# ICCBR 2018


The 26th International Conference on Case-Based Reasoning

July, 09th-12th 2018 in Stockholm, Sweden,

## Workshop Proceedings

Mirjam Minor (Editor)

# Table of Contents

**Doctoral Consortium**

organized by Cindy Marling and Antonio A. Sánchez-Ruiz

# XCBR: First Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems.

Workshop at the
26th International Conference on
Case-Based Reasoning
(ICCBR 2018)

Stockholm, Sweden
July, 2018

Belén Díaz-Agudo,
Juan A. Recio-García,
David W. Aha

**Co-Chairs**

Belén Díaz-Agudo
University Complutense of Madrid, Spain

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

David W. Aha
Naval Research Laboratory, USA

**Programme Committee**

Derek Bridge, University College Cork, Ireland
Pedro A. Gonzlez Calero, UCM, Madrid
Stelios Kapetanakis, University of Brighton, UK
David Leake, Indiana University, USA
Hector Muoz-vila, Lehigh University, USA
Santiago Ontan, Drexel University, USA
Lara Quijano, University Carlos III of Madrid, Spain
Antonio A. Snchez Ruiz-Granados, UCM, Spain
Barry Smyth, University College Dublin, Ireland
Ian Watson, University of Auckland, New Zealand

# Preface

The success of Artificial Intelligence (AI) has led to an explosion of the generation of new autonomous systems with new capabilities like perception, reasoning, planning and acting. Despite the tremendous benefits of these systems, they work as black-box systems and their effectiveness is limited by their inability to explain their decisions and actions to human users. The problem of explainability in Artificial Intelligence is not new but the rise of autonomous intelligent systems has created the necessity to understand how these intelligent systems achieve a solution, make a prediction or a recommendation or reason to support a decision in order to increase users reliability in these systems. Additionally, the European Union included in their regulation about the protection of natural persons with regard to the processing of personal data a new directive about the need of explanations to ensure fair and transparent processing in automated decision-making systems. The goal of Explainable Artificial Intelligence (XAI) is to create a suite of new or modified AI techniques that produce explainable models that, when combined with effective explanation techniques, enable end users to understand, appropriately trust, and effectively manage the emerging generation of Artificial Intelligence systems.

Case-Based Reasoning (CBR) systems have previous experiences in interactive explanations and in exploiting memory-based techniques to generate these explanations that can be successfully applied to the explanation of other AI techniques. Therefore, this first XCBR workshop is dedicated to address the challenges of applying CBR for the explanation of intelligent systems.

The workshop program includes three position papers and five research and application papers representing various approaches about trends, research issues and practical experiences in the use of CBR methods for the inclusion of explanations to several AI techniques using reasoning-by-example. The workshop program includes an invited talk by David Leake, entitled "Applying Explanatory Experience" about previous research on CBR and explanations, including an overview of the most relevant related papers that opens the discussion.

In the position paper *Creating an Explainable Case-Based Reasoning System* from Ian Watson and his team presents an explainer, based on a Knowledge-Light Explanation Framework (KLEF), that uses a lightweight domain knowledge modeling technique through the creation of a local case-base. Adam Johs, et al. survey the literature to facilitate decomposition of what is meant by quality the context of XCBR explanations in their contribution *Measuring explanation quality in XCBR.* The paper by David Menager and Dongkyu Choi *Episodic Memory: Foundation of Explainable Autonomy* presents a theory of episodic memory that explains how intelligent agents can use their personal experience to make known their internal decision making process.

The second session of the workshop covers more applied papers on CBR and explanations. Belen Diaz-Agudo et al. presents *Data explanation with CBR* an ongoing work on the use of CBR to automate and personalize the generation of data explanatory reports. Venkatsampath Raja Gogineni et al. paper provides a

description of an *Application of Case-based Explanations to Formulate Goals in an Unpredictable Mine Clearance Domain*. It is worth mentioning that there are three papers on explanation and recommender systems. *An Approach to Producing Model-Agnostic Explanations for Recommendation Rankings* by Ian Watson et al. is a proposal to develop a recommendation model that uses an ensemble of recommenders to facilitate a model-agnostic means of producing an explanation. Marta Caro-Martnez et al. describes *A Theoretical Model of Explanations in Recommender Systems* and Kyle Martin et al. presented their work *Explainability through Transparency and User Control: A Case-Based Recommender for Engineering Workers* that highlights the trade-off between performance and explainability.

We wish to thank all who contributed to the success of this workshop, especially the authors, the Program Committee, and the editors of the workshop proceedings!

*Belén Díaz-Agudo* July 2018
*Juan A. Recio-García*
*David W. Aha*

# Creating an Explainable Case-Based Reasoning System

Sueyeon Lee, Samule Li, Harry Lim and Ian Watson

Department of Computer Science, University of Auckland, Auckland, New Zealand
{slee681, sli473, hlim448}@aucklanduni.ac.nz,
ian@cs.auckland.ac.nz

**Abstract.** Case-Based Reasoning (CBR) is a naturally intuitive process that uses past experiences to find new solutions. However, CBR struggles to explain the results concerning the domain knowledge, which is necessary to provide more meaningful explanations to the user. The proposed explainer, based on Knowledge-Light Explanation Framework (KLEF), uses a lightweight domain knowledge modelling technique through the creation of a local case-base. This local case-base, in combination with Principal Component Analysis (PCA) weightings, fortiori cases, and odds ratio are used to produce a convincing explanation to the user. To evaluate these methods, we investigated under what conditions the explainer produced a good, average or poor explanation through a proposed set of metrics. After categorising the results, each explanation was compared subjectively, to validate the usefulness of the metric, and ultimately, the explanation itself. It was found that the best explanations were produced on datasets with ordinal data that didn't use the local case-base. However, even the best explanations had limitations, indicating that explanations cannot be evaluated on a purely objective level, and a usability study is required to quantify the effects of a good explanation.

**Keywords:** CBR, explainable, KLEF, PCA.

## 1 Introduction

Case-Based Reasoning (CBR) is the process of determining new solutions based on past experiences. Although CBR itself is an interpretable system and effective at explaining how it achieves its classification, it remains a challenge to explain the solution in regards to the domain knowledge. By integrating domain knowledge with CBR, the system becomes a better explainable system as it mitigates the limits of explainability, allowing users to learn about the domain knowledge, and reveal the interconnected nature of the features that exist. The two combined ultimately provide more useful explanations and therefore trust in the users of the system; if there is no trust, users will not use the system, especially in mission-critical fields.

The proposed explainer system is an explainer based on a Knowledge-Light Explanation (KLE) with Principal Component Analysis (PCA). The explainer builds a knowledge domain in a lightweight manner, where it finds a fixed number of cases most similar to the query case on both sides of the classification, i.e. the boundary.

The explainer then provides convincing argumentation explanation using odds ratio and the fortiori case.

The PCA and odds ratio values are used to strengthen the explanation through ranking the most significant features and describing the multiplicative effects of each feature respectively. This proposed explainer will be tested under diverse conditions to find the advantages and limitations of its explanation.

## 2 Hypothesis

The hypothesis under test is as follows: Under what conditions does a combination of PCA, fortiori case and odds ratio explainer system provide good, average or poor explanations?

Unfortunately, the goodness of an explanation is highly subjective. Therefore, this study will attempt to determine an explanation as objectively as possible. The goodness of an explanation ultimately reflects how useful the user will find the explanation. Therefore, we have defined three metrics that we believe most correspond to a good explanation [1]:

**The explanation provides valid results and correct feature relationships**

Part of an interpretable explanation is one which presents useful data that is accurate. Although an explanation is not defined by the accuracy or validity of the results it produces, it acts as a baseline factor to see if a user can trust the system, thus a crucial component of explainability.

The correct feature relationships will be determined by whether or not the explanation reflects the domain knowledge. The validity will be judged by if the query case was correctly classified and if the PCA and odds ratio values make logical sense.

**The explanation allows the user to understand which differences in features contributed towards the classification**

The understanding of which differences in features contribute towards the classification will be measured concerning how reasonable the explanation is, mainly regarding user's ability to understand which feature values were more important in classifying the case. This will have a main focus on the odds ratios provided and also potentially the insight provided by PCA weightings.

**The explanation allows ease of interpretation from the user**

This measures whether or not the explanation is easy to understand. Although subjective, this will focus on if the user can understand the logic behind the results intuitively. If an explanation produces a confusing explanation, extremely high odds ratio values or lengthy explanations with irrelevant information, users are unlikely to be convinced of the use of the explanation as they cannot interpret it.

### 2.1 PCA

In our system, PCA was used for two primary reasons - to rank the most significant variables in terms of our explanations and to reduce the complexity of the model. In

our numerical datasets, the first principal component accounted for around 70% to 80% variation in the data. We concluded that this explained a sufficient amount of variability, thus, only the first principal component was used in our analysis. In ranking our explanations, the top three weightings are given more emphasis in the explanation. A good explanation should be simplified by extracting the main reasons as to why a classification was made [2], and this is what we aim to do through the ranking system. Therefore, by providing a clearer and more transparent explanation to the user, this will build user trust in the system.

Features with similar values will likely have an odds ratio closer to one. However, if the feature has a significantly larger weighting in the classification of the case, PCA weightings would potentially reflect this.

### 2.2 Local Case Base and Fortiori Cases (KLEF)

After applying PCA, the local case base and fortiori case were implemented using KLEF [3] as the general skeleton of the structure. The framework provided a good base implementation for explanation of CBR systems using the fortiori case and odds ratio. The fortiori case provides convincing explanation through argumentation of the chosen case. This case refers to the case of the same classification of the query case, which is closest to the boundary in the local case base. By arguing that a case closer to the boundary which has less defining features is of the same classification, this provides a reasonable assumption that the query case is a “better" or more definite classification of that class. Furthermore, odds ratios are used to show data to the user, providing more and hopefully stronger evidence for a certain classification. Providing the user with data or facts gives more trust for the user as it creates more transparency between the user and the system, giving the user an indication of the strength and validity of the explanation. To implement these features, the libraries and framework jCOLIBRI [4] and Weka [5] were utilised.

### 2.3 Odds Ratio

The Weka framework was used to run logistic regression. Odds ratios are a crucial component in the explainer as they provide easier interpretable data to improve trust and validity of the explanation to the user. For example, producing an explanation to say "X is 3 times more likely than Y..." is more convincing than an explanation that says "X is more likely than Y..." This also provides more knowledge to the user, and allows the user to also use their own intuition on whether they agree or disagree with the system.

### 2.4 Explanation Generation

The explanation first shows the query case and explanation case to provide transparency to the user; by showing the explanation case, the user is able to see similarities and differences of raw data and use their own intuition and judgement.

Next, the explanation is generated regarding the variables that support the query case. An odds ratio that is greater than one indicates that the query is supported, hence any variables that had this relationship was used for the output. PCA weightings are computed to show which variables are most significant in explaining the variability of the data. The top three weightings are chosen, and if any of the supporting variables are included in these three weightings, then these will be emphasized, thus producing a more convincing argument. All the variables that are not related to the top three PCA weightings will be bullet pointed, variable by variable.

## 3 Evaluation

The UCI breast cancer dataset [6] consists of 699 data points categorising between malignant or benign with 9 features/ attributes. 458 data points are categorised as benign and 241 data points as malignant. The data set contains ordinal data (with numerical values), Space restrictions here limit the reporting of our experiments; a full report can be found at: www.cs.auckland.ac.nz/Rian/X-CBR/X-CBR_FinalReport.pdf

The following is classified as a good explanation:

*The query case was compared to a pre-determined classification (explanation case) where both were determined to be malignant. Uniformity Of Cell Size was identified as the most significant variable in classifying malignant. It was found that a higher value of Uniformity Of Cell Size in the query individual leads to a 2 times higher likelihood of being malignant than the explanation individual.*
*Uniformity Of Cell Shape was identified as the second most significant variable in classifying malignant. It was found that a higher value of Uniformity Of Cell Shape in the query individual leads to a 2 times higher likelihood of being malignant than the explanation individual.*
*Other variables such as: + Single Epithelial Cell Size being higher on the query compared to the explanation means it is 2 times more likely to be malignant. + Normal Nucleoli being higher on the query compared to the explanation means it is 3 times more likely to be malignant.*

This explanation was judged to be good as it met all the criterion in our metrics. All four features correctly correspond to the domain knowledge [7], as the values in the query and explanation align with the relationship defined in the explanation, i.e. higher Uniformity of Cell Size leads to a higher likelihood of being malignant. It also correctly classifies this classification, thus, satisfying the first criterion. Through PCA, we can see that the Uniformity of Cell Size was identified as most significant, which produced a multiplicative effect of two. From a user's perspective, seeing that the most significant variable that classified a malignant person was part of the explanation adds reliability and trust to the user, and they are able to easily identify which features should be focused on. Finally, the ease of interpretation was evaluated by seeing if the multiplicative effects made sense from a user's perspective. All explanations produced were in factors of two and three which, in our metric definition, satisfies the conditions of being easy to interpret.

Results also showed that the goodness of the explanation was heavily dependent on the type of data used. Good explanations were generated on ordinal data, average explanations were generated on nominal data types and poor explanations were generated on skewed numeric data types.

## 4 Conclusion

Through our implementation of a Knowledge Light Explanation Case-Based Reasoner with PCA, we were able to determine under what conditions a combination of PCA, fortiori case and odds ratio provide good, average or poor explanations. It was discovered that generally the dataset being ordinal data created the best explanations in the case of our case based reasoner. In our results, it was noted that PCA had minimal impact on improving our classification on the usefulness of the generated explanation based on the defined metrics.

## References


1. M. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144, 2016.
2. T. Miller, "Explanation in Artificial Intelligence: Insights from the Social Sciences", arXiv preprint arXiv:1706.07269, 2017.
3. C. Nugent, D. Doyle and P. Cunningham, "Gaining insight through case-based explanation", Journal of Intelligent Information Systems, vol. 32, no. 3, pp. 267-295, 2008.
4. jCOLIBRI | GAIA – Group of Artificial Intelligence Applications", Gaia.fdi.ucm.es, 2018. [Online]. Available: http://gaia.fdi.ucm.es/research/colibri/jcolibri. [Accessed: 03-May2018].
5. "Weka 3 - Data Mining with Open Source Machine Learning Software in Java", Cs.waikato.ac.nz, 2018. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 04- Jun- 2018].
6. "UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set", Archive.ics.uci.edu, 2018. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). [Accessed: 04- Jun- 2018].
7. "Analysis and Modeling of Breast Cancer Data", Rpubs.com, 2018. [Online]. Available: https://rpubs.com/ricardosc/breast-cancer.

# An Approach to Producing Model-Agnostic Explanations for Recommendation Rankings

Dylan Hall, Nathan Hur, Jonathan Soulsby, and Ian Watson

The University of Auckland, Auckland 1010, New Zealand
{dhal525, nhur714, jsou754}@aucklanduni.ac.nz
ian@cs.auckland.ac.nz

**Abstract.** In the era of big data, recommender systems are becoming increasingly more important in helping users to navigate the vast amounts of content available to them. This has driven the development of recommender systems to utilise increasing quantities of user data and feedback to produce more personalized and higher quality recommendations. In recent years, regulations and social perspective shifts have brought to light the importance of transparency and explainability in how user data is used by artificial intelligence. At the conjunction of transparency and explainability, we develop our hypothesis to explore how these issues can be addressed: can we develop a recommendation model that uses an ensemble of recommenders to facilitate a model-agnostic means of producing an explanation, without sacrificing accuracy? Our system in future aims to facilitate this by generating natural language explanations which describe how the system's recommended item ranking is produced, by utilizing what we denote a User Profile Frame (UPF) that represent the given weightings on the ensemble. We find that our ensemble approach does not sacrifice accuracy when compared to the individual models, and provides a promising approach for a user-centric and explainable recommendation process.

**Keywords:** XAI, recommender systems, context-aware recommender systems

## 1 Introduction

It is no doubt that AI is becoming more common in the everyday life of modern society. This is especially evident in the case of recommender systems, with millions of users interacting daily with such systems implemented in commercial giants including Netflix, Amazon and TripAdvisor [1].

Although there is substantial research interest in the XAI field due to a recent proposal by DARPA [2] and the slew of benefits promised [3], this report seeks to investigate the more user centric goals of XAI. The public has XAI under its focus because of new EU legislation, GDPR, that mandates data protection and privacy laws for individuals [2]. Given that the evolution of recommender systems is to fulfil the need for more personalised, user-centric recommendations, they are consuming increasingly larger amounts of implicit and explicit information about users. We see a tension between these two paradigms; users need higher quality recommendations, which can

be achieved through personalisation, but also requires transparency of how the increasing amounts of personal data is be utilised to provide recommendations.

Existing research in this area is relatively new, Zeng et al. proposes adding a component of argumentation to promote explainability, they suggest that existing XAI models can answer the question "Why this decision or conclusion?", but they cannot answer "Why not?" [4]. As well as this, Costa et al. suggest a means of generating more convincing explanations through use of a Recurrent Neural Network (RNN) used to produce text emulating user reviews in a domain [5]. With regards to ensemble-based recommender systems; Fortes and Manzato find that combining various types of recommendations in a single model for recommendation using ensemble learning is effective; with their proposed method performing better than their baselines [6]. Finally, Gunter and Bunke successfully demonstrate an improvement in accuracy when using a weighted ensemble voting policy where weights are optimised via a genetic algorithm, although this was achieved with classifier models instead of recommenders [7].

Inspired by related research, this report proposes a combined recommendation model that uses an ensemble of recommenders to facilitate a model-agnostic means of producing an explanation, whilst not sacrificing recommendation accuracy. We denote the set of weights describing the voting power of each recommender as a "User Personalization Frame" (UPF) as it serves the important purposes of providing an explanation of the ranking produced by the combined system, as well as a mechanism that allows a user's feedback on the explanation to control how the system weights their preferences in providing a recommendation list.

## 2 Methodology

At a high-level summary, our prototypical combined recommender system comprises individual collaborative metric learning (CML) models trained on implicit feedback that was converted from an explicit dataset [8]. The combined recommender system served weighted scores of the items for a given user's UPF — generated via a genetic algorithm — which dictated the final item recommendation ranking for the user. Figure 1 shows an overview of this approach, which can include a variety of RecSys methods including CF algorithms and content-based methods, such as case-based reasoning.

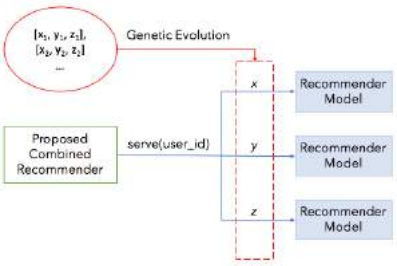



**Fig. 1.** Overview of system architecture

The recommender algorithm we decided to utilise for each of our individual recommender systems was CML, as a means of furthering the user-centered focus of this prototype. We leveraged a free, openly-available movie review dataset (the-movies dataset) provided by Kaggle due to the intuitiveness and user focus of the domain. As suggested in [9], we converted the explicit reviews into implicit actions by creating a threshold for "positive" movie interactions – positive interactions were dictated by reviews of rating 3.0 or greater. To produce the CML models, a relatively new recommender framework, OpenRec [10], was utilized. The OpenRec framework leverages TensorFlow [11] to construct each computational graph, providing the ability to save the individual tensors to disk, which are loaded into the combined recommender. Refer to *https://github.com/Nateeo/datamining* for the scripts used and the individual/combined artifacts.

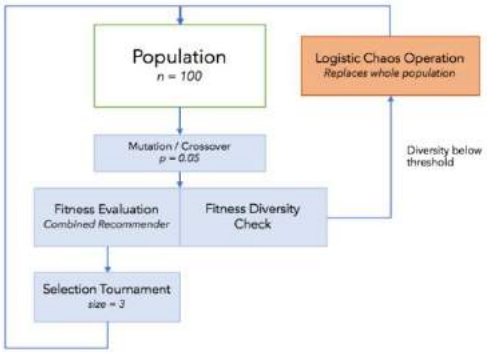



**Fig. 2.** Genetic algorithm approach

The initial training of the ensemble weights follows the success of [7] in the use of genetic algorithms for weighted-vote ensembles. A standard genetic algorithm outlined in Figure 2 was used, where an individual is a tuple of weights representing the UPF. Notably, we use a diversity check and a chaos operator demonstrated in [12] to prevent premature convergence.

It is envisioned that the explanation of the UPF-produced ranking could be generated via something as simple as fuzzy logic. This would be based on the UPF's weightings of the corresponding individual recommender module within the combined system. For example, consider an individual recommender module trained solely on the social network context of users and another trained with popularity, augmented with a time context decay. In this situation for a given user, suppose that their UPF weighting corresponding to the social context was the highest, closely followed by the weight for the time context. Fuzzy logic could then generate an explanation along the lines of "Because you really like what your friends are watching and value popularity of items somewhat…" to explain how the ranked item recommendations were produced. The user could then provide explicit feedback on the given explanation, which would update the corresponding UPF weightings. In addition to this explicit user interaction, the system could utilise implicit data from user actions to update the UPF.

## 3 Initial Results

We obtained NDCG @ 30, AUC, Precision @ 30, and Recall @ 60 for each individual model as well as the combined recommender system for each of the 20 user's test data, using the optimal UPF weightings generated by the genetic algorithm.

In Figure 3, the combined recommender's line graph is filled in grey to compare against the NDCG @ 30 of the individual models. Here it can be seen that the performance of the combined recommender system is comparable to the highest performing individual model, even outperforming it on several users.

Although our initial testing suffered from the problem of sparsity of unknown entries common in offline evaluation of recommender systems [13], our results are consistent with similar evaluation of ensemble approaches in recommender systems and classifiers in the fact that the ensemble does not reduce the overall performance, and can improve it in many applications [6, 7, 13].

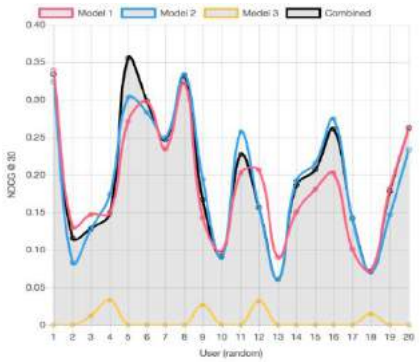



**Fig. 3.** NDCG @ 30 of the models for 20 random users

## 4 Conclusions and Future Work

We have investigated the feasibility of a model-agnostic ensemble-based recommender system that can provide explanations for the output item ranking through utilising ensemble weights corresponding to distinct context aware recommender models. Through our development and evaluation of our system, we have demonstrated that the weighted ensemble does not perform worse in terms of accuracy and ranking relevancy when compared with its constituent models.

We have also discussed the promising characteristics of our approach to generate explanations. Namely, the UPF's allow for simplistic but convincing arguments to standard users, and the nature of our model easily allows for user interaction and customizability in the future. Such characteristics show that this model is able to provide some useful explanations given future development, with the caveat that the explainability revolves around the rankings of the items that the user is shown from the combined model, not explanations of the individual models in the ensemble.

# 5 References


1. Kumar, B., Sharma N.: Approaches, Issues and Challenges in Recommender Systems: A Systematic Review Indian Journal of Science and Technology, vol. 9, 2016.
2. G. D. P. Regulation: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46, Official Journal of the European Union (OJ), vol. 59, pp. 1-88,2016.
3. Samek, W., Wiegand, T., Müller, K.: Explainable Artificial Intelligence: Under-standing, Visualizing and Interpreting Deep Learning Models, arXiv PreprintarXiv:1708.08296, 2017.
4. Zeng, Z., et al.: Building More Explainable Artificial Intelligence with Argumentation,2018.
5. Costa, F., et al.: Automatic generation of natural language explanations, In Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, 2018.
6. Costa F., Manzato, M. G.: Ensemble learning in recommender systems: Combining multiple user interactions for ranking personalization, in Proceedings of the 20th Brazilian Symposium on Multimedia and the Web, 2014.
7. Günter, S., Bunke, H.: Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm, ELCVIA: Electronic Letters on Computer Vision and Image Analysis, vol. 3, (1), pp. 25, 2004.
8. Hsieh, C., Yang, L., Cui, Y., Lin, T., Belongie, S., Estrin, D.: Collaborative Metric Learning, in Proceedings of the 26th International Conference on World Wide Web- WWW '17, 2017.
9. Zhao, Q., Maxwell Harper, F., Adomavicius, G., Konstan, J.: Explicit or Implicit Feedback? Engagement or Satisfaction? A Field Experiment on Machine-Learning Based Recommender Systems, in Proceedings of the 33rd ACM/SIGAPP Symposium On Applied Computing, Track of Recommender Systems: Theory, User Interactions and Applications (SAC 2018), ACM, 2018.
10. Yang, L., et al.: OpenRec: A modular framework for extensible and adaptable recommendation algorithms, in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018.
11. Abadi, M., et al.: Tensor Flow: A system for large-scale machine learning. in OSDI, vol.16, pp. 265-283, 2016.
12. Liu, J., Cai, Z., Liu, J.: A novel genetic algorithm preventing premature convergence by cha-os operator, Journal of Central South University of Technology, vol. 7, (2), pp. 100-103, 2000.
13. Bar, A., et al.: Boosting simple collaborative filtering models using ensemble methods, arXiv Preprint arXiv:1211.2891, 2012.

# Explainability through Transparency and User Control: A Case-Based Recommender for Engineering Workers

Kyle Martin[1 [0000−0003−0941−3111]] ✉, Anne Liret[2 [0000−0003−0620−7240]], and Nirmalie Wiratunga[1 [0000−0003−4040−2496]], Gilbert Owusu[3], Mathias Kern[3]

[1] Robert Gordon University, Aberdeen, Scotland
{k.martin, n.wiratunga}@rgu.ac.uk
[2] BT France, Paris, France
anne.liret@bt.com
[3] British Telecommunications, United Kingdom
{gilbert.owusu, mathias.kern}@bt.com

**Abstract.** Within the service providing industries, field engineers can struggle to access tasks which are suited to their individual skills and experience. There is potential for a recommender system to improve access to information while being on site. However the smooth adoption of such a system is superseded by a challenge for exposing the human understandable proof of the machine reasoning. With that in mind, this paper introduces an explainable recommender system to facilitate transparent retrieval of task information for field engineers in the context of service delivery. The presented software adheres to the five goals of an explainable intelligent system and incorporates elements of both Case-Based Reasoning and heuristic techniques to develop a recommendation ranking of tasks. In addition we evaluate methods of building justifiable representations for similarity-based return on a classification task developed from engineers' notes. Our conclusion highlights the trade-off between performance and explainability.

**Keywords:** Case-Based Reasoning · Recommender Systems · Explainable AI · Information Retrieval · Machine Learning

## 1 Introduction

Within field service provisioning industries there is increasing interest into the empowerment of workers to ensure the right expert knowledge is used at the right level in the decision process. In [12], the scheduling system interactively allocates tasks with empowered engineers thanks to a personalised recommendation system that suggests tasks to an engineer based on their history of completed tasks. However, the increasing complexity of tasks lead to situations where engineers struggle to evaluate accurately the required work on tasks which are nearby and within their skill set. The amount of tasks generated every day across all business divisions can be large, which can further exacerbate the problem engineers face in finding appropriate work with the correct context information at the right time. This question becomes more critical when the type of services are inherently dynamic, such as when high priority tasks are raised that require an

engineer's immediate attention, and require that he must abandon tasks which he might be unable to revisit on time. In a worst case scenario, tasks may miss their deadline.

The motivation of this study is to develop a method to access and prioritize tasks which fall within the engineers' capabilities, experience and are of relevance to the business at that point in time. A recommender system has potential to fill this gap [4], but responses to a fuzzy logic recommender [12] suggested that users resented the lack of clarity behind its recommendations. The ability to explain a system's recommendation, or display a level of transparency which allows the user to understand the reasoning behind that recommendation, encourages trust between a system and its users [13].

In [19] the authors present five goals that an explainable intelligent system must be able to satisfy - transparency, justification, conceptualisation, relevance and learning. Based on these criteria, we observe that a CBR system already achieves 3 (transparency [8], relevance and learning [1]), but does not necessarily answer the remaining 2 (justification and conceptualisation). These criteria and the improvement they may bring to interactive services scheduling motivated the contributions of this paper. A real world dataset from telecommunications services has been used for this case study. For each service request, a number of progression comments until successful or unsuccessful completion are reported by engineers. These offer a large source of unstructured data ("engineer notes"), but are difficult to exploit due to variability in content quality.

With this observation in mind, this paper presents a transparent telecommunications task case-based recommender system which has been extended to incorporate the ides of conceptualisation and justification. To assist users in understanding the necessary concepts for decision-making, we introduce a customizable modular design based upon parallel co-ordinates [9], which considers the input of various similarity assessment modules to develop a recommendation ranking. We combine related case attributes and a local similarities model to improve the conceptual level of recommended objects. To improve the system's ability to justify a decision, we evaluate several text representation learning measures to determine which is most useful to select features for justification.

We offer several contributions in this paper. We (1) present a framework for case-based recommender systems which facilitates conceptualisation inspired by parallel co-ordinates. We (2) showcase an extension to the system that allows user customisation to suit individual or business needs and improves transparency. Lastly, we (3) evaluate methods of developing representations from engineer notes for similarity-based return on the basis of their accuracy and ability to justify a decision. Though presented as a field services recommender, the concept could be adapted to fit other domains.

This paper is split into the following sections. Section 2 discusses the problem in more detail and Section 3 talks about related work. Section 4 outlines the use of conceptual parallel co-ordinates to improve conceptualisation. Section 5 discusses text representation learning methods and analyses their impact on recommendation justification. Finally, in Section 5 we offer some conclusions.

## 2 Learning Similarity from the Experts

As in a number of complex services provisioning organisations, the telecommunications engineering force who carry out the work in the field, gradually form a strong and

concrete expertise in the field of network equipment installation and repair. To ensure service delivery, they traditionally are allocated tasks, that each is, in this scenario, a pre-defined time constrained action to perform on a specific piece of equipment. Field engineers record information about the tasks they have completed in text documents called ”notes”. These notes originally contain necessary information such as location of the task and an overview of the work to be done (Order notes) and are expected to be updated by the engineer when the task is completed with some details of how this was achieved (Closure notes). If a task cannot be completed then the cause of this is also expected to be recorded (Further notes) and lastly, the engineer can enter additional information they feel as necessary (User notes).

This work is motivated by the need to learn similarity from a user's perspective. We believe that by using notes that have been written by engineers themselves as the information source for a similarity metric, the cases retrieved through similarity-based return will be more representative of this point of view. It will also allow greater opportunity for explainability, as it enables potentially generating post-hoc explanations for recommendations based on other engineers' description of tasks. Finally it enables extracting and sharing implicit knowledge to proactively inform about practical work instructions, all over the service and supply chain.

## 3 Related Work

The main motivation behind this paper is the work presented in [19]. In that paper, the authors present the five goals of an explainable intelligent system - transparency (the system can demonstrate the reasoning behind its decision), justification (the system can justify why the proposed solution is better than other potential solutions), conceptualisation (the system can illustrate to the user the meaning of concepts required to understand the decision), relevance (the approach adopted by the system is relevant to the problem) and learning (the solution provided by the system can improve user knowledge).

Anecdotal evidence from a number of sources suggest that Case-Based Reasoning (CBR) systems can already answer 2 of the 5 goals of explainability (transparency, relevance) while facilitating another (learning) by virtue of the architecture itself [6, 17]. We can observe that the solutions offered by a CBR system are always relevant to the presented query, as they draw upon most similar problems that the system has seen before from within the same domain [1]. Furthermore, as CBR systems are based on the concept of reusing solutions to solve similar problems, it is trivial to direct users towards the original solution which led to this point, thus demonstrating transparency of decision-making [8]. We also argue that CBR systems facilitate learning through the method in which they draw on past experience to present a solution to the user. As this is similar to the way in which humans learn [14], it eases uptake of knowledge from the system. We do however acknowledge the argument that a human cannot learn from a solution they do not understand [19], so if a user cannot understand $why$ a presented solution successfully answers a query, then their learning will be inhibited. Thus, we suggest that learning can be most easily obtained by achieving the other 4 goals. As such, we maintain focus on improving justification and conceptualisation.

We argue that a vanilla CBR system does not necessarily answer the remaining two goals - justification and conceptualisation. Although the process which led to the selection of CBR as technical approach is transparent, it is not necessarily clear to the user why the resulting recommendation would be better. Furthermore, although [10] have shown that displaying local similarity scores from known calculations can provide clarity to users regarding the justification of a decision, these do not help in understanding the underlying concepts. Hence we suggest improvements to conceptualisation (inspired by parallel co-ordinate visualisation techniques) and justification (through an evaluation of representation learning methods) aspects in the following subsections.

In this work, we propose to use parallel co-ordinates at a conceptual level with the aim of improving the conceptualisation aspect of field service recommendation systems. Parallel co-ordinates are means to visualise high-dimensional data. They are often used in CBR systems to allow simpler comprehension and comparison of local similarities in the return set [9, 10]. Previously, parallel co-ordinate visualisation methods have been used to improve the explanation of CBR solutions for pharmaceutical tablet formation [10]. In that work, pharmaceutical experts praised the clarity of the method and agreed that the visualisation could be more meaningful and easier to understand than a textual explanation.

However, this method becomes less meaningful if the user cannot understand the features or potentially even the local similarity which is being described. We propose an extension to this practice, by using parallel co-ordinate visualisation techniques at a conceptual level. Similar in essence to the work in [3], where the authors suggest a level of abstraction can be useful for classification, we propose that generalisation of concepts can be a useful tool to improve user understanding for explainability (though we do not use this as a basis to then perform induction for classification as in the original paper). By collecting related local similarities together under a meaningful heading, the user is given context which can improve their understanding of individual similarities and how each contributes to the recommendation as a whole.

In this paper we consider both distributional and distributed approaches to learning representations for text documents as a means to improve a system's ability to justify a decision. Distributional approaches, such as the statistical-based method tf-idf, produce sparse document representations that can be difficult to utilise in machine learning algorithms, but are trivial to relate back to specific features. Distributed approaches, such as document-2-vector (Doc2Vec) [7], a method derived from Word2Vec [11], produce dense representations. However, as they develop a level of abstraction it becomes more difficult to relate these to specific features.

Deep metric learners, like the Siamese Neural Network (SNN) [2], do not learn a representation directly from the text itself. They receive combinations of pre-processed representations (such as that obtained from tf-idf or Doc2Vec) as input to develop embeddings which are optimised based on an objective. This objective is defined by a matching criteria, which does not necessarily have its basis in class knowledge [5]. Deep metric learners develop representations where cases that meet these matching criteria exist close together, whilst cases that do not exist further apart. The developed space is therefore optimised for similarity-based return. Deep metric learners have shown achievements in areas such as face verification [18] and similar text retrieval [15].

## 4 Improving Conceptualisation with Parallel Co-ordinates

In this paper we present a case-based recommender system that uses a parallel co-ordinates approach, inspired from visualisation methods, to contextualise local similarities. Here we use the term 'concept' to describe a subset of local similarities or attributes collected under one descriptive heading. For example, in the suggested system all attributes which are heuristically evaluated to give an idea of task urgency or business priority are grouped under the Business Relevance concept module. Each concept makes use of a subset of local similarities or attributes to develop a score, before all concept scores are combined to develop a recommendation ranking. We use this method to augment the backbone of our recommender, which is based on task similarity knowledge gained from text and is described in the following section.

Collecting local similarities under a meaningful heading allows excellent opportunity for conceptualisation. Instead of being presented with only a recommendation or a general score, users can see how a recommended item is scored across several meaningful concepts. For example, an engineer presented with a top recommendation may have difficulty understanding where it has come from. However, if a score breakdown is provided such that the user can see that the recommendation scored an average of 90% relating to Business Relevance concept and 80% relating to Similarity to Previous Work concept then this could be much more meaningful. Grouping attributes into concepts can also improve understanding of the score if any of the individual attributes or local similarities are not meaningful to the user.

