traces generated from user interaction traces. Those user interaction traces are either user interaction traces originating from StarCraft itself or are player interactions recorded through an interface during active gameplay. StarTrace provides functionality to filter through and visualize StarCraft traces in a number of ways. It furthermore allows the user to create new traces by selecting filtering as well as transformation options for obsels directly in the GUI.

Our goal is to use StarTrace and traces in general to improve the performance of an ML agent that attempts to learn parts of the game. In an example application we showed how trace visualization and transformation provided through StarTrace can lead to better understanding of recorded performance in the game. Eventually we plan to re-integrate knowledge obtained through the tool directly

into the learning process of the agent.

There is also a number of other possible future improvements that have arisen from applying StarTrace to the transformed StarCraft traces. If unit attributes in a trace are not only selectable by unit type or attribute type but for each Unit object in a trace separately, that would give a maximum of control over the filtering options. Section 4.2 already showed how diagrams for several units and attributes from a trace can be displayed simultaneously. We plan on extending this to include the possibility to display several distinct traces at once, eventually with the option of merging values from multiple traces.

A crucial component of a TBS is the possibility to manually identify, store and elaborate on recognition patterns in traces. Among other things, externally appended annotations such as those in Figure 6 could then be added directly with the tool. This feature is the next major planned addition to the tool and will enable a whole new set of potential applications.

References

1. Buro, M., Furtak, T.: Rts games and real-time ai research. In: Proceedings of the Behavior Representation in Modeling and Simulation Conference (BRIMS), Citeseer (2004) 63–70
2. Wender, S., Watson, I.: Applying reinforcement learning to small scale combat in the real-time strategy game starcraft:broodwar. In: Computational Intelligence and Games (CIG), 2012 IEEE Symposium on. (2012)
3. Mille, A.: From case-based reasoning to traces-based reasoning. *Annual Reviews in Control* **30**(2) (2006) 223–232
4. Szczepański, T.: Game ai: micromanagement in starcraft. Master’s thesis, Norwegian University of Science and Technology (2010)
5. Georgeon, O., Henning, M.J., Bellet, T., Mille, A.: Creating cognitive models from activity analysis: A knowledge engineering approach to car driver modeling. In: International Conference on Cognitive Modeling. (2007) 43–48
6. Cordier, A., Mascaret, B., Mille, A.: Dynamic case based reasoning for contextual reuse of experience. In: Provenance-Awareness in Case-Based Reasoning Workshop. ICCBR. (2010) 69–78
7. Lewis, J., Trinh, P., Kirsh, D.: A corpus analysis of strategy video game play in starcraft: Brood war. In: The Annual Meeting Of The Cognitive Science Society (COGSCI 2011). (2011)
8. Weber, B., Ontanón, S.: Using automated replay annotation for case-based planning in games. In: ICCBR Workshop on CBR for Computer Games (ICCBG-Games), Springer (2010)
9. Georgeon, O., Mille, A., Bellet, T.: Analyzing behavioral data for refining cognitive models of operator. In: Database and Expert Systems Applications, 2006. DEXA’06. 17th International Workshop on, IEEE (2006) 588–592
10. Zarka, R., Champin, P.A., Cordier, A., Egyed-Zsigmond, E., Lamontagne, L., Mille, A.: Tstore: A web-based system for managing, transforming and reusing traces. In Luc Lamontagne, J.A.R.G., ed.: ICCBR 2012 TRUE and Story Cases Workshop. (September 2012) 173–182

Toward Addressing Noise and Redundancies for Cases Captured from Traces and Provenance

David Leake and Joseph Kendall-Morwick

Indiana University and Capital University
leake@cs.indiana.edu, jkendallmorwick@capital.edu

Abstract. The use of automatically-captured event sequence information, such as traces and provenance, provides an exciting opportunity for large-scale case capture. Research in process-oriented CBR and trace-based reasoning is studying some key issues for this capture, such as segmenting event streams and representing the complex cases which may result. However, automated case capture methods can pose another challenge, not yet addressed: How to handle noise within the records from which cases are built, and how best to handle the possible proliferation of large and highly similar cases. This position paper presents these problems, argues for their importance, and points to research directions for potential solutions. It proposes that cases from automatically-generated sources will require *internal case maintenance*, and that addressing redundancies in large-scale case capture may require new case representation strategies to help address the potential flood of case information at the case level—within compound cases—in addition to traditional case-base maintenance operating on cases as distinct units.

1 Introduction

Case-based reasoning (CBR) systems depend on the quality of their case bases. CBR research has explored numerous case acquisition methods, ranging from by-hand case generation to capture from generative problem-solving systems, to mining cases from existing data. Automatic case capture methods in trace-based reasoning [1] and in process-oriented CBR provide an interesting alternative to these methods by mining cases from raw event sequences. For example, instrumentation in a car can capture information about driving, providing a basis for generating driving cases [2], and provenance capture systems can capture records on process execution resulting from workflows [3]. The idea of automatically capturing cases by mining sequence records is very appealing for CBR. To the extent such an approach is successful, it could help fulfill a vision for much wider applicability of CBR.

Realizing the vision requires addressing a number of challenges, which are being addressed, for example, by research aimed at determining what parts of a trace should be extracted into a case [2]. This position paper argues for the importance of two additional research problems, which have received little attention: First, developing new methods to repair noise in the steps captured within a case due to flaws in automatic captured processes, and, second, developing methods to address problems of proliferation of highly similar cases. These cannot be satisfactorily addressed by standard

deletion-focused maintenance methods, because of the potential loss of useful information. Addressing these problems will require developing (1) methods for internal case maintenance to address data quality issues within cases, and, (2) case representations which can preserve information about the variations in complex cases without repeating redundant information. Both tasks are closely related, in that variational representations can help in representing and manipulating case information with gaps and uncertainties.

2 Addressing Internal Case Noise

Data quality is widely recognized as an important issue. When automatic case capture methods are applied to complex real-world systems, the quality of captured results depends on sensor capabilities and the vagaries of the capture process. The uncertainties of information capture by physical sensors is well known, but the problem can arise in software systems as well. For example, e-Science performs scientific experiments *in silico* by grid computing, executing large-scale distributed simulations. As such processes are executed, an automated provenance capture system records provenance information. However, the records are far from perfect. A study by Cheah and Plale [4] examined 2890 provenance traces, captured by state-of-the-art methods for provenance capture from distributed computation, from data generated by NASA's Advanced Scanning Microwave Radiometer - Earth Observing System over a 1 month period. The study revealed multiple sources of noise relating to failures of provenance collection instrumentation, including failures which can occur in the capture, storage, and retrieval phases for provenance information, as well as when provenance data from multiple sources are merged. Such errors commonly result in missing data, but may result in redundant data as well: They reported that that nearly half of the steps within the provenance data recorded in the NASA dataset are redundant.

Dropped information results in provenance gaps. Thus generation of cases from traces cannot assume that the process is simply one of selection: It must include augmentation, filling in gaps and potentially replacing erroneously recorded steps. Existing cases may provide information to support these repairs, and methods for case adaptation may prove helpful for repairs. However, the repair task is intrinsically different from the adaptation task: rather than being driven by what is needed in a new situation, it will generally be driven by the need to reconstruct what actually happened, to restore the captured case's integrity. With noisy cases, case integrity also assumes increased importance in retrieval, but the problem becomes subtle: Uncertain aspects of a case may not be important if they will be replaced by adaptation. Thus *internal maintenance*, such as segmentation and repair, must interact with reasoning about needs for case use.

3 Addressing Case Proliferation

Extensive CBR research studies case base maintenance (e.g., [5]), especially how to control case base growth by choosing cases to retain or delete in order to achieve case base compression while maintaining competence. However, for large, complex cases, case deletion may be too coarse-grained. It may be inappropriate to delete a case because:

- Apparently similar cases may differ in small parts, which are nevertheless uniquely useful. For example, in the domain of recipe planning, two custards may have identical preparation except for the ratio of egg whites to yolks. Storing all details of each plan is largely redundant, but deleting either loses an important variant which could be needed.
- Many structured domains have a very large set of features, with some appearing in few cases. There may be a cluster of cases which are highly similar and thus seemingly redundant, and yet contain unique examples of seldom appearing features.

In situations such as the above, clusters of similar cases may need to be retained intact. However, this poses its own problems: Too many similar cases may result in decreased diversity, with nearby cases obscuring important variations. Thus it is important to retain the information of variants while not swamping true differences with overly similar cases. Our position is that achieving this end should involve altering case representations to generalize over a set of similar cases, as discussed in the next section.

4 Proposed Next Steps

We propose abstracting cases into shared structures with annotations for variations. For instance, provenance records can be recorded from the same experiment run multiple times, and produce potentially unique results depending on run-time events and noise in the capture process. The differences between these multiple iterations of what is effectively the same approach could be captured in a generalized case in which the repeated content is only represented once, and each difference between these similar records is recorded as a difference from the main content. This clearly reduces the storage required for the case base. It may also increase the applicability of the case. With traditional representations, the k most relevant cases may only include a small subset of the cluster to which those k cases belong, but a more generalized structure may capture the entire cluster—effectively providing for a flexible constraint on the number of individual past episodes used to solve a problem. The generalized case approach raises numerous research questions concerning how to detect and represent commonalities and differences. As keeping small differences risks retaining noise, retention and reuse strategies may be especially important.