The basic idea is this. Let us say we have a set of concepts, $C$, with each individual concept represented as $c \in C$. Each concept should represent a related collection of case attributes. Therefore, if $A$ were to represent the full set of case attributes, such that $a \in A$, then each concept represents a subset of case attributes, where $c \subset A$. We can represent the concept similarity score, $c_s$, between a given query $q$ and any case $x$ as:

$$c_s(q, x) = \frac{\sum_i^{|A|} a_i(q, x)}{|A|} \tag{1}$$

where $a_i$ represents the local similarity calculation and $|A|$ represents the total number of local similarities or attributes that have contributed towards that concept score.

We can combine the output of all concept scores to create a global score by which to rank cases. This allows us to develop a visualisation similar to that of parallel co-ordinates [9], as in Figure 1. We argue that collecting related attributes under meaningful concept headings allows users to better understand the relationship between local similarities by giving them context. This in turn can help them understand these better at a conceptual level. Furthermore users should be better able to understand the breakdown of the decision-making process of the system, improving overall transparency.

### 4.1 Giving Users Control Over Recommendations

We can facilitate transparency in the presented system by giving the user control over the weighting of the individual components of the recommendation at both a local similarity and concept level. These customization options allow the system to be configured

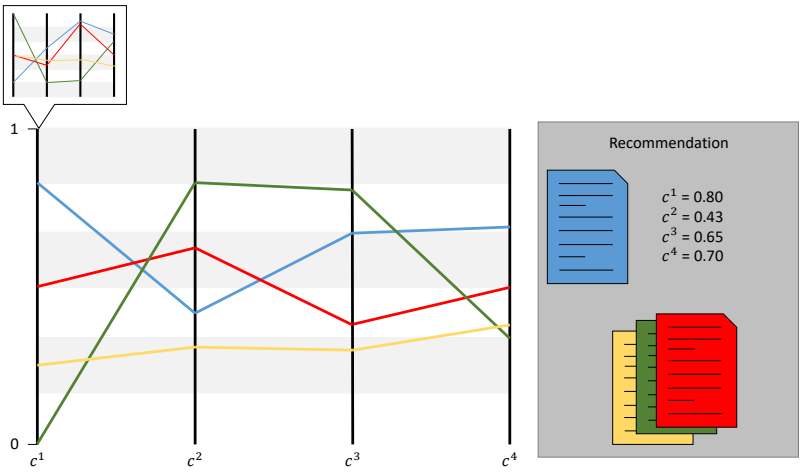



Fig. 1: The concepts of the recommender can be seen to emulate typical similarity visualisation methods. Importantly, each concept acts as a summary for the local similarity methods within it.

to meet both individual user and business needs, as well as encourage trust in the system. In addition, they ensure that the Task Recommender can be altered to meet the needs of the numerous work types within a telecommunication organisation and presents the opportunity to standardise relevant task retrieval across these disciplines. Thus the similarity calculation for any given concept becomes:

$$c_s(q,x) = \frac{\sum_i^{|A|} a_i(q,x) \cdot w_i^a}{|A| \cdot |W^a|} \tag{2}$$

While the overall similarity for a query $q$ to any given case $x$ becomes:

$$q \sim x = \frac{\sum_i^{|C|} c_i(q,x) \cdot w_i^g}{|C||W^g|} \tag{3}$$

By giving the user control over the impact that different concepts have upon the recommendation, we encourage the user to have better understanding of how concepts relate to each other at a summary level, one-step above that of local similarities. This can directly contribute to user learning, one of the goals of an explainable intelligent system. Furthermore the transparency of the system is again highlighted, as it allows the user a better understanding of how the recommendation was made. We plan to evaluate this claim in a subject study with telecommunication engineers in future work.

## 5 Learning a Text Representation for Justification

We examine several methods for learning representations for text documents in regards to their accuracy and their ability to justify a recommendation. Specifically, we consider

a distributional representative (tf-idf), a distributed representative (doc2vec) and a deep metric learner (Siamese Neural Network (SNN)). Furthermore, as deep metric learners require a sample selection strategy to perform most efficiently, we also consider an SNN which uses DYNEE sample selection.

Term frequency-inverse document frequency (tf-idf) calculates a value for each term in a document by dividing the frequency of the term in said document by the percentage of documents which contain that term [16]. As such, for a document representation built from tf-idf, each feature is a value which represents an individual word. It is therefore easy to relate these features back to the raw data to form an explanation. However, tf-idf is not well-suited for extremely large corpus' or vocabulary as this can lead to sparse representations for documents. In addition, it does not explicitly handle synonyms.

Document-2-Vector (Doc2Vec) [7] is an extension of the Word2Vec algorithm [11]. Word2Vec uses contextual knowledge gained from word co-occurrence to build word embeddings for every term in a corpus and develop a metric space where words that have similar contexts exist close together. Using Doc2Vec, the word embeddings for each term in a document are averaged to produce a representation for the document itself. This allows direct similarity comparison between documents within a corpus based upon their contents, whilst avoiding sparseness of representations.

The SNN is a deep metric learner comprised of two matching sub-networks with identical weights and parameters. Examples are input to an SNN in pairs, and are labeled as either positive or negative based on whether the pair satisfies user-defined matching criteria. The metric space which is learned by the SNN is optimised based upon the stated matching criteria, such that positive examples (pairs of instances which adhere to the matching criteria) exist close together, while negative examples (pairs in which members do not fit the matching criteria) exist far apart.

As the SNNs receive pairs as input, there is an additional dimension to training in the form of a pairing strategy. Research has demonstrated that deep metric learners which incorporate a sample selection strategy can offer increased performance. We therefore also considered an SNN supported by DYNEE sample selection. DYNEE is a sample selection method which combines exploitation of the knowledge gained from training thus far and exploration of the feature space to select pairs for training.

### 5.1 Evaluation

In this section we evaluate the vectorial representation of notes gained through each method - tf-idf, Doc2Vec, SNN and an SNN with DYNEE sample selection - to determine which develops the best representation in terms of accuracy and explainability. For the purposes of comparison we have created a simple classification task where notes are classified according to one of four work types.

### 5.2 Experimental Setup

We extracted two months worth of notes generated by telecommunication engineers between March and April 2018. We used Order notes as a means of developing a representation for similarity-based return in a case-based reasoning system. Order notes have the nice property to be present in all tasks, which is not true for any other note type. We

also filtered out any Order note which contained less than 50 characters, as these were judged not to contain enough information to be meaningful. This resulted in a dataset of 1610 notes split into four classes - cabling (227 notes), jointing (789 notes), overhead (503 notes) and power testing (91 notes). These classes represent the work type (i.e. the primary competence of engineer required) which is associated with each note.

The dataset was split into train and test and evaluated using 5-fold cross evaluation. Embeddings for each note were built using each of the above outlined methods[1]. We used k-nearest neighbour for similarity-based return, with $k$ equal to 3. The Doc2Vec feature size was 300. For the SNN implementation using DYNEE, pair selection was repeated every 5 epochs. The $\alpha$ exploitation ratio used for DYNEE was $^{|P|}/_{10}$.

### 5.3 Results

The results can be seen in Table 1. All representations generated by deep metric learners obtain higher accuracy on the classification task. Tf-idf itself does particularly poorly - this is to be expected, as tf-idf does not consider the context of terms whereas doc2vec does. We only display results for SNNs which used input gained from the Doc2Vec model. This was simply because it achieved the best results, though both SNNs using tf-idf as input still outperformed other methods.

Figure 2 illustrates representations of the casebase using a multi-dimensional scaling scatter plot. It confirms that concepts learned by SNNs form better clusters around class boundaries. This also supports the performance gains observed with SNNs.

| Architecture | Accuracy (%) |
|---|---|
| Tf-Idf$_{k-NN}$ | 62.24 |
| Doc2Vec$_{k-NN}$ | 63.79 |
| SNN BASE$_{k-NN}$ | 66.25 |
| SNN DYNEE$_{k-NN}$ | 66.83 |

Table 1: Results of representation learning methods on a classification task.

### 5.4 Explainability of Results

We examined the explainability of each representation in terms of the ability to justify a classification with evidence from the feature set. Using tf-idf we can highlight exactly what terms are similar and have led to finding nearest neighbours by identifying the features that demonstrate most similarity. As each feature directly relates to a term from the documents themselves, it is trivial to find a list of correlating terms. Similarly, using the features from the doc2vec representation we can identify which concepts have seen the most activation, or are most closely related to specific features. However, the SNN is more opaque. After having been input to the network, it becomes difficult to map the meaning of the abstracted output representation back to the original input.

[1]Both SNN sub-network architectures were comprised of 3-layer perceptrons which used ReLU activations and were trained for 250 epochs.

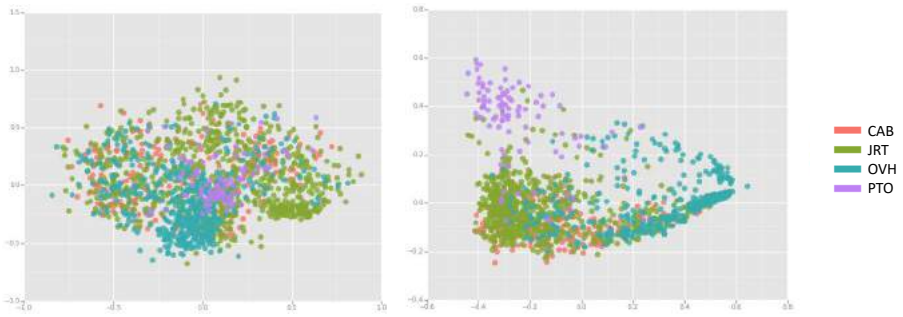



Fig. 2: Representation distribution gained from Doc2Vec (left) and SNN (right).

This leads us to the conclusion that there is a trade-off between performance and explainability. Examining this, we find that the representations built by methods which allow the easiest justification of results (such as tf-idf) are extremely close to the raw features of the data. Meanwhile, the representations built by deeper methods (such as the SNN) tend to achieve greater accuracy as they have underwent a series of abstractions. We can observe there is a trade-off between the clarity that using raw features allows and the performance boost enjoyed by architectures which can develop abstract concepts from data to produce embeddings. This leads us to the suggestion of the explainability-abstraction spectrum. In future work, we plan a subject study with telecommunication engineers to empirically evaluate the justification capacity of different text representation learning methods.

## 6 Conclusion

In this paper we have introduced an explainable task recommender system that is applied to telecommunications service operations flow. This system adheres to the five goals of explainable artificial intelligent systems in its design. We have also presented a method to improve conceptualisation in a CBR system using parallel co-ordinates at a conceptual level. Lastly, we have performed an evaluation of methods to produce representations for text in regards to their ability to justify a decision.

As future work, we are planning a deployment with the intent of obtaining user feedback to evaluate the system on its explainability. Moreover our study confirmed the potential of engineering notes to help drawing up some extra information about tasks. We are in particular motivated to exploit these unstructured data to classify the pieces of work according to actual required knowledge and then recommend workers with on one hand suitable task in need of action, and on the other hand conceptualised form of task context annotated with disturbance likelihood assessment.

## References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications **7**(1), 39 – 59 (Mar 1994)

2. Bromley, J., Guyon, I., LeCun, Y.: Signature verification using a 'siamese' time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7**(4), 669 – 688 (August 1993)
3. Drastal, G., Czako, G., Raatz, S.: Induction in an abstraction space: A form of constructive induction. In: Proc. of the 11th Int. Joint Conference on Artificial Intelligence - Volume 1. pp. 708–712. IJCAI'89, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
4. Isinkaye, F., Folajimi, Y., Ojokoh, B.: Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal **16**(3), 261 – 273 (2015)
5. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Deep Learning Workshop. ICML '15 (July 2015)
6. Kofod-Petersen, A., Cassens, J., Aamodt, A.: Explanatory capabilities in the creek knowledge-intensive case-based reasoner. In: Holst, A., Kreuger, P., Funk, P. (eds.) Volume 173: Tenth Scandinavian Conference on Artificial Intelligence. pp. 28 – 35. Frontiers in Artificial Intelligence and Applications (2008)
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML'14, JMLR.org (2014)
8. Leake, D.B.: Case-Based Reasoning: Experiences, Lessons and Future Directions. MIT Press, Cambridge, MA, USA (1996)
9. Lind, M., Johansson, J., Cooper, M.: Many-to-many relational parallel coordinates displays. In: 2009 13th International Conference on Information Visualisation. pp. 25 – 31 (July 2009)
10. Massie, S., Craw, S., Wiratunga, N.: Visualisation of case-base reasoning for explanation (2004)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
12. Mohamed, A., Bilgin, A., Liret, A., Owusu, G.: Fuzzy logic based personalized task recommendation system for field services. In: Bramer M., Petridis M. (eds) Artificial Intelligence XXXIV. SGAI 2017.Lecture Notes in Computer Science, vol 10630. pp. 300–312. Springer, Cham, Cambridge, UK (December 2017)
13. Muhammad, K., Lawlor, A., Smyth, B.: On the pros and cons of explanation-based ranking. In: Aha, D.W., Lieber, J. (eds.) Case-Based Reasoning Research and Development. pp. 227 – 241. Springer International Publishing, Cham (2017)
14. National Research Council: How people learn: Brain, mind, experience, and school: Expanded edition. National Academies Press (2000)
15. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In: Rep4NLP@ACL (2016)
16. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. pp. 133 – 142 (2003)
17. Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In: Funk, P., González Calero, P.A. (eds.) Advances in Case-Based Reasoning. pp. 389 – 403. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
18. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 815 – 823. CVPR '15, IEEE Computer Society, Washington, DC, USA (June 2015). https://doi.org/doi:10.1109/cvpr.2015.7298682
19. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning - perspectives and goals. Artificial Intelligence Review **24**(2), 109 – 143 (2005)

# Episodic Memory: Foundation of Explainable Autonomy

David H. Ménager and Dongkyu Choi

University of Kansas, Lawrence, KS 66045, USA
{dhmenager, dongkyuc}@ku.edu

**Abstract.** The inner workings of intelligent agents today are often opaque to human users who wish to manage and collaborate with these systems. The lack of transparency prevents users from understanding or predicting the behavior of such agents. In this paper, we present a computational theory of episodic memory with which intelligent agents can explain to the users their personal experience including the internal decision making processes. We provide some initial results from using this framework in question answering and conclude with discussions on related and future work.

**Keywords:** Episodic Memory · Cognitive Architecture · Explainable Autonomy.

## 1 Introduction

Recently, machine learning systems have produced results that have garnered an attentive audience, especially amongst those in industry and government. Unfortunately, many, if not all, of these systems operate like black boxes, of which humans can neither understand, nor predict the internal decision making processes. Users who rely on artificial intelligence systems for recommendations, decisions, or actions will naturally want to understand and predict their behavior. This is especially true for agents existing over extended periods of time because they will likely operate under intermittent or no supervision.

An explainable agent will help remedy this issue, providing information and justifications for its behavior and allowing its user to understand and predict its decision making processes [6]. To demonstrate this capacity, an agent must be able to: summarize its personal history at appropriate levels of abstraction; explain why and when it chose goals to pursue; provide the rationale for how the goals were achieved; discuss how actions were executed in the world; provide details on alternative options for achieving goals; and expose any failures or difficulties the agent faced during planning or execution.

We believe that cognitive systems provide an ideal basis for building such agents. This paper builds on our previous work [11] where we developed episodic memory for a cognitive architecture, Icarus [10], and characterizes the role of this memory in facilitating explainable autonomy. In the following sections, we first review the arhictecure briefly and describe its episodic memory and the

processes that work over it. After that, we explain our approach to explainable agents using episodic memory and describe an ICARUS agent we built using this capacity. We then discuss some related work and future directions for research before we conclude.

## 2 ICARUS Review

ICARUS is a cognitive architecture that provides an infrastructure for modeling intelligent behavior. As such, it makes strong commitments to its knowledge representation, memory, and the processes that operate over them. Some of these commitments are shared with other architectures like Soar [9] and ACT-R[1], but ICARUS uniquely emphasizes on hierarchical knowledge structures and teleoreactive execution. In this section, we briefly review the main aspects of the architecture before we continue our discussions on its episodic memory.

ICARUS has two distinct knowledge structures, concepts and skills, for representing semantic and procedural knowledge, respectively. Concepts describe the relations between objects in the world. ICARUS distinguishes two kinds of concepts. Primitive concepts directly relate objects in the world, while higher-level concepts define relations amongst lower-level concepts. For example, Table 1 shows some sample concepts for an agent playing Minecraft. The first in the list is a primitive concept describing the situation where the agent is carrying an item. It matches against two perceived objects, *self* and *hotbar*, and their attributes and performs a test on them. The second concept is a higher-level one that refers to another concept *entity* in addition to specifying perceptual matching conditions and tests.

**Table 1.** Sample ICARUS concepts and skills for Minecraft.

```
((carrying ?o1 ?o2 ^type ?type ^location ?loc ^size ?size)
 :elements ((self ?o1)
            (hotbar ?o2 type ?type location ?loc size ?size belongs-to ?o1))
 :tests ((> ?size 0)))

((behind-of ?o1 ?self)
 :elements ((self ?self y ?y))
 :conditions ((entity ?o1 ^y ?y1))
 :tests ((> ?y (+ padding* ?y1))))

((about-face-to ?o1)
 :elements ((self ?self))
 :conditions ((behind-of ?o1) (on-vertical-axis ?o1))
 :actions ((*turn-left))
 :effects ((not (behind-of ?o1)) (on-vertical-axis ?o1) (front-of ?o1)))

((turn-around ?o1)
 :elements ((self ?self))
 :conditions ((behind-of ?o1))
 :subskills ((about-face-to ?o1) (stop-turning))
 :effects ((on-vertical-axis ?o1) (front-of ?o1) (not (turning ?self))))
```

Icarus's skills are procedural descriptions for how to achieve goals. As the last two items in Table 1 show, skill definitions are condition-action pairs with associated effects. Primitive skills like *about-face-to* describe direct actions the system should take in order to achieve the desired effects under the conditions described. Higher-level skills like *turn-around* have subgoal decompositions that, in turn, triggers other skills.

During run time, Icarus operates in cycles. As Figure 1 shows, on each cycle, Icarus receives sensory information from the environment and posits them into its perceptual buffer. From there, the system elaborates a belief state by engaging the inference engine. All beliefs are inferred in a bottom-up fashion to ensure that all possible inferences are made. Once the system completes this step, it selects a skill that achieves one or more of its goals and instantiates the skill as its intention. Then Icarus executes that intention until the goal is achieved, or the skill is no longer executable. The architecture continues operating in cycles until all its goals are satisfied. In the next section, we now turn our attention to the recent extension for episodic memory.

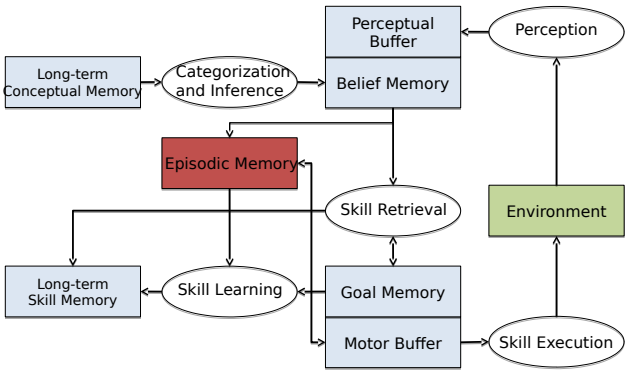



**Fig. 1.** A diagram showing Icarus's memories and processes.

## 3 Episodic Memory in Icarus

The episodic memory in Icarus is a long-term, cue-based memory that the system uses to encode and retrieve records of events. As shown in Figure 2, the memory is a compound structure composed of a state-intention cache ($\rho$), a concept frequency forest ($\mathcal{F}$), and an episodic generalization tree ($\mathcal{T}$).

The state-intention cache is an ordered sequence of belief state and intention pairs, which stores a complete, detailed history of what the agent observed and what it did. This cache is reminiscent of an episodic buffer from the psychological literature [3] that is believed to process the recent past. The traces that are collected into the cache must be processed for *interesting* events, which Icarus determines using its concept frequency forest. This forest is a collection of concept frequency trees that are indexed by the agent's location predicate

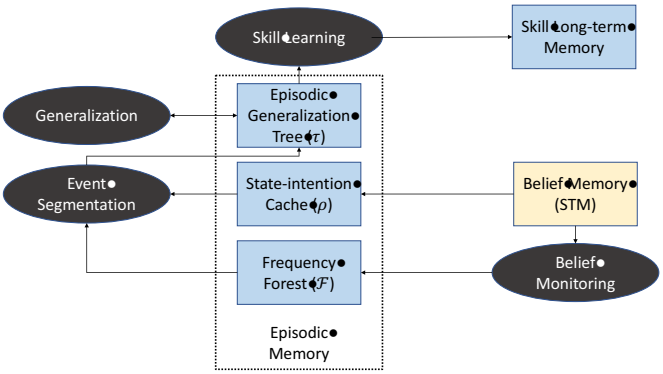



**Fig. 2.** Block diagram depicting ICARUS's episodic memory components and information flow starting from the belief memory.

at their roots. Each of these trees tracks the concept instances the agent observed at a particular location and maintains conditional probabilities for them. Using these trees, the system can form expectations to observe certain concept instances and not to encounter other instances in a particular location. Such expectations then allow the agent to segment events properly, just as humans are believed to perceive the beginning of a new episode when their near-term expectations are violated [8].

When one or more unexpected concept instances occur, the system creates a new episode based on these. An episode is a tuple that includes the beginning and end states, the set of significant beliefs in the end state, and the number of times the episode has occurred. This follows Tulving's [15] notion of an episode being a mental construct composed of a sequence of observed changes, with subjectively defined start and end points. For example, in Blocks World, the system might have in its episodic memory an episode like the following:

$$
\begin{aligned}
&episode(state_1, state_2, ..., state_n)\\
&\quad start\text{: } ((on\ a\ b))\\
&\quad end\text{: } ((not\ (on\ a\ b)))\\
&\quad significant\text{-}beliefs\text{: } ((not\ (on\ a\ b)))\\
&\quad count\text{: } n
\end{aligned}
$$

Observe that the head of the episode contains a list of pointers to belief states stored in the cache where this episode occurred. The associated intention for the episode can be found by following the pointer to the cache and looking at the executed intention at that state. The start and end fields define the state sequence boundaries for the episode. Theoretically, the start and end points can be arbitrarily far apart, but they are assumed to be consecutive in the current implementation. The significant beliefs field contains all the positive and negated beliefs that violated the agent's expectations and, therefore, are deemed interesting to the agent.

The last remaining component of Icarus's episodic memory, namely, the episodic generalization tree, groups episodes according to their similarities in a hierarchy. The root node of this tree is the most general episode that can instantiate all other episodes. When a new episode is created, the system checks the generalization tree from the root node to find where to insert the episode. If there exists a generalized episode in the tree that can instantiate the new episode, the system attempts to insert the new one under this generalized episode. If any of the children can, again, instantiate the new episode, Icarus moves to its children. If a child node is identical to the new episode, the system simply increment the count for that child episode. Otherwise, the new episode is inserted as a new child episode.

Once the insertion of a new episode happens, the system searches for any possible generalizations among the siblings of the new episode. This might result in a new generalized episode that covers a subset of the sibling episodes. This allows the system to maintain a series of generalized episodes at different levels for each fully instantiated episode at the leaf node, which Icarus can use when generating explanations at different levels of abstractions. We believe the episodic memory structures and processes we reviewed so far facilitates explainable autonomy in Icarus agents. In the next section, we continue our discussion in this direction.

## 4 Episodic Memory for Explainable Autonomy

Episodic memory is a domain independent case-based reasoning mechanism. In this section, we provide details about the episodic phenomena we wish to model, and present some theoretical postulates to explain them. We provide insight on feasible cognitive structures for achieving explainable autonomy using episodic memory, thereby making clear the relationship between episodic memory and case-based reasoning. Lastly, we present preliminary evidence showing that agents with episodic memory capabilities can in fact be explainable agents. The behaviors we think are most suited for explainable autonomy are:

***Summarization*** People summarize their past experiences at appropriate levels of abstraction. They discuss information with appropriate details, rather than spew low-level information about quotidian happenings that may overwhelm their counterparts.

***Question answering*** People are also able to answer a variety of questions about their internal decision making processes and behaviors. Question answering is not a one-off behavior. People can provide more detailed information when asked, and they can also answer follow-up questions about their goals beliefs and intentions at varying levels of detail.

***Prospective thinking*** Lastly, people can think about and discuss with others what they might do in the future. Psychological evidence from [2] shows that young children's ability to think about their future depends on what they remember about their past.

We would like to create agents that demonstrate this range of behaviors. We supplement the behavioral aspects related to explainable autonomy with theoretical postulates that attempt to explain them:

**Explainability is facilitated by episodic memory contents.**
An agent records its personal experiences as episodes, which are stored and organized inside its episodic memory. Processing episodic memory contents allows an agent to expose details about its past experience.

**These contents describe an agent's internal and external states.**
Episodes contain descriptions of an agent's beliefs about the world as well as descriptions of the agent's goals and intentions for achieving its ends.

**Episodic memory structures are domain independent.**
Cognitive structures such as goals, beliefs, intentions, and episodes are abstract symbolic structures that are grounded eventually.

The first postulate in the theory emphasizes the role of knowledge and information processing in explainable autonomy. In contrast to popular machine learning techniques, the cognitive systems approach recognizes that explainable agency is not purely a matter of inputs and outputs, but multi-step mental processes that involve many different modules in the cognitive architecture. The second and the third postulates expand the first by specifying some representational properties of the mental structures reasoned over by episodic memory processes. Next, we present some preliminary findings that show an agent explaining its behavior in a computer game, Minecraft.

## 5 Icarus Agent for Explainable Autonomy

In this paper, we focus our attention on creating agents that answer questions about their past. Answering questions requires that an agent retrieve episodic memory contents, process the rememberings to evaluate their relevance to the question, and synthesize a response from the relevant information. The episodic memory in Icarus supports cue-based retrieval to allow the agent to recall its personal past. Retrieval operates on the generalization tree and is triggered deliberately. Deliberate retrieval is goal-directed, meaning that the system has skills for remembering. For instance consider the skill for answering how a goal was accomplished:

*inform-how*(?*o*1)
  *elements: ((question ?o1 type goal how t))*
  *conditions: ((uninformed ?o1)*
  *actions: ((*explain-goal-achievement ?o1)*
  *effects: ((not (uninformed ?o1))*

This skill allows an agent to a question about how it achieved a goal. Given that the questioner is wants to know how the goal was achieved, if the system

explains how, then the questioner will no longer be uninformed about the goal. The action **explain-goal-achievement* involves the agent retrieving episodes from its memory, and synthesizing text that gets output to the screen. To clarify the episodic nature of these explanations, consider Algorithm 1 which shows how the agent retrieves episodes that match a goal description. Observe that on line 5, the episode cache pointers are being collected, so the agent can recover its previously executed intentions.

**Algorithm 1** REMEMBER($\rho, \tau, goal$)

```
1: remembered ← *RETRIEVE(τ, goal)
2: idx ← ∅
3: ints ← ∅
4: for all episode in remembered do
5:     collect episode.cache-pointers into idxs
6: for all idx in idxs do
7:     executed-intention ← ρ.intentions[idx]
8:     if not NULL(executed-intention) then
9:         collect executed-intention into ints
10: return ints
```

We tested our explainable agent in the popular video game Minecraft[12]. Minecraft is an open-world, sandbox game. Players can create and experience new worlds by collecting resources, building structures, discovering new lands, etc. We placed our agent in a room with a zombie, and told the agent to defend itself against the threat. After the agent kills the zombie, we asked the agent a series of questions about what happened. The following is a transcript of the interaction, where we present English translations of the questions in bold and then the agent's response. On any given question, the system describe all experiences that are relevant, but does not make clear the sequential relationship amongst its experiences.

**How did you achieve a situation where the zombie1 is not present?**

- `I did (ATTACK ZOMBIE1) to achieve (NOT (ENEMY ZOMBIE1)).`
- `I did (KILL ZOMBIE1) to achieve (NOT (ENEMY ZOMBIE1)).`

In this response, the system exposes that it did two different procedures for clearing the room of the zombie. Next we discover why two different procedures were followed.

**Why did you attack zombie1?**

- `I executed (ATTACK ZOMBIE1) because I wanted to achieve ((NOT (ENEMY ZOMBIE1))).`
  `All I had to do make it executable was to have:`
  `((ARMED-AND-DANGEROUS ME) (NEXT-TO ZOMBIE1 ME))`

**Why did you kill zombie1?**

```
– I executed (KILL ZOMBIE1) because I wanted to achieve:
  ((NOT (ENEMY ZOMBIE1))).
  All I had to do make it executable was to have:
  ((ARMED-AND-DANGEROUS ME))
  I could have done (ATTACK ZOMBIE1), but at the time
  ((NEXT-TO ZOMBIE1 ME) (ARMED-AND-DANGEROUS ME))
  were unsatisfied conditions.
```

Notice how the system reveals that there are two different ways to rid the room of a zombie. Notice also that the agent makes known alternatives that it could have chosen and explains why they were not selected. Next, we find out about the structure of (KILL ?O1) and (ATTACK ?O1).

**How did you actually attack zombie1?**

```
– I executed (ATTACK ZOMBIE1) by performing motor functions:
  ((*ATTACK 'ZOMBIE1))
```

The attack skill specifies low-level details on what motor functions to perform to complete the task.

**How did you actually kill zombie1?**

```
– I executed (KILL ZOMBIE1) by following steps:
  ((OFFENSIVE-GO-TO ZOMBIE1) (ATTACK ZOMBIE1))
```

The agent reveals that (KILL ?O1) is a more abstract description of how to remove a zombie from a room than attack, because (ATTACK ?O1) is contained as a subskill of (KILL ?O1). In Minecraft, zombies move around or move to the player they are trying to eat so, at times the more generic (KILL ?o1) procedure is applicable, and other times, the zombie is in a situation, where the system can directly execute primitive actions to achieve its ends.

**Why did you nominate being armed and dangerous as a goal?**

```
– I nominated (ARMED-AND-DANGEROUS ME) because it was a necessary
  condition for (KILL ZOMBIE1).
```

In this question, the agent is explain its reasons for trying to achieve situations in the world. In the current implementation, goals are either given to the agent as a top-level goal, or the system can nominate goals to achieve which it thinks will allow it to achieve its top-level goal.

Our ICARUS agent is able to describe details about its decision making process that allow a user to predict its behavior and make expectations about what tasks the system can accomplish. Observe that a system without episodic memory capabilities cannot answer questions about its behavior in this level of detail. So, it seems that researching the role of episodic memory and its related processes is a promising direction for creating explainable collaborative agents.

## 6 Related Work

Efforts to create systems that explain their internal motivations and justify their actions are not new [4, 14]. The case-based reasoning community has a rich history of building such systems, so we review some of these systems here. We begin with work in case-based reasoning, then move to discuss episodic memory in cognitive systems.

One early work in case-based reasoning attempted to explain how the system designed physical devices [5]. This work used meta-cases to store a trace of the cognitive processing done during problem solving. These meta-cases could be retrieved to explain to a user the reasoning behind the agent's design choices. This research is closely related to our work, but unlike their system, our work uses a domain independent episodic memory.

Work by Molineaux and Aha [13] describes a goal reasoning system with episodic memory-like capabilities for learning event models. While the system bears much resemblance to our work including its ability to be surprised, capacity to generate explanations using episodic memory-like structures, the system is only used for learning event models. Explanations are used internally and are comprised of observation-action sequences. The system does not generate explanations of its own behavior. All models that the system learns are related to explaining exogenous events, and the system does not remember its actions in in relation to the goals it was persuing.

There are a few other work that incorporated episodic memory into a cognitive architecture. Most notably, Soar has an episodic memory that contains sequentially ordered snapshots of the agent's working memory. Although their work does not make theoretical commitments on how episodic memory explains human behavior, it does present a series of design decisions that are inspired by psychological evidence.

## 7 Future Directions

Our aim is to create systems that can explain their behavior to humans in a natural way. Now that we have presented our theoretical position on the role of episodic memory in explainable autonomy, and shown how the theory can be implemented, we propose a research agenda that can guide our efforts towards meaningful progress.

We would like to turn our attention to summarization. To some extent, our agent can already explain its behavior in an abstract manner because it can abstract away low-level details when describing how it killed a zombie, but we want our system to synthesize its history into summaries. Much like in story telling, we want to design systems can that tell users stories about the events they experience.

Lastly, we would like our agent to accept feedback from users about how to make decisions. And, in conjunction with modeling these behaviors, we should simultaneously devise and debate theories that explain these behaviors. In so doing, we advance our understanding of the nature of intelligence.

## 8 Conclusions

In this paper, we argued that episodic memory capabilities allow intelligent agents to explain their internal decision making processes to users. We adopted the cognitive systems paradigm and showed evidence that supports our claim. Additionally, we also provided a theory of episodic memory that explains aspects of how people think about their personal future, summarize events, and answer questions about heir past. We also showed how the implementation of the episodic memory in Icarus has been guided by psychological evidence. We also provided some thoughts on promising directions that could yield new insights into explainability.

## References

1. Anderson, J.R., Lebiere, C.: The Atomic Components of Thought. Erlbaum, Mahwah, NJ (1998)
2. Atance, C.M.: Young children's thinking about the future. Child Development Perspectives **9**(3), 178–182 (2015)
3. Baddeley, A.: The episodic buffer: a new component of working memory? Trends in Cognitive Sciences **4**(11), 417–423 (2000)
4. Doyle, D., Tsymbal, A., Cunningham, P.: A review of explanation and explanation in case-based reasoning. Tech. rep., Trinity College Dublin, Department of Computer Science (2003)
5. Goel, A.K., Murdock, J.W.: Meta-cases: Explaining case-based reasoning. In: European Workshop on Advances in Case-Based Reasoning. pp. 150–163. Springer (1996)
6. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA) (2017)
7. Johnson, M., Hofmann, K., Hutton, T., Bignell, D.: The Malmo platform for artificial intelligence experimentation. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 4246–4247. IJCAI/AAAI Press, New York (2016)
8. Kurby, C.A., Zacks, J.M.: Segmentation in the perception and memory of events. Trends in Cognitive Sciences **12**(2), 72–79 (2008)
9. Laird, J.E.: The Soar Cognitive Architecture. MIT Press, Cambridge, MA (2012)
10. Langley, P., Choi, D.: A unified cognitive architecture for physical agents. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence. pp. 1469–1474. AAAI Press, Boston, MA (2006)
11. Ménager, D., Choi, D.: A robust implementation of episodic memory for a cognitive architecture. In: Proceedings of the Thirty-Eighth Annual Conference of the Cognitive Science Society. pp. 620–625. Cognitive Science Society, Austin, TX (2016)
12. Microsoft: Minecraft. https://minecraft.net/en-us/ (2009)
13. Molineaux, M., Aha, D.W.: Learning unknown event models. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 395–401. AAAI Press, Quebec City (2014)
14. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning–perspectives and goals. Artificial Intelligence Review **24**(2), 109–143 (2005)
15. Tulving, E.: Elements of Episodic Memory. Oxford University Press (1983)

# Application of Case-based Explanations to Formulate Goals in an Unpredictable Mine Clearance Domain

Venkatsampath Raja Gogineni, Sravya Kondrakunta, Matthew Molineaux and Michael T. Cox

Wright State University, Dayton OH 45435, USA
Gogineni.14@wright.edu

**Abstract.** In many domains of expertise, having a pre-defined case base of explanations is a storage-light and computationally cost-effective way to generate most probable explanations of unusual events and problems. These explanations can help an agent to generate goals in response to developing problems, and to describe the motivations behind new goals or changes to old goals for the benefit of third parties. This paper presents an agent that uses case-based explanations and goal generation to improve its success in locating and responding to underwater mines in a critical channel.

**Keywords:** Case-base explanation, goal reasoning, explanation patterns.

## 1 Introduction

Agents in a mine clearance domain must perform critical surveillance tasks with high accuracy in deep waters where communication and observability are limited. Due to unpredictable and dangerous events such as explosions in unexplored areas, intelligent behavior is required to understand and respond to the environment. In this domain, explanation is useful for both monitoring the environment and engendering trust in human operators who have only intermittent contact with the agent. In this paper, we consider the problems of understanding, reacting, and communicating for a deliberative mine hunting agent that must respond to such unpredictable events.