PO-CBR researchers have recommended retrieval strategies aimed at addressing efficiency when using large case-bases of structured cases [6, 7]. We now recommend the development of accompanying strategies for case representation which not only should further this goal, by decreasing the size of the case base, but also enable improved performance by providing new opportunities for case-mining and re-use strategies.

The basic approach we propose is a case representation which allows for compressing multiple cases into a single structure, disposing of redundant data, but also retaining the small differences between cases collapsed into this single representation as metadata, enabling the original cases to be re-derived. Such structures provide several advantages, touching on different phases of the CBR cycle:

- Eliminating redundant data increases feasibility of computationally expensive structural comparison and index-based retrieval algorithms, increasing the amount of time the system can devote to re-use.

- If such representations are lossless, full advantage can be taken of the small differences between highly similar cases, eliminating the problem of reduced competence from case-base maintenance.
- Discovery of common structural elements, along with the deviations from these elements, assists with the identification of potential noise within recorded cases.
- Adaptation of cases derived from generalized structures can also benefit from the knowledge stored in these structures reflecting common points of divergence from the typical model, also enabling learning from past failures.

Such a representation must be designed with mining, retrieval, and re-use in mind, since there are significant implications for each. Determining which structural features to collapse is a difficult problem from a case-mining perspective, since there is significance to these features beyond their common, syntactic properties. This is directly related to studies of frequent substructure mining [8], and relates as well to some of the foundational research on human knowledge structures, such as MOPs [9], from the early cognitive science study of CBR. Structural comparison may not be as simple to implement for retrieval purposes if important aspects of cases are hidden in metadata. Analyzing the relevance of multiple cases derived from a generalized structure should benefit from knowledge of which structural features are common among the consolidated cases, minimizing repetition in the similarity assessment process. Enabling case use at different levels of abstraction may be useful as well, and also relates to the issue of how to associate process traces with particular events.

References

1. Mille, A.: From case-based reasoning to traces based reasoning. Technical Report 2281, LIRIS, University of Lyon (2006)
2. Mathern, B., Mille, A., Cordier, A., Cram, D., Zarka, R.: Towards a Knowledge-Intensive and Interactive Knowledge Discovery Cycle. In Luc Lamontagne, J.A.R.G., ed.: Proceedings of the ICCBR-12 Workshop TRUE and Story Cases: Traces for Reusing Users' Experiences. (September 2012) 151–162
3. Simmhan, Y.L., Plale, B., Gannon, D.: Karma2: Provenance management for data driven workflows. *International Journal of Web Services Research*, Idea Group Publishing **5** (2008)
4. Cheah, Y.W., Plale, B.: Provenance analysis: Towards quality provenance. 8th IEEE International Conference on eScience 2012 (10/2012 2012)
5. Leake, D., Wilson, D.: Categorizing case-base maintenance: Dimensions and directions. In Cunningham, P., Smyth, B., Keane, M., eds.: *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, Berlin, Springer Verlag (1998) 196–207
6. Bergmann, R., Minor, M., Islam, M.S., Schumacher, P., Stromer, A.: Scaling similarity-based retrieval of semantic workflows. In: *Proceedings of the ICCBR-12 Workshop on Process-Oriented Case-Based Reasoning*. (2012)
7. Kendall-Morwick, J., Leake, D.: On tuning two-phase retrieval for structured cases. In: *Proceedings of the ICCBR-12 Workshop on Process-Oriented Case-Based Reasoning*. (2012)
8. Yan, X., Yu, P.S., Han, J.: Graph indexing: a frequent structure-based approach. In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. SIGMOD '04, New York, NY, USA, ACM (2004) 335–346
9. Schank, R.: *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, Cambridge, England (1982)

The ICCBR 2013 Doctoral Consortium

At the
Twenty-First International Conference on
Case-Based Reasoning
(ICCBR 2013)

Saratoga Springs, U.S.A.
July 2013

Thomas Roth-Berghofer and Rosina Weber (Eds.)

Chairs

Thomas Roth-Berghofer
University of West London, UK

Rosina Weber
Drexel University, USA

Mentors/Reviewers

Klaus-Dieter Althoff, DFKI/University of Hildesheim, Germany
Ralph Bergmann, University of Trier, Germany
Béatrice Fuchs, Université Lyon 1, France
Pedro González-Calero, Complutense University of Madrid, Spain
Luc Lamontagne, Laval University, Canada
David Leake, Indiana University, USA
Jean Lieber, LORIA - INRIA Lorraine, France
Cindy Marling, Ohio University, USA
Stefania Montani, University Piemonte Orientale, Italy
Hector Munoz-Avila, Lehigh University, USA
Enric Plaza, IIIA, Spain
Barry Smyth, University College Dublin, Ireland
Ian Watson, University of Auckland, New Zealand
Nirmalie Wiratunga, Robert Gordon University, Scotland
Qiang Yang, HKUST, Hong Kong

Preface

The objective of a doctoral consortium (DC) is to nurture the interests of students (and others) who recently started studying a specific research field. A DC provides participants with an opportunity to describe and obtain feedback on their research, future work plans, and career objectives from senior researchers and peers. For the case-based reasoning (CBR) community, the DC is very important because it provides a forum for the community to welcome, guide, and encourage junior researchers who may become active (and even leading) community members.

The International Conference on CBR (ICCBR) held its first DC at ICCBR-09; this is the 5th ICCBR DC. In the first two years, we naturally had few participants (4 and 3, respectively). We addressed this in 2011 (e.g., with increased publicity), and were fortunate to host 10 participants. We thought that was a good number, yet we had 16 last year, which greatly exceeded our expectations. This year we have 7 participants, which is consistent with the submission level of the conference.

We advertised the DC widely to identify prospective participants and asked them to submit: (1) a 3-page Research Summary; (2) a 1-page CV; (3) a 1-page statement on their DC expectations; and (4) a 1-page letter of support from their advisor(s). The summary requires students to describe their objective, progress, and plans using the conferences publishing format, the CV describes the applicants experience, the expectations requires the applicant to consider what they may share or learn at the DC, and the letter ensures that advisors are aware of this event. Our PC reviewed each application; all were found to be CBR-relevant and were invited to participate. We assigned a mentor per student, matching mentors who could provide valuable feedback from a different perspective (including nationality). Mentors provided iterative guidance/feedback on students presentations prior to the DC.

At the DC, each student gave a 15-minute talk on their Research Summary, followed by a 10-minute Q/A session (on presentation skills and content) led by their mentor. (Each mentor was asked to attend at least 2 students' presentations, thus allowing them to also attend co-timed events.) Also, senior researchers gave presentations to provide the students with insights on community interests and career opportunities. We thank Barry Smyth and Mirjam Minor for giving Career Reflection presentations. Finally, a group lunch and dinner provided students with a relaxed opportunity to chat with other conference participants (the DC was open to all ICCBR registrants).

We thank the PC and mentors for their participation and assistance. We hope that it enhanced each students interest in studying CBR. We strongly encourage them to participate in future ICCBR conferences and related venues. We wish them well!

June 2013

Thomas Roth-Berghofer

Rosina Weber

Preference-Based Case Based Reasoning

Amira Abdel-Aziz

Philipps University, Marburg, Germany
amira@mathematik.uni-marburg.de

1 Introduction

Preference-based CBR is a new approach to case-based reasoning, in which knowledge representation and problem solving are realized on the basis of preference information. This approach is appealing, mainly because case-based experiences naturally lend themselves to representations in terms of preference relations, even when not dealing with preference information in a literal sense. The flexibility and expressiveness of a preference-based formalism well accommodate the uncertain and approximate nature of case-based problem solving. The approach mainly consists of inferring preferences for candidate solutions in the context of a new problem, given such preferences in similar situations; thus, the basic “chunk” of experience stored in a case base is (pairwise) preferences over candidate solutions that are “contextualized” by a corresponding target problem. The advantages of a preference-based approach to problem solving in comparison to the more conventional constraint-based one have been discussed in [2]. It is argued that in many contemporary application domains, the user has really little knowledge about the set of possible or feasible solutions. As the user does not know what the best achievable plan could be or which product or document is the best one, it is difficult for the user to characterize the solution or its characteristics properly. The result of this would be either that the user will ask for an unachievable goal which does not correspond to an available solution, or the user will ask for too little and in return receive a solution which can be improved. Related to this regard, came the motivation for my current work as well as my planned future work.

2 Finished Work

The work done so far is a continuation of recent work regarding a preference-based approach discussed in [1], on a formalization of preference-based CBR. This approach focused on an essential part of the methodology: a method to predict a most plausible candidate solution given a set of preferences on other solutions, deemed relevant for the problem at hand. More specifically, the method consists of inferring preferences for candidate solutions in the context of a new problem, given knowledge about such preferences in similar situations. In [3], we went one step further by embedding this method in a more general, search-based problem solving framework. In this framework, case-based problem solving is

formalized as a search process in which a solution space is traversed through the application of adaptation operators, and the choice of these operators is guided by preference information collected in previous problem solving episodes. Our finished work so far is about applying an inference procedure, which specifically consists of inferring preferences for candidate solutions in the context of a new problem. The statistical approach of the maximum likelihood estimation was used in the mentioned framework to aid the search for the best possible solution, using the preferences (solutions) which were created and stored in the case base. The effectiveness of this approach is illustrated in two case studies shown in [3], one from the field of bioinformatics and the other one related to the computer cooking domain. Another application was the use of preference-based CBR as a search method to explore a large dataset of protein binding sites (CavBase), and find a solution in a much shorter time as compared with other search methods [9].

3 Future Work

From our perspective, case-based experience can be modeled in terms of preference information in a quite convenient way and moreover, case-based inference can be realized quite elegantly in the form of preference processing. The vision and ambition of my work, is to develop an alternative generic methodological framework for case based reasoning, on the basis of formal concepts and methods for knowledge representation and problem solving with preferences. This framework can then be used in several applications in different domains, as was shown in [3].

3.1 Case Base Organization and Maintenance

It is clear that simply storing each encountered problem along with a list of associated (pairwise) preferences is not optimal, especially since a case base of that type may quickly become too large and hamper efficient case retrieval. In CBR, this problem has been addressed by methods for case base maintenance [4]. Such methods seek to maintain the problem solving competence of a case base, mainly by using case base editing strategies [5, 6], including the removal of misleading (noisy) or redundant cases and the summarization of a set of cases by a single representative (virtual) case. Our idea is to transfer existing approaches for case base maintenance from conventional case bases to preference case bases. Formalizing the concepts of redundancy, developing efficient methods for organizing and maintaining case bases in preference-based CBR is one of the goals of my future work.

3.2 Learning of Similarity Measures

Another aspect to consider in my future work, is the question of how to access a corresponding case base to support the current problem solving process. In our

case, where problems are not associated with single solutions but rather with preferences on solutions, an obvious generalization of this type of inference is to combine the preferences associated with the nearest neighbors into a preference relation on candidate solutions for the query. As a consequence of this type of case-based inference, the success of a CBR system crucially depends on the specification of a suitable similarity measure. In my future work, the problem of learning a (global) similarity measure in an automatic way [7, 8] is planned to be addressed.

3.3 Search Methods

Once a subset of (presumably) most relevant problems has been retrieved, case-based inference proceeds by combining in one way or the other, the solutions of these problems into a candidate solution for the query problem. The type of aggregation procedure which is applied to this end strongly depends on the structure and representation of solutions, and on the type of preference relation defined on the solution space. Moreover, it is important to recall that problems are not associated with single solutions but rather with preferences over solutions. Putting this into consideration, it is very important to determine which problem-solving algorithms are best used in the context of preference-based CBR. The next step in my work will also include investigation of different search methods that are best used in preference-based CBR.

References

1. Hüllermeier, E., Schlegel, P.: Preference-Based CBR: First Steps toward a Methodological Framework. ICCBR (2011)
2. Brafman, R., Domshlak, C.: Preference Handling-An Introductory Tutorial. AI Magazine 30(1), (2009)
3. Abdel-Aziz, A., Cheng, W., Strickert, M., Hüllermeier, E.: Preference-Based CBR: A Search-Based Problem Solving Framework. ICCBR (2013)
4. Smyth, B., McKenna, E.: Competence Models and the Maintenance Problem. Computational Intelligence 17(2):235-249, (2001)
5. McKenna, E., Smyth, B.: Competence-Guided Editing Methods for Lazy Learning. In Proceedings of 14th European Conference on Artificial Intelligence, Berlin 60-64, (2000)
6. Delany, S., Cunningham, P.: An Analysis of Case-Base Editing a Spam Filtering System. In Proceedings of European Conference on Case-Based Reasoning, Heidelberg 128-141, (2004)
7. Stahl, A., Gabel, T.: Optimizing Similarity Assessment in Case-Based Reasoning. In Proceedings of 21st National Conference on Artificial Intelligence, Boston, MA, USA (2006)
8. Richter, M.: Foundations of Similarity and Utility. In Proceedings of 20th International FLAIRS Conference, Key West, Florida (2007)
9. Abdel-Aziz, A., Strickert, M., Fober, T., Hüllermeier, E.: Preference-Based CBR: A Search-Based Problem Solving Framework. ICCBR Workshop (2013)

Case-based Learning of Ontology-based Goal-Driven Autonomy Knowledge

Dustin Dannenhauer

Department of Computer Science and Engineering, Lehigh University, Bethlehem PA
18015, USA

1 Introduction

Goal-driven autonomy (GDA) is a conceptual model for reasoning over goals in autonomous agents. The model makes use of a planner (treated as a black box) which produces both a plan and corresponding expectations. GDA reasons about goals in four steps: discrepancy detection, explanation, goal formulation, and goal management. The process of a GDA agent begins with the goal management component sending a goal to the planner. The planner then produces a plan and expectations for what should be true in the future. The agent begins executing the plan and, concurrently, the discrepancy detection component observes any expectations that are not satisfied. When this happens, the explanation component generates an explanation for why the discrepancy occurred and sends the explanation to the goal formulation component, which may create a new goal. This new goal is sent to the goal management component, which then adjusts the priority of each goal and decides which goal(s) to send to the planner.

Almost all work on GDA has been non-hierarchical (for example, discrepancy detection in the ARTUE system uses set-difference) nor has previous work made use of an ontology. We propose using an ontology to represent the state of the environment in order to provide more knowledge-rich reasoning within GDA. At this time we are not committed to a particular formalism for an ontology, and are currently looking into using a Web Ontology Language capable of expressing Description Logics (OWL-DL). Real-time strategy (RTS) games are one domain where the game state could be represented as an ontology. The following example demonstrates how each component of GDA could benefit from using an ontology in the RTS game Starcraft:

1. **Discrepancy Detection:** A hypothetical plan achieving the goal “surround the enemy base” could produce an expectation such as “the agent controls regions 5 and 6”. The discrepancy detector could identify if this expectation was met by adding the expectation statement to the ontology and checking consistency. The ontology would contain facts such as in what region each unit is located. The ontology would also contain a description logic rule stating that: “A region belongs to a player if there is at least 1 unit in the region and that player owns every unit in the region.”

2. **Explanation:** Continuing the example above, the explanation component could identify which part of the rule was not satisfied. If there were no units in the region, then a valid explanation could be that somehow our units were not able to travel to the region (perhaps there is an enemy force between the agent’s units and the target region). But if the other part of the rule failed, that not all the units were ours, that would be evidence for a different explanation, perhaps that the enemy controls the target region.
3. **Goal Formulation:** Either explanation would provide rich knowledge for the goal formulation component. If the explanation was that agent’s units were not able to travel to the region, the new goal could be to send a more powerful force (or send different types of units). The units chosen could be those specifically effective at defeating the enemy units that destroyed the first force. Which units are effective against other units can be represented in the ontology. But if the explanation was that the enemy controlled the target region, then perhaps the new goal should be drastically different. The new goal could be to develop a sneak attack on the enemy base from behind using air units.
4. **Goal Management:** The goal management component could take the goals produced from the goal formulation component and lower the priority of the current goal. Perhaps the new information in the ontology warrants abandoning the “surround the enemy base” goal and instead a better goal would be to build up the agents army.

Most GDA systems reason at the unit level of a game (i.e. unit 3 is at (5,6)) and are too detailed to be meaningful for the user. Ontologies would allow one to abstract information such as labelling a region as “contested”, “owned by the agent” or “owned by the enemy”. This level of granularity is easier for a human to understand and interact with the system. Additionally, some researchers argue that STRIPS representations are too limited to represent events that happen in the real world, more structured representations are needed in order to capture more real world constraints. While ontologies are knowledge-rich and more expressive than STRIPS, a recurring problem is knowledge engineering. Most work on GDA has assumed that GDA knowledge is given by the user or domain expert, aside from the work done where expectations and goal formulation are learned via reinforcement learning and where goals are formulated from cases made from Starcraft game traces. To reduce the knowledge engineering burden of creating ontologies, we propose using automated text-extraction tools to build initial ontologies and use case-based learning to refine the ontologies. For self-contained domains such as Starcraft, a significant amount of textual information is available online which could be used to build an ontology. There has been previous work that demonstrated extracting text from the manual for the RTS game Civilization II to create rules based on the strategy described in the manual [2]. Additionally, there are automated tools for creating semantic web ontologies which may be able to be used to create initial ontologies.

2 Research Plan

We plan to manually design an ontology with the help of a domain expert for a GDA agent that can play Starcraft. This will most likely require new GDA algorithms to integrate such an ontology into the GDA agent. We will then use others' strategies for extracting text to automatically build ontologies from text data. Such ontologies will likely contain errors, which we will address with case based learning that will refine the ontology by playing episodes of the game, finding discrepancies, and repairing knowledge errors in the ontology. We expect the GDA agent with the hand crafted ontology will outperform the agents with automatically created ontologies and that the GDA agent with a refined ontology will outperform the agent with the unrefined ontology.