Our agent for this domain is called *GATAR (Goal-driven Autonomy for Trusted Autonomous Reasoning)*. GATAR plans to achieve its goals, then executes each step in this plan after checking to confirm that its preconditions are met. The actions constitute GATAR's *expectations* about the world, when they do not match the current observations of the world, it tries to recognize their causes and predict their effects on the agent's goals. These cognitive capabilities provide two benefits: first, they help GATAR to intelligently respond to such events and prevent their recurrence; second, they help GATAR communicate the rationale behind its behavior to third parties. This second benefit is critical to building trust when working with humans [1]. While the GATAR agent is the primary focus of this paper, we expect lessons learned and results to be generalizable; we expect intelligent explanation-based behavior with deliberately selected goals to be useful in other critical domains like surveillance, medicine and autonomous driving.

One approach to responding to unpredictable events is to generate contingent plans in advance that cover all possibilities. Unfortunately, this is computationally intractable

in most domains and handling all the plan possibilities for an unexpected event can be overwhelming. However, in any specific domain of interest, an abstractly defined case-base of explanations by domain experts can be retrieved and adapted to explain the unpredictable event.

In this paper we present an approach by which an agent can respond to unpredictable events using case-based reasoning [20] for explanation. Here explanation provides a causal basis for goal generation and enables communicative rationale for goal changes to third parties. This approach follows *Goal Directed Autonomy (GDA)* principles [2,3,4,5,6], in which agents respond to *discrepancies* (i.e., agent expectation failures) by generating explanations and generating goals based on those explanations.

In Section 2, we describe a mine clearance domain along with the goals of an agent, followed by a discussion of unpredictable events. A description of explanatory cases, their retrieval and an approach for goal formulation follows in Section 3. Section 4 presents the evaluation and empirical results, related research is in Section 5. The conclusion and future scope completes the paper in Section 6.

## 2 The Mine Clearance Domain

Our approach is implemented in the mine clearance domain, which is simulated using MOOS-IVP [8], software that provides complete autonomy for marine vehicles.

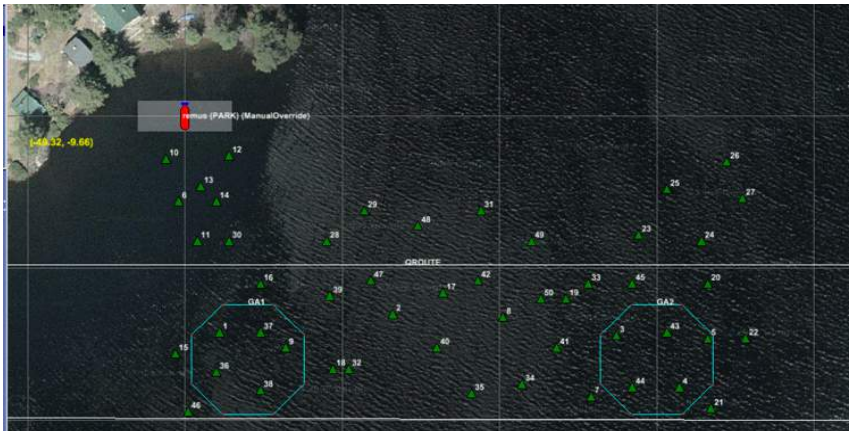



*Figure 1 Mine Clearance Domain with two clearance areas in the Q-route*

Figure 1 shows the simulation of the mine clearance domain with a GATAR agent directing a Remus unmanned underwater vehicle. The Q-route is a safe passage for ships to enter and leave the port and is represented as a rectangular area in Figure 1. GA1 and GA2 are the two octagonal shaped areas where mines are expected to exist, while the triangular shaped objects are the mines. The goals of the agent are to survey and clear mines in GA1 and GA2. The Remus has a sonar sensor with a specific width of 10 units and a length of 5 units to detect mines. This domain is setup in such a way that the mines are manually placed in and outside the Q-route.

### 2.1 Problems and Unpredictable Events in MOOS-IvP

In the mine clearance domain, several events often co-occur simultaneously, and many events cannot be predicted based on knowledge available to an agent. These events might affect the agent itself or the mission of the agent. Explanations help the agent to recognize these events, decide whether they are problematic, and respond to them. Specific responses are not within the scope of this paper; however, we will look at several events which will and will not be a problem to the agent.

In this domain, an enemy ship lays mines randomly around the area to hurt friendly ships. A mine in a Q-route presents a problem because friendly ships will subsequently traverse the Q-route and may be damaged by such a mine. Removing such mines is an explicit goal for GATAR within areas GA1 and GA2. However, a mine discovered outside the Q-route or within it but outside of GA1 and GA2 might or might not be a problem to the agent. The agent must decide the difference between the two, and explanation is used for these decisions.

## 3 Case-based Explanation Patterns

In our work, each case in GATAR's case-base is an abstract *explanation pattern (XP)* [7] engineered by experts for a specific domain. An *XP* is a data structure that represents a causal relationship between two states and/or actions; each action/state is abstractly defined with variables to be adapted during or after case retrieval. In GATAR, an action or state is referred to as a *node* and different types of nodes are described based on their role in an XP, as follows:

- *Explains node*: An unpredictable action/state that is observed.
- *Pre-XP node*: Action/state that is observed along with the explains node.
- *XP-asserted node*: An action, state, or XP that contributes to the explanation's cause. In the case of a causing XP, the effects of the cause XP can be seen as direct causes, and the causes of that XP indirect causes, of the effect XP.
- *Internal node*: Action/state that links the XP-asserted nodes with both the explains and the pre-XP nodes.

An explanation pattern represents a causal structure in which XP-asserted nodes form an antecedent, and a consequent is made up of pre-XP nodes and an explains node; we say that XP-asserted nodes cause the associated explains and pre-XP nodes. The internal nodes form the connecting links between antecedent and consequent.

### 3.1 Retrieving, Reusing and Revising Explanation Patterns from a Case-base

Case-based reasoning follows a formal four-step process to retrieve, reuse, revise and retain cases [20]. Since we assume that the cases defined by the experts in the domain serve our purpose, retention is not much of a concern in this domain. However, *XP's* are retrieved, reused and revised. In GATAR, explanations are retrieved by the *Meta-AQUA* component [9], a story understanding system that tries to explain discrepancies in a story through use case-based explanations. We have integrated this system with the *MIDCA (Metacognitive Integrated Dual Cycle Architecture)* component [10], a cognitive architecture that perceives and acts directly on the world, to examine the interaction between explanation generation (by Meta-AQUA) and goal formulation

(provided by MIDCA); for the purposes of this paper, we refer to the combined system as the GATAR agent.

Actions and states in the world are perceived by the MIDCA component and sent to Meta-AQUA in the form of a *story*. A story with an unpredictable event retrieves an abstract *XP* from the case-base. However, a story here is not a natural language text but a frame representation of action and state pairs. However, MIDCA identifies an event as an unpredictable event if it does not match the *expectations* of the world observations by the agent.

An abstract *XP* is retrieved when Meta-AQUA unifies its explains node with each new unpredicted state or action, if the unification turns out to be true then the pre-XP nodes of the corresponding case are unified with the observations of the corresponding states or actions from the story, if they turn out to be true then the specific *XP* is retrieved. Moreover, the retrieved abstract *XP* is reused by adapting the variables in the consequent with the world observations and back-chaining the variables to fill the antecedents. However, if the XP-asserted nodes in the reused *XP* contain hypothetical information they can be revised when the new knowledge is obtained from the story.

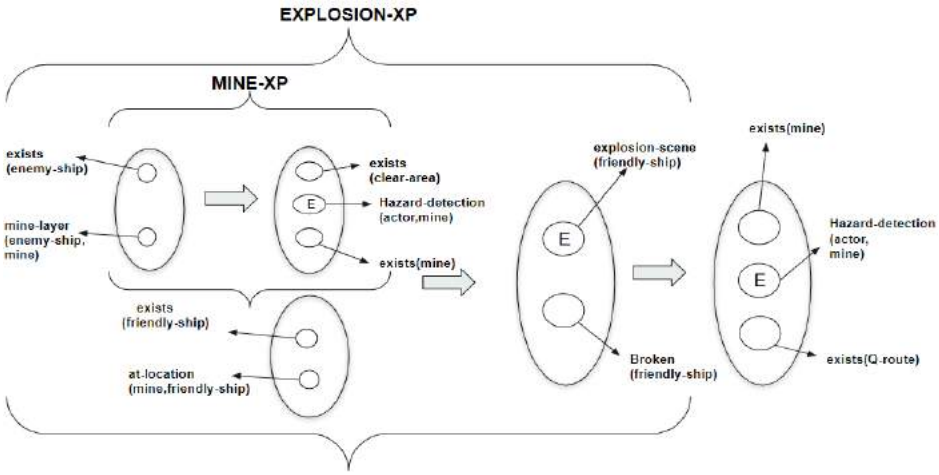



*Figure 4 Problem -XP*

Figure 4 depicts an *XP* named PROBLEM-XP from the case-base for the mine clearance domain. Although retrieval of this *XP* follows the same procedure discussed above, composite *XPs* such as this merit further discussion and have some exceptions. An *XP* within an *XP* can act as a node, although it might not be an action/state but all the actions/states in an *XP* falls under XP-asserted nodes when looked at the entire composite *XP*. However, this is not the case when looking at them individually. PROBLEM-XP refers to a simpler explanation pattern, MINE-XP, which can be used to explain a hazard-detection action in a clear-area, which is an unpredictable event. In this XP, hazard-detection is an explains node, which forms part of the antecedent of MINE-XP, along with the pre-XP nodes clear-area and mine, which describe state that is observed alongside the hazard detection when MINE-XP applies. The antecedent of

MINE-XP is composed of a mine-layer action, enemy-pilot state, and enemy-ship state, all of which are XP-asserted nodes.

EXPLOSION-XP in Figure 4 is used to explain the explosion of a friendly ship. In this XP, the consequent is composed of an observed action explosion-scene(friendly-ship) (the explains node) and a state of broken(friendly-ship), a pre-XP node that represents broken pieces of a ship. The antecedent of EXPLOSION-XP includes three XP-asserted nodes: the hazard-detection action and the state nodes "exists(friendly-ship)" and "at-location(mine, friendly-ship)". Note that a hazard-detection action is used as an XP-asserted node by EXPLOSION-XP, and as an explains node by MINE-XP; this allows them to chain together.

Finally, PROBLEM-XP explains that there is a problem when a mine coexists with a Q-route; a problem exists in this situation because friendly ships will travel through a Q-route and be damaged by the mine. The antecedent of PROBLEM-XP is EXPLOSION-XP. PROBLEM-XP is specialized and retrieved from the case base by GATAR whenever a story from MIDCA unifies with PROBLEM-XP's explains and pre-XP nodes.

```
(story-input
(survey.4
    (actor  (value remus))
    (location (value ga1)))
((state
    (at_location.7  (co-domain  (value transit))
                (domain  (value remus)))
    (enabled.4  (domain  (value remus))
    (hazard_at_location.1  (co-domain  (value transit))
                        (domain  (value mine6))))

(hazard-detection.1    (actor  (value remus))
                          (location  (value transit))
                          (object  (value mine6)))"")
((state
(at_location.8  (co-domain  (value transit))
                  (domain  (value remus)))
(enabled.5  (domain  (value remus)))
(hazard_at_location.2  (co-domain  (value transit))
                          (domain  (value mine6))))
)
```

*Figure 5 A story for detecting mine at the Q-route.*

Figure 5 gives a story representation for a mine detection in a surveilled Q-route by a Remus in area GA1. This story consists of a series of actions, each of which is followed by a world state observed by MIDCA.. Note that a similar story of mine detection that occurs during transit, but not in a Q-route, will not retrieve PROBLEM-XP but will retrieve MINE-XP.

```
For the state frame PROBLEM-XP.12593:

The VALUE facet of the ACTOR role is REMUS.12595.
The VALUE facet of the OBJECT role is MINE.12597.
The VALUE facet of the LOCATION role is QROUTE.12599.
The VALUE facet of the EXPLAINS role is HAZARD-DETECTION.12621.
The VALUE facet of the PRE-XP-NODES role is (MINE.12597 QROUTE.12599).
The VALUE facet of the INTERNAL-NODES role is NIL.0.
The VALUE facet of the XP-ASSERTED-NODES role is (EXPLOSION-XP.12601).
```

*Figure 6 Part of Retrieved PROBLEM-XP*

Figure 6 shows a specialized PROBLEM-XP with all the placeholders of the observations in the natural language fed to MIDCA. REMUS.12595 is the specific instance of the Remus in the story; similarly, all nodes in Figure 6 are specialized observed actions/states including the antecedent EXPLOSION-XP whose representation can be expanded and will be discussed in the section 4.2.

### 3.2 Goal Formulation from a Retrieved XP

Goal formulation is essential for an intelligent agent to respond to unpredictable events; in GATAR, we perform formulation by chaining backward on each of the antecedents of retrieved XPs until we reach all the antecedents that GATAR can respond to. Antecedent nodes include actions and or states; therefore, when GATAR wishes to prevent an undesired consequent from recurring, it considers as potential goals the elimination of antecedent actors or objects that participate in antecedent states. The potential goals are generated using the removal mapping function that takes in the actors or objects and outputs the goals that eliminate them.

```
For the state frame EXPLOSION-XP.12601:
 (Which is in the ANTECEDENT slot of frame PROBLEM-XP.12593)

The VALUE facet of the ACTOR role is SHIP.12603.
The VALUE facet of the OBJECT role is MINE.12597.
The VALUE facet of the LOCATION role is QROUTE.12599.
The VALUE facet of the EXPLAINS role is EXPLOSION-SCENE.12653.
The VALUE facet of the PRE-XP-NODES role is (BROKEN.0).
The VALUE facet of the INTERNAL-NODES role is NIL.0.
The VALUE facet of the XP-ASSERTED-NODES role is (MINE-XP.12607 AT-LOCATION.12670 SHIP.12603).
```

*Figure 7 Specialized EXPLOSION-XP from the PROBLEM-XP*

In the mine domain, future PROBLEM-XP events are prevented by preventing the antecedent EXPLOSION-XP. Figure 7 describes a specialized XP, EXPLOSION-XP.12601; we consider its antecedents, AT-LOCATION.12670, and MINE-XP.12607, to discover potential goals.

```
For the state frame AT-LOCATION.12670:
 (Which is in the ANTECEDENT slot of frame EXPLOSION-XP.12601)

The VALUE facet of the DOMAIN role is SHIP.12603.
The VALUE facet of the CO-DOMAIN role is MINE.12597.
```

*Figure 8 Specialized AT-LOCATION from the EXPLOSION-XP*

AT-LOCATION.12670 in Figure 8 has the domain SHIP.12603, which represents the friendly ship, and the co-domain MINE.12597, which represents the mine that caused the explosion in the Q-route. Two possible goals that respond are to either 1) prevent the ship from arriving at the Q-route or 2) remove the mine from the Q-route. Since GATAR has the means to remove a mine, it feeds the removal mapping function with the mine to generate a goal *cleared_mines(Q-route)* to remove the mine from the Q-route.

```
For the state frame MINE-XP.12607:
 (Which is in the ANTECEDENT slot of frame EXPLOSION-XP.12601)

The VALUE facet of the ACTOR role is REMUS.12595.
The VALUE facet of the OBJECT role is MINE.12597.
The VALUE facet of the LOCATION role is QROUTE.12599.
The VALUE facet of the EXPLAINS role is HAZARD-DETECTION.12621.
The VALUE facet of the PRE-XP-NODES role is (QROUTE.12599 MINE.12597).
The VALUE facet of the INTERNAL-NODES role is NIL.0.
The VALUE facet of the XP-ASSERTED-NODES role is (MINE-LAYER.12612 ENEMY-PILOT.12614 ENEMY-SHIP.12618).
```

*Figure 9 Specialized MINE-XP of the frame EXPLOSION-XP*

The consequent of MINE-XP, as shown in Figure 9, has already happened: HAZARD-DETECTION.12621. To prevent MINE-XP, GATAR considers removing both the actor "enemy-pilot" and "enemy-ship" who performed the MINE-LAYER.12612 action by feeding them to the goal removal mapping function to obtain the goals *apprehend(enemy-pilot)* and *apprehend(enemy-ship)*.

### 3.3 Explaining Goals to Third Parties

Having a causal knowledge structure such as an XP to formulate goals provides an ability for an agent to explain the basis of having a goal to third parties. Although the communication is not in the form of a natural language sentence but the form of an antecedent causing a consequent deliberately provides the third parties with the source actions and states for the cause of unexpected events and the motivations behind the agent to generate a goal. The GATAR has the ability to provide a specialized XP when queried by third parties for the basis of having a generated goal.

## 4 Evaluation of Case-based Explanation in the GATAR Agent

As previously mentioned, GATAR retrieves explanatory cases with the Meta-AQUA component, a system that explains discrepancies (various types of expectation failures [21]) in an input. The Remus unmanned underwater vehicle in Figure 1 is controlled through MOOS-IvP simulator by MIDCA, which repeatedly sends these actions and observations of the environment to Meta-AQUA in the form of a *story*. Whenever unpredictable events occur, Meta-AQUA retrieves a case, adapts it to the given situation and sends the adapted explanation to MIDCA. MIDCA then reasons about the cause of the specialized explanation and generates corresponding goals to achieve in the world. GATAR's two main goals are to clear mines in two specific areas (GA1 and GA2). For purpose of measuring performance, we award GATAR one point for each mine neutralized within the Q-route, whether inside or outside GA1 and GA2.

However, mines outside the Q-route do not improve its score. We use this metric to examine how the performance of GATAR is influenced by explanation generation and goal formulation.

The evaluations are performed according to the above defined evaluation metrics. We compared two agents: GATAR using case-based explanation and the goal generation approach, and a baseline MIDCA agent that performs neither, ignoring unexpected events entirely. The agents are evaluated based on their performance during mine clearance using different deadlines. Table 1 depicts the number of mines cleared by the agent with and without explanations for various time limits. The time limits vary from 10 to 80 with increasing step size of 10, so there are 8 observations in total for each case. As we can see GATAR performed identically to the baseline agent with a short deadline, but with more time performed better. Differences in performance between the two agents can be explained by the fact that as time goes on, more unexpected mines are encountered in the Q-Route, which GATAR responds to, but the baseline does not.

*Table 1 Comparing GATAR's performance with and without explanations.*

| | **Performance (# mines cleared)** | |
|---|---|---|
| **Deadline (time)** | **GATAR** | **Baseline** |
| 10 | 2 | 2 |
| 20 | 4 | 4 |
| 30 | 6 | 4 |
| 40 | 8 | 6 |
| 50 | 10 | 8 |
| 60 | 12 | 10 |
| 70 | 15 | 10 |
| 80 | 17 | 10 |

## 5 Related Work

Explanation patterns were introduced by Schank in 1982 [7] and were later used in the story understanding systems SWALE [11] [12] and AQUA [13]. SWALE is a case-based approach to explanation of discrepancies in a story that retrieves, adapts and stores explanation patterns. SWALE demonstrated an early technique for ruling out competing explanations using memory knowledge. AQUA (Asking Questions and Understanding Answers) operates by first questioning missing knowledge in a story, then using explanation patterns to understand the answers.

Roth-Berghofer et al's [14] work on classifying explanations and their use-cases according to the user's intentions is one of the theoretical research directions towards explanations in case-based reasoning (see also [22]). This paper introduces the concept of "explanation goals" that are used to decide when and what the system should explain to users based on their expectations. In future research, we will investigate application of these techniques to prevent the system from repeatedly explaining the same type of unexpected events to a user who is already familiar with them. This paper

also talks about different kinds of explanations and classifies them into four different knowledge containers, all of which are used to generate explanations based on the user's goals or intentions.

In [1], a robot adapts its behavior to gain trust in human machine teaming using the approach of case-based reasoning. In addition, Floyd and Aha [15] presented an approach to explain such adaptations based on an operator's feedback, and evaluated their system based on how closely the explanations aligned with the operator's feedback. Our interest in generating explanations of the intelligent behavior of an agent aligns closely with the interests of this paper.

There has also been a lot of research on goal driven autonomy. In [2], Paisner et al. argue that whenever a discrepancy is found, an agent needs to explain the anomaly and that explanation should be used to generate goals. This concept is demonstrated in a modified blocks world domain. Their goal generation system uses different types of goals, including "K-track goals", which come from specific knowledge structures such as action models and explanation patterns, and "D-track goals" are implicit in the system. An A-star statistical inference method is used to generate D-track goals and K-track uses explanation patterns in Meta-AQUA to generate goals.

## 6 Conclusion and Future Scope

In this paper we discussed an approach to generating intermediate goals for unpredictable events using case-based explanation patterns in the underwater domain. Having a causal structure, these explanation patterns are human-understandable, so they help to communicate why GATAR chooses intermediate goals.

In future work, we will consider the problem of explanation selection when two or more cases are retrieved for the same unpredictable event. We will also consider the problem of switching explanations when a selected XP is proven false. We will also explore approaches to automated learning of abstract explanation patterns. This will relax the demand for domain engineering and give GATAR flexibility to respond to unknown unpredictable events.

## References


1. Floyd, M. W., Drinkwater, M., Aha, D. W.: Trust-guided behavior adaptation using case-based reasoning. Naval Research Laboratory. Washington, United States (2015).
2. Paisner, M., Cox, M., Maynord, M., Perlis, D.: Goal-driven autonomy for cognitive systems. In: Proceedings of the Annual Meeting of the Cognitive Science Society. Vol. 36. No. 36. (2014).
3. Molineaux, M., Klenk, M., Aha. D. W.: Goal-Driven Autonomy in a Navy Strategy Simulation. AAAI, pp. 1548-1554 (2010).
4. Dannenhauer, D., Munoz, A. H.: Raising Expectations in GDA Agents Acting in Dynamic Environments. IJCAI, pp. 2241-2247 (2015).
5. Cox, M. T.: Goal-Driven Autonomy and Question-Based Problem Recognition. In: Proceedings of the 2nd Annual Conference on Advances in Cognitive Systems, pp. 29-45. Maryland, USA (2013).
6. Munoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., Molineaux, M.: Applying Goal Driven Autonomy to a Team Shooter Game. FLAIRS Conference (2010).

7. Schank, R. C.: Explanation patterns: Understanding mechanically and creatively. Psychology Press (2013).
8. Richard, M.B.: MOOS-IvP Autonomy Tools User's Manual Release 4.2. 1. Massachusetts Institute of Technology. Sea Grant College Program (2011).
9. Ram, A., & Cox, M. T.: Introspective reasoning using meta-explanations for multistrategy learning (1994). In R. S. Michalski & G. Tecuci (Eds.), Machine learning: A multistrategy approach IV (pp. 349-377). San Francisco: Morgan Kaufmann.
10. Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., Perlis, D.: MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy. AAAI (2016).
11. Schank, R. C., Leake, D. B.: Creativity and learning in a case-based explainer. Artificial intelligence 40(1-3), 353-385 (1989).
12. Leake, D. B.: Evaluating explanations: A content theory. Psychology Press (2014).
13. Ram, A.: AQUA: Questions that drive the explanation process. Georgia Institute of Technology (1993).
14. Roth-Berghofer, T. R., Jörg Cassens.: Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In: International Conference on Case-Based Reasoning. Springer, Berlin, Heidelberg (2005).
15. Floyd, M. W., Aha, D. W.: Incorporating transparency during trust-guided behavior adaptation. In: International Conference on Case-Based Reasoning. Springer, Cham (2016).
16. Cox, M. T., Burstein, M. H.: Case-based Explanations and the Integrated Learning of Demonstrations. Künstliche Intelligenz journal, 22(2):35-37 (2008).
17. Cox, M. T., Ram, A.: Introspective multistrategy learning: On the construction of learning strategies. Artificial Intelligence 112(1-2), 1-55 (1999).
18. Cox, M. T.: Introspective multistrategy learning: Constructing a learning strategy under reasoning failure. PhD thesis. Georgia, Atlanta (1996).
19. Johnson B., Floyd M.W., Coman A., Wilson M.A., Aha D.W. Goal Reasoning and Trusted Autonomy. In: Abbass H., Scholz J., Reid D. (eds) Foundations of Trusted Autonomy. Studies in Systems, Decision and Control, vol 117. Springer, Cham (2018).
20. de Mántaras, R. L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A., & Watson, I. (2006). Retrieval, reuse and retention in case-based reasoning. Knowledge Engineering Review, 20(3), 215-240.
21. Cox, M. T. (1997). An explicit representation of reasoning failures. In D. B. Leake & E. Plaza (Eds.), Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning (pp. 211-222). Berlin: Springer.
22. Aamodt, A. (1994). Explanation-Driven Case-Based Reasoning. In S.Wess, K.Althoff, M.Richter (eds.): Topics in case-based reasoning (pp 274-288).Berlin: Springer.

# A Theoretical Model of Explanations in Recommender Systems *

Marta Caro-Martinez, Guillermo Jimenez-Diaz, and Juan A. Recio-Garcia

Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Spain
email: martcaro@ucm.es, gjimenez@ucm.es, jareciog@ucm.es

**Abstract.** Explanations in recommender systems are essential to improve user confidence in recommender systems. In this work, we propose a theoretical model to categorize explanations in recommender systems. Although this work is sustained by previous explanation taxonomies, our model includes concepts not considered in current literature. Moreover, we make a novel contribution regarding the formalization of this model, as our long-term goal is to build an ontology that will be integrated into a development methodology to guide the implementation of explanations in recommendation systems.

## 1 Introduction

Nowadays, people can find a huge amount of information on the Internet. With the appearance of online shops, like Amazon or eBay, and platforms of entertainment consumption, like Spotify or Youtube, internet users are able to get products in an easy and fast way. However, the amount of products offered by these platforms is immense and it can hinder the task of finding the product that the user really wants. Recommender systems alleviate this problem. They help users to find products that can be interesting to them. Thus, the users make better decisions thanks to the recommendations provided by the system.

Many times, users do not trust recommender systems as they do not know how the recommendation has been carried out and the reasons why a product has been recommended. This problem causes a bad user experience and the system is not used as we would expect. Therefore, explanations for justifying recommendations are necessary for helping the user to understand the system behaviour.

The goal of our work is to introduce a new categorization model for explanation systems in order to help to design successful explanations. Our approach provides a refinement with respect to previous models studied during our work. We enhance those models with some concepts not considered in previous works and we include specifications of novel concepts. The ideas and weaknesses found in current literature are specified in Section 2.

* Supported by the Complutense University of Madrid (Group 910494) and Spanish Committee of Economy and Competitiveness (TIN2017-87330-R).

Additionally, our current work follows our group research line [25, 15] presenting a first attempt to formalize a model that includes the semantic description of explanations in recommender systems. The resulting ontology, together with the RecoLibry framework will be the basic buildings blocks of a future platform for the generation of recommender systems analogous to COLIBRI Studio.

The paper is structured as follows. Section 2 reviews state-of-the art literature about explanations in recommender systems. Next, Section 3 describes our model. Finally, Section 4 describes the way the model will be validated and Section 5 concludes the paper.

## 2 Background

In this work, we have studied several publications where we can find different ways of explaining recommendations. Some of them are studies that propose ways of classifying explanations [14, 21, 12, 30, 22, 29, 6, 10].

First, we studied the survey by [30], where we found an analysis of the different purposes of explanations in recommender systems. The survey describes the criteria that *good* explanations should have, different ways of presenting recommendations and how users interact with recommender systems. In [12], we also found the properties described by Tintarev *et al.* but applying this knowledge on different ways of presenting the explanations for recommendations. Thanks to this study and the proposal in [14], that describes several explanation approaches for a movie recommender system, we found new ways of visualizing explanations, a facet that we will consider in our model. The work in [6], compares two models of explanation: a normative model and a pragmatic model. The normative model is focused on transparency and user understanding, while the pragmatic model is designed for usability. This approach helps us to refine features that our model should have. Finally, we studied several explanation taxonomies, proposed in [21, 22, 10]. The most detailed one is the systematic review detailed in Nunes *et al.* It presents detailed descriptions of different explanation types according to many aspects. [22] present new concerns about explanations in social recommender systems and [10] introduce new facets about the recommendation types and the importance of knowing these types for explaining recommendations. In [29], that provides a study about the effectiveness of explanatory information and how it is evaluated, we can also find some interesting details about recommender systems and types of knowledge sources for explaining recommendations.

Our work is focused on building a classification model for explanations in recommender systems. This model should help to design successful explanations for recommender systems. In order to ease the understanding of the model proposed here, it is necessary to define some important concepts related to recommender systems. The main entities involved in recommender systems are *users* and *items*. *Users* are the people who are interested in getting new items and who interact with the system to explore the products of the platform. *Items* are the products available on the recommendation system: food, clothes, hotels, experiences, etc. On one hand, the interactions carried out by users in the recommendation

system are one of the most exploited sources of knowledge to recommend new items to these users. One way widely used to identify the user preferences is the *ratings*, a form of feedback of the user about an item. If one user rates an item with a good score, then this item (and its features) will be suitable for this user. On the other hand, another source of knowledge involved in the recommendation process is item descriptions. Every item has its own *content*, an extensive representation of the item, for example, a textual description. Moreover, the content can be represented by a set of *features* that defines the item: its price, colour or size, among others.

Frequently, the recommendation is supported by the *similarities* between users or items. The similarities are the coincidences between two elements, for example, between item features or user ratings. Although, it is common to recommend items similar to the ones that the user interacted with before, some recommendation approaches suggest items on the basis of the user *preferences*. The preferences are the user requirements elicited before interacting with the recommender. Moreover, every user is part of the *community* defined by the recommender system. The behaviour of both every single user and the community can be employed in the recommendation process. Finally, the recommendation can be enhanced using the information extracted from the environment and the circumstances of the user and her interaction with third-party systems. We call this as *contextual information*.

## 3 Theoretical Model

Thanks to the study that we have carried out, we propose a new model for classifying explanations, which introduces four main classification facets: motivation, knowledge containers, generation and presentation. Figure 1 shows an overview of the classification model developed in our work. We adopt the vocabulary used to define ontologies in order to formalize our classification [7]. With it, we solve the lack of formalization found in the previous studies. Concretely, we use the following ontology relationships:

- *Composition*. Some concepts of our ontology are compositions of other concepts that define them from different points of view. For example, we define the presentation facet as a composition of 4 concepts (format, argumentation, detail and interaction) that describes how an explanation is presented to the user.
- *Subconcept*. A subconcept is an extension of a concept following a *"is-a"* relationship. Subconcepts are not disjoint as explanations can be classified into several concepts. For example, our model shows "Positive" and "Negative" as subclasses of the "Argumentation" aspect.
- *Instance*. An instance is a concrete instantiation of a concept. For example, the "Format" concept, has 4 instances: "Natural language", "Schematic", "Visual" and "Other". An explanation can be characterized by several instances of the same concept.

Next, we detail the main concepts of the model in the following sections.

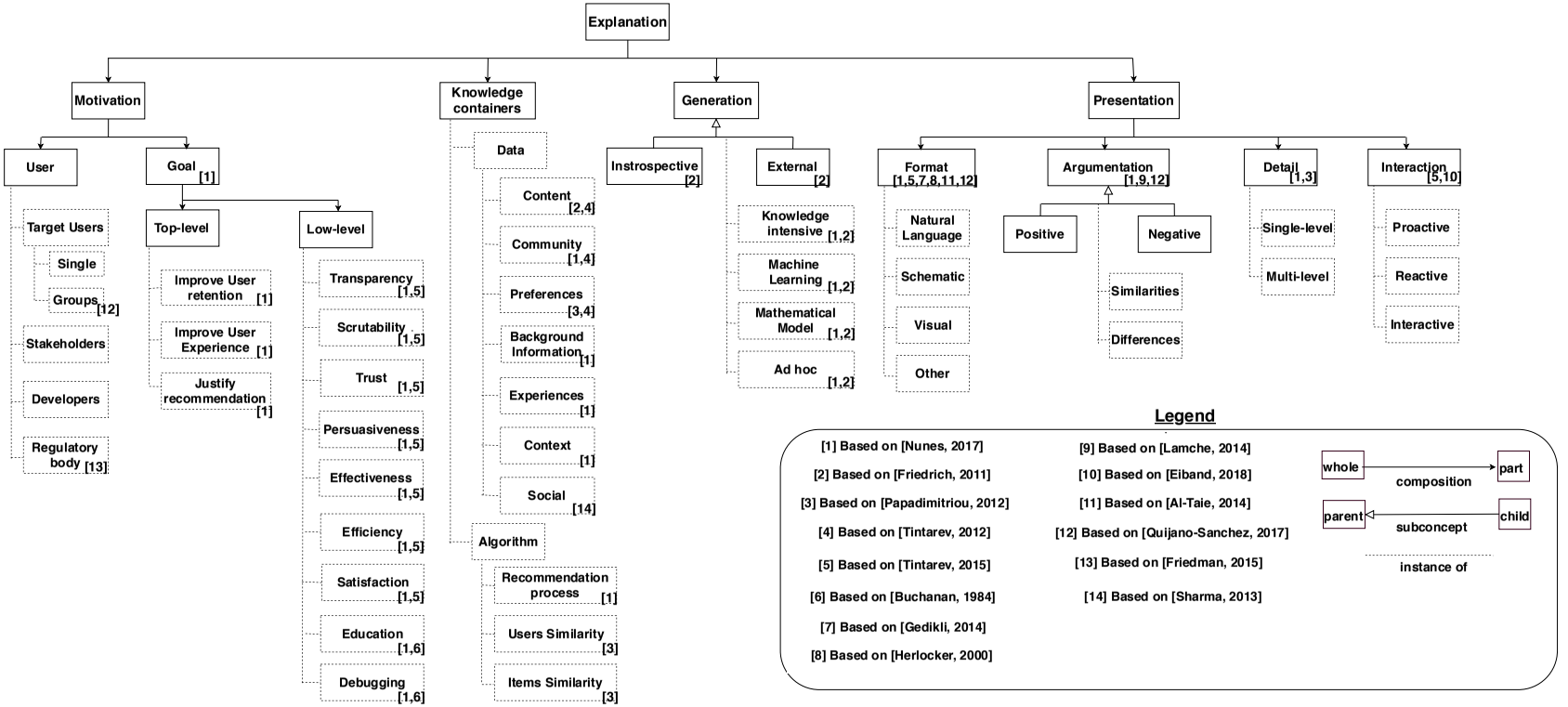



Fig. 1: Classification Model for explanations in recommender systems.

### 3.1 Motivation

Motivation refers to the goal of the explanation according to the target user. Both ideas, goal and user, are the two major subconcepts that characterize explanations according to the motivation.

**3.1.1 User.** Recommender systems are supposed to satisfy user needs. However, explanations do not have to be aimed exclusively at the **Target User**, the people who receive the recommendations made by the system. Our model also classifies users as **Stakeholders** (the people interested in the success of the recommender system), **Developers** (the people who need to know how the recommender is working in order to debug it) and **Regulatory body** (which represents the legislative and government organizations that need to know how recommender systems work in order to regulate their transparency and their correct use [9]).

**3.1.2 Goal.** The goal of an explanation is closely related to the user to whom the explanation is targeted. The goals can be divided into two subconcepts: top-level goals and low-level goals. The top-level goals are focused on the user, while the low-level ones help to achieve the top-level goals.

Top-level goals distinguish three groups, inspired by Nunes and Jannach's proposal [21]: **Improve user retention** (to increase the probabilities of a user to return to the recommendation system), **Improve user experience** (to help users to make good decisions and to enjoy the recommendation activity) and **Justify recommendation** (to support the recommendation provided, helping the user to understand why an item was recommended).

Our low-level goals match with the criteria for designing good explanations defined in [30] and [3]. We have included the following low-level goals for the explanations: **Effectiveness** [30] (the explanation helps the user to find the items that she needs), **Efficiency** (it helps the user to make decisions faster), **Trust** [30, 3] (it increases the user confidence in the system), **Scrutability** [30] (the user will be able to provide feedback when the system provides a wrong recommendation), **Persuasiveness** [30, 3] (it convinces the user that the recommendation is appropriate), **Transparency** [30, 3] (it specifies how and why a recommendation is made), **Education** [3] (users can learn something about the recommender system using the explanation) and **Debugging** [3] (it allow users to identify bugs in the recommendation system).

### 3.2 Knowledge containers

Explanations for recommender systems can be created based on different types of knowledge sources or containers [26, 33]. We have observed that explanations can use information extracted from two types of sources: from the data available in the recommender system or from the algorithm used in the recommendation.

**3.2.1 Data.** These explanations are based on the information available at the recommender system. We have found different types of data used in explanations: **Content** (the explanation refers to the item description, like "This shirt is white and its price is 8.99 €."), **Preferences** (it refers to the item features, considering the known user preferences [22, 29], like 'This shirt is white, a colour that you like, and its price is 8.99 €, less than 10 €, your limit price."), **Community** (it refers to the information extracted from the behaviour of the community of users interacting with the recommender system [21, 29], like "This shirt is one of the shop's top-sellers"), **Background Information** (The data is extracted from the domain [21], like information from reviews or external interactions), **Experiences** (explanations that refer to user past experiences and system history [21], like "This shirt is white and you bought white shirts yesterday"), **Context** (it refers to the dynamic features that describe the requirements of target users, like weather restrictions) and **Social** (the explanation refers to user's relationships [27, 24], like "Your brother also bought this shirt").

**3.2.2 Algorithm.** The explanations are based on the information that the system collects from the recommendation algorithm. The classes proposed for this knowledge container are: **Recommendation Process** (the explanation shows how the algorithm performs a recommendation [21, 6]), **User Similarity** (it justifies the recommendation through the interactions that similar users have carried out with the recommended items [22]) and **Item Similarity** (it is based on the similarity that exists among recommended items and the items that the user previously interacted with [22]).

### 3.3 Generation

Explanations for recommender systems can be implemented using different processes and techniques. During the analysis of these techniques, we observed that recommender systems commonly act in two different ways: as a black box system, when it does not show how it works to the user to whom recommends the items, or as a white box system, which is transparent to the users and it allows them to know the way the recommender algorithm operates [10].

We can classify explanations as **Introspective**, when the explanations are generated by the recommendation algorithm itself because it acts as a white box system [10], or **External**, if the recommender system acts as a black box system, so the explanation system has to implement a new technique for creating justifications, different from the underlying recommendation algorithm [10].

No matter whether the system operates as black box or as white box, different techniques can be used to generate the explanations. Therefore, explanation systems can be also classified by the method that they use for generating explanations. The types of explanation according to their explanation algorithms are [21]: **Knowledge-based Explanation** (based on techniques that use a knowledge source for solving problems), **Machine Learning Explanation** (explanation uses information about past experiences to explain recommendations),

**Mathematical model Explanation** (based on mathematical models to justifying recommendations) and **Ad-hoc Explanation** (the explanation algorithm provides the explanations required by the recommender system).

### 3.4 Presentation

The presentation aspect represents the way the system displays the explanation to the user. Among all of the publications studied, we have identified four different subconcepts related with the presentation: the display format, the argumentation, the level of detail and the way of interaction.

**3.4.1 Display Format.** The explanations can be shown in different ways, more or less simple. The following ones are the main formats that we have found in literature [14, 12, 21, 30, 1, 24]: **Natural Language** (the justification for a recommendation is displayed as a text, commonly based on templates), **Schematic** (explanations are shown in a simplified text format, like tables, logs or ratings), **Visual** (explanations use a more graphic way as charts or histograms) and **Other** (which employ more innovative display formats, like audio, video or even augmented reality[11]).

**3.4.2 Argumentation.** Explanations can show different outlooks for a recommended item. They can be **Positive** (they justify why a recommended item is suitable for the user with positive arguments) or **Negative** (explanations show features that do not fit user preferences and tastes using negative argumentation) [16, 21, 24]. Additionally, these arguments can be supported using **Similarities** or **Differences** between item features and user preferences and tastes.

For example, "This shirt is yellow, that is not your favourite colour, but it might also like it" is a negative explanation based on differences that can also be suitable for the user when the explanation needs to justify the diversity or the serendipity. However, "This shirt is green, the colour that you are looking for" is positive when the explanation highlights the similarities between user likes and item features.

**3.4.3 Level of detail.** It is the degree of specification about a recommendation included in its explanation. The level of detail can vary in different explanations, depending on what the user needs to know or what the developer wants to show. Due to the level of detail is a subjective aspect, we take into account the knowledge base, the number of visualization types and the length of the explanation for modelling this concept.

As the authors in [21, 22] state, we have considered **Single-level** (explanations that use a single knowledge base, have a single way to display it and/or are a short explanation with few details) and **Multi-level** (explanations that use several knowledge containers, have different visualization formats and/or they are a long explanation with many details, like Tag cloud by [12]) detailed explanations.

**3.4.4 Interaction** The different ways of interaction depend on how the user obtains explanations through the system [30, 6]. The main ways of interaction are: **Proactive** (when the explanation is shown with the recommendation, like Amazon, eBay or Netflix), **Reactive** (when the user asks for the explanation of the recommendation received, like the conversational system in [19]) or **Interactive** (when the explanation is already available to the user and she can interact with the explanation for getting more information, like Shopr [16])

## 4 Model validation

To validate the proposed model, we have studied and classified 23 different explanation approaches collected from the literature. Some of them are explanation approaches for classification methods or machine learning techniques since we also wanted to validate the extensibility of our model in other fields. In some cases, we do not have enough information in order to complete every aspect within our model.

- "Explainable Movie Recommendation Systems by using Story-based Similarity" [17] ($A$).
- "Explaining Recommendations by Means of User Reviews" [4] ($B$).
- "Explaining Complex Scheduling Decisions" [18] ($C$).
- "Explaining Contrasting Categories" [23] ($D$).
- "Explaining smart heating systems to discourage fiddling with optimized behaviour" [28] ($E$).
- "The design and validation of an intuitive confidence measure" [31] ($F$).
- "Interactive Explanations in Mobile Shopping Recommender Systems" [16] ($G$).
- "A Review of Explanation and Explanation in Case-Based Reasoning" [5]. We have studied the following CBR systems mentioned in this publication: CARES ($H$), DIRAS ($I$) and MOCAS ($J$).
- "How should I explain? A comparison of different explanation types for recommender systems" [12] ($K$).
- "Explaining Collaborative Filtering Recommendations" [14]. We have studied seven explanations methods proposed in this work: Histogram with grouping ($L$), Neighbour ratings histogram ($M$), Table of neighbours rating ($N$), MovieLens Percent confidence in prediction ($O$), Number of neighbours ($P$), Overall percent 4+ ($Q$), and Overall average rating ($R$).
- "Tagsplanations: Explaining Recommendations Using Tags" [32] ($S$).
- "A case-based reasoning system for aiding detection and classification of nosocomial infections" [13] ($T$).
- "A framework for Explanation of Machine Learning Decisions" [2] ($U$).
- "Great Explanations: Opinionated Explanations for Recommendations" [20] ($V$).
- "Knowledge-based systems, viewpoints and the world wide web" [8] ($W$).

| System | Motivation | | Knowledge Container | Generation | Presentation | | | | Interaction |
|---|---|---|---|---|---|---|---|---|---|
| | User | Low-level Goal | | | Display Format | Perspective | Positive / Negative | Detail Level | |
| A | - | - | - | Knowledge-intensive | External | - | - | - | - | - |
| B | - | Transparency | Background information | Machine Learning | External | - | - | - | - | - |
| C | Target User: Single | - | Content | Knowledge-intensive | Introspective | Natural Language / Visual | Similarities | Positive | Multi-level | Proactive / Reactive |
| D | Target User: Single | - | Content | Machine Learning | Introspective | Natural Language / Visual | Similarities | Positive | Multi-level | Proactive |
| E | Target User: Single | Trust | Preferences | Knowledge-intensive | Introspective | Natural Language / Schematic / Visual | Similarities | Positive | Multi-level | Proactive / Reactive |
| F | Target User: Single | Debugging (validation) | Experiences | Knowledge-intensive | External | Schematic | - | - | Single-level | - |
| G | Target User: Single | Transparency / Scrutability | Preferences | Ad hoc | External | Natural Language / Visual | Similarities / Differences | Positive / Negative | Multi-level | Proactive / Reactive / Interactive |
| H | Target User: Single | - | Experiences | Knowledge-intensive | Introspective | Natural Language | Similarities | Positive | Multi-level | Proactive |
| I | Target User: Single | - | Experiences | Knowledge-intensive | Introspective | Schematic | Similarities | Positive | - | Proactive |
| J | - | - | Experiences | Knowledge-intensive | Introspective | - | - | - | - | - |
| K | Target User: Single | - | Content / Preferences | Machine Learning | External | Visual | Similarities / Differences | Positive / Negative | Multi-level | - |
| L | Target User: Single | Effectiveness, Satisfaction, Transparency | Users similarity | Machine Learning | Introspective / External | Visual | Similarities | Positive | Single-level | - |
| M | Target User: Single | Effectiveness, Satisfaction, Transparency | Users similarity | Machine Learning | Introspective | Visual | Similarities | Positive | Single-level | - |
| N | Target User: Single | Effectiveness, Satisfaction, Transparency | Users similarity | Machine Learning | Introspective | Schematic | Similarities | Positive | Single-level | - |
| O | Target User: Single | Effectiveness, Satisfaction, Transparency | - | Machine Learning | Introspective | Schematic | Similarities | Positive | Single-level | - |
| P | Target User: Single | Effectiveness, Satisfaction, Transparency | Users similarity | Machine Learning | Introspective | Schematic | Similarities | Positive | Single-level | - |
| Q | Target User: Single | Effectiveness, Satisfaction, Transparency | Community | Machine Learning | Introspective | Schematic | Similarities | Positive | Single-level | - |
| R | Target User: Single | Effectiveness, Satisfaction, Transparency | Community | Machine Learning | Introspective | Schematic | Similarities | Positive | Single-level | - |
| S | Target User: Single | Transparency, Effectiveness, Efficiency | Content / Preferences | Ad hoc | External | Visual | Similarities | Positive | Multi-level | - |
| T | Target User: Single | - | Experiences | Knowledge-intensive | Introspective | - | - | - | - | - |
| U | - | - | Content | Ad hoc | External | Natural Language | Similarities | Positive | Single-level | - |
| V | Target User: Single | Efficiency, Persuasiveness | Background information / Experiences / Preferences | Knowledge-intensive | Introspective | Schematic | Similarities / Differences | Positive / Negative | Multi-level | Proactive / Interactive |
| W | Target User: Single / Stakeholders | - | Recommendation Process | Knowledge-intensive | Introspective | Schematic | Similarities / Differences | Positive / Negative | Single-level | - |

Table 1: Classification of approaches studied according to our model. Dash values ("-") represents that we do not have enough information to define this concept.

According to the motivation aspect, in most of the approaches studied the user who receives the explanation is a single target user. We have found only an approach whose user is a stakeholder ($W$), although it also explains decisions to a single target user. Only a few studies just describe the main goal of the explanations using low-level goals. The most usual goal is "Transparency".

From the publications studied, the knowledge containers based on data are more used than the algorithm-based ones. "Content" type seems to be one of the most used knowledge containers and it is commonly used in combination with "Preferences". We have found only an explanation approach based on "Recommendation Process" as knowledge container.

According to the "Generation" facet, we can see that the classification is equally distributed between the "Introspective" and the "External" values. Regarding the explanation algorithm, the most used is "Knowledge-intensive". There are some systems whose algorithms are "Ad-hoc", but we have not found systems using a "Mathematical model".

In reference to the "Display format" concept, several studies combine "Natural language" (used to explain by default) with "Visual" (to explain with more details when users ask for it, using a reactive explanation). Regarding the argumentation facet, we can observe that "Similarities" are commonly used with a "Positive" argumentation, while "Differences" are used in "Negative" argumentations. The "Level of detail" is also distributed between high detailed and low detailed explanation methods. Finally, we have not found too much information according to the way of interaction facet.

## 5 Conclusions

In this work, we have defined a classification model for explanations in recommender systems. Our model includes features and concepts considered in previous works and it is improved with additional aspects and some formalization knowledge that other models lack. The main goal of this formalization is to set the foundations of an ontology to complement our frameworks jCOLIBRI and RecoLibry with explanations. This way, we expect that our model will be an useful tool to guide the design and development of explanations for recommender systems.

Our model includes the main aspects that explanations should have. In this work, we detail what these aspects are and we define the types of explanations according to them. We have also classified different approaches found in the literature in order to prove the validity of our model. We have found that most of the approaches studied do not describe information about all of our model concepts. Moreover, we have not found approaches for a few categories proposed. As future work, we will research in depth to achieve examples for all the types defined here.

## References


1. Mohammed Z Al-Taie and Seifedine Kadry. Visualization of explanations in recommender systems. *Journal of Advanced Management Science Vol*, 2(2):140 – 144, 2014.
2. Chris Brinton. A framework for explanation of machine learning decisions. In *IJCAI-17 Workshop on Explainable AI (XAI)*, page 14.
3. Bruce G Buchanan and Edward H Shortliffe. Explanation as a topic of AI research. *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*, page 331, 1984.
4. Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Explaining recommendations by means of user reviews. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
5. Dónal Doyle, Alexey Tsymbal, and Pádraig Cunningham. A review of explanation and explanation in case-based reasoning. Technical report, Trinity College Dublin, Department of Computer Science, 2003.
6. Malin Eiband, Hanna Schneider, and Daniel Buschek. Normative vs pragmatic: Two perspectives on the design of explanations in intelligent systems. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
7. Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.
8. Ian Finch. Knowledge-based systems, viewpoints and the world wide web. In *IEE Colloquium on Web-Based Knowledge Servers (Digest No. 1998/307)*, pages 8/1–8/4, Jun 1998.
9. Arik Friedman, Bart P. Knijnenburg, Kris Vanhecke, Luc Martens, and Shlomo Berkovsky. *Privacy Aspects of Recommender Systems*, pages 649–688. Springer US, Boston, MA, 2015.
10. Gerhard Friedrich and Markus Zanker. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98, 2011.
11. Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. Mobile recommender systems in tourism. *Journal of network and computer applications*, 39:319–333, 2014.
12. Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should I explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.
13. HJ Gómez-Vallejo, B Uriel-Latorre, M Sande-Meijide, B Villamarín-Bello, Reyes Pavón, F Fdez-Riverola, and Daniel Glez-Peña. A case-based reasoning system for aiding detection and classification of nosocomial infections. *Decision Support Systems*, 84:104–116, 2016.
14. Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.
15. Jose L. Jorro-Aragoneses, Belén Díaz-Agudo, Juan A. Recio-García, Diego M. López-Gutierrez, and Gineth M. Ceron-Rios. RecOnto: An ontology to model recommender systems and its components. In *2017 International Conference on Tools with Artificial Intelligence*, pages 815–821. IEEE, 2017.
16. Béatrice Lamche, Ugur Adıgüzel, and Wolfgang Wörndl. Interactive explanations in mobile shopping recommender systems. In *Joint Workshop on Interfaces and Human Decision Making in Recommender Systems*, page 14, 2014.

17. O-Joun Lee and Jason J Jung. Explainable movie recommendation systems by using story-based similarity. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
18. Jeremy Ludwig, Annaka Kalton, and Richard Stottler. Explaining complex scheduling decisions. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
19. David McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, 2005.
20. Khalil Muhammad, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. Great explanations: Opinionated explanations for recommendations. In *International Conference on Case-Based Reasoning*, pages 244–258. Springer, 2015.
21. Ingrid Nunes and Dietmar Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3-5):393–444, 2017.
22. Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, 24(3):555–583, 2012.
23. Michael Pazzani, Amir Feghahati, Christian Shelton, and Aaron Seitz. Explaining contrasting categories. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
24. Lara Quijano-Sanchez, Christian Sauer, Juan A Recio-Garcia, and Belen Diaz-Agudo. Make it personal: a social explanation system applied to group recommendations. *Expert Systems with Applications*, 76:36–48, 2017.
25. Juan A. Recio-García, Belén Díaz-Agudo, and Pedro A. González-Calero. *The COLIBRI Platform: Tools, Features and Working Examples*, pages 55–85. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
26. Michael M Richter and Rosina O Weber. *Case-based reasoning*. Springer, 2016.
27. Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144. ACM, 2013.
28. Simone Stumpf, Simonas Skrebe, Graeme Aymer, and Julie Hobson. Explaining smart heating systems to discourage fiddling with optimized behavior. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
29. Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, 2012.
30. Nava Tintarev and Judith Masthoff. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook*, pages 353–382. Springer, 2015.
31. Jasper van der Waa, Jurriaan van Diggelen, and Mark Neerincx. The design and validation of an intuitive confidence measure. In *Workshop On Explainable Smart Systems (EXSS)*, 2018.
32. Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 47–56. ACM, 2009.
33. Markus Zanker and Daniel Ninaus. Knowledgeable explanations for recommender systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 657–660. IEEE, 2010.

# Data Explanation with CBR *

Belén Díaz-Agudo, Juan A. Recio-Garcia, and Guillermo Jimenez-Díaz

Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Spain
email: belend@ucm.es, jareciog@fdi.ucm.es, gjimenez@ucm.es

**Abstract.** Data scientists interpret large amounts of data, apply analysis algorithms and communicate the insights using visuals and textual descriptions to help people understand the details. The field of Data Visualization investigates graphical data representations that reinforce human cognition and detection of causal relationship and patterns between data. In this paper we study how to apply CBR to automate and personalize the generation of data explanation reports. This position paper describes our preliminary work defining a general methodology based on CBR and a case study in the medical domain. The CBR system reuses explanation report templates that include text and data visualization charts. During the adaptation process our CBR system chooses the visuals that better fits the explanation goal, the type of user and the specific characteristics of the input data.

## 1 Introduction

The problem of explainability in Artificial Intelligence is not new but the rise of the amount of data we get through sensors and wearables, and the use of black box autonomous intelligent systems making diagnosis and predictions with these data, has created the necessity to visualize, tell and understand both the data and how these intelligent systems work. Big (and *thick*) Data concept works on the principle that the more quality data you get, the more you know about any situation and the more reliably you can gain new insights and make predictions about what will happen in the future. Predictions involve using advanced analytics technology, building models, machine learning and data mining algorithms, running simulations and finding patterns. Visualizing and explaining the data patterns is helpful to make users understand and trust the system and its predictions.

Data scientists interpret large amounts of data, apply analysis algorithms and communicate the insights using visuals and textual descriptions to help people understand the details. Even when working with very different data sets, there are similarities in the process of analyzing data, visualizing interesting results and communication of the findings [1]. The field of Data Visualization (DataViz) investigates graphical data representations that reinforce human cognition and

* Supported by the UCM (Research Group 921330) and the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R)

detection of causal relationship and patterns between data [14]. We have seen the rise of the so-called narrative visualizations that helps us to tell stories with data.

Psychological studies [6] discuss the important role that visual aids can play in communicating health-related information and how different types of visuals influence the way people encode and contextualize information. Visual aids can even lead to enduring changes in attitudes and behavioral intentions toward certain health behaviors. Authors have shown that people consider risk information easier to understand and recall when it is presented using a visual aid. Furthermore, it typically takes people less time to understand information when it is presented visually than when it is presented numerically.

In this paper we study how to apply Case-based Reasoning (CBR) to automate and personalize the generation of data explanatory reports. We present our ongoing work on a general CBR methodology and a case study in the medical domain. Electromyography (EMG) is a diagnostic procedure to assess the health of muscles and the nerve cells that control them (motor neurons). Motor neurons transmit electrical signals that cause muscles to contract. An EMG uses tiny devices called electrodes to transmit or detect electrical signals, and translates these signals into graphs, sounds or numerical values that a specialist interprets. Given a set of data collected through the electrodes during a certain exercise, our system dynamically adapts the visualization charts, detects the areas to zoom and generates explanatory reports by retrieving and modifying previous reports.

Data visualization is regularly promoted for its ability to reveal and tell stories within data, yet these "data stories" differ in important ways from traditional forms of storytelling [15]. Data storytelling has been recently used as a communication and marketing data driven strategy and it has shown to be an effective approach for sharing insights gained by studying specific data sets [11]. Data storytelling is a structured approach for communicating data insights, and it is defined as "the practice of building a narrative around a set of data and its accompanying visualizations to help convey the meaning of that data in a powerful and compelling fashion" [15]. Data storytelling [3] (Figure 1) involves a combination of three key elements: data, visuals, and narrative. The three elements combine and work together. When visuals are applied to data, they can *enlighten* the audience to insights not seen without charts or graphs. Too often data storytelling is interpreted as just visualizing data effectively, however, it is much more than just creating visually-appealing data charts. When narrative is coupled with data, it helps to *explain* to your audience what is happening in the data and why a particular insight is important [3]. Data storytelling has been extensively used in marketing and sales to hook your audience with a engaging and effective presentation. However, generating written reports in the medical domain has different goals. There are different levels to introduce storytelling into a scientific presentation without putting the axe to scientific accuracy and integrity. In our current research we deal with a basic generation of data explanation using visuals and texts but we do not connect the data with metaphors, emotions and narrative.

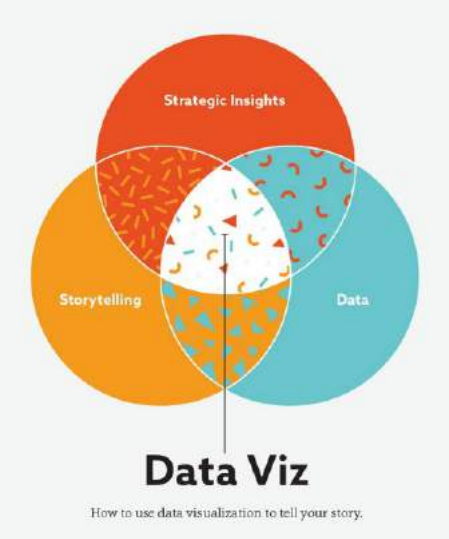



Fig. 1: Data Visualization

We aim to generate visual reports to explain the information that the data tell us about a patient and her context, patterns in data, how a patient evolves in time, how she relates with its context, why this patient is similar or different regarding others. We propose a CBR system based on report templates that are personalized, which include data visualization as part of the story. As part of the adaptation process, our CBR system chooses the visual techniques that are better than others for a particular goal (temporal evolution, current state, comparison with other patients), for the type of user –patient, doctor, or data scientist– and for the type of data characterization.

## 2 Related work

Natural Language Generation (NLG) takes structured data as input and produces natural language text. Our approach relates with research on NLG based on templates [12]. The underlying idea is that texts often follow conventionalized patterns. These patterns can be encapsulated in *schemas*, which are template programs which produce text plans. Schema are derived from a target text corpus, by breaking up these texts into messages, and trying to determine how each message can be computed from the input data. The schema-based approach to NLG has striking parallelism to CBR approaches to problem solving, in that existing previous solutions, such as those obtained from a corpus of target texts, are extracted and prepared so they can be reused to solve future problems, much in the same way as cases in a case-base are prepared from previous problem solutions. In our research group CBR has been previously used for poetry generation

[2] and story plot generation [7]. The system reuses existing stories to produce a new story that matches a given user query. The plot structure is obtained by a CBR process over a case base of tales and an ontology of explicitly declared relevant knowledge. The resulting story is generated as a sketch of a plot described in natural language by means of NLG techniques. In [18] authors propose a CBR approach reusing explanation reports from previous transportation incidents.

Also related with our methodology, the study described in [13] refines the traditional typology of data stories from the journalistic perspective and also identifies core templates to good data journalism practice. Authors characterize each of the 44 cases and determine the type of the stories and the nature of technologies employed to create them.

In [14] authors describe an approach and a prototype to improve data driven knowledge transfer in presentation tools by applying information visualization concepts. They discuss some of the shortcomings of the current presentation tools and introduce an interactive data visualization solution providing effective narrative visualizations in presentations. Their tool offers the possibility of manually changing the chart type, applying filters on the data or adjusting the focus dynamically during the presentation time. The presenter breaks free from any predefined visualization and can explore and discuss the data without restrictions. Our approach is related to it in the sense of interaction and flexibility, although, taking advantage of the CBR methodology, we aim to provide automatic reasoning and advanced functionality extending the approach with the recommendation of the most appropriate visualization graphics, learning from the manual interaction of the user, and the automatic rendering of a narrative text explaining the data. Many of the existing works are presentation oriented [10,14]. Our approach is oriented to visual narrative, i.e., the rendering of data storytelling and visualization of the input data. The novelty of the proposal described in this paper is the use of CBR as an approach that uses the data, charts and user characterization to retrieve and personalize the visuals. Besides, CBR is appropriate for lazy learning from the manual interaction of the user with the data. It allows the system to tag data with expert knowledge captured through the interactions with data charts.

## 3 Methodology

In the situation under research, we start with a large number of measured variables that we need to visualize and explain in a report with possible different underlying goals depending on the final user. We propose a CBR approach to generate visual narrative reports to explain the input data set, and allows comprehension and decision making processes with data. Next we summarize the main steps of the methodology. Section 4 describes the CBR system, and Section 5 describes the case study. First step in the methodology is data characterization according to syntactic and semantic tags (see Section 3.1). Next, chart characterization according to the visualization goals and its adequacy to be used with a certain type of input data (see section 3.2). Design the CBR system as a

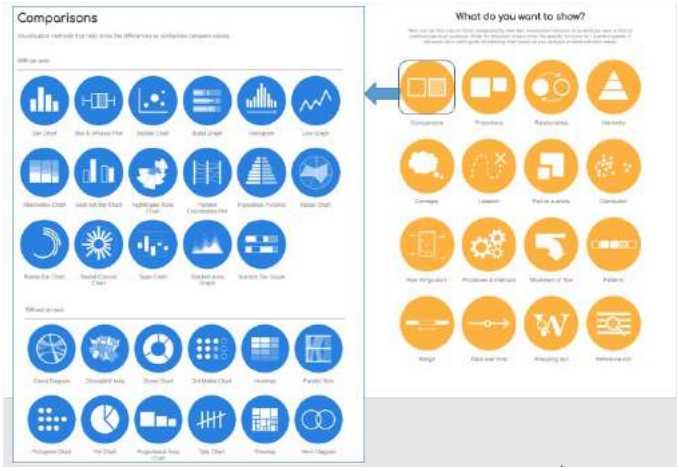



Fig. 2: Selection of charts (*The Data Visualisation Catalogue*) by function

intelligent visualization tool that is able to decide the best way to explain and visualize the input data to the target user. The tool is interactive and allows the user to adjust the scale to zoom in and out, change colours and sizes. Capturing the expert interaction is useful as a source of knowledge for learning.

### 3.1 Data characterization

The data characterization phase aims at recognizing and classifying the input data variables and apply Machine Learning techniques to capture data features, and patterns, predictions, anomalies and dependencies within the data variables. It uses data analytics, filtering and cleaning techniques, looking at data on different time periods, studying patterns and how data evolves, anomaly detection to identify items which do not conform to an expected pattern, studying how the data relates with context, and making a description of the features that are more representative to be included in the report. A variable in a data set can be quantitative discrete (countably infinite or finite), quantitative continuous, categorical or qualitative, or ordinal. Besides, we define a set of features to characterize the input data series values in two levels:

- Syntactic tags describing statistics about data, peaks of the distribution, average, percentiles, variance, regression lines, ranges, and others.
- Semantic tags describing information about the data meaning. For example, in our case study we tag the data with information about the patient, his/her physical build, physical features, the type of workout doing during the monitoring session, and others. This is related with the concept of *thick data* that is used to pinpoint the idea that syntactic tags alone are not enough.

### 3.2 Chart characterization

Our approach characterizes both data and charts and provides with a set of tags to help the system to choose the right type of chart for explaining a certain data set. A well-chosen and well-crated visual easily tells the story in a single glance but finding the right chart is not always easy. The process of information visualization is always related with an specific goal, but note that, even if we are able to precisely characterize visualization functions, how well they work depends not only on the data structure but also on the actual values. For example, recommending a scatter-plot or a connected scatter-plot will depend on the data. A connected scatter-plot is recommended if the values change relatively smoothly and in slightly unexpected ways, but it is not recommended if the resulting connected chart includes a large number of tangled lines. We have reviewed the extensive literature in the area. Hearst [9] identified some of the goals we want to cover in this case study: making large and complex data set coherent, presenting information from various viewpoints and at several levels of details, supporting visual comparison and presenting the information in context. In his book, Few [5] proposed eight messages to show numeric data together with the type of visualization that is suitable for each message. The messages include time series, rankings, part-to-whole, deviation, distribution, correlation, geospatial messages and nominal comparison. Bar charts are quite versatile and they enable nominal comparisons, comparison of relative point values, rankings, frequency distributions (histograms) and deviation (since bars can also go below the horizontal axis). Line charts requires quantitative variable across the x-axis and they are useful when variables have contiguous values and further allow us to visualize time series, and the area under each line can optionally be filled with a colour. As an addition to bar charts, box plots can also be used for showing deviation. Pie charts are useful for emphasizing differences in proportion among a few numbers, and allow the user to visualize how categories relate to the total amount of data. Regular bar charts and stacked bar charts are also useful for this task. Scatter plots are provided for showing correlation as they convey overall impression of relationship between two variables. The book [4] reviews the spectrum of graph types available in Excel, and give a guide and a "chart chooser cheat sheet" to determine, beyond the default options, which one most appropriately fits specific data stories, and easy steps for making the chosen graph. We have used the visualization functions included on *The Data Visualisation Catalogue*[1]. It offers a list of charts categorized by their data visualization functions, that is the message that the chart is able to communicate to an audience. As they notice, the allocation of each chart into specific functions is not a perfect system and it is only a guide for selecting a chart based on the communication needs. Figure 2 shows the example of the visualization methods that the catalogue suggests for the comparison function, i.e, the charts that are more suitable to help show the differences or similarities between values. Additionally to the comparison, we will include the following visualization functions

[1] https://datavizcatalogue.com/

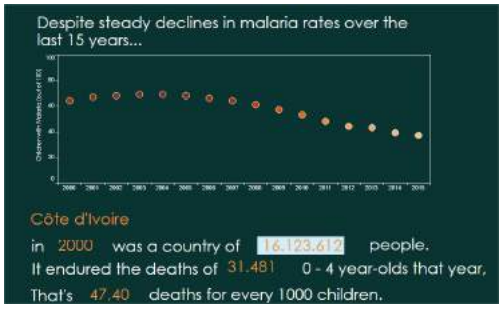



Fig. 3: Example by J. Richards (http://www.jnnyrchrds.com/)

to classify charts in our reports: proportions, relationships, hierarchy, concepts, part-to-a-whole, distribution, patterns, range and data over time. In our case study we concentrate on the functions: comparison, relationships, distribution, range and data over time.

## 4 CBR cycle

The case base is a set of explanatory report templates with a fixed part of text and images (e.g., logo) and several variability hooks or *gaps*, both in the visuals and in the text (see Figure 3), like the patient personal data from the personal record, charts, colours, sizes of text and lines in charts and others. Each template includes a textual explanation of the input data set and its visual representation.

The CBR cycle of the proposed system (Figure 4) runs as follows:

1. The input query describes:
   - Data set with its annotated characteristics.
   - For whom we are generating the report.
   - Which are the main visualization functions.
2. Retrieval of the template report more suitable for the query.
3. Adaptation of the variability hooks or gaps in the template. Each gap in the template has a semantic description of the appropriate filler. We fill the text gaps with data values, or predefined text fragments, for titles and labels. In contrast, we use a substitution method [8] for chart gaps, where a local query is used to search the more suitable chart for the visualization function and data. For example, if a chart gap is intended for comparing numeric continuous values, line or bar charts are used to fill the gap; on the other hand, proximity graphs are used to show user in context. The selection of colours, line pattern and sizes is also very relevant and it is adjusted to the data characteristics, ranges, outstanding values, etc. We take into account research in information visualization that tries to exploit findings in Gestalt psychology in order to facilitate knowledge transfer [14]. Details are not included in this paper, as we have not complete the formalization of the description language for query and the chart gaps.

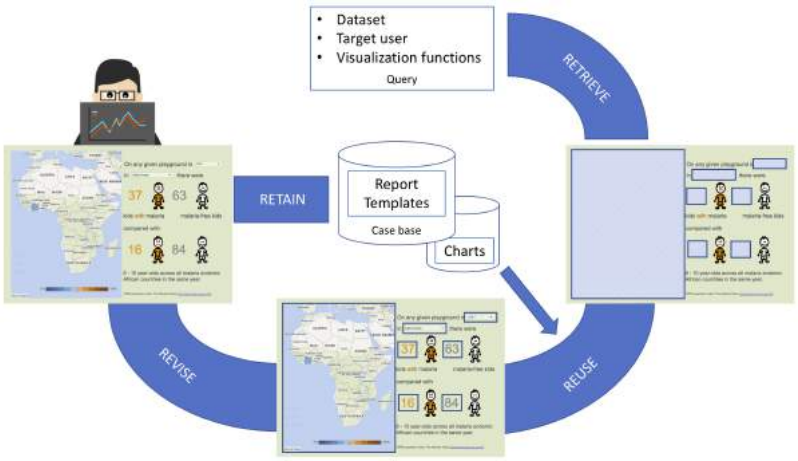



Fig. 4: CBR cycle

After the chart is retrieved and the visual features are adjusted to the data, the proposed chart is built with input data and it is included in the corresponding gap of the report.

4. Revise of the report and learn from the interaction. Static visualizations have long been used to support storytelling, usually in the form of diagrams and charts embedded in a larger body of text. In this format, the text conveys the story, and the image typically provides supporting evidence or related details[15]. Current technological advances and research in DataViz led to more dynamic and interactive visualizations that emphasizes on the exploration and discovery of meaningful relations between data sets [14].

   Although we use different ML techniques to find patterns in the data, the expert remains a very valuable source of knowledge that is captured for later use in other stakeholders. In our interactive visualization tool we plan to include a learning functionality based on the observation and memorization of the expert's interaction with the data in the visualization: what area has the expert enlarged to see details? Why? Does the expert think the system has chosen the most appropriate visualization? Has it served to detect the problem? We propose using the categorization proposed by the literature [17,19] of the frequently used interaction techniques in information visualization: overview, zoom, select, explore, extract, reconfigure, encode, abstract, elaborate, filter and connect.

   Regarding the chart customization, the system chooses the initial values for the visual features, like line size, colour and thickness, according to the data and the chart gap description in the template. However, interactivity allows the system to learn from the user revision and the new properties can be stored with the case to be reused.

## 5 Case study: mDurance

We apply our methodology to a case study of explaining EMG data. Electromyography (EMG) is a diagnostic procedure to assess the health of muscles. An EMG uses tiny devices called electrodes to transmit or detect electrical signals, and translates these signals into graphs, sounds or numerical values that a specialist interprets. The power of these data counteracts with the fact that only an expert on EMG can understand them accurately and interpret the results.

mDurance [2] is a digital health tool that helps to verify the health of patient muscles. It is aimed at professionals in physical health, sports and well-being. It offers a portable electromyograph sensor, a mobile application, cloud storage and data analysis through a web application. The tool is responsible for the acquisition of the electromyographic signal from the sensors and noise filtering. In addition, it provides analytics and graphical visualization tools for this data. Our ongoing work applies the methodology described in this paper for the generation of personalized explanation reports using templates.

Given a set of data collected through the electrodes during a certain exercise, our system generates an explanatory report that dynamically adapts the visualization charts.

At the current state we are working with few prototypical report templates in the case base. The input query describes the type of report and includes a characterization of the target user (doctor, patient or data scientist) and what are the main functions of the report, patient current state and evolution, comparison with context or fraud detection. The reports are aimed at the medical professionals involved in the diagnosis and treatment, therefore it is essential to convey in the generated report the complete information available about the patient.

The generated report explains the data and allows non expert users to verify muscle abnormalities, evaluate muscle performance and detect muscle asymmetries, assessing state of health, motivation and education on habits among many other utilities. Explanation of data in context gives people points of comparison to understand how the data relates to larger trends, other geographies, treatments, similar patients, etc. Figure 5 shows the structure of a patient explanatory report after an EMG diagnosis test, including text and charts.