The following is the research plan:

1. Perform a literature overview on GDA research (done)
2. Investigate a test platform for experimentation with RTS games (done)
3. Investigate use of ontologies in the context of GDA (in progress)
4. Implement a system exhibiting the results of (3) and test it in an RTS game
5. Perform a literature review of ontology extraction
6. Investigate automated extraction techniques of ontologies from textual/web information sources
7. Apply (6) on Starcraft strategy repositories
8. Investigate case-based learning techniques to refine ontologies from episodic knowledge
9. Perform an empirical evaluation on Starcraft comparing: GDA with hand-crafted ontologies vs. with automatically extracted ontologies

3 Progress

We have currently looked into taking a case-based reasoning approach to the knowledge engineering burden from an IBM's Watson perspective (using scoring algorithms) for the goal management/selection component of GDA. This relies on the presence of oracles (possibly humans) and uses evidence scoring algorithms to rank goals in order to choose the best one. We are currently working towards creating an expert given ontology for Starcraft for step 3 in our research plan.

References

1. Aha, D.W., Molineaux, M., & Klenk, M. (2011). Goal-driven autonomy. 2011 NRL Review, 164-166.
2. Branavan, S. R. K., David Silver, and Regina Barzilay. "Learning to win by reading manuals in a monte-carlo framework." (2011).

Due to space limitations, relevant GDA references cannot be included. For a brief overview of GDA see [1]

Recommending Research Profiles for Multidisciplinary Academic Collaboration

Sidath Gunawardena

College of Information Science and Technology, Drexel University
3141 Chestnut Street, Philadelphia, PA
sg349@drexel.edu

1 Introduction

Advances in technology can provide access to data sources that present opportunities to apply Case-based Reasoning (CBR) to solve new problems. My research addresses two challenges that may arise in such situations: (1) the available data may only present positive examples and (2) the domain knowledge on negative instances may be incomplete. Both these issues present challenges to CBR systems when trying to learn weights. My research addresses these problems in the context of designing and developing a CBR recommender system for multidisciplinary research collaboration, based on a collection of funded multidisciplinary grant proposals.

This problem differs from that of Expert Location Systems (ELS) as such systems seek to solve a short term knowledge need [3]. The collaboration recommender must consider the combination of the characteristics of all collaborators involved in determining its success. Recommender systems for collaborators in similar disciplines leverage the social networks and the similarity between researchers and research topics [11]. A social network-based approach to recommend multidisciplinary collaborations uses cross-domain topic models, but requires topic models for each domain considered [10].

The recommender system provides an academic researcher with the characteristics of potential collaborators as described by the three features job rank, institution, and research interest. The recommended collaborators complement the characteristics of the researcher such that the resulting collaboration is analogous to previously observed collaborations found in the data. This problem lacks ranking or other metrics avail in other recommendation contexts [2, 9] that can provide a means of determining what constitute a poor recommendation. The two main research challenges to be addressed are applying CBR to a context with only positive instances and of the weak domain theory related to the characteristics of negative instances.

The data contains positive instances only

Typical learning algorithms used in CBR to learn weights require a dense dataset containing both positive and negative instances [1]. When the available data is only positive instances, such as in the case of data on funded grant proposals. This precludes the use of feedback algorithms.

Weak domain theory on negative instances

The literature on collaboration lacks sufficient knowledge that can be systemized to determine what makes a collaboration unlikely to succeed, i.e., a negative example of collaboration.

2 Research Plan

To address the two challenges from the previous section this research has the following goals:

Determine a method for reasoning with positive instances

The data present challenges to supervised learning methods. Unsupervised learning methods can provide a starting point by determining which cases are similar and which are outliers.

Investigate Characteristics of Negative Instances

The available domain knowledge presents a starting point that can be leveraged and combined with machine learning methods. This requires an investigation of machine learning methods suited to data with negative instances, namely Single Class Learning Methods (SCL) [4].

Methodology

1. Determine a case structure and suitable distance measures to facilitate the application of CBR to this problem context
2. Investigate the use of unsupervised methods to determine which cases are more suitable for CBR
3. Leverage this knowledge to facilitate the application of a feedback algorithm to learn from the data
4. Make the grant dataset conducive to the application of SCL methods, and thereby learn the characteristics of potential negative instances.
5. Validate the knowledge learned via a survey of domain experts

3 Research Progress

The following progress has been made to address the two challenges of this research:

Reasoning with Positive Instances

Two approaches have been developed for reasoning with only positive instances. They are based on the tenet that case bases where similar problems have similar solutions are more conducive for CBR. To leverage this, density-based clustering methods are used on the problem and solution spaces of the data to identify cases occurring in high and low density areas. The resulting clusters are used in an approach to learn

feature weights in [5, 6], while in [7] the outlier cases are used as negative instances to allow for the use of feedback algorithms.

Characteristics of Negative Instances

My study of the literature on collaboration [7] serves as a starting point to inform a SCL based approach to learn the characteristics of negative instances of collaboration [8]. These characteristics denote the entire collaboration, and thus are not used to learn weights. Instead they are used to determine if a recommended collaboration is representative of a negative instance and should be rejected. The knowledge learned through this research is currently being validated by surveying practitioners in the field, namely academic faculty with grant review experience.

My Research brings a contribution to the field of CBR by allowing for the application of CBR systems that use weights in a context that was problematic before; one where the cases are only positive instances.

References

1. Aha, D. W.: Feature weighting for lazy learning algorithms. In: Liu H, Motoda H (eds.) *Feature Extraction, Construction and Selection: A Data Mining Perspective*, pp. 13–32. Kluwer, Norwell, MA. (1998)
2. Baccigalupo, C., Plaza, E.: A case-based song scheduler for group customized radio. In *Case-Based Reasoning Research and Development*, pp. 433-448. Springer Berlin Heidelberg. (2007)
3. Becerra-Fernandez, I. Searching for experts on the web: A Review of Contemporary Expertise Locator Systems, *ACM Transactions on Internet Technology*, 6(4), pp. 333-355. (2006)
4. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 213-220. ACM. (2008)
5. Gunawardena S, Weber R. O.: Reasoning with organizational case bases in the absence negative exemplars. In *ICCBR2012: 2nd Workshop on Process-Oriented Case-Based Reasoning*, Lyon, France, pp. 35 – 44. (2012)
6. Gunawardena S. Weber. R. O.: Applying CBR principles to reason without negative exemplars. *FLAIRS* 26. (2013)
7. Gunawardena S, Weber R. O., Agosto D. E.: Finding that special someone: interdisciplinary collaboration in an academic context. *Journal of Education for Library and Information Science*, 51(4), 210-221. (2010)
8. Gunawardena, S. Weber, R. O., Stoyanovich, J.: Learning feature weights from positive cases. *ICCBR 2013* (accepted)
9. Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J.: A case-based solution to the cold-start problem in group recommenders. *Case-Based Reasoning Research and Development*, 342-356. (2012)
10. Tang, J., Wu, S., Sun, J., Su, H. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1285-1293. ACM. (2012)
11. Xu, Y., Hao, J., Lao, R., Ma, J., Xu, W. Zhao, D.: A personalized researcher recommendation approach in academic contexts. *PACIS*. (2010).

Using Ensembles of Adaptations for Case-Based Reasoning

Vahid Jalali

School of Informatics and Computing, Indiana University
Bloomington IN 47408, USA
vjalalib@cs.indiana.edu

1 Introduction

The generic Case-based Reasoning (CBR) process includes retrieving similar past problems, re-using the solutions of the retrieved cases, adapting the solutions if required and possibly incorporating the results into the system's knowledge [1]. The proposed dissertation research studies methods for improving the performance of the case adaptation step, developing a knowledge-light ensemble-based adaptation approach for CBR applied to regression tasks.

For knowledge-rich domains with structure-based case representation adaptation plays a significant role in the success of a CBR system. However, for domains with simpler case representation (e.g. feature-based representation) reliance on simple case adaptation is common. For example, when CBR is applied to regression tasks (e.g. k-NN) the value of a case can be estimated by calculating a distance-weighted average of the values of the top k similar cases. However, other methods that elaborate more on generating and using adaptations for case-based regression have also been introduced and studied [2, 3]. These methods generate adaptations by comparing pairs of cases in the case base and apply appropriate adaptations to the generated solutions for the input queries. For example, for predicting apartment rental prices, if two apartments differ in that one has an additional bedroom, and consequently a higher price, the differences in the prices of those apartments can be used to adapt the price of a different apartment retrieved for predicting the rental price of another apartment with an additional bedroom.

To complement methods focusing on generating adaptation rules, it is possible to focus on improving retrieval. Adaptation-guided retrieval (AGR) methods [4] use adaptation knowledge to retrieve adaptable cases for solution generation. As an example, CABAMAKA system introduced by d'Aquin [5] et al. uses adaptation guided retrieval principle and knowledge discovery techniques from databases (in form of frequent itemset extraction) to retrieve and apply adaptations. However, there are at least two major differences between such an approach to case-based regression versus relying on similarity principal: First, AGR or any method relying on frequent itemset extraction such as CABAMAKA is suited for domains with symbolic or boolean features and consequently when applied to domains with numeric features, discretizing the feature values may result in

information loss; Second, AGR requires domain expert knowledge to connect problem and solution features through adaptation knowledge which makes the whole process in the best case semi-automated which is in contrast to the premise of the knowledge light approaches (e.g. [2, 3]) that can be fully automated.