## 6 Conclusions

There is a very active research area on data visualization and data storytelling for presenting and communicating data effectively. In this paper we have described our current work on a CBR methodology and a case study to generate personalized visual explanation reports for different types of users. Data represent signals captured by an electromyograph sensor to produce a record called an electromyogram. The signal data together with information about the patient

[2] http://www.mdurance.eu/


Fig. 5: mDurance reports

and the movement during the session are analyzed and explained in a visual report. The underlying goal of the proposed system is to explain and to strengthen a viewer's understanding of the underlying data, which might be hard to interpret in its raw form. Visual reports explain the data and their context, patterns, evolution in time, why this patient is similar or different regarding others. As part of the adaptation process, our CBR system chooses the visual techniques that are better than others for a particular goal, for the type of user and for the type of data characterization. We are working in the formalization of description languages for describing extended queries and the chart gaps in the templates. Besides we will improve the retrieval and reuse methods to be able to combine several templates from the case base. Moreover, our goal is to improve the template case base to introduce more narrative elements and apply our methodology to other domains. From research in psychology the use of stories to communicate data is related with the promotion on the use of metaphors. Metaphor is the phenomenon whereby we talk and, potentially, think about something in terms of something else. In the book [16] authors discuss metaphor as a common linguistic occurrence that is central to many different types of information communication. As a future work we plan to study the use of a case base of metaphors to enrich the narrative of the text generation process.

## References


1. J. Christensen. Effective data visualization: The right chart for the right data, and data visualization: A handbook for data driven design. *Technology|Architecture + Design*, 1(2):242–243, 2017.
2. B. Díaz-Agudo, P. Gervás, and P. A. González-Calero. Poetry generation in colibri. In *European Conference on Case-Based Reasoning*, pages 73–87. Springer, 2002.
3. B. Dykes. Data storytelling: The essential data science skill everyone needs. *Forbes.com*.
4. S. Evergreen. Effective data visualization: The right chart for the right data. SAGE publications, 2016.
5. S. Few. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, USA, 2nd edition, 2012.
6. R. Garcia-Retamero and E. Cokely. Communicating health risks with visual aids. *Current Directions in Psychological Science*, 22(5):392–399, 2013.
7. P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás. Story plot generation based on CBR. *Knowl.-Based Syst.*, 18(4-5):235–242, 2005.
8. P. A. González-Calero, M. Gómez-Albarrán, and B. Díaz-Agudo. A substitution-based adaptation model. In *Challenges for Case-Based Reasoning - Proceedings of the ICCBR'99 Workshops, Seeon Monastery, Germany, July 27-30, 1999*, pages 17–26, 1999.
9. M. Hearst. Information visualization: Principles, promise, and pragmatics. CHI 2003 Tutorial http://www.chi2003.org/tutorial-details.html, 2003.
10. R. Kosara. Presentation-oriented visualization techniques. *IEEE Computer Graphics and Applications*, 36(1):80–85, 2016.
11. R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *Computer*, 46(5):44–50, May 2013.
12. K. R. McKeown. Discourse strategies for generating natural-language text. Artificial Intelligence 27(1):1-41., 1985.
13. A. Ojo and B. Heravi. Patterns in award winning data storytelling. *Digital Journalism*, pages 1–26, 2017.
14. R. Roels, Y. Baeten, and B. Signer. Interactive and narrative data visualisation for presentation-based knowledge transfer. In G. Costagliola, J. Uhomoibhi, S. Zvacek, and B. M. McLaren, editors, *Computers Supported Education*, pages 237–258, Cham, 2017. Springer International Publishing.
15. E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, Nov 2010.
16. E. Semino, Z. Demjén, A. Hardie, S. Payne, and P. Rayson. Metaphor, cancer and the end of life: A corpus-based study, 2017.
17. H. Siirtola. *Interactive visualization of multidimensional data*. Tampereen yliopisto, 2007.
18. G. Sizov, P. Öztürk, and E. Marsi. Let me explain: Adaptation of explanations extracted from incident reports. *AI Communications*, 30(3-4):267–280, 2017.
19. J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1224–1231, 2007.

# Measuring Explanation Quality in XCBR

Adam J. Johs, Meaghan Lutts, Rosina O. Weber

College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, USA
{ajj37, mkl83, rw37}@drexel.edu

**Abstract.** As part of our motivation to advance societal acceptance of and trust in explainable artificial intelligence (XAI)—namely, explainable case-based reasoning (XCBR) systems—we recognize the criticality of ascertaining how to properly approach explanation quality measurement. In this paper, we search the literature to facilitate decomposition of what is meant by explanation quality in the context of XCBR explanations. Of the various elements used to frame explanation quality in XCBR, *relevance* emerged as one of the most frequently used concepts. We draw from correlative research in the field of information science to illustrate the additional intricacies that should be considered when approaching measurement of explanation *relevance* in XCBR.

**Keywords:** Explainable Case-Based Reasoning, Explanation Quality, Explanation Relevance, Explainable Artificial Intelligence, Case-Based Explanation, Explanation, Quality, Relevance, Case-Based Reasoning

## 1 Introduction

Motivated by our desire to advance societal acceptance of and trust in explainable artificial intelligence (XAI), namely, explainable case-based reasoning (XCBR) systems, we set out to search the literature to understand how measurement of explanation quality should be approached in XCBR. We perceive *quality* as markedly contextual and subjective, and hence, as requiring a nuanced approach to measurement. Even in deferring to definitions like that of [1]—where quality is defined as "*fitness for purpose*"—it became evident through our literature search that decomposing the meaning of explanation quality is paramount to understanding how explanation quality in XCBR should be measured.

## 2 Highlights from the Literature

Informed by our basic awareness of the XCBR landscape and bibliography recommended in the calls for 2017 and 2018 Workshop on XAI, our search focused on literature in the field of XCBR including seminal publications grounded in cognitive science. Our literature search was facilitated via Summon® (i.e., a tool for searching the collections of Drexel University Libraries), Google Scholar, and Google Search. Search queries such as *explanations AND case-based reasoning*, *explanations AND CBR*, *explainable case-based reasoning*, and *case-based explanation* served to preliminarily steer search efforts. As our search matured and XCBR sources were

identified, citation chaining (forward and backward) was performed to expand retrieval results. Once completed, a total of 66 publications in XCBR were retrieved [2–67]; only one publication from the field of expert systems [68] was included as part of this initial search, given that [68] was explicitly referenced in multiple XCBR publications where explanation quality was discussed [e.g., 14, 36, 52].

We proceeded to review the publications collected to ascertain how researchers in XCBR have characterized or framed explanation quality when approaching measurement of explanation quality. For each publication reviewed, we annotated the terms and concepts frequently used by the author(s) to characterize explanation quality. Results were compiled into a single table then subsequently visualized[1] given the comprehensiveness of the table. Figure 1 is a word cloud visualization portraying the range of elements identified throughout the included XCBR literature to frame explanation quality. The terms and concepts depicted in Fig. 1 are not exhaustive, but simply representative of the most frequently occurring terms and concepts used throughout the included XCBR literature.

For consistency and illustration, whenever we encountered in the referred literature instances in the form of *good*, *relevant*, *useful*, etc., we annotated such terms and concepts in the noun forms as seen in Fig. 1. In line with our position as to the contextual and subjective nature of *quality*, we noted observable variation as to how authors characterize explanation quality in XCBR.


**Fig. 1.** A word cloud visualizing the various elements used throughout the included XCBR literature to frame explanation quality. A word cloud is a visualization utilized to illustrate the frequency that a word appears in a given text—the larger the word, the more frequent the appearance.

## 3 Relevance in XCBR Explanations

The spectrum of elements used in the included XCBR literature to frame explanation quality are illuminating. Though *usefulness* was noted as the most frequent element used in the included XCBR literature to characterize explanation quality, we specifically draw attention to the concept of *relevance* and reserve discussion for *usefulness* for future work. We believe what is required to measure the *relevance* of an

[1]https://www.jasondavies.com/wordcloud/

explanation is almost an entirely distinct exercise than, for example, measuring the *convincingness* of an explanation, which is opinion-oriented. Furthermore, according to [60], an explanation is relevant to an event when the explanation is "*a causal chain leading to the event*." This perspective seems directly related to the source of the event to be explained. On the other hand, Leake further elaborates that explanations should be focused solely on anomalies instead of explaining every existing solution [28]. Correspondingly, *relevance* to an anomaly can be evaluated according to whether an explanation resolves the conflict of belief existing at the core of the anomaly [61]. The conflict of belief stems from the notion that an anomaly reflects a discrepancy between a user's expectation and a solution. Therefore, an explanation fulfills a user information need to adjust such discrepancy [28]. Intrinsically, relevance would also then become a matter of user opinion.

In the information science domain, scholars have long grappled with the notion of *relevance* [70–81]. Some research [77] has concluded criteria of *relevance* "*are spatially and temporally bound to, and internally constructed by, the user*," whereas other efforts [71] propose that judgments of *relevance* evolve as user interaction with a system progresses. Additional research [70] has explicated on the subjectivity of *relevance*, while other research [80] has attempted to reframe *relevance* from a lens of linguistic pragmatics, underscoring the purpose of *relevance* as: "*produc[ing] in the mind of a user valuable cognitive effects without undue processing effort*."

We believe incorporating findings from the information science literature can advance and yield insight for research in XCBR explanation quality measurement. Recognizing how the temporal, evolutionary, and idiosyncratic aspects of *relevance* influence whether an explanation is deemed *relevant* is crucial, particularly in how the role of users are concerned. Understanding how the characteristics of individuals and stages of user-system interaction impact *relevance* is fundamental to learning how to properly approach explanation quality measurement. Such intricacies are further illustrative of the nuance required when approaching measurement of explanation quality when framed in terms of *relevance*, versus approaching explanation quality when framed in terms of *convincingness* (as an example).

## 4 Conclusion and Future Work

In accordance with our motivation to advance societal acceptance of and trust in XAI, we set out to explore how measurement of explanation quality should be approached in XCBR. In response to the results of our literature search, we learned of the various terms and concepts used by XCBR researchers to characterize explanation quality. Of the elements identified, *relevance* emerged as one of the concepts used most frequently to frame explanation quality in XCBR. We subsequently juxtaposed *relevance* (as used in the included XCBR literature) with correlative findings from the information science domain, underscoring the aspects of *relevance* we believe crucial to furthering XCBR explanation quality measurement. Given that *usefulness* is central to the relation between problems and solutions in CBR [82], we also intend to extend our efforts to elements of explanation quality beyond that of *relevance*; examining *explanation usefulness* as part of future work holds intrigue considering the observations noted from Fig. 1 and discussion of *usefulness* in CBR provided by [82].

As noted through our literature search, current measurement of explanation quality tends to be approached from a binary perspective—that is, *is of quality* versus *is not of quality*. While many authors converged on common terms to denote explanation quality, few provided a scale or method to holistically measure explanation quality. We suggest a need for a scale of explanation quality and framework for measurement aimed at increasing confidence in and use of XCBR systems in line with [10, 40]. Further investigation would likely shed light on the feasibility of developing contextually-centered metrics for explanation quality measurement, for example, by learning from the innovative approaches adopted as part of the creation of the *explanation competence* evaluation measure in [62] and *explanation utility* metric in [12]. The notion of *explanation strength* or how *compelling* an explanation is, as discussed in terms of measures for ranking explanations in recommender systems [40–42], will also prove valuable to future efforts; notably, [40] explicitly divorced *relevance* from *strength*, leading us to peg [40] as key to understanding how the successful decomposition of explanation quality in XCBR can be approached.

Overall, we aim to extend our analyses into adjacent and distinct domains (e.g., the broader context of XAI, cognitive science, and other related fields) as is seen in [14, 28, 48, 64, 65, 69]. Existing frameworks and metrics of explanation *effectiveness* in XAI [83] strongly correlate with how measuring the *relevance* of explanations in XCBR can be approached. Streams of explanation research in cognitive science and psychology [84–87], robotics [88–89], CBR and deep learning [33], and intelligent systems [90–91] provide a basis for which we can expand upon. Ultimately, our overarching goal is to conduct a granular, synthesizing survey of the literature for further insight into the factors and criteria that influence how explanation quality should be measured in XCBR.

## Acknowledgements

We would like to express our sincerest appreciation to the anonymous reviewers for the remarkably insightful feedback provided.

## References

1. Juran, J., Godfrey, A.: Juran's quality handbook. 5th edn. McGraw-Hill, New York (1999).
2. Aamodt, A.: Explanation-driven retrieval, reuse, and learning of cases. In Richter, M., Wess, S., Althoff, K., Maurer, F., eds.: EWCBR-93: First European Workshop on Case-Based Reasoning. Number SR-93-12 (Technical Report). Kaiserslautern, University of Kaiserslautern, pp. 279–284 (1993).
3. Aamodt, A.: Explanation-driven case-based reasoning. In: Stefan Wess, K.D.A., Richter, M. (eds.): Topics in Case-Based Reasoning, Berlin, Springer-Verlag (1994).
4. Aamodt, A.: Knowledge-intensive case-based reasoning in CREEK. In: Funk and Gonzalez–Calero, (eds.) pp. 1–15. Springer, Berlin, Heidelberg (2004).
5. Armengol, E., Palaudàries, A., & Plaza, E.: Individual prognosis of diabetes long-term risks: A CBR approach. Methods of Information in Medicine. Special issue in prognostic models in Medicine. vol. 40, pp. 46–51 (2001).
6. Bahls, D., & Roth-Berghofer, T.: Explanation support for the case-based reasoning tool myCBR. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence.

July 22–26, 2007, Vancouver, British Columbia, Canada., The AAAI Press, pp. 1844–1845. Menlo Park, California, (2007).
7. Bergmann, R., Pews, G. & Wilke, W.: Explanation-based similarity: A unifying approach for integrating domain knowledge into case-based reasoning for diagnosis and planning tasks. In: Topics in Case-Based Reasoning: Proceedings EWCBR 1993, pp. 182–196 (1993).
8. Caruana, R., Kangarloo, H., Dionisio, J. D., Sinha, U., & Johnson, D.: Case-based explanation of non-case-based learning methods. Proceedings of the AMIA Symposium, pp. 212–215 (1999).
9. Cassens, J.: Knowing what to explain and when. In Gervas, P. & Gupta, K. (eds.): Proceedings of the ECCBR 2004 Workshops. pp. 97–104, Technical Report 142-04, Departmento de Sistemas Informaticos y Programacion, Universidad Complutense de Madrid, Madrid, Spain (2004).
10. Cunningham, P., Doyle, D., & Loughrey, J.:. An evaluation of the usefulness of case-based explanation. In: Ashley, K. D. & Bridge, D. G. (eds.) ICCBR, vol. 2689 of LNCS, pp. 122–130, Springer, Berlin (2003).
11. Doyle, D.: A knowledge-light mechanism for explanation in case-based reasoning, PhD thesis, Trinity College Dublin, Department of Computer Science (2005).
12. Doyle, D., Cunningham, P., Bridge, D., & Rahman, Y.: Explanation oriented retrieval. In: Funk, P. & Calero, P. (eds.) Advances in Case-Based Reasoning (Procs. of the Seventh European Conference on Case-Based Reasoning), pp. 157–168, Springer: Berlin (2004).
13. Doyle, D., Cunningham, P., & Walsh, P.: An evaluation of the usefulness of explanation in a CBR system for decision support in bronchiolitis treatment. Computational Intelligence, vol. 22(3–4), pp 269–281 (2006).
14. Doyle, D., Tsymbal, A., & Cunningham, P.: A review of explanation and explanation in case-based reasoning. Technical Report TCD-CS-2003-41, Trinity College Dublin (2003).
15. Green, M., Ekelund, U., Edenbrandt, L., Björk, J., Forberg, J. L., & Ohlsson, M.: Exploring new possibilities for case-based explanation of artificial neural network ensembles. Neural Networks, vol. 22(1), pp. 75–81 (2009).
16. Gu, M., & Aamodt, A.: Explanation-boosted question selection in conversational CBR, http://www.idi.ntnu.no/~agnar/publications/eccbr04-expl-ws.pdf, last accessed 2018/5/15 (2004).
17. Gu, M., Aamodt, A., & Tong, X.: Component retrieval using conversational case-based reasoning. Proceedings of ICIIP-2004, International Conference on Intelligent Information Processing, Beijing, China (2004).
18. Hammond, K. J.: CHEF: A model of case-based planning. AAAI-86 Proceedings (1986a).
19. Hammond, K. J.: Learning to anticipate and avoid planning problems through the explanation of failures, AAAI-86 Proceedings (1986b).
20. Hammond, K. J.: Case-based planning: A framework for planning from experience. Cognitive Science, vol. 14, pp. 385–443 (1990a).
21. Hammond, K. J.: Explaining and repairing plans that fail. Artificial Intelligence, vol. 45(1–2), pp. 173–228 (1990b).
22. Kass, A., Leake, D.: Types of explanations, Tech. Rep. ADA183253, DTIC Document (1987).
23. Kass, A.: Developing creative hypotheses by adapting explanations. PhD thesis, Yale University. Northwestern University Institute for the Learning Sciences, Technical Report 6 (1990).
24. Kofod-Petersen, A., Cassens, J., & Aamodt, A.: Explanatory capabilities in the CREEK knowledge-intensive case-based reasoner. Frontiers in Artificial Intelligence and Applications, vol. 173, pp 2835 (2008).
25. Leake, D. B.: Evaluating explanations. In AAAI-88 Proceedings, pp. 251–255 (1988).

26. Leake, D. B.: Goal-Based Explanation Evaluation. Cognitive Science, vol. 15, pp. 509–545 (1991a)
27. Leake, D. B.: An indexing vocabulary for case-based explanation. In Proceedings AAA1-91, pp. 10–15. American Association for Artificial Intelligence, Anaheim (1991b).
28. Leake, D. B.: Evaluating explanations: A content theory. Lawrence Erlbaum Associates, 2014. Psychology Press, New York, NY (1992).
29. Leake, D. B.: Focusing construction and selection of abductive hypotheses. IJCAI, pp. 24–29 (1993).
30. Leake, D. B.: Adaptive similarity assessment for case-based explanation. The International Journal of Expert Systems Research and Applications, vol. 8(2), pp. 165–194 (1995a).
31. Leake, D. B.: Towards goal-driven integration of explanation and action. In Ram, A., & Leake, D. B., Goal-driven learning. Mass: MIT Press, Cambridge (1995b).
32. Leake, D., & McSherry, D.: Introduction to the special issue on explanation in case-based reasoning. Artificial Intelligence Review, vol. 24, pp. 103–108 (2005).
33. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. AAAI Conference on Artificial Intelligence. (2017).
34. Louis, S., Mcgraw, G., & Wyckoff, R. O.: Case-based reasoning assisted explanation of genetic algorithm results. Journal of Experimental & Theoretical Artificial Intelligence, vol. 5(1), pp. 21–37 (1993).
35. Massie, S., Craw, S., Wiratunga, N.: A visualisation tool to explain case-base reasoning solutions for tablet formulation. In: Macintosh, A., Ellis, R., Allen, T. (eds.) AI 2004, pp. 222–234. Springer, London (2005).
36. Maximini, R., Freßmann, A., & Schaaf, M.: Explanation service for complex CBR applications. In: Gonzalez-Calero, P. A. & Funk, P. (eds.), Advances in Case-Based Reasoning, pp. 302–316. Springer-Verlag, Berlin, Heidelberg (2004).
37. McSherry, D.: Interactive case-based reasoning in sequential diagnosis. Applied Intelligence, vol. 14, pp. 65–76 (2001).
38. McSherry, D.: Similarity and compromise, 5th International Conference on Case-Based Reasoning. K. D. Ashley & D. G. Bridge (eds.) LNAI 2689, pp. 122–130, Springer Verlag, (2003).
39. McSherry, D.: A lazy learning approach to explaining case-based reasoning solutions. In B. Díaz Agudo and I. Watson (Eds.): ICCBR 2012, LNCS 7466, pp. 241–254 (2012).
40. Muhammad, K., Lawlor, A., Smyth, B: On the use of opinionated explanations to rank and justify recommendations. FLAIRS Conference 201, pp. 554–559 (2016).
41. Muhammad, K., Lawlor, A., Smyth, B: On the pros and cons of explanation-based ranking. ICCBR 2017, pp. 227–241 (2017).
42. Muhammad, K., Lawlor, A., Smyth, B: A multi-domain analysis of explanation-based recommendation using user-generated reviews. FLAIRS Conference 2018 (2018).
43. Nugent, C., & Cunningham, P.: A case-based explanation system for black-box systems. Artificial Intelligence Review, vol. 24(2), pp 163–178 (2005).
44. Nugent, C., Cunningham, P., & Doyle, D.: The best way to instill confidence is by being right. In H. Muñoz-Avila & F. Ricci (eds.), Case-based reasoning, research and development, 6th International Conference on Case-Based Reasoning, ICCBR 2005, Chicago, IL, USA, 23–26 August 2005. Proceedings, LNCS (vol. 3620, pp. 368–381 (on line 863)). Springer (2005).
45. Nugent, C., Doyle, D., & Cunningham, P.: Gaining insight through case-based explanation. J Intell Inf Syst, vol. 32, pp. 267–295 (2009).
46. Olsson, T., Gillblad, D., Funk, P., & Xiong, N.: Case-based reasoning for explaining probabilistic machine learning. International Journal of Computer Science and Information Technology, vol. 6(2), pp 87–101 (2014).

47. Ong, L. S., Sheperd, B., Tong, L.C., Seow-Choen, F., Ho, Y. H., Tong, L.C., Ho Y. S, Tan, K.: The colorectal cancer recurrence support (CARES) System. Artificial Intelligence in Medicine, 1vol. 1(3), pp. 175–188 (1997).
48. Öztürk, P., Munoz-Avila, H., & Aamodt, A.: Explanation of opportunities. Workshop Program at the 22nd International Conference on Case-based Reasoning. Cork, Ireland (2014).
49. Packer, K. B.: Using hypertext and case-based explanation to help learners access explanations to unexpected grammar forms encountered in native speech examples. Dissertation 3508 (2012).
50. Plaza, E., Armengol, E., & Ontañón, S.: The explanatory power of symbolic similarity in case-based reasoning. Artif. Intell. Rev., vol. 24, pp 145–161 (2005).
51. Rissland, E. L.: The fun begins with retrieval: explanation and CBR. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 1–8. Springer, Heidelberg (2006).
52. Roth-Berghofer, T. R.: Explanations and case-based reasoning: Foundational issues. In Funk, P., Calero, P. A. G., eds.: Advances in Case-Based Reasoning, pp. 389–403, Springer-Verlag (2004).
53. Roth-Berghofer, T., & Bahls, Daniel.: Explanation capabilities of the open source case-based reasoning tool myCBR, pp 23–34 (2008).
54. Roth-Berghofer, T. R., & Cassens, J.: Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In H. Muñoz-Ávila & F. Ricci (Eds.), Case-Based Reasoning Research and Development (vol. 3620, pp. 451–464). Berlin, Heidelberg: Springer Berlin Heidelberg (2005).
55. Sauer, C. S., Hundt, A., & Roth-Berghofer, T.: Explanation-aware design of mobile myCBR-based applications. In B. D. Agudo & I. Watson (Eds.), Case-Based Reasoning Research and Development (vol. 7466, pp. 399–413). Berlin, Heidelberg: Springer Berlin Heidelberg (2012).
56. Schank, R. C. & Abelson, R.: Scripts, plans, goals and understanding, Lawrence Erlbaum Associates, Hillsdale, New Jersey (1977).
57. Schank, R. C.: Dynamic Memory: A theory of reminding and learning in computers and people. Cambridge University Press, Cambridge (1982).
58. Schank, R. C.: Explanation: A first pass. Defense Technical Information Center Technical Report. YALEU/CSD/RR #330 (1984).
59. Schank, R. C.: Explanation patterns – Understanding mechanically and creatively. New York: Lawrence Erlbaum (1986).
60. Schank, R. & Leake, D.: Creativity and learning in a case-based explainer. Artificial Intelligence vol. 40(1–3): pp. 353–385 (1989).
61. Schank, R. C., Kass, A., & Riesbeck, C. K.: Inside case-based explanation. Hillsdale, NJ: Lawrence Erlbaum (1994).
62. Sizov, G., Öztürk, P., & Bach, K.: Evaluation of explanations extracted from textual reports. In: FLAIRS Conference March 2016, pp. 425–429 (2016).
63. Sooriamurthi, R., & Leake, D.: Towards situated explanation [Research abstract]. In: AAAI-94 Proceedings (1994)
64. Sørmo, F., & Cassens, J.: Explanation goals in case-based reasoning. In: Proceedings of the ECCBR 2004 Workshops (Technical Report 142–04). Universidad Complutense de Madrid, Departamento de Sistemas Informáticos y Programación (2004)
65. Sørmo, F., Cassens, J., & Aamodt, A.: Explanation in case-based reasoning–Perspectives and goals. Artificial Intelligence Review, vol. 24(2), pp. 109–143 (2005).
66. Spinelli, M., & Schaaf, M.: Towards explanations for CBR-based applications. In Andreas Hotho and Gerd Stumme (eds.) Proceedings of the LLWA Workshop 2003, pp. 229–233, Karlsruhe, Germany, 2003. AIFB Karlsruhe (2003).

67. Zenobi, G., & Cunningham, P.: An approach to aggregating ensembles of lazy learners that supports explanation. In Craw, S. & Preece, A. (eds.) Advances in Case-Based Reasoning: Proceedings ECCBR 2002, pp. 436–447, Springer, Berlin Heidelberg (2002).
68. Swartout W., & Moore J.: Explanation in second generation expert systems, Second generation expert systems, J. David, J. Krivine, & R. Simmons (eds.) Springer, Berlin, pp. 543–585 (1993).
69. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Retrieved from https://arxiv.org/pdf/1706.07269.pdf (2017).
70. Abbott, R.: Subjectivity as a concern for information science: A Popperian perspective. Journal of Information Science, vol. 30(2), pp. 95–106 (2004).
71. Borlund, P.: The concept of relevance in IR. Journal of the American Society for Information Science & Technology, vol. 54(10), pp. 913–925 (2003).
72. Buckland, M. K.: Relatedness, relevance and responsiveness in retrieval systems. Information Processing and Management, vol. 19(3), pp. 237–41 (1983).
73. Nolin, J.: 'Relevance' as a boundary concept: Reconsidering early information retrieval. Journal of Documentation, vol. 65(5), pp. 745–67 (2009).
74. Saracevic, T.: Relevance: A review of and a framework for the thinking on the notion in information science. Journal of the American Society for Information Science, vol. 26(6), pp. 321–43 (1975).
75. Saracevic, T.: Relevance: a review of the literature and a framework for thinking on the notion in information science. Part II: Nature and manifestations of relevance. Journal of the American Society for Information Science & Technology, vol. 58(13), pp. 1915–33 (2007a).
76. Saracevic, T.: Relevance: a review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. Journal of the American Society for Information Science & Technology, vol. 58(13), pp. 2126–2144 (2007b).
77. Schamber, L., Eisenberg, M. B., & Nilan, M. S.: A re-examination of relevance: Toward a dynamic, situational definition, Information Processing & Management, vol. 26(6), pp 755–776 (1990).
78. Taylor, A., Zhang, X., & Amadio, W. J.: Examination of relevance criteria choices and the information search process. Journal of Documentation, vol. 65(5), pp. 719–744 (2009).
79. Taylor, A.: User relevance criteria choices and the information search process. Information Processing & Management, vol. 48(1), pp. 136–153 (2012).
80. White, H. D.: Relevance in theory. In M. J. Bates & M. N. Maack (Eds.), Encyclopedia of library and information sciences (3rd ed.; vol. 6; pp. 4498–4511). Taylor & Francis., New York (2010).
81. Wilson, P.: Situational relevance. Information Storage and Retrieval, vol. 9(8), pp. 457–471 (1973).
82. Richter, M. M., & Weber, R. O.: Case-based reasoning: A textbook. Springer-Verlag, Berlin Heidelberg (2013).
83. Gunning, D.: Explainable artificial intelligence (XAI). DARPA/I2O, Program Update November 2017 (2017).
84. Vasilyeva, N., Wilkenfeld, D., & Lombrozo, T.: Contextual utility affects the perceived quality of explanations. Psychonomic Bulletin & Review, pp. 1436–1450 (2017).
85. Giffin, C., Wilkenfeld, D., & Lombrozo, T.: The explanatory effect of a label: Explanations with named categories are more satisfying. Cognition, vol. 168, pp. 357–369 (2017).
86. Lombrozo, T.: Explanatory preferences shape learning and inference. Trends in Cognitive Sciences, vol. 20, pp. 748–759 (2016).
87. Lombrozo, T. & Gwynne, N. Z.: Explanation and inference: Mechanistic and functional explanations guide property generalization. Frontiers in Human Neuroscience, vol. 8 (2014).

88. Hayes, B., and J. A. Shah: Interpretable models for fast activity recognition and anomaly explanation during collaborative robotics tasks, International Conference on Robotics and Automation (ICRA), Singapore, IEEE, 05/2017 (2017).
89. Hayes, B., and J. A. Shah: Improving robot controller transparency through autonomous policy explanation, 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2017), 03/2017 (2017).
90. Hoffman, R.R., & Klein, G: Explaining explanation, part 1: Theoretical foundations. IEEE Intelligent Systems, vol. 32, pp. 68–73 (2017).
91. Hoffman, R.R., Mueller, S.T., & Klein, G.: Explaining explanation, part 2: Empirical foundations. IEEE Intelligent Systems, vol. 32, pp. 78–86 (2017).

# Preface

The successes of Deep Learning have not gone unnoticed and are increasing inspiring novel methods and techniques that exploit Deep Learning for the benefit of Case-based reasoning (CBR). The potentials of Deep Learning for CBR are numerous and include potential improvement in knowledge aggregation and feature extraction for case representation, efficient indexing and retrieval architectures as well as assisting with case adaptation.

In this second edition of the ICCBR Workshop on Case-Based Reasoning and Deep Learning, 3 papers have been accepted for presentation from researchers from Germany and UK. The first contribution presents a system that uses deep convolutional neural networks (CNNs) for classifying images of skin lesions in a CBR system for early detection of melanoma. The proposed CNN approach is used to improve retrieval of new input images in the CBR system called DePicT Melanoma CLASS, which is designed to support users with more accurate recommendations relevant to their requested problem (e.g., image of affected area). The second contribution presents an approach for personalised human activity recognition from wearable sensor data using a neural network architecture called a matching network. support set. A important advantage of matching networks over standard k-Nearest Neighbour is their ability to iteratively learn feature representations that maximise classification performance, given a chosen similarity metric. This contribution presents a study of 5 different similarity metrics used with matching networks for personalised HAR. The third contribution addresses an important challenge in HAR which is the fact that a variable number of sensors could be present during training and after deployment of a HAR system. This challenge is addressed using a Neural Translator, capable of generating missing sensor data from any available sensors at test time. The contribution demonstrates the use of the Neural Translator for HAR using k-Neaerest Neighbour classifier with promising results

July 2018

Sadiq Sani, Stewart Massie and Nirmalie Wiratunga
Program Chairs
Case-Based Reasoning and Deep Learning Workshop, 2018

# Organization

## Organizing Committee

Sadiq Sani (Robert Gordon University, UK)
Stewart Massie (Robert Gordon Univerity, UK)
Nirmalie Wiratunga (Robert Gordon Univerity, UK)

## Program Committee

Daniel Lopez-Sanchez (Universidad de Salamanca, Spain)
Kerstin Bach (Norwegian University of Science and Technology, Norway)
Marc Pickett (Google, USA)
Michael Floyd (Knexus Research Corporation , USA)

# Enriching CBR recommender system by classification of skin lesions using deep neural networks

Sara Nasiri, Julien Helsper, Matthias Jung and Madjid Fathi

Department of Electrical Engineering & Computer Science, University of Siegen
Institute of Knowledge Based Systems & Knowledge Management
Hölderlinstr. 3, 57076 Siegen, Germany
sara.nasiri@uni-siegen.de,
(julien.helsper,matthias.jung)@student.uni-siegen.de,
fathi@informatik.uni-siegen.de

**Abstract.** An approach to classify skin lesions using deep learning for early detection of melanoma in a CBR system is proposed. This approach has been employed for retrieving new input images from the case base of DePicT Melanoma CLASS to support users with more accurate recommendation relevant to their requested problem (e.g., image of affected area). The efficiency of our system has been verified by utilizing the ISIC archive dataset in analysis of skin lesion classification as a benign and malignant melanoma.

**Keywords:** Machine learning · Deep learning · Image classification · Case retrieval · Melanoma skin cancer.

## 1 Introduction

Although case-based reasoning (CBR) has been applied in a number of medical systems, only a few systems have been developed for melanoma. e.g., the CBR system of Nicolas et al. used rules to answer medical questions based on the knowledge extracted from image data [12]. The survival rates of melanoma from early to terminal stages vary between 15 and 65% [2]; therefore, having the right information at the right time via early detection is essential to surviving this type of cancer. Accordingly, developing decision support systems has become a major area of research in this field [9]. The best path to early detection is recognizing new or changing skin growths, especially those that appear different from other moles [1]. Even after treatment, it is very important for patients to keep up on their medical history and records. In this paper, we propose a hybrid CBR system and evaluate its performance on the skin lesions classification (benign or malignant). DL helps researchers absolutely to treat and detect diseases by analyzing medical data (e.g., medical images). One of the representative models among the various deep-learning models is a convolutional neural network (CNN) which is also integrated with CBR for classification [8]. Although convolutional networks outperform other methods in many recognition tasks and in the

classification of particular melanomas [15] [4], deep networks generally require thousands of training samples (labled classes). This classification of skin cancer is comparable to that of dermatologists detected [4]. In the proposed system, deep neural networks used as a case classifying in the context of CBR methodology and its retrieval process. Therefore this hybrid system has the advantages of deep learning (DL) and CBR and benefits from both. Our paper is organized as follows: Section 2, briefly describes the background in terms of preliminary system called DePicT Melanoma CLASS. The proposed system is followed in Section 3 by a description of image processing and classification with deep neural networks. This Section explains the tools and dataset which we have used for the implementation of our system and discusses the results and evaluation. Finally, Section 4 concludes the paper.

## 2 DePicT Melanoma CLASS

Various skin lesions classification systems have been developed using support vector machines (SVMs) and k nearest neighbor (k-NN) like interactive object recognition methodologies to perform border segmentation [13], extract global and local features and apply Otsus adaptive thresholding method [6]. Sumithra et al. utilized SVM and k-NN for skin cancer classification based on region-growing segmentation with results of (f-measure) 46% and 34%, respectively [14]. In the previous study, DePicT CLASS [10] was used to retrieve the textual components of requested problems and classify melanoma images using region growing method based on SVM and k-NN to support patients and health providers in managing the disease [11]. The case base of DePicT Melanoma CLASS is built based on the textual information about melanoma from the AJCC [1] staging, melanoma skin cancer information data base [2] and melanoma images from the ISIC archive dataset[3]. Each case has a word association profile for main keywords extracted from melanoma textbooks and reports (fifteen melanoma-related papers and books) from which case descriptions and references are built. The case structure comprises a case description (melanoma taypes and stages) and recommendation (e.g., related images and treatment plan) including image features, segmentation processes, reference images, identified keywords, and a word association profile.

## 3 Proposed System

In this study, we applied an end-to-end CNN framework (machine learning system) to detect malignant melanoma using images from ISIC archive dataset. As shown in Fig. 1, the goal of our proposed system is classifying new images which is enriched by using DL.