My dissertation research studies augmenting case-based regression with adaptations based on ensembles of rules (which is proven to increase accuracy [6]) generated from various types of case comparison. The following are the issues that should be considered in case-based regression using the case comparison heuristic for generating adaptations, and which I am addressing in my research:

- Determining cases that are worthwhile to compare which includes:
 - determining/considering confidence in the values of the adaptations' composing cases . For example, if a case is incorporated to the case base as the result of solving a new problem, the system's confidence in the estimated solution for that case should be considered if that case is later used for generating adaptation rules
 - pre-computing of adaptations versus using an on-demand lazy method for generating them
- Retrieving cases and adaptations for building solutions:
 - Modifying case representation. This includes deciding about the features to be used and their values (e.g. normalizing/standardizing feature values).
 - Similarity measures to be used for retrieving cases and adaptations.
 - As for AGR methods, in case-based regression it is possible that the most similar cases to the input problem are not the best cases to build the solution from. For example, it is possible that no quality adaptation rule exists for adapting the solution of a highly similar case to the input problem while for a relatively dissimilar case there exist more accurate adaptations that overall can yield a better result.
 - Number of cases and adaptations to be used for building a solution.
 - Considering the context of the input problem, cases to adapt and the composing cases of adaptations in the regression process. Because the same changes in the problem specification part of cases can affect the solutions differently based on the context. For example, in the real estate domain, the value of a property may increase differently by adding an additional bedroom depending on its location

2 Research Plan

I am studying the potential of CBR for applications such as regression in domains with simpler case representation in which the role of adaptation is often neglected. I am specially interested in finding automatic knowledge-light methods for generating adaptations from case comparison and using those adaptations in adjusting the values of the retrieved cases to adapt.

The key hypothesis that I am exploring in my research are finding methods to generate an optimal number of adaptations by comparing a limited pairs of

cases and studying retrieval methods for both cases and adaptations that can yield accurate estimations with low computational cost. All issues listed at the end of the previous section are considered for answering these questions in my research.

3 Progress

The following are what I have accomplished so far:

- Studying the role of neighborhood selection in case-based regression [7]
- Implementing EAR (Ensemble of Adaptations for Regression), a case-based regression system that generates adaptations from case comparison and uses an ensemble of adaptations for adjusting the case values
- Identifying and testing different neighborhood selection methods for selecting cases to adapt
- Identifying and testing different neighborhood selection methods for generating adaptations
- Comparing the performance of EAR versus other machine learning methods
- Primary experiments aimed at studying the effect of confidence in the values of cases on the performance of EAR.
- Identifying the data characteristics that determine the efficiency of EAR.

References

1. Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision, and retention in CBR. *Knowledge Engineering Review* **20**(3) (2005)
2. Hanney, K., Keane, M.: The adaptation knowledge bottleneck: How to ease it by learning from cases. In: *Proceedings of the Second International Conference on Case-Based Reasoning*, Berlin, Springer Verlag (1997)
3. McSherry, D.: An adaptation heuristic for case-based estimation. In: *Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning. EWCBR '98*, London, UK, UK, Springer-Verlag (1998) 184–195
4. Smyth, B., Keane, M.: Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence* **102**(2) (1998) 249–293
5. d’Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Case base mining for adaptation knowledge acquisition. In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, San Mateo, Morgan Kaufmann (2007) 750–755
6. Wiratunga, N., Craw, S., Rowe, R.: Learning to adapt for case-based design. In: *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning. ECCBR '02*, London, UK, UK, Springer-Verlag (2002) 421–435
7. Jalali, V., Leake, D.: An ensemble approach to instance-based regression using stretched neighborhoods. (In press)
8. Bay, S.D.: Combining nearest neighbor classifiers through multiple feature subsets. In: *Proc. 15th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA (1998) 37–45

Workflow extraction from textual process descriptions

Pol Schumacher

Goethe University Frankfurt - Institute for Computer Science - Business Information
Systems
D-60325 Frankfurt am Main, Germany
`schumacher@cs.uni-frankfurt.de`

1 Introduction

For my thesis I investigate the area of workflow extraction from textual process descriptions. Workflow extraction is the transformation of a process description formulated in natural language into a formal workflow model. There are two main questions I want to investigate. The first one is if the quality of the extracted workflow is high enough to use them for retrieval, user-guidance etc. and what are the limitations. The second questions is how transferable the extraction techniques developed, what needs to be adapted to extract workflow of a new domain.

Recently Process oriented Case-Based Reasoning (POCBR) emerged and approaches to handle procedural knowledge were developed [1, 2]. My work is part of a joint project of the University of Trier and the Goethe University Frankfurt. Whereas our partner group is investigating novel retrieval approaches my work is mainly located in the area of case acquisition by means of workflow extraction.

A lot of how-to communities raised in the internet [3]. People write and share how-tos in those communities. A how-to is an instruction to perform a certain task, similar to an instruction in a manual. These how-tos describe a process and contain therefore procedural knowledge. Unfortunately this knowledge is stored in natural language. Current approaches to use this knowledge e.g. for retrieval or automatic adaptation process these texts as general texts and do not take into account their procedural character.

A workflow is a set of activities which are ordered by a control-flow. A control-flow can be parallel, disjunctive, or repetitive. An activity can have a set of input- and output-products. Input products are consumed and output products are produced by a task. Workflow extraction can be divided into three phases. In the first phase we employ standard NLP software to perform a linguistic analysis. In the second phase we try to identify the different elements of a workflow. At the end we try to build the control- and data-flow. The data-flow defines the flow of products or information through the workflow.

Workflow extraction faces several challenges. First, textual descriptions of processes are frequently incomplete. People often omit certain details because they can be inferred with implicit knowledge. Unfortunately it is impossible

to capture all this implicit knowledge. A second problem relates to the type of content which we are processing, the user generated content. This content contains more grammatical and orthographical errors than authored content (e.g. newspaper articles). One of the main challenges in workflow extraction is the evaluation. Due to different granularities and the paraphrasing problem, it is necessary to employ a human expert for the evaluation which makes it expensive. The paraphrasing problem is the problem that the same process can be described by different workflows. The granularity problem is the problem of handling the different levels of abstraction which can be used to formalize a process using a workflow.

The automatically extracted workflows can be used in different scenarios. A workflow execution engine can be used to execute the workflows, they can be used as knowledge for reasoning and they can support the evaluation of new reasoning approaches. The first domain which is investigated is the domain of cooking as it is frequently used in artificial intelligence research. A second domain which is investigated is the domain of computer how-tos. A third domain is strived but not yet finally decided.

2 Research plan

Current forms of procedural knowledge: It is necessary to get an overview about existing communities. Which communities exists and what are the domains. Another point is to determine under which form these descriptions are published and if they can be retrieved automatically.

Build repository with textual description: For the development and the evaluation we need a repository of process descriptions for at least two domains.

Experiment with different nlp tools: There exists different different types of nlp software. Before developing the prototype the appropriate software must be determined.

Build prototype: The prototype is used to generate workflows for evaluation. In addition it can be used to create test repositories for our partners in the project.

Evaluate prototype: The prototype is evaluated in the cooking domain. In a second step the software is adapted to another domain and evaluated again. Are there any differences in the result and how big is the effort to adapt the software to the new domain.

Analyse Paraphrasing: We need to identify cases of paraphrasing in real textual description and analyse the impact on workflow. In addition we need to develop an approach to handle it.

Hierarchical workflow representation: Improve the current workflow representation to handle different granularities.

Create workflow using a planning approach: Developing a planning domain for cooking and build workflow by handling them as planning problem. This approach allows to create a second independent corpus of workflows which might be used to evaluate the extraction process.

Workflow simulation test bench: Develop a system which can simulate the execution of workflows and introduce noise(e.g. missing ingredients) in the execution.

Compare applicability of the two corpora for evaluation: Analyse the differences between an evaluation with extracted workflows and an evaluation workflows generated by planning. Can both corpora be used for the same kind of evaluations?

3 Progress

- Several how-to communities were investigated. For two communities the how-tos were crawled and transformed to an easy to handle xml format. The first community was the cooking community allrecipes.com with about 37 000 cooking recipes. The second one was the general purpose community wikihow.com with about 140 000 how-tos.
- Two prototypes with reduced functionality were evaluated. One was built using GATE as NLP tool and the other one using SUNDANCE as NLP tool. The software which used SUNDANCE performed better [4].
- A framework for workflow extraction was developed on top of the SUNDANCE NLP tool. The framework is flexible and allows to adapt the software for a new domain. A prototype for the cooking domain was developed.
- Two evaluation were performed with the cooking prototype. In the first evaluation we investigated if incomplete workflow can be used as modelling help for workflow modellers, which seems to be the case [5]. The second evaluation was focused on the data-flow and showed that the construction of complete data-flow is a complex task which needs a lot of effort.