[1] American Joint Committee on Cancer: Melanoma of the Skin Staging
[2] Melanoma Skin Cancer, https://www.cancer.org/cancer/melanoma-skin-cancer.html
[3] https://isic-archive.com/

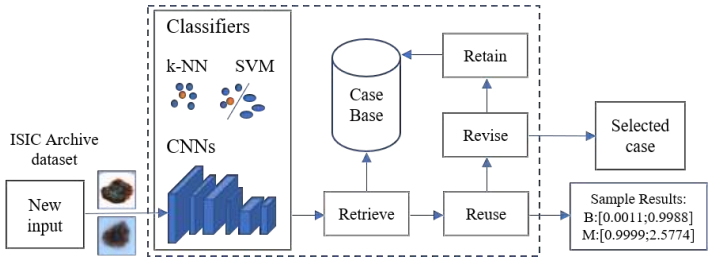



**Fig. 1.** The proposed CBR system.

CNNs are created of several convolutional layers (involving linear and non-linear operators), pooling, inner products which are fully connected layers and losses like softmax and the architecture for its state-of-the-art has many parameters. As shown in Fig. 2, we built a 16-layer model which contains thirteen convolutional (series of conv layers: conv1-conv5) and three fully connected (FC) layers. The input image is the first layer ($h \times w \times d$ which $h \times w$ is the pixel size and $d$ is the color channel, here is $256 \times 256 \times 3$). The configuration of our network is also illustrated at the bottom of Fig. 2 regarding the filter kernel and feature map size represented by a three-dimensional array ($h \times w \times d$ which $h$ and $w$ are spatial dimensions and d is the number of channels). Our requested problem is a two-way classification problems: malignant and benign classes, therefore, the last fully connected layer (FC8) have 2 channels which is the same as the number of classes (B and M, see also Fig.1 - Sample Results), while the first-two fully connected layers (FC6 and FC7) have 4,096 channels. Therefore, the case representation is generated based on the features coming out of the last layer of our network. After creation of the whole case base based on the labeled images, in retrieval process of new input image, a distance-based method can work over the case representation to find the matched class.

For applying deep-learning, we have utilized Caffe[4][5] which is a deep learning framework developed by Berkeley AI Research (BAIR)[5]. As it mentioned the ISIC Archive dataset containing images of benign and malignant melanoma was used for image-processing and classification. In the first round, 300 images for training and 100 for testing were utilized. By continuing the previous study, we have first used this dataset and then increased the training and test images in the second round to 1400 and 400, respectively. A total of 1400 dermoscopy images comprising malignant (978 images) and benign (422 images), were analyzed in this study to train our classifier.

Matlab (2017a) was utilized to develop DePicT Melanoma CLASS, in particular with the use of Image Processing Toolbox, Parallel Computing Toolbox, Matlab Compiler and Coder, and App Designer. For comparing CNN with the

[4] http://caffe.berkeleyvision.org/
[5] http://bair.berkeley.edu/

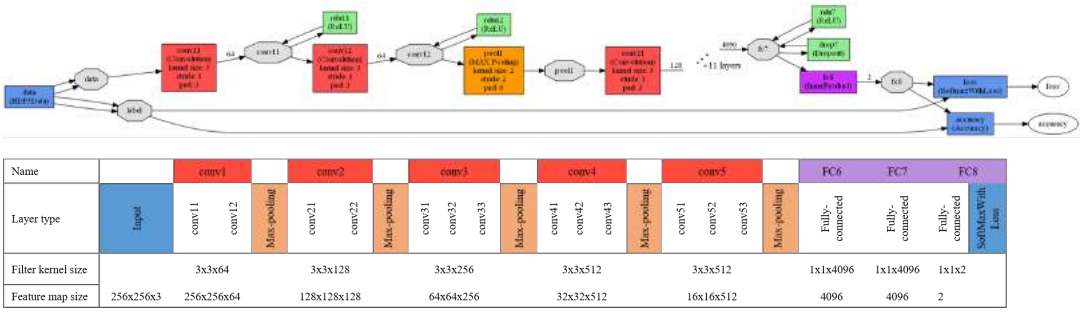



**Fig. 2.** Layer view and a MNIST digit classification example of a Caffe network - Sixteen layers excluding max-pooling.

previous classifiers, we have done the third and fourth test (200 images for B & 200 for M). DePicT Melanoma CLASS achieved appropriate results. Its performance in terms of comparison of its evaluation scores in these four tests is shown in Table 1.

**Table 1.** The comparison of evaluation scores (precision, recall (sensitivity), specificity, f-measure and accuracy) of DePicT Melanoma CLASS.

| Classification and retrieval | TP | TN | FP | FN | Pre. | Rec.(Sen.) | Spec. | F-m. | Acc. |
|---|---|---|---|---|---|---|---|---|---|
| k-NN: 1st test (300, 100) | 30 | 34 | 16 | 20 | 0.65 | 0.6 | 0.68 | 0.62 | 0.64 |
| SVM: 2nd test (300. 100) | 25 | 37 | 13 | 25 | 0.66 | 0.5 | 0.74 | 0.57 | 0.62 |
| CNN: 3rd test (300,100) | 35 | 33 | 17 | 15 | 0.67 | 0.70 | 0.66 | 0.68 | 0.68 |
| CNN: 4th test (1400,400) | 185 | 167 | 33 | 15 | 0.85 | 0.92 | 0.83 | 0.88 | 0.88 |

## 4 Conclusion and Outlook

In this paper, a training and half-testing method were deployed for composing a comparatively accurate CNN model from sample images of ISIC archive dataset. Although further data analysis is necessary to improve its accuracy, CNN would be helpful for the early detection of malignant melanoma. Analysis of the results obtained by testing a melanoma dataset suggests that our enriched case-based system (via CNN which made case classification more efficient) for detecting malignant melanoma is fit for the purpose of supporting users by providing relevant information. Further work will involve extending the training phase by using more images and more classes (different types and stages of melanoma skin cancers). The retrieval phase could also be further developed based on the new classes in regenerating a case representation.

## References


1. American Cancer Society, Cancer Facts and Figures 2017. Genes and Development, 21(20), 2525–2538 (2017).
2. Ali, A.R. and Deserno, T. A Systematic Review of Automated Melanoma Detection in Dermatoscopic Images and its Ground Truth Data. Proceedings of SPIE, 8318I, doi: 10.1117/12.912389 (2012).
3. Coit, D. G. et al. NCCN Guidelines Insights: Melanoma, Version 3.2016. Journal of the National Comprehensive Cancer Network : JNCCN, 14(8), 945–58 (2016).
4. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118 (2017).
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093, (2014).
6. Kavitha, J. C., Suruliandi, A., Nagarajan, D., and Nadu, T. Melanoma Detection in Dermoscopic Images using Global and Local Feature Extraction. International Journal of Multimedia and Ubiquitous Engineering, 12(5), 19–28 (2017).
7. Lee, T., Ng, V., Gallagher, R., Coldman, A., and McLean, D. Dullrazor: A software approach to hair removal from images. Computers in Biology and Medicine, 27(6), 533–543 (1997).
8. López-Sánchez, D., Corchado, J. M., González Arrieta, A., A CBR System for Image-Based Webpage Classification: Case Representation with Convolutional Neural Networks, FLAIRS 2017, AAAI Press, 483–488 (2017).
9. Masood, A. and Al-Jumaily, A. A. Computer aided diagnostic support system for skin cancer: A review of techniques and algorithms. International Journal of Biomedical Imaging, Volume 2013, Article ID 323268, doi.org/10.1155/2013/323268 (2013).
10. Nasiri, S., Zenkert, J., and Fathi, M. Improving CBR adaptation for recommendation of associated references in a knowledge-based learning assistant system. Neurocomputing, 250, 5–17 (2017).
11. Nasiri, S., Jung, M., Helsper, J. and Fathi, M. Detect and Predict Melanoma Utilizing TCBR and Classification of Skin Lesions in a Learning Assistant System In: Rojas I., Ortuo F. (eds) Bioinformatics and Biomedical Engineering. IWBBIO 2018. Lecture Notes in Computer Science, vol 10813. Springer, 531–542, (2018).
12. Nicolas, R., Vernet, D., Golobardes, E., Fornells, A., Puig, S., and Malvehy, J. Improving the Combination of CBR Systems with Preprocessing Rules in Melanoma Domain. In Workshop Proceedings of the 8th International Conference on Case-Based Reasoning, 225–234 (2009).
13. Sabouri, P., GholamHosseini, H., Larsson, T., and Collins, J. A cascade classifier for diagnosis of melanoma in clinical images. In Annual: International Conference of the IEEE Engineering in Medicine and Biology Society, 6748–6751 (2014).
14. Sumithra, R., Suhil, M., and Guru, D. S. Segmentation and classification of skin lesions for disease diagnosis. In Procedia Computer Science, volume 45, 76–85 (2015).
15. Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. CoRR, abs/1505.0, 234–241 (2015).
16. Yu C, Yang S, Kim W, Jung J, Chung KY, et al. Acral melanoma detection using a convolutional neural network for dermoscopy images. PLOS ONE 13(3): e0193321, (2018).

# Study of Similarity Metrics for Matching Network-Based Personalised Human Activity Recognition

Sadiq Sani[1], Nirmalie Wiratunga[1], Stewart Massie[1], and Kay Cooper[2]

[1] School of Computing Science and Digital Media,
[2] School of Health Sciences,
Robert Gordon University,
Aberdeen AB10 7GJ, Scotland, UK
{s.sani,n.wiratunga,s.massie,k.cooper}@rgu.ac.uk

**Abstract.** Personalised Human Activity Recognition (HAR) models trained using data from the target user (subject-dependent) have been shown to be superior to non personalised models that are trained on data from a general population (subject-independent). However, from a practical perspective, collecting sufficient training data from end users to create subject-dependent models is not feasible. We have previously introduced an approach based on Matching networks which has proved effective for training personalised HAR models while requiring very little data from the end user. Matching networks perform nearest-neighbour classification by reusing the class label of the most similar instances in a provided support set, which makes them very relevant to case-based reasoning. A key advantage of matching networks is that they use metric learning to produce feature embeddings or representations that maximise classification accuracy, given a chosen similarity metric. However, to the best of our knowledge, no study has been provided into the performance of different similarity metrics for matching networks. In this paper, we present a study of five different similarity metrics: Euclidean, Manhattan, Dot Product, Cosine and Jaccard, for personalised HAR. Our evaluation shows that substantial differences in performance are achieved using different metrics, with Cosine and Jaccard producing the best performance.

## 1 Introduction

Automatic recognition and tracking of human activity using wearable sensors is increasingly being adopted for health care applications e.g. management of chronic low back pain in SelfBACK [1] [1]. An important consideration for HAR applications is classifier training, where training examples can either be acquired from a general population (subject-independent), or from the target user of the system (subject-dependent). Previous works have shown using subject-dependent data to result in superior performance [2–4]. Matching networks [6]

[1]The SelfBACK project is funded by European Union's H2020 research and innovation programme under grant agreement No. 689043.

have been successfully applied for efficiently learning personalised HAR models [5]. Given a (typically small) support set of labelled examples, matching networks are able to classify an unlabelled example by reusing the class labels of the most similar examples in the support set. A key advantage of matching networks is that they use metric learning to produce feature embeddings or representations that maximise classification accuracy, given a chosen similarity metric. Thus, it is important to investigate how different similarity metrics affects the performance of matching networks for personalised HAR. Accordingly, in this paper, we present a study of five different similarity metrics used with matching networks.

## 2 Personalised HAR using Matching Networks

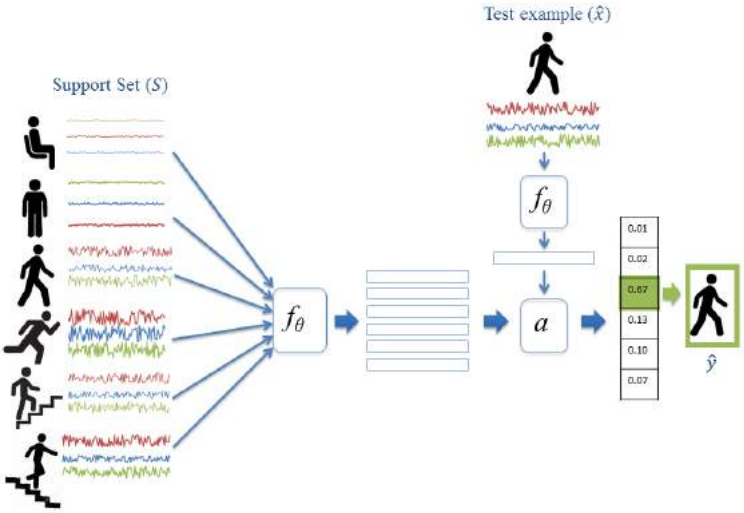



**Fig. 1.** Illustration of matching network for HAR.

The aim of matching networks is to learn a model that maps an unlabelled example $\hat{x}$ to a class label $\hat{y}$ using a small support set $S$ of labelled examples. This is illustrated in Figure 1. Given a set of instances $X = \{x|x \text{ is an instance vector}\}$, a set of class labels $L = \{y|y \text{ is a class label}\}$, an embedding function $f_\theta$ which in this is case a neural network parameterised by $\rho$, the function $a$ is an attention mechanism that takes the embedded representation of a test instance and a support set $S$ and returns a probability distribution $P(y|\hat{x}, S)$ over class labels $y$ of instances in $S$. To train the matching network for personalised HAR, we also define a set of users $U$ where each user $u_j \in U$ is comprised of a set of labelled examples as follows:

$$u_j = \{(x, y)|x \in X, y \in L\} \tag{1}$$

Next we define a set of training instances $T_j$ for each user $u_j$ as follows:

$$T_j = \{(S_j, B_j)\} \tag{2}$$

i.e., $T_j$ is made up of user-specific support and target set pairs $S_j$ and $B_j$ respectively, where $S_j = \{(x, y)|(x, y) \in u_j\}$ and $B_j = \{(x, y)|(x, y) \in u_j, (x, y) \in S_j\}$. Note that the set of labels in $S_j$ is always equivalent to $L$ because we are interested in learning a classifier over the entire set of activity labels. Accordingly, $S_j$ contains $m$ examples for each class $y \in L$ and the cardinality of $S_j$ is $|S| = m \times |L|$. Both $S_j$ and $B_j$ are sampled at random from $u_j$ $l$ times to create $T_j$. Each $B_j$ is used with it's respective $S_j$ by classifying each instance in $B_j$ using $S_j$ and computing loss using categorical cross entropy. The network is trained using stochastic gradient descent and back propagation.

## 3 Similarity Metrics

Matching networks use a similarity metric to match a given test instance to the most similar instances in a support set. In the following subsections, We discuss five of the most popular similarity metrics used in literature.

### 3.1 Euclidean

Euclidean distance is perhaps the most popular metric used for estimating similarity between items represented as numerical vectors. The Euclidean metric gives the distance between any two points in n-dimensional space as the length of a straight line connecting those two points. Euclidean distance can be converted to similarity simply by taking the inverse as shown in Equation 3.

$$Euclidean(\hat{x}, x) = \frac{1}{\sum\sum (\hat{x}_j \sim x_j)^2 + 1} \tag{3}$$

### 3.2 Manhattan

The Manhattan distance between two items is computed as the sum of absolute differences between the values of their dimensions. This can also be converted to a similarity by taking the inverse as shown in Equation 4. In comparison with Euclidean, the Manhattan metric is less susceptible to large differences in values in few dimensions.

$$Manhattan(\hat{x}, x) = \frac{1}{\sum |\hat{x}_j \sim x_j| + 1} \tag{4}$$

### 3.3 Cosine

Cosine metric estimates similarity between two items by measuring the angle between their vectors in n-dimensional space. Cosine similarity can be computed as shown in Equation 5.

$$Cosine(\hat{x}, x) = \frac{\sum_j^n \hat{x}_j x_j}{\sqrt{\sum_j^n \hat{x}_j^2}\sqrt{\sum_j^n x_j^2}} \tag{5}$$

### 3.4 Dot Product

Dot product measures the projection of one vector onto another in n-dimensional coordinate space as shown in Equation 6. Unlike cosine similarity, dot product is not normalised and thus takes into account both angle and magnitude of the two vectors.

$$DotProduct(\hat{x}, x) = \sum \hat{x}_j x_j \tag{6}$$

### 3.5 Jaccard

The Jaccard metric measure similarity between finite sets as the ratio of the size of the intersection to the size of the union of the sets. The general form of the Jaccard metric for finding similarity between numerical vectors is provided in Equation 7

$$Jaccard(\hat{x}, x) = \frac{\sum \hat{x}_j x_j}{\sum \hat{x}_j^2 + \sum x_j^2 \sim \sum \hat{x}_j x_j} \tag{7}$$

## 4 Evaluation

Evaluation is conducted on a dataset of 50 users with 9 activity classes and about 3 minutes of activity data per class. We adopt a hold-out validation strategy where 8 out of the 50 users are randomly selected for testing. To simulate user provided samples for creating personalised support sets, we hold out the first 30 seconds of each test user's data for creating the support set. This leaves approximately 150 seconds of data per activity which are used for testing, Performance is reported using macro-averaged F1 score.

In the evaluation, we explore the performance of matching network for personalised HAR using the five similarity metrics presented in Section 3. Five different matching networks are trained, each using one of the similarity metrics. All matching networks are identical except for the similarity difference in similarity metric are all trained for the same number of epochs. Results are presented in Table 1. It can be observed that the best results are achieved using Cosine and Jaccard. Euclidean produces a reasonably close third place performance while both Manhattan and Dot Product are a distant forth and fifth place

respectively. Note that both Cosine and Jaccard metrics are normalised by the magnitude of the vectors involved in the similarity. This suggests that similarity metrics that do not take into account differences in vector magnitudes tend to work better for this application.

**Table 1.** Results of different algorithms showing F1 scores.

| *Metric* | Euclidean | Manhattan | Cosine | Dot Product | Jaccard |
|---|---|---|---|---|---|
| *F1 Score* | 0.757 | 0.696 | **0.788** | 0.694 | 0.783 |

## 5 Conclusion

In this paper, we have presented a comparative study of 5 different similarity metrics for personalised HAR using matching networks. Results show cosine and Jaccard metrics to produce the best classification performance. Our work suggests that the choice of similarity metric is a very important consideration for matching networks and that the performance of metrics that do not consider differences in vector magnitude (e.g. Cosine and Jaccard) are superior to metrics that take into account vector magnitude (e.g. Euclidean, Manhattan and dot product).

## References


1. Bach, K., Szczepanski, T., Aamodt, A., Gundersen, O.E., Mork, P.J.: Case representation and similarity assessment in the selfback decision support system. In: Proceedings of 24th International Conference on Case-Based Reasoning. pp. 32–46. Springer International Publishing (2016)
2. Berchtold, M., Budde, M., Gordon, D., Schmidtke, H.R., Beigl, M.: Actiserv: Activity recognition service for mobile phones. In: Proceedings of International Symposium on Wearable Computers. pp. 1–8 (2010)
3. Jatoba, L.C., Grossmann, U., Kunze, C., Ottenbacher, J., Stork, W.: Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In: Proceedings of 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 5250–5253 (2008)
4. Sani, S., Wiratunga, N., Massie, S., Cooper, K.: knn sampling for personalised human activity recognition. In: Proceedings of International Conference on Case-Based Reasoning. pp. 330–344. Springer (2017)
5. Sani, S., Wiratunga, N., Massie, S., Cooper, K.: Personalised human activity recognition using matching networks. In: Proceedings of International Conference on Case-Based Reasoning. Springer (2018)
6. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D.: Matching networks for one shot learning. In: Proceedigns of Advances in Neural Information Processing Systems. pp. 3630–3638 (2016)

# Improving Human Activity Recognition with Neural Translator Models

Anjana Wijekoon ✉, Nirmalie Wiratunga, and Sadiq Sani

Robert Gordon University, Aberdeen AB10 7GJ, Scotland, UK
{a.wijekoon, n.wiratunga, s.a.sani}@rgu.ac.uk

**Abstract.** Multiple sensor modalities provide more accurate Human Activity Recognition (HAR) compared to using a single modality, yet the latter is more convenient and less intrusive. It is advantages to create a model which learns from all available sensors; although it is challenging to deploy such model in an environment with fewer sensors, while maintaining reliable performance levels. We address this challenge with Neural Translator, capable of generating missing modalities from available modalities. These can be used to generate missing or "privileged" modalities at deployment to improve HAR. We evaluate the translator with k-NN classifiers on the SelfBACK HAR dataset and achieve up-to 4.28% performance improvements with generated modalities. This suggests that non-intrusive modalities suited for deployment benefit from translators that generate missing modalities at deployment.

**Keywords:** Human Activity Recognition · Machine Learning · Privileged Learning

## 1 Introduction

Reasoning with multi-modal sensor data is an active area of research with applications fielded in multiple domains, including Human Activity Recognition(HAR), Robotics and Interactive Natural Interfaces. Typically HAR applications are related to tracking or monitoring movements such as ambulatory activities [5], activities of daily living [1] or exercises [6]. Inertial sensors and ambient sensors are mainly used in such applications to track user activity. For HAR, having multiple modalities is advantageous as it captures contextually richer representations. However access to all sensor modalities at deployment can be restricted due to ease of use or erroneous behaviours. This poses an interesting challenge of effectively deploying reasoning models with fewer modalities, compared to the number of modalities used in training.

We address this challenge as a Privileged Learning (PL) [8] problem. PL defines an additional feature space (Privileged Information) that improves classification performance, but is only available during training. It resembles how humans learn better with a teacher. In HAR we recognise this additional feature space as "Privileged Sensor Modalities", that are available during training

but not after deployment. Our initial evaluations suggested that, simply ignoring privileged modalities result in poor performance. Therefore we recognise the need for estimating privileged sensor modalities at deployment. We also learnt that there is no significant linear correlation between sensor modalities, eliminating the possibility of using a simpler estimation method such as linear regression to generate the missing data. Accordingly we implementing a generative neural networks inspired translator that can learn non-linear mapping between modalities.

Recent literature suggest the use of generative models in image/video captioning [9], language translation [7] and time-series forecasting [3] with Recurrent Neural Networks (RNNs). We did not observe any advantage of using RNN as a translator given that our data has no significant temporal dependencies. Generative Adversarial Networks (GANs) is another upcoming generative model, applied successfully in image generation from random noise [2]. Yet GANs fail to generate an output influenced by the input sensor data and the class. Our architecture closely resembles Auto-encoders, which are successfully applied in audio and video reconstruction [4]. The goal of Auto-encoders is to build an abstract feature representation of given data. In contrast we focus on learning mappings between different sensor data in order to transform one to another.

We will introduce Privileged Learning for HAR and our Neural Translator in Section 2. Successive sections will present the SelfBACK Dataset, Experiment Design and Results. Finally we will discuss future improvements in Conclusion.

## 2 Privileged Learning with Neural Translator

We illustrate Privileged Learning (PL) for HAR referring to the two modalities of the SelfBACK dataset [1]; Wrist (W) and Thigh (T). Let $X_W$ and $X_T$ represent input modalities. We select $X_T$ as the privileged sensor modality due to its intrusiveness in real life and comparatively better performance in HAR.

Figure 1 A refers to the training stage of the classification model where both sensors' data is available. Figure 1 C illustrates deployment of the classification model where only $X_W$ is present. If we only use $X_W$ to recognise an activity at deployment, the model performance is highly penalised. Accordingly we train a translator which learns the mapping between two sensor data streams $X_W$ and $X_T$; which is done in parallel to the training of the classification model. More generally, this mapping can be between any number of sensor modalities. The input layer consists of features representing modalities that are present at test time and the output estimates the missing modalities.

Figure 1 B illustrates the Neural Translator in detail, which uses the wrist modality $X_W$ to estimate the thigh modality $X_T^-$. We use a fully connected neural network to estimate privileged modality, where it learns a neural mapping between its input and output layers. A single hidden layer is introduced to learn

[1] The SelfBACK project is funded by European Union's H2020 research and innovation programme under grant agreement No. 689043. More details available: http://www.selfback.eu. The dataset is publicly accessible from https://github.com/selfback/activity-recognition

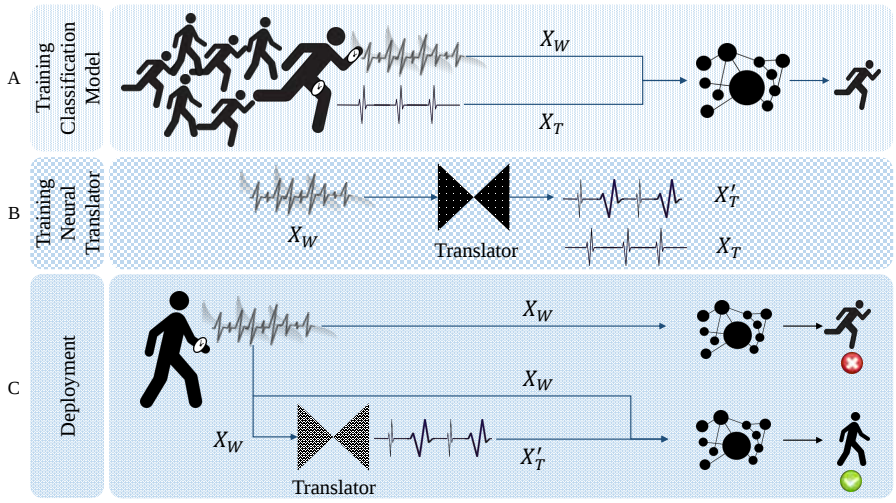



**Fig. 1.** Training Classification Model with Privileged Sensor Data

the feature mapping from input to the output units. During training, given an input, the network learns to generate a representation of the output modality that is as close to the actual values. This is enforced by using a loss function of Mean Squared Error (MSE) between predicted output $X_T^-$ and expected output $X_T$. We will refer to the Neural Translator as $T^N$.

## 3 SelfBACK Dataset

SelfBACK Dataset is compiled with two tri-axial accelerometer data streams, belonging to 6 activity classes. Accelerometers were mounted on the right-hand wrist and thigh of each subject (thus forming 2 modalities). The data was recorded at $100Hz$ sampling rate with 34 individuals. We perform three pre-processing steps on the dataset to prepare it for the translator and the classifier. First we use a sliding window size of 3 seconds with no overlap to create instances. Next we convert three-dimensional raw data instances into single dimension Discrete Cosine Transform (DCT) feature vectors of size 180. It conveniently simplifies the task of the translator where the mapping is learnt between two abstract feature representations instead of raw data. Finally data is normalised to ensure that the k-NN classifiers are unaffected by scalar differences between different modalities.

## 4 Experiment Design

We use k-NN as the classifier which provides interpretable results compared to a neural network. We apply Leave-One-Person-Out (LOPO) cross validation for

the classifier and the translator. We use three configurations with no privileged modalities as the baseline; and use accuracy of classification to study the contribution of the translator in the performance gains of HAR.

We experiment on different number of hidden units, while maintaining the number of hidden layers to one. We confirm that a narrow hidden layer supports learning better mappings between sensors by discarding arbitrary noise. In addition we observe that the translator is over-fitting to training data when increasing the number of hidden layers (thus increasing number of trainable parameters). Accordingly we identify the most optimal architecture for $T_N$ as 1 hidden layer of 96 units.

We follow naming convention $f(X_i/X_j)$ to indicate a classification model trained with set of modalities $X_i$; and $X_j$ are privileged modalities. Here $X_j = \in$ indicates that none of the modalities were considered as privileged after deployment. $X_j = T$ indicates that thigh is a privileged modality, and at deployment it will be estimated with Neural Translator $T_N(W/T)$ which estimates thigh data from original wrist data.

## 5 Results

We present baseline results of classification with no privileged modalities on Figure 2. Baseline results confirm that thigh is clearly the privileged sensor modality for the SelfBACK dataset.

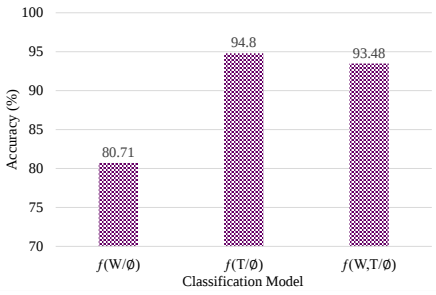



**Fig. 2.** Baseline classification results

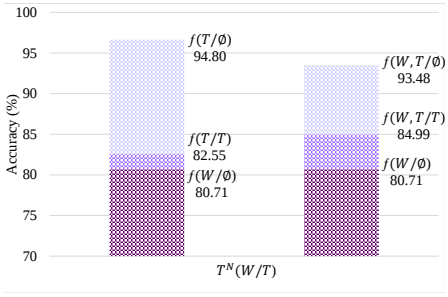



**Fig. 3.** Classification with $T_N$

Figure 3 shows classification results with $T_N(W/T)$. Here each bar shows the lower and upper bounds set by the baselines. Upper bound uses the actual data instead of the estimated after deployment; whilst the lower bound is when the privileged modality is not used for HAR. Ideally we want the translator to improve upon the lower bound to get closer to the upper.

The first bar shows that we can achieve 1.84% improvement over $f(W/\in)$ using estimated thigh on a model trained with original thigh data ($f(T/T)$). The second bar shows that the Neural Translator has significantly improved

the performance by 4.28% over $f(W/\in)$ using estimated thigh data on a model trained with original wrist and thigh data ($f(W,T/T)$); brining it closer to the upper bound set by $f(W,T/\in)$.

These results suggest that a classifier trained with multiple modalities, can be used with a single or smaller subset of modalities in deployment. It is not only possible but improves performance significantly. The Neural Translator learns the non-linear correlations between input and output modalities, discarding ambiguities and noise of the source modalities. As a result the estimated modalities improve performance of the HAR classification at deployment.

## 6 Conclusion

We introduced the Neural Translator to improve HAR by augmenting missing modalities with estimated data. Our results show significant improvement of performance with estimated sensor data in k-NN classification. In addition to estimating privileged modalities, this versatile method can be used to augment incomplete data due to noise or technical faults. We believe there is further opportunity to improve Neural Translator with other deep learning techniques, which we plan to address in future. Finally this work demonstrates that translators can minimise sensors at deployment while improving performance which contributes towards an sustainable HAR solution.

## References

1. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J.d.R., Roggen, D.: The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Rec. Letters **34**(15), 2033–2042 (2013)
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in NIPS. pp. 2672–2680 (2014)
3. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traff c speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies **54**, 187–197 (2015)
4. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the ICML-11. pp. 689–696 (2011)
5. Sani, S., Massie, S., Wiratunga, N., Cooper, K.: Learning deep and shallow features for human activity recognition. In: Int. Conf. on Knowledge Science, Engineering and Management. pp. 469–482. Springer (2017)
6. Sundholm, M., Cheng, J., Zhou, B., Sethi, A., Lukowicz, P.: Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In: Proc. of the 2014 ACM Int. Joint Conf. on pervasive and ubiquitous computing. pp. 373–382. ACM (2014)
7. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in NIPS. pp. 3104–3112 (2014)
8. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural networks **22**(5), 544–557 (2009)
9. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR, 2015 IEEE Conf. on. pp. 3156–3164. IEEE (2015)

# Introduction to the ICCBR 2018 Workshop on Synergies between CBR and Machine Learning

This workshop follows up the second workshop on synergies between Case-Based Reasoning (CBR) and data mining held at ICCBR 2014 in Frankfurt, Germany and the workshop on synergies between CBR and knowledge discovery held at ICCBR 2016 in Atlanta, USA. This series of workshops focuses on studying the multi-faceted and evolving relationships between case-based reasoning and machine learning. It can be considered as the third edition in this series.

At the core of CBR lies the ability of a system to learn from past cases. However, CBR systems often incorporate machine learning methods, for example, to organize their memory or to learn adaptation rules. In turn, machine learning systems often utilize CBR as a learning methodology, for example, through a common set of problems with the nearest-neighbor method and reinforcement learning. Meanwhile, the machine learning community, which is tightly coupled with knowledge discovery, has historically included CBR among the types of instance-based learning.

This third Workshop on Synergies between CBR and Machine Learning was dedicated to studying in-depth the possible synergies between case-based reasoning and machine learning. It also aimed at identifying potentially fruitful ideas for cooperative problem-solving where both CBR and knowledge discovery researchers can compare and combine methods. In particular, new advances in machine learning may help CBR to advance its field of study and play a vital role in its future.

These proceedings also combine those of the workshop on Evolutionary Computation and CBR. The goal of this workshop was to foster communication between researchers in these two areas and provide a forum to identify opportunities and challenges in both Evolutionary Computation and CBR which can be solved using each other.

Six papers have been selected this year for presentation during ICCBR workshops and inclusion in the Workshops Proceedings. They deal with learning pathology-genes pairs using analogical reasoning [Devignes et al.], designing distributed k-NN similarity measures for Big Data CBR [Barua et al.], knowledge based extraction from traces for model comparison [Leonardi et al.], improving adaptation knowledge discovery by exploiting negative cases [Gillard et al], evolutionary algorithms and CBR for solving a variation of the stable roommates' problem [Lane et al.], and online learning with reoccurring drifts from a CBR perspective [Al Ghossein et al.].

These papers report on the research and experience of twenty-two 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 synergies between CBR and machine, and promise very interesting discussions and interaction among major contributors of CBR research.

July 6, 2018

*Isabelle Bichindaritz*
*Cindy Marling*
*Stefania Montani*
*Hayley Borck*
*Kerstin Bach*

# From knowledge-based trace abstraction to process model comparison

G. Leonardi[a], M. Striani[b], S. Quaglini[c], A. Cavallini[d], S. Montani[a]

[a]DISIT, Computer Science Institute, University of Piemonte Orientale, Alessandria, Italy
[b] Department of Computer Science, University of Torino, Italy
[c]Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Italy
[d]IRCCS "C. Mondino", Pavia, Italy - on behalf of the Stroke Unit Network (SUN) collaborating centers

**Abstract.** Process model comparison can be exploited to assess the quality of organizational procedures, to identify non-conformances with respect to given standards, and to highlight critical situations. Sometimes, however, it is difficult to make sense of large and complex process models, while a more abstract view of the process would be sufficient for the comparison task. In this paper, we show how process traces, abstracted on the basis of domain knowledge, can be provided as an input to process mining, and how abstract models (i.e., models mined from abstracted traces) can then be compared and ranked, by adopting a similarity metric able to take into account penalties collected during the abstraction phase.
The overall framework has been tested in the field of stroke management, where we were able to rank abstract process models more similarly to the ordering provided by a domain expert, with respect to what could be obtained when working on non-abstract ones.

## 1 Introduction

Nowadays, many information systems record data about the executed business process instances in an *event log*, which stores the sequences (*traces* [12] henceforth) of activities that have been completed at the organization. Event logs can be provided as an input to *process mining* [12] algorithms, a family of a-posteriori analysis techniques able to extract non-trivial knowledge from these historic data; within process mining, *process model discovery*, in particular, takes as input the log traces and builds a process model, focusing on its control flow constructs. Classical process mining algorithms, however, provide a purely syntactical analysis, where activities in the traces are processed only referring to their names. Activity names are strings without any semantics, so that identical activities, labeled by synonyms, will be considered as different, or activities that are special cases of other activities will be processed as unrelated. On the other hand, the capability of relating *semantic structures* such as ontologies to activities in the log can enable process mining techniques to work at *different levels*

*of abstraction* (i.e., at the level of instances or concepts) and, therefore, to mask irrelevant details, to promote reuse, and, in general, to make process analysis much more flexible and reliable. Interestingly, *semantic process mining*, defined as the combination of semantic processing capabilities and classical process mining techniques, has been recently proposed in the literature (see, e.g., [11, 3]). However, most contributions are still at a purely theoretical level. Moreover, while more work has been done in the field of semantic *conformance checking* (another branch of process mining) [5], to the best of our knowledge *semantic process model discovery* needs to be further investigated.

Following these considerations, we propose a **trace abstraction mechanism**, able to map activities in the log traces to terms in an ontology, so that they can be converted into higher-level concepts by navigating a hierarchy, up to the desired level. Consecutive activities that abstract as the same concept are also merged into the same abstracted *macro-activity*, properly managing delays and other activities in-between. Abstracted traces are then given as an input to *process mining*: in particular, we resort to classical algorithms embedded in the open source framework ProM [13].

Once the process model of a given organization has been obtained, it is useful to compare it to other organizations' ones, as well as to existing standards. In the medical field, for instance, the analysis of the patient management processes actually implemented in practice supports quality of service evaluation. Evaluating the provided service is a key task in a competitive helthcare market, where hospitals have to focus on ways to deliver high quality care while at the same time reducing costs. Specifically, the actual process model, mined from the organization event log, can be compared to the reference clinical guideline, to verify the existence and the entity of changes, possibly due to local resource constraints, or sometimes to medical errors. Moreover, a ranking of different hospitals' process models can support audit activities and resource assignment. Sometimes, however, it is difficult to make sense of large and complex process models, while a more abstract view of the processes themselves would be sufficient for the comparison task.

In order to address both the need for process model comparison and the need for abstraction, we have **extended a similarity metric** we defined in our previous work [8], in order to allow for comparison and ranking of **models mined from abstracted traces**. Interestingly, process model comparison has been addressed also in [7], which makes use of a normalized version of the graph edit distance, but exploits just syntactical information in the definition of the edit operation costs. The use of semantic information in process model comparison and retrieval is proposed in [1], a system working on workflows represented as semantically labeled graphs. The work in [1] is however more focused on the data flow, which was not considered in our current application. Moreover, none of the previous works affords the abstraction issue.

In this paper, we also report on our experimental work in the field of stroke care, where we were able to rank abstract process models (i.e., models mined from abstracted traces) more similarly to the ordering provided by a domain

expert, with respect to what could be obtained when working on non-abstract ones.

## 2 Knowledge-based trace abstraction

In our framework, knowledge-based trace abstraction has been realized as a multi-step procedure, which extends the one described in [9].

The first step relies on **ontology** mapping. In detail, the ontology (see figure 1 for an excerpt) comprises: (i) a *goal taxonomy*, composed by a set of classes, representing the main goals in stroke management. These main goals can be further specialized into subclasses, according to more specific goals (relation "is-a"; e.g., "Early Relapse Prevention" is a subgoal of "Prevention"); (ii) an *activity taxonomy*, composed by all the activities that can be logged in stroke management traces (in "is-a" relation with the general class "Activity"); (iii) a set of "aimsTo" relations, which formalize that an activity can be executed to implement a (sub)goal. Multiple "aimsTo" relations could connect a given activity to different goals (e.g., CAT (Computer Aided Tomography) can implement "Monitoring" or "Timing" even within the same guideline).

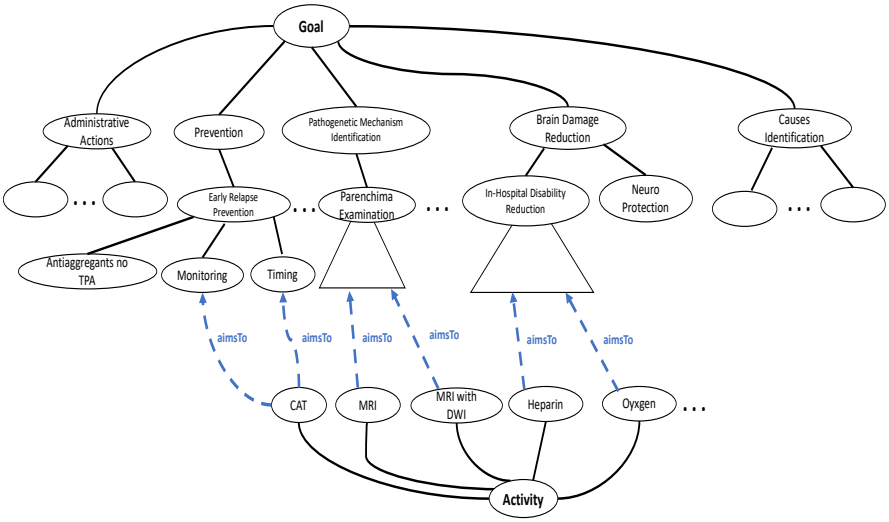



**Fig. 1.** An excerpt from the stroke domain ontology

In the case of multiple "aimsTo" relations, the proper goal to be used to abstract a given activity is then selected by a **rule base**. Contextual information (i.e., the activities that have been already executed on the patient at hand, and/or her/his specific clinical conditions) is used to activate the correct rules.

Once the correct goal of every activity has been identified, trace abstraction can be completed. The level in the ontology to be chosen for abstraction (e.g., a very general goal, such as "Prevention", or a more specific one, such as "Early Relapse Prevention"), has to be specified as an input by the user. In this last step, when a set of consecutive activities on the trace abstract as the same goal, they have to be merged into the same abstracted ***macro-activity***, labeled as the common goal at hand. A macro-activity is an abstracted activity that covers the whole time span of multiple activities, and is labeled as their common goal in the ontology, at the specified abstraction level. As a special case, the macro-activity can abstract a single activity as its goal. This procedure requires a proper treatment of *delays* (i.e., of time intervals between two activities logged in the trace, within which no other activity takes place), and of activities in-between that implement a different goal (*interleaved activities* henceforth). Specifically, the procedure to abstract a trace operates as follows.
For every activity $i$ in the trace:
(1) $i$ is abstracted as the goal it implements (at the ontology level selected by the user); the macro-activity $m\_i$, labeled as the identified goal, is created;
(2) for every element $j$ following $i$ in the trace: (i) if $j$ is a delay, its length is added to a variable $tot - delay$, that stores the total delay duration accumulated so far during the creation of $m\_i$; (ii) if $j$ is an interleaved activity, its length is added to a variable $tot - inter$, that stores the total interleaved activities durations accumulated so far during the creation of $m\_i$; (iii) if $j$ is an activity that, according to domain knowledge, abstracts as the same goal as $i$, $m\_i$ is extended to include $j$, provided that $tot - delay$ and $tot - inter$ do not exceed domain-defined thresholds. $j$ is then removed from the activities in the trace that could start a new macro-activity, since it has already been incorporated into an existing one;
(3) the macro-activity $m\_i$ is appended to the output abstracted trace which, in the end, will contain the list of all the macro-activities that have been created by the procedure.

The variables $tot - delay$ and $tot - inter$, accumulated during abstraction, are also provided as an output attribute of each macro-activity. As discussed in section 4, they will be used as a penalty in abstracted process model similarity calculation.

## 3 Process mining

In our approach, process mining is implemented resorting to the well-known tool ProM, extensively described in [13]. ProM (and specifically its newest version ProM 6) is a platform-independent open source framework that 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. For the work described in this paper, we have exploited ProM's Heuristic Miner [14]. Heuristic Miner is a plug-in for process model discovery, able to mine process models from logs. It receives as input the log, and considers the order of the activities within every single trace.

It can mine the presence of short-distance and long-distance dependencies (i.e., direct or indirect sequence of activities), and information about parallelism, with a certain degree of reliability. The output of the mining process can be visualized as a graph with two types of nodes: activity nodes and gateway nodes - these last ones representing AND/XOR join/fork points. Currently, we have chosen to rely on Heuristics Miner, because it is known to be tolerant to noise, a problem that may affect medical logs (e.g., sometimes the logging may be incomplete). Anyway, testing of other mining algorithms available in ProM 6 is foreseen in our future work.

## 4 Process model comparison

In our framework, we have extended a metric we described in [8], to permit the comparison of models mined from abstracted traces as well. In the following, we will first summarize the initial contribution [8], and then illustrate the extensions.
**Initial contribution**. Since mined process models are represented in the form of graphs, we have defined a distance that extends the notion of graph edit distance [2]. Such a notion calculates the minimal cost of transforming one graph into another by applying edit operations, i.e., insertions/deletions and substitutions of nodes, and insertions/deletions and substitutions of edges. While string edit distance looks for an *alignment* that minimizes the cost of transforming one string into another by means of edit operations, in graph edit distance we have to look for a *mapping*. A mapping is a function that matches (possibly by substituting) nodes to nodes, and edges to edges. Unmatched nodes/edges have to be deleted (or, dually, inserted in the other graph). Among all possible mappings, we will select the one that leads to the minimal cost, having properly quantified the cost of every type of edit operation. In particular, when considering node mapping, activity nodes will only be mapped to activity nodes, while gateway nodes will be mapped to gateway nodes. Formally, 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 $|Ni|$ and $|Ei|$ be the number of nodes and edges of graph $Gi$. Let $M$ be a partial injective mapping (see [4]) 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 $M$.

In our approach, the fraction of inserted/deleted nodes, denoted $fskipn$, the fraction of inserted/deleted edges, denoted $fskipe$, and the average distance of substituted nodes, denoted $fsubn$, are defined as follows:

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

where $|skipn|$ is the number of inserted or deleted nodes;

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

where $|skipe|$ is the number of inserted or deleted edges;

$$fsubn = \frac{2 - (\sum_{n,m \in M_A} dt(n,m) + \sum_{x,y \in M_G} dg(x,y))}{|subn|}$$

where $M_A$ represents the set of mapped activity nodes in the mapping $M$, $M_G$ represents the set of mapped gateway nodes in $M$; $dt(n,m)$ is the distance between two activity nodes $m$ and $n$ in $M_A$, and $dg(x,y)$ is the distance between two gateway nodes $x$ and $y$ in $M_G$; $|subn|$ is the number of substituted nodes.

In detail, $dt(n,m)$ is a proper knowledge-intensive distance definition, to be chosen on the basis of the available knowledge representation formalism in the domain at hand. Currently, we are adopting Palmer's distance [10].

To calculate $dg(x,y)$ we proceed as follows:

1. if $x$ and $y$ are nodes of different types (i.e., a XOR and an AND), their distance is set to 1;
2. if $x$ and $y$ are of the same type (e.g., two ANDs), we have to calculate the difference between their incoming/outgoing activity nodes. The distance between a pair of activity nodes is still calculated exploiting Palmer's distance [10]. Further details can be found in [8].

Finally, the average distance of substituted edges $fsube$ is defined as follows:

$$fsube = \frac{2 - \sum_{(n1,n2),(m1,m2) \in M} (|r(e1) - r(e2)| + |p(e1) - p(e2)| + |m(e1) - m(e2)| + |s(e1) - s(e2)|)}{4 - |sube|}$$

where edge $e1$ (connecting node $n1$ to node $m1$) and edge $e2$ (connecting node $n2$ to node $m2$) are two substituted edges in $M$; $|sube|$ is the number of substituted edges; $r(ei)$ is the reliability of edge $ei$ [14]; $p(ei)$ is the percentage of traces that crossed edge $ei$; $m(ei)$ and $s(ei)$ are statistical values (mean and standard deviation of the elapsed times) calculated over all the occurrences of the $ni \rightarrow mi$ pattern (i.e., $ni$ directly followed by $mi$) in the traces, and normalized in $[0,1]$ dividing by the duration of the longest $ni \rightarrow mi$ pattern in the log. If one of these parameters is unavailable (e.g., reliability is unavailable because Heuristic Miner was not used), its contribution is simply set to 0. Different/additional parameters learned by a miner could be considered as well in $fsube$ in the future.

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

$$ext_{edit}(M) = \frac{wskipn - fskipn + wskipe - fskipe + wsubn - fsubn + wsube - fsube}{wskipn + wskipe + wsubn + wsube}$$

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 any mapping between these graphs. To find the mapping that leads to the minimal distance we resort to a greedy approach, in order to limit

computational costs. It can be shown that the algorithm works in cubic time on the number of nodes of the larger graph [4].

**Extensions**. The novel extensions we present in this paper basically lead to substitute $fsubn$ with a more complete definition, where average abstraction penalties are also summed up when considering the mapping of two activity nodes (which, in this new version, more properly represent macro-activity nodes). We introduce the following definitions:

***Delay Penalty***. Let $n$ and $m$ be two macro-activities, that have been matched in the mapping. Let $average_{delay_n} = \frac{\sum_{i=1}^{k} length(i)}{numtraces}$ be the sum of the lengths of all the $k$ delays that have been incorporated into $n$ in the abstraction phase, divided by the number of abstracted traces that include $n$ (and let $average_{delay_m}$ be analogously defined). Let $maxdelay$ be the maximum, over all the abstracted traces, of the sum of the lengths of the delays incorporated in a macro-activity. The Delay Penalty $delay_p(n, m)$ between $n$ and $m$ is defined as:

$$delay_p(n, m) = \frac{|average_{delay_n} - average_{delay_m}|}{maxdelay}$$

As for interleaved activities penalty, we operate analogously to delay penalty, by considering the average lengths of the interleaved activities that have been incorporated within the involved macro-activities in the abstraction phase.

***Interleaving Length Penalty***. Let $n$ and $m$ be two macro-activities, that have been matched in the mapping. Let $average_{inter_n} = \frac{\sum_{i=1}^{k} length(i)}{numtraces}$ be the sum of the lengths of all the $k$ interleaved activities that have been incorporated into $n$ in the abstraction phase, divided by the number of abstracted traces that include $n$ (and let $average_{inter_m}$ be analogously defined). Let $maxinter$ be the maximum, over all the abstracted traces, of the sum of the lengths of the interleaved activities incorporated in a macro-activity. The Interleaving Length Penalty $interL_p(n, m)$ between $n$ and $m$ is defined as:

$$interL_p(n, m) = \frac{|average_{inter_n} - average_{inter_m}|}{maxinter}$$

Then, formally, $fsubn$ becomes

$$fsubn = \frac{2 - (\sum_{n,m \in M_A} \varphi(n.m) + \sum_{x,y \in M_G} dg(x, y))}{|subn|}$$

where

$$\varphi(n, m) = \frac{dt(n, m) + delay_p(n, m) + interL_p(n, m)}{3}$$

Notably, the abstraction penalty contributions might also be weighted differently from $dt(n, m)$, if the domain expert suggests to give more/less importance to Palmer's distance between macro-activities with respect to abstraction penalties themselves.

## 5 Experimental results

In this section, we describe the experimental results we have conducted, in the application domain of stroke care. The available event log was composed of more than 15000 traces, collected at the Stroke Unit Network (SUN) collaborating centers of the Lombardia region, Italy. The number of traces in the different Stroke Units (SUs) of the network varied from 266 to 1149. Traces were composed of 13 activities on average. For our validation study, we asked a SUN stroke management expert to provide a ranking of some SUs (see table 1, column 1), on the basis of the quality of service they provide, with respect to the top level SU. Such a ranking was based on her personal knowledge of the SUs human and instrumental resource availability (not on the process models); therefore, it was qualitative, and "coarse-grained", in the sense that more than one SU could obtain the same qualitative evaluation. The top level SU will be referred as H0 in the experiments. The expert identified 6 SUs (H1-H6) with a high similarity level with respect to H0; 5 SUs (H7-H11) with a medium similarity level with respect to H0; and 4 SUs (H12-H15) with a low similarity level with respect to H0. The ordering of the SUs within one specific similarity level is not relevant, since, as observed, the expert's ranking is coarse-grained. It is instead important to distinguish between different similarity levels. We then mined the process models of the 16 SUs by using Heuristic Miner, both working on non-abstracted traces, and working on abstracted traces. We ordered the two available process model sets with respect to H0, resorting to the extended similarity metric presented in section 4, globally obtaining two rankings. In the experiments, we set all the weights to 1, except for $wskipn$ and $wskipe$, which were set to 2. Indeed, the domain expert suggested to strongly penalize missing macro-activities or missing connections, with respect to substitutions. Moreover, in $fsubn$ abstraction penalty contributions were weighted 20% of $dt(n, m)$ (abstraction penalties were simply set to zero when comparing models mined from non-abstracted traces). As regards abstraction thresholds, on the other hand, they were common to all traces in the log, and set on the basis of medical knowledge.

Results are shown in table 1. Column 1 shows the expert's qualitative evaluation (similarity with respect to the reference SU H0), and lists the corresponding SUs names; columns 2 and 3 show the ranking obtained by relying on our distance definition, mining the process models on non-abstracted and abstracted traces, respectively. In particular, abstraction was conducted at level 2 in the ontology (where level 0 is "Goal"). When working on models mined from non-abstracted traces, we correctly rate two process models in the high similarity group (33%), zero process models in the medium similarity group (0%), and one process model in the low similarity group (25%, column 2). When working on models mined from abstracted traces, on the other hand, we correctly rate three process models in the high similarity group (50%), three process models in the medium similarity group (60%), and one process model in the low similarity group (25%, column 3). In summary, when working on abstracted traces, our extended distance definition, able to properly consider abstraction penalties as well, leads to rankings that are closer to the qualitative ranking provided by

**Table 1.** Ordering of 15 SUs, with respect to a given query model. Correct positions in the rankings with respect to the expert's qualitative similarity levels are highlighted in bold.

| Qual. Similarity - Medical expert | Non-abstract | Abstract |
|---|---|---|
| High - H1 | **H2** | **H5** |
| High - H2 | H9 | **H2** |
| High - H3 | H7 | H12 |
| High - H4 | H8 | **H3** |
| High - H5 | H13 | H8 |
| High - H6 | **H6** | H13 |
| Medium - H7 | H15 | **H7** |
| Medium - H8 | H5 | H6 |
| Medium - H9 | H1 | **H11** |
| Medium - H10 | H12 | **H10** |
| Medium - H11 | H4 | H15 |
| Low - H12 | H3 | **H14** |
| Low - H13 | H11 | H1 |
| Low - H14 | H10 | H4 |
| Low - H15 | **H14** | H9 |

the human expert, and therefore allows to better classify the quality of service provided to patients by the different SUs.

## 6 Conclusions

In this paper, we have described a framework able to semantically abstract traces, and provide them as an input to process mining. Models mined from abstracted traces can then be compared and ranked, by adopting an extended process model similarity metric, which can take into account abstraction phase penalties as well. In the experiments, we mined the process models of some SUs by using Heuristic Miner, both working on non-abstracted traces, and working on abstracted ones. We then ordered the two available process model sets with respect to the model of the best equipped SU in the SUN network, resorting to the extended metric. We verified that, when working on abstracted traces, distance calculation leads to a ranking that is closer to the qualitative one provided by a domain expert, thus better classifying the quality of service provided to patients by the different SUs. In the future, we plan to test the approach in different application domains as well, after having acquired the corresponding domain knowledge. Finally, an abstraction mechanism directly operating on process models (i.e., on the graph, instead of the log), may be considered, possibly along the lines described in [6], and abstraction results will be compared to the ones currently enabled by our framework.

## References


1. R. Bergmann and Y. Gil. Similarity assessment and efficient retrieval of semantic workflows. *Information Systems*, 40:115–127, 2014.
2. H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689694, 1997.

3. A. K. Alves de Medeiros and W. M. P. van der Aalst. Process mining towards semantics. In T. S. Dillon, E. Chang, R. Meersman, and K. P. Sycara, editors, *Advances in Web Semantics I - Ontologies, Web Services and Applied Semantic Web*, volume 4891 of *Lecture Notes in Computer Science*, pages 35–80. Springer, 2009.
4. R. Dijkman, M. Dumas, and R. Garca-Banuelos. Graph matching algorithms for business process model similarity search. In U. Dayal, J. Eder, J. Koehler, and H. Reijers, editors, *Proc. International Conference on Business Process Management*, volume 5701 of *Lecture Notes in Computer Science*, pages 48–63. Springer, Berlin, 2009.
5. M. A. Grando, M. H. Schonenberg, and W. M. P. van der Aalst. Semantic process mining for the verification of medical recommendations. In V. Traver, A. L. N. Fred, J. Filipe, and H. Gamboa, editors, *HEALTHINF 2011 - Proceedings of the International Conference on Health Informatics, Rome, Italy, 26-29 January, 2011*, pages 5–16. SciTePress, 2011.
6. C. Günther and W. van der Aalst. Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors, *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer, 2007.
7. 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.
8. S. Montani, G. Leonardi, S. Quaglini, A. Cavallini, and G. Micieli. A knowledge-intensive approach to process similarity calculation. *Expert Syst. Appl.*, 42(9):4207–4215, 2015.
9. S. Montani, M. Striani, S. Quaglini, A. Cavallini, and G. Leonardi. Semantic trace comparison at multiple levels of abstraction. In D. W. Aha and J. Lieber, editors, *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, volume 10339 of *Lecture Notes in Computer Science*, pages 212–226. Springer, 2017.
10. M. Palmer and Z. Wu. Verb Semantics for English-Chinese Translation. *Machine Translation*, 10:59–92, 1995.
11. C. Pedrinaci, J. Domingue, C. Brelage, T. van Lessen, D. Karastoyanova, and F. Leymann. Semantic business process management: Scaling up the management of business processes. In *Proceedings of the 2th IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, California, USA*, pages 546–553. IEEE Computer Society, 2008.
12. W. van der Aalst. *Process Mining. Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
13. 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, Berlin, 2005.
14. A. Weijters, W. van der Aalst, and A. Alves de Medeiros. *Process Mining with the Heuristic Miner Algorithm, WP 166*. Eindhoven University of Technology, Eindhoven, 2006.

# Improving Adaptation Knowledge Discovery by Exploiting Negative Cases: First Experiment in a Boolean Setting

Tristan Gillard, Jean Lieber, and Emmanuel Nauer

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

**Abstract.** Case-based reasoning usually exploits *positive* source cases consisting of a source problem and its solution that is known to be a correct for the problem. The work presented in this paper addresses in addition of positive case exploitation, the exploitation of *negative* cases, i.e. problem-solution pairs where the solution is an incorrect answer to the problem, which can be acquired when the case-based reasoning (CBR) process fails. An originality of this work is that positive and negative cases are used both for adaptation knowledge (AK) discovery using closed itemsets built on variations between cases. Experiments show that exploiting negative cases in addition to positive ones improves the quality of the AK being extracted and, so, improves the results of the CBR system.

**Keywords:** adaptation knowledge discovery, closed itemset extraction, case mining, negative cases, case-based reasoning

## 1 Introduction

Case-based reasoning (CBR) [14] aims at solving a new problem—the *target problem*—thanks to a set of cases (the *case base*), where a case is a pair consisting of a problem and a solution to this problem. A *source case* is a case from the case base, consisting of a *source problem* and one of its solutions. The classical approach to CBR consists in selecting source cases *similar* to the target problem and adapting them to solve it. The adaptation step may use different approaches, one of them is the use of adaptation knowledge (AK). Acquiring AK is, in this case, a crucial issue.

Most of the times, AK discovery for CBR focuses on the exploitation of *positive* source cases [5, 3, 2, 7]. A *positive* source case is a source case such that the solution is a correct solution to the problem (according, e.g. to a human expert). However, a case base may also contains *negative* source cases. A *negative* source case is a source case such that the solution is an incorrect solution to the problem. Such negative cases can for example be acquired at the retain step of the classical 4R (retrieve, reuse, revise, retain) CBR process [1], when the CBR process fails and returns an incorrect solution.

This paper presents an approach exploiting at the best all the existing cases of the case base, the positive ones but also the negative ones, in order to improve the $AK$ discovery, under the hypothesis that the better the $AK$ quality is, the better the results of the CBR system that will use it will be. This work is based on the approach proposed in [2] for extracting $AK$. The approach is based on closed itemset (CI) extraction on

| | Apple | PieCrust | PuffPastry | Sugar | Cream |
|---|---|---|---|---|---|
| $r^1$ | × | × | | × | |
| $r^2$ | × | × | | × | × |
| $r^3$ | | | × | × | × |
| $r^4$ | × | | × | | |

Table 1: An example of formal context representing ingredients used in recipes.

variations between cases. The originality of our work lies in the use of CI extraction to take into account negative cases.

The paper is organized as follows. Section 2 introduces the motivations and the preliminaries for this work, introducing CI extraction and CBR with related work. Section 3 describes our approach for exploiting positive and negatives cases in an AK discovery process. Section 4 presents the evaluation of our approach through experiments and discusses the results. Section 5 points out lines for future research.

## 2 Motivation and preliminaries

Cooking CBR systems, as the ones which have participated to the *Computer Cooking Contest* (e.g. TAAABLE [4]) are typical of systems that are concerned by the objective of this work. Indeed, it has been showed that adapting cooking recipes benefits from the use of AK [7]. Moreover, feedback may be collected about the system results. For example, the TAAABLE system provides a result interface allowing the users to evaluate whether a recipe adaptation is correct or not [15]. This approach to manage correct and incorrect adaptations is a way to collect positive and negative cases, which can both be stored in the case base (with an appropriate label).

### 2.1 Itemset extraction

Itemset extraction is a collection of data-mining methods for extracting regularities into data, by aggregating object items appearing together. Like FCA [10], itemset extraction algorithms start from a *formal context* $K$, defined by $K = (G, M, I)$, where $G$ is a set of objects, $M$ is a set of items, and $I$ is the relation on $G \times M$ stating that an object is described by an item [10]. Table 1 shows an example of context, in which $4$ recipes are described by the ingredients they require: $G$ is a set of $4$ objects (recipes $r^1$, $r^2$, $r^3$, and $r^4$), $M$ is a set of $5$ items (ingredients `Apple`, `PieCrust`, `PuffPastry`, `Sugar`, and `Cream`).

An *itemset* $I$ is a set of items, and the *support* of $I$, $\texttt{supp}(I)$, is the number of objects of the formal context having every item of $I$. $I$ is frequent, with respect to a threshold $\rho$, whenever $\texttt{supp}(I) \geq \rho$. $I$ is closed if it has no proper superset $J$ ($I \subsetneq J$) with the same support. For example, $\{\texttt{Apple}, \texttt{PieCrust}\}$ is an itemset and $\texttt{supp}(\{\texttt{Apple}, \texttt{PieCrust}\}) = 2$ because exactly $2$ recipes require both

| | Apple$^{-}$ | Apple$^{=}$ | Apple$^{+}$ | PieCrust$^{-}$ | PieCrust$^{=}$ | PieCrust$^{+}$ | PuffPastry$^{-}$ | PuffPastry$^{=}$ | PuffPastry$^{+}$ | Sugar$^{-}$ | Sugar$^{=}$ | Sugar$^{+}$ | Cream$^{-}$ | Cream$^{=}$ | Cream$^{+}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $V^{12}$ | | × | | | × | | | | | | × | | | | × |
| $V^{13}$ | × | | | × | | | | × | | | × | | | | × |
| $V^{14}$ | | × | | × | | | | × | | × | | | | × | |

Table 2: Formal context for ingredient variations in pairs of recipes $(r^1, r^2)$, $(r^1, r^3)$ and $(r^1, r^4)$.

Apple and PieCrust. However, {Apple, PieCrust} is not a CI, because {Apple, PieCrust, Sugar} has the same support. For $\rho = 2$, the frequent CIs (FCIs) of this context are {Apple, PieCrust, Sugar} and {Sugar, Cream}.

For our experiments, we use CORON [16], a software platform which implements efficient algorithms for symbolic data mining and especially FCI computation.

### 2.2 Exploiting case variations for AK discovery

Exploiting case variations is not a new idea. [11] introduces this approach of AK learning based on pairwise comparisons of cases. This approach has been applied in various domains such as chemistry [3], medicine [5] or cooking [2, 7].

For an ordered pair of cases $(\mathtt{c}^1, \mathtt{c}^2)$, the approach consists in representing what features have to be removed ($-$), kept ($=$) and added ($+$) to transform $\mathtt{c}^1$ into $\mathtt{c}^2$. For example, for a pair $(r^i, r^j)$ of recipes described each by the ingredients they require, the representation of the variation is denoted by $V^{ij}$, where $V^{ij}$ represents the set of variations of ingredients from $r^i$ to $r^j$. Each ingredient $ing$ is marked by $-$, $=$, or $+$:

- $ing^- \in V^{ij}$ if $ing$ is an ingredient of $r^i$ but not of $r^j$.
- $ing^+ \in V^{ij}$ if $ing$ is an ingredient of $r^j$ but not of $r^i$.
- $ing^= \in V^{ij}$ if $ing$ is an ingredient of both $r^i$ and $r^j$.

Table 2 shows the ingredient variations for 3 ordered pairs of recipes described in Table 1. An AK discovery process based on FCIs can be run on such a binary table which constitutes a formal context. Each extracted FCI produces an adaptation rule (AR) ar with a support supp(ar), i.e. the number of $V^{ij}$ containing ar. For example, for $\rho = 2$, {PieCrust$-$, PuffPastry$+$} is an FCI which produces an adaptation rule consisting in replacing the pie crust by a puff pastry.

### 2.3 Assumptions and notations about CBR

Let $\mathcal{P}$ and $\mathcal{S}$ be two sets. A *problem* (resp., a *solution*) is an element of $\mathcal{P}$ (resp., of $\mathcal{S}$). The existence of a binary relation with the semantics "has for solution" is assumed. In this paper, this relation is assumed to be functional, guided by the idea to fully automatize the evaluation process. Let f be the function from $\mathcal{P}$ to $\mathcal{S}$ such that $\mathtt{y} = \mathtt{f}(\mathtt{x})$ if y is the solution of x. A *case* is a pair $(\mathtt{x}, \mathtt{y}) \in \mathcal{P} \times \mathcal{S}$ such that $\mathtt{y} = \mathtt{f}(\mathtt{x})$.

A CBR system on $(\mathcal{P}, \mathcal{S}, \mathtt{f})$ is built with a knowledge base $\mathtt{KB} = (\mathtt{CB}, \mathtt{DK}, \mathtt{RK}, \mathtt{AK})$ where CB is the case base (a finite set of cases), DK is the domain knowledge, RK is the

retrieval knowledge (in this work, $\mathtt{RK} = \mathtt{dist}$, a distance function on $\mathcal{P}$), and $\mathtt{AK}$ is the adaptation knowledge that will take the form of adaptation rules.

A CBR system on $(\mathcal{P}, \mathcal{S}, \mathtt{f})$ aims at associating to a target problem $\mathtt{x}^t$ an $\mathtt{y}^t \in \mathcal{S}$, denoted by $\mathtt{y}^t = f_{\text{CBR}}(\mathtt{x}^t)$. The function $f_{\text{CBR}}$ is intended to be an approximation of $\mathtt{f}$. It is built thanks to the following functions:

- the retrieval function, with the profile $\mathtt{retrieval} : \mathtt{x}^t \mapsto (\mathtt{x}^s, \mathtt{y}^s) \in \mathtt{CB}$;
- the adaptation function, with the profile $\mathtt{adaptation} : ((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^t) \mapsto \mathtt{y}^t \in \mathcal{S}$; it is usually based on $\mathtt{DK}$ and $\mathtt{AK}$. $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^t)$ is an *adaptation problem*.

Thus $f_{\text{CBR}}(\mathtt{x}^t) = \mathtt{adaptation}(\mathtt{retrieval}(\mathtt{x}^t), \mathtt{x}^t)$.

With no domain and adaptation knowledge ($\mathtt{DK} = \mathtt{AK} = \emptyset$), the adaptation consists usually of a mere copy of the solution. This process is called *null adaptation*:

$$\mathtt{null\_adaptation} : ((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^t) \mapsto \mathtt{y}^s$$

**Adaptation principle using adaptation rules.** Generally speaking, an adaptation rule $\mathtt{ar}$ is a function mapping an adaptation problem $((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^t) \in \mathtt{CB} \times \mathcal{P}$ to $\mathtt{y}^t \in \mathcal{S} \cup \{\mathtt{failure}\}$. Two cases of failure ($\mathtt{y}^t = \mathtt{failure}$) are considered: (i) no $(\mathtt{x}^s, \mathtt{y}^s) \in \mathtt{CB}\}$ such as $\mathtt{dist}(\mathtt{x}^s, \mathtt{x}^t) \leq \rho$, with $\rho$ a given threshold, is returned by the retrieval function, and (ii) no AR $\mathtt{ar}$ is applicable on this adaptation problem. Else, $\mathtt{y}^t$ is a proposed solution to $\mathtt{x}^t$, by adaptation of $(\mathtt{x}^s, \mathtt{y}^s)$ according to $\mathtt{ar}$. A support $\mathtt{supp}(\mathtt{ar}) \geq 0$ is associated to a rule $\mathtt{ar}$; the higher is $\mathtt{supp}(\mathtt{ar})$, the more $\mathtt{ar}$ is preferred.

The adaptation consists in selecting the subset $\mathtt{AAR}$ of $AK$ of applicable adaptation rules with maximum support: $\mathtt{ar} \in \mathtt{AAR}$ iff $\mathtt{ar}((\mathtt{x}^s, \mathtt{y}^s), \mathtt{x}^t) \neq \mathtt{failure}$ and there exists no $\mathtt{ar}' \in \mathtt{AAR}$ such that $\mathtt{supp}(\mathtt{ar}') > \mathtt{supp}(\mathtt{ar})$.

### 2.4 Boolean setting

We have presented at the beginning of section 2 the interest of AK discovery for a real life application. However, evaluating how an AK approach improves the results of the CBR system in such a concrete application is difficult because experiments require humans (users or experts) who have to evaluate the quality or the validity of the system answers (see for example [8] where the evaluation process implies users who have to judge if the TAAABLE cooking system returns correct or incorrect answers in the context of collaborative knowledge acquisition). Such a human based evaluation has many inconveniences: (a) specific interfaces have to be built to guarantee a blind evaluation, (b) it is time consuming, especially compared to an automated evaluation, (c) it requires available users and/or experts, and (d) the coverage of the evaluation is limited because of points (b) and (c).

For these reasons, we use in this work a Boolean setting in which all experiments can be automatized using Boolean functions as $\mathtt{f}$. Let $\mathbb{B} = \{0, 1\}$ be the set of Boolean values. The Boolean operators are denoted by the connector symbols of propositional logic: for $a, b \in \mathbb{B}$, $\neg a = 1 - a$, $a \wedge b = \min(a, b)$, $a \vee b = \max(a, b)$, $a \oplus b = |b - a|$ ($\oplus$ is the exclusive or) and $a \Leftrightarrow b = \neg(a \oplus b)$.

Let $p \geq 0$. The Hamming distance $H$ on $\mathbb{B}^p$ is defined by $H(a, b) = \sum_{i=1}^{p} |b_i - a_i|$. For example, with $p = 5$, $H((0, 1, 0, 0, 1), (1, 1, 0, 1, 1)) = 2$.

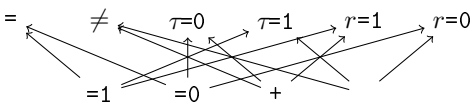



Fig. 1: The generalization/specialization hierarchy of variations.

Let $m, n \in \mathbb{N}^*$, $\mathcal{P} = \mathbb{B}^m$, $\mathcal{S} = \mathbb{B}^n$ and $\mathtt{f} : \mathcal{P} \to \mathcal{S}$, be a Boolean function to be approximated. A CBR system is considered on $(\mathcal{P}, \mathcal{S}, \mathtt{f})$ with $\mathtt{DK} = \emptyset$, $\mathtt{RK} = \mathtt{dist}$, the Hamming distance on $\mathcal{P}$, and $AK$ a set of adaptation rules.

**Adaptation rule language.** The AR language used in this work is based on the notion of variations between Booleans, as described hereafter. Given $\tau, r \in \mathbb{B}$ ($\tau$ stands for *left*, $r$ for *right*), the variation from $\tau$ to $r$ is represented by *variation symbols*. Each of the 4 ordered pairs $(\tau, r)$ is represented by a primary variation symbol $v$:

- $(\tau, r) = (1, 0)$ is represented by $v =$ ;
- $(\tau, r) = (0, 1)$ is represented by $v =$ +;
- $(\tau, r) = (0, 0)$ is represented by $v =$ =0;
- $(\tau, r) = (1, 1)$ is represented by $v =$ =1.

Each primary variation symbols is linked to 3 inferred variation symbols. Fig. 1 shows the generalization links between the primary variation symbols and the inferred ones. For example, $v =$ is linked to $\neq$, stating that $\tau \neq r$, to $\tau$=1 and to $r$=0, stating that $\tau$ (resp. $r$) is equal to 1 (resp. 0).

Given two cases $\mathtt{c}^1 = (\mathtt{x}^1, \mathtt{y}^1)$ and $\mathtt{c}^2 = (\mathtt{x}^2, \mathtt{y}^2)$, the set of variations $V^{12}$ from $\mathtt{c}^1$ to $\mathtt{c}^2$ is encoded by the set of the expressions $\mathtt{x}_i^v$ and $\mathtt{y}_j^w$ such that $v$ (resp., $w$) is a variation symbol from $\mathtt{x}_i^1$ to $\mathtt{x}_i^2$ (resp., from $\mathtt{y}_j^1$ to $\mathtt{y}_j^2$). For example, if $(\mathtt{x}^1, \mathtt{y}^1) = ((0, 1), 0)$ and $(\mathtt{x}^2, \mathtt{y}^2) = ((0, 0), 1)$ then $V^{12} = \{\mathtt{x}_1^{=0}, \mathtt{x}_1^{=}, \mathtt{x}_1^{\varphi=0}, \mathtt{x}_1^{r=0}, \mathtt{x}_2^{-}, \mathtt{x}_2^{\neq}, \mathtt{x}_2^{\varphi=1}, \mathtt{x}_2^{r=0}, \mathtt{y}_1^{+}, \mathtt{y}_1^{\neq}, \mathtt{y}_1^{\varphi=0}, \mathtt{y}_1^{r=1}\}$.