References

1. Minor, M., Montani, S., Recio-García, J.A.: Editorial: Process-oriented case-based reasoning. Information Systems (In Press)
2. Dufour-Lussier, V., Le Ber, F., Lieber, J., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. Information Systems (In Press)
3. Plaza, E.: On reusing other people’s experiences. *Künstliche Intelligenz* **9**(1) (2009) 18–23
4. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: Workshop Proceedings: WWW’12, Lyon, France (2012)
5. Schumacher, P., Minor, M.: Hybrid extraction of personal workflow. In: Konferenzbeiträge der 7. Konferenz Professionelles Wissenmanagement, Passau, Germany (2013)

A Case-Based Reasoning Approach to Text Generation

Josep Valls-Vargas

Computer Science Department
Drexel University
Philadelphia, PA, USA 19104
josep@valls.name

1 Introduction

Natural Language Generation (NLG) is one of the longstanding problems in Artificial Intelligence. Natural language text generation can be described as a multi-step process that could be generalized as a processing pipeline with the following stages [9]: 1) *Document planning* determines the content and structure of the document; 2) *Microplanning* decides words and syntactic structures to communicate the previously defined content; 3) *Surface realization* maps internal structures into actual text embodying all decisions related to the grammar and morphology of a particular language. There has been much work done on the first two stages of the NLG pipeline, but, regarding surface realization, existing work tend to rely on annotated templates or rule-based systems. Text generation systems may use annotated templates with gaps that are filled by the system with the underlying information that needs to be conveyed [5].

In my work, I have been exploring an alternative approach, inspired in CBR ideas, based on automatic text adaptation. The main idea is that a given piece of text can be automatically modified by the system to convey the desired information. This method provides great advantages, the most prominent being a simplification or removal of the template generation and annotation processes, but at the same time poses additional problems.

There are several areas that may limit the performance of such a system:

1. **Text Retrieval:** Surface realization through text adaptation starts from an existing text snippet. The original text snippet to be adapted needs to be able to accommodate the content and structure defined in the previous stages of the NLG pipeline.
2. **Text Adaptation:** Token replacement techniques can effectively modify an existing text snippet but in order to ensure proper text syntax and coherence (i.e. verb tense or noun-pronoun coordination) higher level transformations may be required.
3. **Limitations of NLP:** Natural Language Processing (NLP) techniques can be used to automatically parse unannotated text into structures the system can later use for NLG. This shifts the burden of manual annotation of the text from expert to the system but the output of state-of-the-art NLP systems is not yet reliable.

2 Research Plan

The goal of my research will be to address the three tasks identified in the previous section. By addressing these issues I expect to be able to develop new frameworks for NLG and surface realization that do not depend on annotated templates. Some benefits of using readily available corpora and NLP techniques instead of hand-crafted templates would be language-agnostic and domain-independent systems, among others.

Specifically I would be interested in researching the following areas:

1. **Textual CBR (TCBR)**: There has been an increased interest in the TCBR community in the past few years to seek and go beyond retrieval, although these are still the exception [4]. Two representative efforts in this line are the CR2N system [1], and some recent work in jCOLIBRI [8]. A major component on my research is to develop a general-purpose framework for text adaptation. The motivation of this is to provide any system to request arbitrarily modifications on text snippets suitable for a variety of tasks. The framework should contain algorithms that are general in nature and do not require specific domain or language information encoded and can work using readily available NLP tools and a set of provided examples (or another form of corpora to learn from) to automatically build the cases.
2. **High Level NLP**: NLP is usually broken into several sub-tasks. Some of the tasks in the Information Extraction (IE) pipeline [2] annotate text with syntactic structures. Systems such as Part-Of-Speech (POS) taggers [6] or syntactic parsers [10], capable of learning models from generic corpora are readily available. TCBR systems operating on the output of these systems could work at a higher level and enable disambiguation of complex text structures, conflict resolution, interoperability between systems and language-agnostic processing. Other CBR techniques could also be used at this level in order to combine or convert from other knowledge sources (i.e. writing style or symbolic commonsense knowledge).
3. **Automatic Text Evaluation**: Another component I would like to work on and that would become an essential part of a CBR cycle (for the *Revision* step) would be automatic text evaluation. Text evaluation for text adaptation tasks should ensure that a processed output conveys the desired information and that the generated text is grammatically correct and semantically coherent. I plan on using statistical and probabilistic language models that could be learned automatically from provided examples (or another form of corpora).
4. **Computational narrative**: I would like to focus on the specific domain of computational narrative and computer games. Although CBR has been used for plot generation [3], fiction and narrative pose specific particularities for NLP and NLG (i.e. anthropomorphism, metaphors or subjective morality). Text-based computer games provide an interesting testbed and an opportunity to explore text processing from a different perspective. Also, a concrete, simplified domain will help assessing the system in the beginning before trying to generalize it.

3 Progress

The following progress has been made on my proposed research:

- **Natural Language Generation through Case-based Text Modification:** We presented CEBETA [11], a case-based approach to text modification capable of several number, gender and tense transformations on a given sentence. In our system, pairs of plain text sentences implementing specific transformations represent cases. NLP tools are used to automatically process the cases and infer the required text transformation routines.
- **Interactive Narrative:** I integrated a version of CEBETA into RIU, an interactive narrative system [7]. RIU uses computational analogy to find mappings between scenes and then performs replacements for the matched tokens in the textual description of the scenes. My module performed the requested replacements plus any other that would be required in order to maintain grammatical correctness and coherence.

References

- [1] Adeyanju, I., Wiratunga, N., Lothian, R., Sripada, S., Lamontagne, L.: Case retrieval reuse net (CR2N): An architecture for reuse of textual solutions. In: IC-CBR. pp. 14–28 (2009)
- [2] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02) (2002)
- [3] Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on cbr. In: Macintosh, A., Ellis, R., Allen, T. (eds.) Applications and Innovations in Intelligent Systems XII, pp. 33–46. Springer London (2005)
- [4] Gervás, P., Hervás, R., Recio-García, J.A.: The role of natural language generation during adaptation in textual cbr. In: Workshop on Textual Case-Based Reasoning: Beyond Retrieval, in 7th International Conference on Case-Based Reasoning (ICCBR07). p. 227235. Northern Ireland (08/2007 2007)
- [5] Hovy, E.H., Arens, Y.: Automatic Generation of Formatted Text, vol. Readings in intelligent user interfaces (1998)
- [6] Marie-Catherine de Marneffe, B.M., Manning, C.D.: Generating typed dependency parses from phrase structure parses. vol. LREC (2006), <http://nlp.stanford.edu/software/lex-parser.shtml>
- [7] Ontañón, S., Zhu, J.: Story and Text Generation through Computational Analogy in the Riu System. In: AIIDE. pp. 51–56. The AAAI Press (2010)
- [8] Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Textual cbr in jcolibri: From retrieval to reuse. In: Wilson, D.C., Khemani, D. (eds.) Proceedings of the ICCBR 2007 Workshop on Textual Case-Based Reasoning: Beyond Retrieval. pp. 217–226 (August 2007)
- [9] Reiter, E., Dale, R.: Building Natural Language Generation Systems (2000)
- [10] Schmid, H.: Probabilistic part-of-speech tagging using decision trees (1994), <http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf>
- [11] Valls-Vargas, J., Ontañón, S.: Natural language generation through case-based text modification. In: Case-Based Reasoning Research and Development, pp. 443–457. Springer (2012)

Towards an Artificial Teammate for Supporting and Conducting Arguments with Analogies and Cases in Biologically Inspired Design

Bryan Wiltgen

Design & Intelligence Laboratory, School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA, USA
bryan.wiltgen@gatech.edu

1 Introduction

When you are on a team with others and you want to persuade your teammates about something, what do you do? Among other things, you may argue for your opinion using an analogy or a case. Although they do not use the term case, Ramage et al. [2] refer to these kinds of arguments as “Resemblance Arguments” (p. 238). There are two types of these arguments: “1. **Arguments by analogy**, in which the arguer likens one thing to another by using a metaphor or imaginative comparison” (p. 238, emphasis theirs), and “2. **Arguments by precedent**, in which the arguer likens a current or proposed event or phenomenon to a previous event or phenomenon” (p. 238, emphasis theirs).

I am studying this topic in the context of collaborative, conceptual, biologically inspired design. Biologically inspired design (a.k.a., BID) is a design methodology where designers take inspiration from nature to aid their designing. By collaborative, I mean that a team conducts BID. By conceptual, I mean the designing occurs during the conceptual phase of design.

I intend to explore the relationship between the knowledge and perspectives held by a person (and/or between people) and arguments involving analogies and/or cases. For example, to what extent do the knowledge and perspective differences between people and what those people know about each other’s knowledge and perspectives influence resemblance arguments? Differences may be large between teammates with different cultures (e.g., different professional disciplines) because their knowledge and perspectives are less likely to intersect than between people of the same culture. Therefore, BID represents an interesting context for this topic because, at least in a class where we have studied BID, BID involves interdisciplinary design teams.

In addition, I would like to explore the relationship between design (via biologically inspired design) and arguments involving analogies and cases. For example, what are the strengths, weaknesses, and/or potential dangers associated with making resemblance arguments in BID?

I also intend to develop an interactive, software technology called CICADA, which stands for the Collaborative and Interactive Computer Assistant for Design with Analogies. I will describe CICADA in sub-section 2.2.

2 Plan, progress, and remaining work

In this section, I briefly describe the parts of my research. For each part, I describe the plan for that part, the progress I have made, and the projected remaining work for that part.

2.1 Explore resemblance arguments in collaborative, conceptual BID

I plan to develop an information-processing model of resemblance arguments in collaborative, conceptual BID. Although I have not yet settled upon the specifics of my model or its particular goals, I intend for my model to address research questions like the following. (1) What are the contents of the mental models of design teammates engaged in argumentation using resemblance arguments before, during, and after the argument? (2) How does the design change because of the argument? (3) How does the nature of the analogy (e.g., the concreteness of the source analog, or the kind of inference(s) claimed) impact the argument and its outcome(s)?

I plan to develop my model using a combination of two methods. For my first study, I will conduct qualitative, discourse analysis of data collected through participant-observation in a student BID team during a class on the topic. Afterwards, I will conduct laboratory studies to continue my investigation of this topic. I already have conducted the participant-observation, collected the data, and started preliminary analysis for the first study. However, much work remains to analyze this data. I have not begun serious development of the second study.

2.2 Develop an artificial teammate for collaborative, conceptual BID

I plan to develop a web-based, interactive, software tool called CICADA. CICADA will serve both as a research platform to help me investigate resemblance arguments in this context and as a kind of artificial teammate, aiding human designers in their designing. As of now, I intend for CICADA to consist of at least three modes: Workspace, Modeler, Version Tracker, and Arguer.

Workspace mode will allow the user to investigate all the models and model versions associated with each team to which a user belongs, and it will provide access to CICADA's other modes.

Modeler mode will support knowledge modeling. Upon saving a model (either a new model or an edited model), a user will be required to justify their new version. Only one user at a time will be able to work on a particular model.

Version Tracker mode will allow users to compare temporally adjacent versions of a model and view the justification made for going to an initial version or for going from the old version to the new version.

In Arguer mode, CICADA will suggest new model versions on its own and suggest new model versions based upon some input (e.g., a desired change, such as reducing energy consumption). In both cases, CICADA will include justification for the suggested new model version. To do these things, CICADA will compare the current model and the input (if given) with past adjacent model versions, the differ-

ences in past adjacent versions, and their justifications. CICADA will draw from previous modeling done using it, including modeling done from users outside of the current user's team.

3 Expected contributions

In this section, I describe what I expect my research to contribute.

To my knowledge, my information-processing model will be the first such model in this context to focus specifically on collaboration. I project that my model will (1) inform the design of CICADA and (2) improve our understanding of resemblance arguments in collaborative, conceptual BID and, in turn, improve our understanding of collaboration in collaborative, conceptual BID. Improved understanding of resemblance arguments and/or collaboration in this context may inform its teaching and practice. For example, imagine that my analysis reveals that arguments by analogy are too persuasive and may lead teams to make erroneous decisions when used. Upon learning about this finding, BID educators may then develop instruction to encourage designers to be skeptical of arguments by analogy.

CICADA will contribute to designers in collaborative, conceptual BID as a prototype tool that they may use for their designing. In addition, I plan for CICADA to aid in the second study that I plan to do to continue investigating resemblance arguments in collaborative, conceptual BID. CICADA also extends and applies in a new context ideas from a prior work [1] in automated analogical design.

Acknowledgments

I thank the reviewers for their feedback. I also thank Ashok K. Goel, Cherish Weiler, Michael Helms, Swaroop Vattam, David Majerich, other members of DILab, Jeanette Yen (the coordinator of the MS/ISyE/MSE/PTFe/BIOL 4740 class), and the anonymous student design team from my first study. I am grateful to the US National Science Foundation for funding through the NSF CreativeIT grant (#0855916) titled "Computational Tools for Enhancing Creativity in Biologically Inspired Engineering Design."

References

1. Bhatta, S., Goel, A.: Learning Generic Mechanisms for Innovative Strategies in Adaptive Design. *The Journal of the Learning Sciences*, 6(4), pp. 367-396 (1997)
2. Ramage, J.D., Bean, J.C., Johnson, J.: *Writing arguments: a rhetoric with readings*. 9th edition. Pearson Education, Inc. (2012)

Scientific Reproducibility

A Reproducibility Process for Case-Based Reasoning

David W. Aha¹ and Odd Erik Gundersen²

¹Navy Center for Applied Research in AI;
Naval Research Laboratory (Code 5514); Washington, DC 20375

²Verdande Technology; Trondheim, Norway
david.aha@nrl.navy.mil | odderik@verdandetechnology.com

Abstract. The reproducibility of empirical studies is a cornerstone of a mature research discipline, as exemplified in the hard sciences. It is also gaining a foothold in computer science. We seek to raise awareness of its importance for case-based reasoning (CBR) research. In this paper, we briefly describe motivations for encouraging reproducible CBR research and the process that we propose to assess the reproducibility of volunteered studies to be presented at ICCBR 2013.

Keywords: Case-based reasoning, reproducibility, empirical studies

1. Introduction

Science can advance through the development, testing, verification, and substantiation of theoretical claims. But while the first three tasks can be conducted by the originators of a research contribution, the responsibility for substantiation must be shouldered by their peers, and requires a concerted effort. In the case-based reasoning (CBR) research community, as in many others, the results of an investigation are published based solely on the researchers' description of their work, which must be sufficiently convincing to the reviewers. However, it is not the reviewers' task to *confirm* a study's conclusions by reproducing their investigation. In fact, reproduction of published results is rarely attempted, and in cases where it is attempted, the outcome is seldom communicated. This is worrisome; what if a paper's conclusions cannot be confirmed, or can be rejected? Given that this has happened to a Nobel Laureate (NY Times, 2010) (albeit the results concerned were unrelated to the prize-winning research) perhaps we are all at risk unless some reproducibility process provides us with objective feedback, preferably *before* we publish our work.

We introduce a reproducibility process for CBR research. We provide some background in §2, describe the process we plan to enact for the 2013 International Conference on CBR¹ in §3, and attempt a brief call to arms in §4.

¹ <http://www.iccbr.org/iccbr13>

2. Motivations for Reproducibility

The topic of reproducible research is not a new idea. In fact, it is an essential part of the scientific method (Descartes, 1637). Reproducibility means that researchers at one laboratory can independently replicate and confirm the results found by a group at another laboratory. It is a hallmark of good science, and has long been encouraged in many natural science disciplines. *Science*² and other leading journals require authors to provide supplementary materials that will permit other researchers to replicate their study. For example, Molineaux, Thach, and Aha (2008) describe how they replicated a study published by Sachs et al. (2005) in *Science* on using a Bayesian network learning algorithm to model a protein-signaling network. The materials they used included data and multiple software components left behind by Sachs et al., but the task was still surprisingly difficult, and they required extensive discussions with Sachs to identify details that were already partly lost to memory. Likewise, replicating studies published in the CBR literature can be challenging.

This topic has recently been given greater attention in computer science. For example, since 2008 the SIGMOD conference series features an Experimental Reproducibility effort³ whose goal is to ensure that conference papers are “reliable”, and whose premise is that experimental papers are “most useful when their results have been tested and generalized by objective third parties”. The Very Large Data Bases Conference series adopted this effort beginning in 2013. Several journals require authors to ensure that their results are reproducible (e.g., *IEEE Transactions on Signal Processing*, *Biostatistics*) (Freire *et al.*, 2012). Competitions⁴ have been held to develop tools for assisting with creating “executable” papers. Several meetings have been held on scientific reproducibility, such as the 2011 ICIAM Workshop on Reproducible Research⁵ and the 2012 Workshop on Reproducibility and Experimental Mathematics⁶. Some institutions (e.g., ETH Zurich) post guidelines for their researchers on how to make their work reproducible, and journal special issues have been devoted to reproducibility (e.g., (Fomel & Claerbout, 2009)).

Reproducibility is increasingly sought because it offers many benefits to researchers and their community. Some of these include:

Public Confirmation: Reproducibility committees acknowledge publications whose experiments are reproducible. This may positively impact the reputations of a researcher, their institution, and their research field more generally.

Increased Citations: Initial data suggests that publications which include reproducible experiments may have higher impact and visibility than others (Vandewalle *et al.*, 2009), although this claim requires more analysis.

Standards for Reproducible Research: Encouraging reproducibility in a research community will yield the following byproducts:

- Guidelines on how to conduct and report reproducible empirical studies;

² www.science.org

³ <http://www.sigmod.org/2012/reproducibility.shtml>

⁴ <http://executablepapers.com/>

⁵ <http://www.stodden.net/AMP2011/>

⁶ <http://icerm.brown.edu/tw12-5-rcem>

- Repositories of data and software used in experimental studies; and
- Tools (e.g., COLIBRI Studio (Recio-García *et al.*, 2012)) to assist authors with conducting reproducible studies.

Even repositories alone, without connection to specific investigations, can dramatically impact a research field by simplifying the task of conducting comparative studies on common data sets. For example, this has been demonstrated in the machine learning (ML) community by the UCI Repository of Machine Learning Databases (Frank & Asuncion, 2010), Weka (Hall *et al.*, 2009), scikit-learn (Pedregosa *et al.*, 2011), and mloss⁷. While easily accessible repositories may also cause unintended side-effects (e.g., an imbalanced emphasis on easily performed empirical studies with little scientific impact (Langley, 2011)), they can assist with conducting reproducibility studies, which is a goal of MLcomp⁸.

Faster Progress: The documents, data, and software corresponding to reproducible investigations may serve as building blocks for subsequent research. Newcomers and students may be able to more quickly come up to speed in their familiarity with the community’s methods of practice and software infrastructure. This may also increase the clarity of an investigation’s contributions, thus reducing the frequency of publishing redundant studies.

Barriers to reproducibility also exist (Stodden, 2010). Unlike the community-focused motivations for encouraging reproducibility, barriers primarily focus on personal concerns. For example, some barriers to sharing code and data include:

- The time required for documentation and cleansing
- Responding to questions concerning them
- Not receiving attribution
- Patenting considerations
- Legal barriers (e.g., copyright)
- The time required to solve privacy concerns
- Potential loss of future publications and competitor advantage

Stodden argues that some of these barriers can be addressed through cultural change, such as when institutions or research communities require investigators to follow procedures to ensure their investigations are reproducible. If this could be accomplished for CBR research, then several of the aforementioned benefits may be enjoyed by the CBR community. However, concerns about reproducibility in the CBR literature are in their formative stages; we need to take small steps initially, retrieving and adapting reproducibility processes from other communities, and revising them as needed.

Towards this goal, we describe a modest process that we will enact for ICCBR 2013. The experience gained by executing this process can be used to improve a reproducibility process for future meetings. If successful, then similar but improved processes may be used in future meetings.

⁷ <http://www.mloss.org> (Machine Learning Open Source Software)

⁸ <http://www.mlcomp.org> (Machine Learning Comparisons)

3. Initial Reproducibility Process

We will conduct a process for confirming the results of voluntarily-provided empirical studies reported at ICCBR 2013. Our broad goals concern:

- **Scientific Quality:** To promote good practices in research and reporting.
- **Trust:** To increase the confidence that reported results are accurate and provide assurance to others that they can apply the studied methods.
- **Standards:** To clarify how studies can be designed to enhance reproducibility.
- **Efficiency:** To reduce the frequency of redundant empirical research efforts.
- **Examples:** To highlight studies of CBR research that has been reproduced.
- **Growth:** To establish a dialogue and gather experience on this topic within the CBR community.

The steps of the process that we will follow, which are also posted on the conference web site, are described next.

3.1 Preparation

First, we will form a small Reproducibility Committee (RC) whose members can assist us in this process, either by serving as consultants or by assisting with attempts to acquire materials and repeat experiments on their machines. The RC will collaboratively discuss the reproducibility process to be pursued, and refine it as needed.

3.2 Solicitation

After decisions are made on which submissions are accepted for presentation at ICCBR 2013, we will contact the authors of accepted papers, explain the objectives of the RC, and invite (but *not* mandate) them to submit their papers (if they describe empirical studies) for reproducibility testing. Our objective is to interest a few groups of authors to submit their studies.

3.3 Execution

For each set of authors who volunteer, we will assign a member of the RC who will acquire all materials required for replication (e.g., data, scripts, system code, analysis code, and documentation). Authors will be asked to provide details of their empirical study, any software or hardware constraints (e.g., operating system, licensing, processor), and any constraints on sharing their materials (i.e., some tests may require signing non-disclosure or alternative agreements).

Given this information, the RC will then identify which studies can be feasibly replicated (prior to the conference), and by which RC members, with careful attention to prevent conflicts of interest (i.e., RC members will not be permitted to participate in conducting reproducibility tests for studies in which they participated or for which they would be biased). With input from the RC, the Reproducibility Committee Co-chairs will then assign two or more RC members to replicate each of the selected studies. The co-chairs will also record explanations on their selection of studies and assignments.

During the testing period, the co-chairs will monitor and manage the process, obtaining additional RC members and reassigning studies as needed. They will provide the RC with a guide on how to conduct and assess the results of reproducibility testing. RC members will complete a short (unpublicized) report when reporting their results, and discuss these with the co-chairs.

When testing is completed, the co-chairs will discuss the report and findings for each reproducibility attempt with the corresponding set of volunteer authors, and address any concerns that may arise. The authors will be asked if they permit the RC to publicize information on their reproducibility attempt. If the authors give their consent, then the process described in §3.4 will include discussion pertinent to their study. However, if they decline, then our reproducibility attempt for their work will not be publicized.

For consenting authors, the RC will provide awards in the following categories:

- *Reproducible*: This means that the RC succeeded in reproducing the central results of the submitted study.
- *Shareable*: This means that the experimental components of the study have been made available to the community, with a URL indicated on the ICCBR 2013 web site. While we anticipate that some (e.g., academic) studies can be shared, others (e.g., from industry) may not be shareable due to concerns of IP, security, etc.

3.4 Dissemination

For only those sets of authors who consented, we will publicize and disseminate information on this process in three ways. First, the RC will document this process in a short report that will be posted on the ICCBR 2013 web site. It will include explanations of any study components that the RC succeed or failed to reproduce, and describe lessons learned and recommendations for future CBR reproducibility efforts.

Second, we will present a poster at the conference that summarizes the objectives and results of the RC. Our objectives will be to advertise and obtain feedback on this process (e.g., suggestions for future efforts).

Finally, we will give a short plenary presentation at the conference, in which we will describe the RC's objectives, process, and results, and announce the awards.

4. Conclusions

*“Reproducibility of carefully documented experiments is a cornerstone of the scientific method, and yet is often lacking in computational mathematics, science, and engineering. Setting and achieving appropriate standards for reproducibility in computation poses a number of interesting technological and social challenges.”*⁹

Empirical studies in CBR are essential and should be reproducible, as this would have many benefits for the community. When feasible we should share the materials of our published empirical studies (e.g., documentation, data, scripts, system code, analysis code, and methods for generating results summaries), packaged to allow others to

⁹ <http://icerm.brown.edu/tw12-5-rcem>

objectively reproduce our results. However, this vision would require substantial preparation, including standardizing methods for sharing and documenting these materials.

We outlined a modest, voluntary process for assessing the reproducibility of empirical studies reported at ICCBR 2013. It is meant to be a “proof-of-concept” attempt; lessons learned could be used to improve future such processes.

Reproducibility is not easily achieved, even in more established communities; a recent study found that only 11% of 53 studies on preclinical cancer research were reproducible (Begley & Ellis, 2012). Yet we should address reproducibility challenges head-on; discovering the reasons for non-reproducible studies (e.g., due to erroneous data assumptions, faulty software or procedures, or ambiguously worded claims) could serve as cases for others to learn from. We encourage the community to create an infrastructure for reproducible research (e.g., standards, repositories, evaluation tools).

Acknowledgements

Thanks to NRL and Verdande Technology for supporting this research, and to ICCBR 2012 invited speaker Yolanda Gil for providing inspiration. Thanks also to David McSherry for his thorough comments on earlier drafts. Finally, thanks to the ICCBR 2013 Co-Chairs, Santiago Ontañón and Sarah Jane Delaney, for allowing us to conduct this process at ICCBR 2013. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or the DoD.

References

- Begley, C.G., & Ellis, L.M. (2012). Drug development: Raise standards for preclinical cancer research. *Nature*, **483**, 531-533.
- Descartes, R. (1637). Discourse on the method for reasoning well and for seeking truth in the sciences.
- Fomel, S., & Claerbout, J. (2009). Guest editors' introduction: Reproducible research. *Computing in Science Engineering*, **11**(1), 5-7.
- Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository. [archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- Freire, J., Bonnet, P., & Shasha, D. (2012). Computational reproducibility: State-of-the-art, challenges, and database research opportunities. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 593-596). Scottsdale, AZ: ACM Press.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, **11**(1). [www.cs.waikato.ac.nz/ml/weka]
- Langley, P. (2011). The changing science of machine learning. *Machine Learning*, **82**(3), 275–279.
- Molineaux, M., Thach, D., & Aha, D.W. (2008). *Further derivations of causal protein-signaling networks* (Technical Report AIC-08-004). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

- NY Times (2010). Nobel laureate retracts two papers unrelated to her prize. [www.nytimes.com/2010/09/24/science/24retraction.html?_r=1&emc=eta1].
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830. [scikit-learn.org/stable]
- Recio-García, J.A., Díaz-Agudo, B., & González-Calero, P.A. (2012). Reproducibility of CBR applications in COLIBRI. In *Proceedings of the Seventeenth UK Workshop on Case Based Reasoning*. Brighton, UK: Springer.
- Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., & Nolan, G. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**(5721), 523-529.
- Stodden, V. (2010). *The scientific method in practice: reproducibility in the computational sciences* (Working Paper 4773-10). Cambridge, MA: Massachusetts Institute of Technology, Sloan School of Management.
- Vandewalle, P., Kovacevic, J., & Vetterli, M. (2009). Reproducible research in signal processing: What, why, and how. *IEEE Signal Processing Magazine*, **26**(3), 37-47.

Sponsors



General Electric



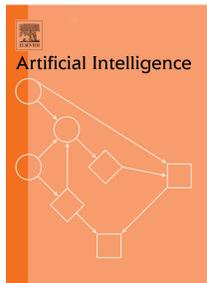
Association for
the Advancement of
Artificial Intelligence



Knexus Research
Corporation



Empolis
Information Management



Artificial Intelligence
Journal



Drexel University