An AR $\mathtt{ar}$ is a set of expressions $\mathtt{x}_i^v$ and $\mathtt{y}_j^w$. At least one $\mathtt{x}_i^v$ and one $\mathtt{y}_j^w$ are required in $\mathtt{ar}$ to be used for adaptation: $\mathtt{ar}$ is applicable on an adaptation problem $((\mathtt{x}^\mathtt{s}, \mathtt{y}^\mathtt{s}), \mathtt{x}^\mathtt{t})$ if there exists $\mathtt{y}^\mathtt{t} \in \mathbb{B}^n$ such that $V^{st} \supseteq \mathtt{ar}$ (where $V^{st}$ represents the variation from $(\mathtt{x}^\mathtt{s}, \mathtt{y}^\mathtt{s})$ to $(\mathtt{x}^\mathtt{t}, \mathtt{y}^\mathtt{t})$). If it is applicable, then its application consists in choosing such a $\mathtt{y}^\mathtt{t}$. If several $\mathtt{y}^\mathtt{t}$'s exist, the chosen one is the closest to $\mathtt{y}^\mathtt{s}$ according to the Hamming distance on $\mathcal{S} = \mathbb{B}^n$, meaning that if $\mathtt{ar}$ gives no constraint on some $\mathtt{y}_j^\mathtt{t}$ then $\mathtt{y}_j^\mathtt{t} = \mathtt{y}_j^\mathtt{s}$. For example:

if $\mathtt{ar} = \{\mathtt{x}_1^{-}, \mathtt{x}_2^{=}, \mathtt{y}_1^{+}\}$, $(\mathtt{x}^\mathtt{s}, \mathtt{y}^\mathtt{s}) = ((1, 0, 0), (0, 0))$ and $\mathtt{x}^\mathtt{t} = (0, 0, 1)$
then $\mathtt{ar}$ is applicable on $((\mathtt{x}^\mathtt{s}, \mathtt{y}^\mathtt{s}), \mathtt{x}^\mathtt{t})$ and $\mathtt{ar}((\mathtt{x}^\mathtt{s}, \mathtt{y}^\mathtt{s}), \mathtt{x}^\mathtt{t}) = \mathtt{y}^\mathtt{t} = (1, 0)$

## 3 $AK$ discovery using positive and negative cases

An originality of this work is to build an $AK$ discovery for CBR exploiting both *positive* and *negative* source cases. For a case $c = (\mathtt{x}, \mathtt{y}) \in \mathtt{CB}$, the case $c$ is said *positive*

if $\mathtt{y} = \mathtt{f}(\mathtt{x})$ and *negative* if $\mathtt{y} \neq \mathtt{f}(\mathtt{x})$. We denote $\mathtt{CB}^+$ (resp. $\mathtt{CB}^-$), the set of positive (resp. negative) cases of CB, with $\mathtt{CB} = \mathtt{CB}^+ \cup \mathtt{CB}^-$.

Starting from the two sets of cases $\mathtt{CB}^+$ and $\mathtt{CB}^-$, ordered pairs of cases $(\mathtt{c}^1, \mathtt{c}^2)$ are formed, with $\mathtt{c}^1 = (\mathtt{x}^1, \mathtt{y}^1) \in \mathtt{CB}^+$ and $\mathtt{c}^2 = (\mathtt{x}^2, \mathtt{y}^2) \in \mathtt{CB}$ such that $\mathtt{x}^1 \neq \mathtt{x}^2$. Each such pair is encoded by a set $V^{12}$ of the variations from $\mathtt{x}_i^1$ to $\mathtt{x}_i^2$ and from $\mathtt{y}_j^1$ to $\mathtt{y}_j^2$, as presented above. When $\mathtt{c}^2 \in \mathtt{CB}^+$, the variations from $\mathtt{c}^1$ to $\mathtt{c}^2$ can be considered as a positive example of AR (i.e. the application of the AR will produce a correct answer). When $\mathtt{c}^2 \in \mathtt{CB}^-$, the variations between $\mathtt{c}^1$ and $\mathtt{c}^2$ can be considered as a negative example of AR (i.e. the application of the AR produces an incorrect answer).

### 3.1 Exploiting positive examples

The AR learning process based on FCI extraction takes in input a set of $V^{ij}$, which will be used to build the formal context. In this work and especially for the evaluation, we have used two approaches to build the formal context. The first one consists in using each $V^{ij}$ as an object with only the primary variations as properties. This approach will be denoted by $AK^+$ in the following. The second approach consists in extending $AK^+$ by using also the more general variations that can be inferred from the primary ones as object properties. This is a classical AK approach for extracting more general adaptation rules (i.e. rules with a higher support) (see e.g. [6]). This second AK approach will be denoted by $AK^{+I}$ in the following.

### 3.2 Exploiting negative examples

The objective of the $AK^{+I}$ approach is the extraction of more general ARs. However, when an AR is too general, its application is likely to give an incorrect answer. This is for example the case when the general rule consisting in replacing a pie crust by a puff pastry is applied to a salted tart recipe. This observation motivates the exploitation of negative examples for filtering too general ARs.

Exploiting negative examples in a learning process requires a specific approach. Some machine learning approaches such as, for example the version space model introduced by Mitchell [13], considers a training set composed of positive and negative examples in order to learn a binary classification model. The idea of the version space model is to build a space of hypotheses (represented by a disjunction of logical sentences) such that a hypothesis covers all positive examples and no negative ones. More recently, Ganter and Kuznetsov established the link between the version space model and FCA [9, 12]. Our exploitation of negative examples in order to extract ARs is based on the same idea introduced in these related works: generating AR covering positive examples without covering negative ones. A first approach to address this issue consists in generating AR only on a formal context built on positive examples and then removing rules which cover at least one negative example. Let $V^{e^-}$ be the set of variations of the negative example $e^-$, ar is removed if $e^- \supseteq \mathtt{ar}$. However, the complexity of this approach (in $O(|\mathrm{AR}| \times |\mathtt{CB}^-|)$) leads us to consider another more efficient way to compute ARs consistent with negative examples. For this, we take advantage of the efficiency of the FCI extraction algorithms by adding to the formal context the negative

examples and by removing, before generating the ARs, the CIs which extents contain at least one negative example. In the following, this appproach exploiting both positive and negative examples will be denoted by $AK^{+-}$ when no inferred variations are used and $AK^{+-I}$ when they are.

## 4 Evaluation

The objective of the evaluation is to study, on various types of Boolean functions, how exploiting negative cases in addition to positive ones improves the results of the CBR system. Experimental results are presented and discussed.

### 4.1 Experiment setting

In the experiment, $\mathcal{P} = \mathbb{B}^8$ and $\mathcal{S} = \mathbb{B}$. Functions $\mathtt{f}$ are randomly generated using the following generators that are based on the three main normal forms, with the purpose of having various types of functions:

`CNF` $\mathtt{f}$ is generated in a conjunctive normal form, i.e., $\mathtt{f}(\mathtt{x})$ is a conjunction of $n_{\text{conj}}$ disjunctions of literals, for example $\mathtt{f}(\mathtt{x}) = (\mathtt{x}_1 \vee \neg\mathtt{x}_7) \wedge (\neg\mathtt{x}_3 \vee \mathtt{x}_7 \vee \mathtt{x}_8) \wedge \mathtt{x}_4$. The value of $n_{\text{conj}}$ is randomly chosen uniformly in $\{3, 4, 5\}$. Each disjunction is generated on the basis of two parameters, $p^+ > 0$ and $p^- > 0$, with $p^+ + p^- < 1$: each variable $\mathtt{x}_i$ occurs in the disjunct in a positive (resp. negative) literal with a probability $p^+$ (resp., $p^-$). In the experiment, the values $p^+ = p^- = 0.1$ were chosen.

`DNF` $\mathtt{f}$ is generated in a disjunctive normal form, i.e., it has the same form as for `CNF` except that the connectors $\wedge$ and $\vee$ are exchanged. The parameters $n_{\text{disj}}$, $p^+$ and $p^-$ are set in the same way.

`Pol` is the same as `DNF`, except that the disjunctions ($\vee$) are replaced with exclusive or's ($\oplus$), thus giving a polynomial normal form. The only different parameter is $p^- = 0$ (only positive literals occur in the polynomial normal form).

The case base $\mathtt{CB} = \mathtt{CB}^+ \cup \mathtt{CB}^-$ is generated randomly, with the values for their sizes:$|\mathtt{CB}^+| \in \{16, 32, 48\}$, i.e. $|\mathtt{CB}^+|$ is between $\frac{1}{16}$ and $\frac{3}{16}$ of $|\mathcal{P}| = 2^8 = 256$, and $|\mathtt{CB}^-| \in \{0, \frac{|\mathtt{CB}^+|}{2}, |\mathtt{CB}^+|\}$.

Each positive source case $(\mathtt{x}, \mathtt{y})$ is generated as follows: $\mathtt{x}$ is randomly chosen in $\mathcal{P}$ with a uniform distribution and $\mathtt{y} = \mathtt{f}(\mathtt{x})$. Each negative source case $(\mathtt{x}, \mathtt{y})$ is generated as follows: $\mathtt{x}$ is randomly chosen as for positive source cases and $\mathtt{y} = \neg\mathtt{f}(\mathtt{x})$.

The FCIs are computed with a support threshold of $8\%$ of the number of the positive examples forming the formal context for the approaches using inferences on variations ($AK^{+I}$ and $AK^{+-I}$), and $5\%$ of the number of the positive examples for $AK^+$ and $AK^{+-}$. For example, for $|\mathtt{CB}^+| = 32$ and $|\mathtt{CB}^-| > 0$, the threshold is $32 \times 31 \times 5\% = 49.6$. These thresholds, set experimentally, are a good compromise to compute a large set of rules in each approach in a limited execution time.

Five adaptation approaches are tested: $AK^+$, $AK^{+-}$, $AK^{+I}$, $AK^{+-I}$ and $NN$, the classical nearest neighbor approach with the `null_adaptation` for adaptation

| | | prec (%) | | | | | | | | | car (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lvert CB^+\rvert =$ | 16 | | | 32 | | | 48 | | | 16 | | | 32 | | | 48 | | |
| | $\lvert CB^-\rvert =$ | 0 | 8 | 16 | 0 | 16 | 32 | 0 | 24 | 48 | 0 | 8 | 16 | 0 | 16 | 32 | 0 | 24 | 48 |
| CNF | $NN$ | 85 | 85 | 85 | 87 | 87 | 87 | 88 | 88 | 88 | 79 | 79 | 79 | 86 | 86 | 87 | 88 | 88 | 88 |
| | $AK^+$ | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 | 84 |
| | $AK^{+-}$ | 84 | 87 | 88 | 84 | 89 | 91 | 84 | 90 | 93 | 84 | 87 | 87 | 84 | 88 | 88 | 84 | 88 | 88 |
| | $AK^{+I}$ | 79 | 79 | 79 | 77 | 77 | 77 | 80 | 80 | 80 | 79 | 79 | 79 | 77 | 77 | 77 | 80 | 80 | 80 |
| | $AK^{+-I}$ | 79 | 85 | 86 | 77 | 87 | 90 | 80 | 88 | 91 | 79 | 85 | 86 | 77 | 87 | 89 | 80 | 88 | 91 |
| DNF | $NN$ | 83 | 83 | 82 | 84 | 84 | 84 | 86 | 86 | 85 | 76 | 76 | 76 | 83 | 83 | 84 | 86 | 85 | 85 |
| | $AK^+$ | 82 | 81 | 81 | 82 | 82 | 82 | 81 | 81 | 81 | 82 | 81 | 81 | 82 | 82 | 82 | 81 | 81 | 81 |
| | $AK^{+-}$ | 82 | 84 | 86 | 82 | 87 | 91 | 81 | 90 | 94 | 82 | 84 | 86 | 82 | 86 | 86 | 81 | 86 | 85 |
| | $AK^{+I}$ | 72 | 71 | 71 | 72 | 72 | 72 | 71 | 71 | 71 | 72 | 71 | 71 | 72 | 72 | 72 | 71 | 71 | 71 |
| | $AK^{+-I}$ | 72 | 83 | 84 | 72 | 84 | 88 | 71 | 86 | 91 | 72 | 82 | 84 | 72 | 83 | 87 | 71 | 86 | 90 |
| POL | $NN$ | 67 | 67 | 67 | 69 | 69 | 69 | 71 | 71 | 71 | 61 | 61 | 62 | 69 | 69 | 69 | 71 | 71 | 71 |
| | $AK^+$ | 58 | 58 | 58 | 59 | 59 | 59 | 64 | 64 | 64 | 58 | 58 | 58 | 59 | 59 | 59 | 64 | 64 | 64 |
| | $AK^{+-}$ | 58 | 68 | 75 | 59 | 79 | 90 | 64 | 87 | 96 | 58 | 68 | 70 | 59 | 68 | 64 | 64 | 66 | 60 |
| | $AK^{+I}$ | 42 | 42 | 43 | 40 | 40 | 40 | 38 | 38 | 38 | 42 | 42 | 43 | 40 | 40 | 40 | 38 | 38 | 38 |
| | $AK^{+-I}$ | 42 | 59 | 68 | 40 | 67 | 83 | 38 | 78 | 91 | 42 | 59 | 67 | 40 | 66 | 75 | 38 | 76 | 79 |

Table 3: prec and car of the five approaches for the three generators for different case base sizes.

function. The retrieve and adaptation processes attempt to adapt, for $NN$, the 3 source cases which are the closest ones to the target problem according to dist with a maximal distance of 2 on $\mathcal{P}$. For the three approaches based on AK, there is no threshold on the maximal distance: all source cases for which AR can be applied participate to solve the problem. A vote on the results computed from the retrieved cases is used to associate a unique element $y \in \mathcal{S}$. Moreover, a vote is also used when using AR: 3 ARs with the higher supports are used to adapt each of the source cases and the most frequent result wins.

All the approaches are evaluated according to two measures: the precision prec and the correct answer rate car. Let ntp be the number of target problems posed to the system, na be the number of (correct or incorrect) answers (ntp $-$ na is the number of target problems for which the system fails to propose a solution), and nca be the number of correct answers (according to the generated function f) and the predicted answer. So, the precision prec is defined as the average of the ratios $\frac{\texttt{nca}}{\texttt{na}}$, and the correct answer rate car is defined as the average of the ratios $\frac{\texttt{nca}}{\texttt{ntp}}$. The average of prec and car are computed on more than $10^5$ problem solvings for each function generator.

### 4.2 Results and discussion

Table 3 presents the precision and correct answer rate of the five approaches for the differents function generators with various sizes of $CB^+$ and $CB^-$. The results show that exploiting negative cases improves the CBR system results using inferences or not (comparison of $AK^{+-}$wrt. $AK^+$ and $AK^{+-I}$ wrt. $AK^{+I}$).

For the prec measure, approaches based on AK built only on positive examples ($AK^+$, $AK^{+I}$) never gives better results than the $NN$ baseline approach, mainly due to the extraction of too general rules. By contrast, when introducing in these approaches the exploitation of negative examples, the prec measure increases: around $+10\%$ for the best results of $AK^+$ for CNF and DNF functions and even around $+25\%$ for $AK^+$ for Pol functions. The results show also that from the precision point of view, $AK^{+-}$ gives always better results than $AK^{+-I}$.

For the car measure, approaches based on AK built only on positive examples ($AK^+$, $AK^{+I}$) give again results under those of the $NN$ approach. When exploiting negative examples, the car increases. An important point is that these car scores have to be considered in regards to the prec score because increasing the precision usually has a negative impact on the car measure (i.e. the system answers better but less times). That is why the results of the approaches based on negative examples must be highlighted because, except for the Pol functions, the increasing of the precision takes place without decreasing the car. The car measure also shows the interest of the $AK^{+-I}$ approach in the case of Pol functions. $AK^{+-I}$ is less precise than $AK^{+-}$ but is able to provide, most of the times, better answers than the $NN$ approach and with a better car than the $AK^{+-}$. Examination of the detailed results shows that the $AK^{+-}$ car under the $NN$ car is due to an insufficient number of ARs because of the support thresholds that have been chosen for the experiments. It is reasonable to think that the same kinds of results can be obtained for Pol functions than for the CNF and DNF functions by decreasing the support thresholds for building more ARs.

## 5 Conclusion

Most of works addressing $AK$ discovery for CBR focuses on the exploitation of *positive* source cases. In this paper, we have presented an approach based on CI extraction on variations between cases, to exploits at the best all the existing cases of the case base, the positive ones but also the negative ones, in order to improve the $AK$ discovery (which involves that the retain step of CBR does not choose only the positive cases, but labels the cases as "positive" and "negative").

The results of the first experiments are encouraging: they show that exploitating negative cases, when available, allows to compute an AK of better quality which use improves the results of the CBR system. This study is only a first step in the negative case exploitation issue for AK discovery. Indeed, the experiments are based on many factors that must be studied in details in order to optimize the process, e.g. the impact of the number of cases used to solve a target problem when using AK (in this study, all cases of the case base participates), or the impact of the AR support in the results.

Another ongoing work relies on the combination of these approaches to get even better results. For example, it is reasonable to think that building this combination based on a preference relation on the approaches will increase the final CBR system results. So, if the preferred method provides a solution, this is the solution returned by the combination function, else the second preferred method is tried, and so on. This makes sense since a method may provide results unfrequently but with high plausibility and should be tried before a method providing frequent results with lower plausibility. The

prec score seems to be, in this case, a good candidate to establish the preference order: $AK^{+-}$ preferred to $AK^{+-I}$, preferred to $NN$.

## References


1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications **7**(1) (March 1994) 39–59
2. Badra, F., Cordier, A., Lieber, J.: Opportunistic Adaptation Knowledge Discovery. In McGinty, L., Wilson, D.C., eds.: 8th International Conference on Case-Based Reasoning - ICCBR 2009. Volume 5650 of Lecture Notes in Computer Science., Seattle, États-Unis, Springer (2009) 60–74
3. Berasaluce, S., Laurencco, C., Napoli, A., Niel, G.: An Experiment on Mining Chemical Reaction Databases. In Le Thi, H.A., Dinh, T.P., eds.: Modelling, Computation and Optimization in Information Systems and Management Sciences - MCO'04, Metz, France, Hermes Science Publishing, London (2004) 535–542
4. Cordier, A., Dufour-Lussier, V., Lieber, J., Nauer, E., Badra, F., Cojan, J., Gaillard, E., Infante-Blanco, L., Molli, P., Napoli, A., Skaf-Molli, H.: Taaable: a Case-Based System for personalized Cooking. In Montani, S., Jain, L.C., eds.: Successful Case-based Reasoning Applications-2. Volume 494 of Studies in Computational Intelligence. Springer (2014) 121–162
5. d'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Case base mining for adaptation knowledge acquisition. In Veloso, M.M., ed.: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07), Morgan Kaufmann, Inc. (2007) 750–755
6. Dufour-Lussier, V., Lieber, J., Nauer, E., Toussaint, Y.: Improving case retrieval by enrichment of the domain ontology. In: 19th International Conference on Case Based Reasoning - ICCBR'2011, London, United Kingdom (2011)
7. Gaillard, E., Lieber, J., Nauer, E.: Adaptation knowledge discovery for cooking using closed itemset extraction. In: The Eighth International Conference on Concept Lattices and their Applications - CLA 2011, Nancy, France (2011)
8. Gaillard, E., Lieber, J., Nauer, E.: How Managing the Knowledge Reliability Improves the Results of a Reasoning Process. In: 16th European Conference on Knowledge Management - ECKM 2015, Udine, Italy (2015) 10 pages
9. Ganter, B., Kuznetsov, S.O.: Hypotheses and version spaces. In Ganter, B., de Moor, A., Lex, W., eds.: Conceptual Structures for Knowledge Creation and Communication, Berlin, Heidelberg, Springer Berlin Heidelberg (2003) 83–95
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)
11. Hanney, K., Keane, M.T.: Learning adaptation rules from a case-base. In Smith, I., Faltings, B., eds.: Advances in Case-Based Reasoning – Third Eur. Workshop, EWCBR'96. LNAI 1168, Springer Verlag, Berlin (1996) 179–192
12. Kuznetsov, S.O.: Complexity of learning in concept lattices from positive and negative examples. Discrete Applied Mathematics **142**(1) (2004) 111 – 125
13. Mitchell, T.: Version Space: An Approach to Concept Learning,. PhD thesis, Stanford University (1978)
14. Richter, M.M., Weber, R.O.: Case-based reasoning, a textbook. Springer (2013)
15. Skaf-Molli, H., Desmontils, E., Nauer, E., Canals, G., Cordier, A., Lefevre, M., Molli, P., Toussaint, Y.: Knowledge Continuous Integration Process (K-CIP). In: 21st WWW Conference - Semantic Web Collaborative Spaces workshop, Lyon, France (2012) 1075–1082
16. Szathmary, L., Napoli, A.: CORON: A Framework for Levelwise Itemset Mining Algorithms. Supplementary Proc. of The Third International Conference on Formal Concept Analysis (ICFCA '05), Lens, France (2005) 110–113

# SOLVING A VARIATION OF THE STABLE ROOMMATES PROBLEM USING EVOLUTIONARY ALGORITHMS

RAYMOND BERGER
*Computer Science, Eckerd College, 4200 54th Ave S.*
*St. Petersburg, FL 33711 US*

TREVOR LANE
*Computer Science, Eckerd College, 4200 54th Ave S.*
*St. Petersburg, FL 33711 US*

HOLGER MAUCH
*Computer Science, Eckerd College, 4200 54th Ave S.*
*St. Petersburg, FL 33711 US*

Abstract: This paper introduces an evolutionary algorithm (EA) to solve a variation of the Stable Roommates Problem where we use ideas from Case Based Reasoning (CBR) to evaluate roommate compatibility between a pair of students. The variation requires that roommates be paired based on a set of quantifiable survey questions and answers, as opposed to based on a list of desired roommates. It can be generalized and used to create near-optimal pairings based on any set of objects with quantifiable attributes where the attributes of two objects should show as little difference as possible, or equivalently where the attributes of two objects should be of maximum similarity.

More formally, the input to the problem are a set $V$ of $n$ objects (students to be paired up as roommates), where $n$ is a positive even integer, and a set of attribute values for each object (e.g., whether smoking or not, the type of pet one owns, allergies, typical bedtime, level of tidiness, etc..). The desired output of the problem is a partition of $V$ into $n/2$ subsets $V_i$, each of size 2 (called the "pairing"), such that the sum of the distances of the paired objects is minimal. If the sets of attribute values are consolidated into a $n \times n$ distance or cost matrix $c_{i,j}$ then this problem is known in the literature as the non-bipartite weighted matching problem [3, p.255].

Preliminary results on real data from the incoming Eckerd College class of 2022 show that the evolutionary algorithm successfully found pairings for 500 students faster than pairing found by conventional methods which involved manual assignment with support of spreadsheet software.

Keywords: evolutionary computing, evolutionary algorithm, stable roommates problem, non-bipartite weighted matching problem, case-based reasoning

## 1. Introduction

### 1.1. *Background*

Every year, Eckerd College has over 500 new students join the school, most of which choose to live on campus. In order to make sure students have the best college experience they should live with a roommate they can get along with. As such, the housing department sends a survey to incoming students with about a dozen questions pertaining to personal things such as when they prefer to go to bed and wake up, if they listen to music outloud, do they smoke, do they want to have a lot of people in their room, and more. Currently, the housing department takes these results in an excel sheet and manually pairs them up. The Stable Roommates Problem **[2],** is very similar to the problem described above. However, while the stable roommates problem asks each person to give a list of preferred roommates, our problem is that each person answers a survey about their preference. On the surface, these problems may seem similar but in our problem the students do not know each other and so they must be matched solely based on their preferences. Our goal was to create an algorithm that could take a list of n students with M quantified answers to questions and output a list where every student is paired with a roommate. The focus was on making sure that on average, roommates had good compatibility, as opposed to ensuring that each pair had some minimum compatibility.

The search space of this problem grows very rapidly with $n$, the number of students to be matched. Assuming that $n$ is a positive even integer, we count *(n-1)*(n-3)*...*5*3*1* different complete matchings, so the size of the search space is $\prod_{i=1}^{n/2}(2i-1) = \frac{n!}{2^{n/2}\,(n/2)!}$. We will use an evolutionary algorithm to explore and exploit this huge search space. There are two main reasons why we prefer to use an evolutionary algorithm compared to a customized version of a traditional algorithm for the non-bipartite weighted matching problem as described e.g. in [3, p.255]. First, the evolutionary algorithm offers more flexibility in terms of obtaining non-optimal solutions that might be preferable for practical reasons: sometimes a second-best, or third-best solution that is only slightly worse turns out to be better in practice because of unexpected minor constraints not considered by the model. Second, the evolutionary algorithm is easier to understand and implement than the traditional algorithm. The benefits of using evolutionary computation in software development projects are, among others, higher chance of successfully completing the project (i.e., smaller risk of failure), shortened development time, and robustness which still produces good results in the presence of minor bugs or defects [4].

*Goal:* The college wants students to be comfortable with their roommates.
*Problem:* The current process of pairing up incoming students is done by hand, which is time-consuming, error-prone, and because of the large size of the search space, only a tiny portion of it can be explored. Making an error of pairing up extremely incompatible roommates can impact the retention rate of first year students.

*Solution:* Create an evolutionary algorithm to generate a near optimal solution in a short amount of time. This will free up an estimated 40-60 hours of labor within the housing department and ensure a strong improvement in compatibility compared to years past.

**2. Evolutionary Approach(es) to solve the problem.**

**2.1. Standard Genetic Algorithm**

We have modified a standard GA in several ways to formulate our solution. The standard pseudocode of a GA is as follows:
1. Generate initial population from a predetermined domain.
2. Calculate Fitness.
3. Select chromosomes based on a selection mechanism like tournament selection.
4. With probability P (C) two parent chromosomes are crossed over at a random crossover point (one-point crossover). Create two offspring from parents.
5. Mutate with a chance P (m) by changing an allele in the chromosome.
6. Replace old population with offspring. Most fit individual from old population automatically gets passed onto next generation (elitism).

**3. Implementation**
With our current implementation, we already obtained good results without the use of a crossover operator. In future work we will experiment with variations of an edge crossover operator or an order crossover operator [6, pp.72-73] that ensures that relative order is preserved and that individual roommate pairs (i.e., certain edges) are preserved rather than split. The rest of the implementation uses mostly components of the standard GA.
**3.1** ***Problem Representation:*** Fixed-size array of roommates
**3.2** ***Mutation Operator****:* When a mutation occurs, two random students (array elements) are swapped.
**3.3** ***Individual****:* An individual $x$ is made up of the entire set of roommate pairs, represented as an array of indexes as roommates. For example, $x$=[2,3,5,1,4,6] represents the pairing {{2,3},{1,5},{4,6}}.

**3.4** ***Selection Method****:* Elitism Selection. Individuals are ranked and sorted by average compatibility score. The highest ranking individuals are placed in the elitist pool, with the size of the pool varying based on parameters given in the main method. In the results section, we experiment with different sizes of the parent selection pool.

**3.5** ***Fitness Function****:* The fitness of an individual $x$ is calculated by averaging the compatibility score of each roommate pair in the set. The compatibility score of a pair is calculated based off each student's answers to the personal preference survey used by the housing department.

For all weighting purposes (weights of questions, and weights of answers to questions) we relied on the expertise of the housing staff and our personal experience as resident advisors to predict

what might make good matches. Our core assumption is that people would rather live with someone who is similar to them than someone who is very different from them.

So in mathematical terms the fitness function reads:

$$f(x)= \frac{1}{(n/2)} \sum_{i=1}^{n/2} \left( \frac{1}{|Q|} \sum_{j=1}^{|Q|} w_j c_j(x_{2i-1}, x_{2i}) \right),$$

Where the following variables are used:

$x$: an individual of the population

$n$: total number of students (even)

$|Q|$: number of questions in the personal preference survey

$w_j$: weight of question j

and the compatibility subscore between two students $s$ and $t$ for question $j$ is defined as

$c_j(s,t)=1-|v_{j,a(s)}-v_{j,a(t)}|$

where

$V_{j,a(s)}$ is the value of student $s$'s response $a(s)$ for question $j$.

*Example Question:*

*Sample Question:* "How clean do you keep your room?"
*Answers:* Cluttered (0) , Mostly Messy (30), Generally Neat and Tidy (100)
*Student 1: Cluttered (0) Student 2: Mostly Messy (30)*
*Abs(00-30) = 30*
*100 - 30 = 70*
*.7 or 70% compatibility*

Currently, each question is weighted equally (i.e., if there are $|Q|$ questions, each question carries a weight of $1/|Q|$), however, allowing different weights is an option we are exploring for future research. Within each question we use different "weights" (i.e., values $V_{j,a(s)}$ ) for student's answers.

Initially, we assigned answers a numeric value between 1-5 to calculate the difference of each student's answers. We were unable to assign questions different weights which lead to imbalances. For example, the compatibility of a regular smoker vs someone who hates smoking was impacting the compatibility score of the pair just as much as if someone prefers to wake up before 10AM on weekends vs someone who prefers to wake up after 10AM on weekends.

Eventually, we moved to a scale of 0-100 to allow for more flexibility in number of answers and answer weights. We used our own discretion assigning the weights of answers, as some answers to questions were much more polarizing than others.

Appendix 1 includes the specific weights we assigned to each specific question. If we felt a question/answer combination did not have an intuitive relationship to compatibility, we set the weight at 50 so it did not interfere with the compatibility scores.

**4. Experimental results and discussion**

For our results, we ran our SGA using the data set from 2016 of incoming students provided by the Director of Housing at Eckerd College.
For each metric, we ran the evolutionary algorithm 20 times, and took the average fitness score at each generation.

Figure 4.1 shows that our evolutionary algorithm reaches a high level of fitness in just a few hundred generations. The mutation probability P(m) was set to .0025. We ran the algorithm for 500 generations 20 times. For each generation, 1-500, we took the best fitness from each run and averaged them. The initial fitness score, in generation one, was .74. By generation 500, the average population fitness for population size 30 was .97, for population size 10 it was .936. Further tests with more generations may be run in the future to evaluate the maximum compatibility of the sample roommates, however, the marginal fitness score of each generation decreases significantly after a few hundred generations. Although we currently do not have access to the roommate pairings that were matched by hand by the housing department, we recognize

that there is a significant improvement in compatibility from the initial population average to the final population average.

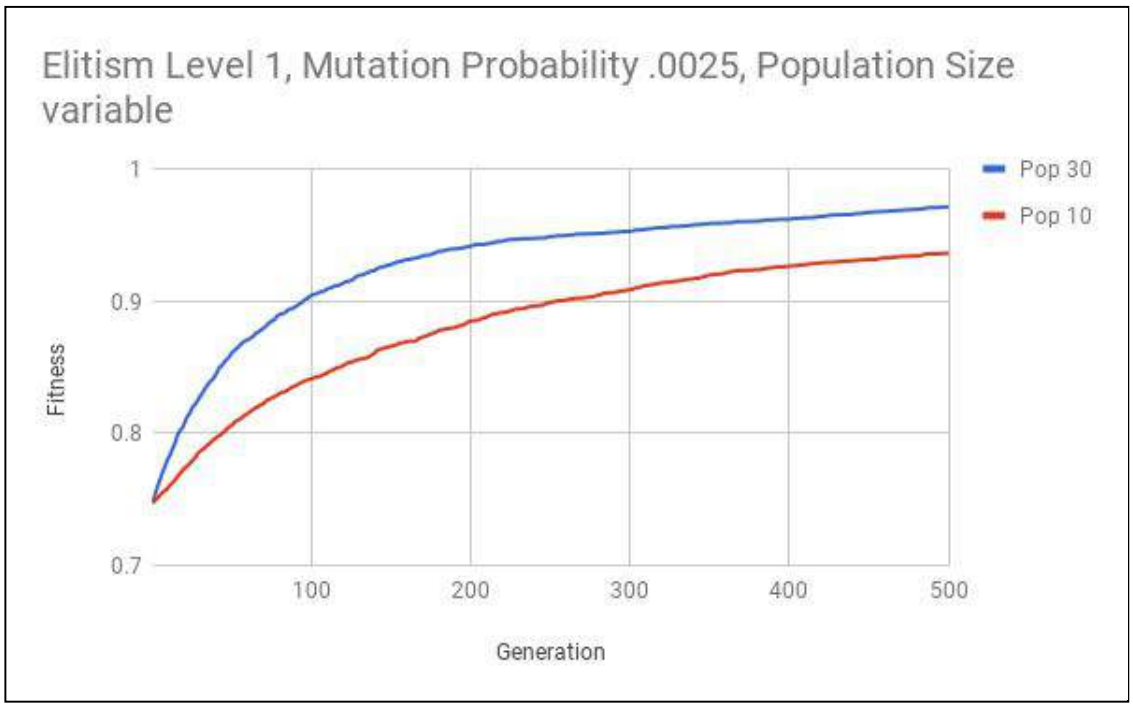



Figure 4.1 Graph showing how population size impacts the fitness over generations.

**5. Conclusion and Future Research**

From the observations discussed in section 4 we found that our model of pairing roommates is superior to the manual matching process. Through our algorithm, we found a set of roommate pairings with a very high level of average compatibility. Another interesting point for further studies will be to identify a suitable crossover operator to make the evolutionary algorithm more efficient. Further research could be in the area of setting minimum constraints and how different questions could be weighted. In particular, we envision to let our system use as input the data stored in the database of historical information, compute solutions, and derive improved weights for questions, and improved weights for answer options. Then we could store established cases with a granularity of the dormitory level in our database, so that e.g. a wellness dorm would utilize a different set of weights than say a pet-friendly dorm. Continued knowledge acquisition would simply mean to gather and store cases [5, p.308] and lead to continued improvements of pairings of students over time that also takes into account the physical dorm space available. In order to use CBR more directly in the future we envision to add to our database additional satisfaction measures; e.g., (i) number of students that have requested a reassignment ("divorce") from their current roommate, (ii) satisfaction level as measured by a survey of the housing students, (iii) satisfaction level as measured by a survey of the housing administration (residential

advisors and other staff). The hope is to discover and use cases like "student athlete and early riser are compatible in wellness dorm" or even rules of the sort like "smokers and nonsmokers are never satisfied when living together (and thus should not be roommates.)" In other words we retain information about both success (compatible matches from the past) and failure (incompatible matches from the past) of previous solutions. Then we can (1) Retrieve an appropriate case, (2) Reuse it (modification might be needed, e.g., replace 'student athlete' feature with 'wellness-conscious' if we face the situation of a newly built wellness dorm), (3) Revise it after the application (e.g., the student-athlete case to the wellness-conscious student), and (4) Retain the solution along with its compatibility score by saving it into the database.

**6. References**


1. David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, (1989).
2. R.W. Irving, *An efficient algorithm for the "stable roommates" problem*. Journal of Algorithms 6(4), 577–595, 1985.
3. Christos H. Papadimitriou, Kenneth Steiglitz, *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs, N.J., 1982.
4. Holger Mauch, *Efficient Junior Software Developers use Evolutionary Computation*, in: Proceedings of the 2017 Conference on Industry, Engineering and Management Science (IEMS2017), pp.57-63.
5. George F. Luger, *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, Sixth Edition, Pearson, 2008.
6. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, Second Edition, Springer, Berlin, 2015.

**Appendix 1**

This appendix includes all of the sample questions asked in the survey, the possible answers, and the weights we assigned to those answers.

| | |
|---|---|
| Do you smoke? | Absolutely not: 0<br>Occasionally or socially: 70<br>All the time: 100 |
| Do you object to a roommate who smokes? | Yes: 0<br>No: 100 |
| Please describe your room: | Cluttered: 0<br>Mostly Messy: 30<br>Generally neat and tidy 100 |
| You're leaving for eckerd tomorrow, you… | Haven't even started packing: 0<br>Still have a few last minute things to pack: 40<br>Had bags packed for weeks: 100 |
| When do you usually go to bed? | Before 10pm: 0<br>10pm to Midnight: 40<br>After Midnight: 100 |
| On a typical weekend, when do you usually wake up? | Before 10am: 0<br>In between: 30<br>After 11: 100 |
| How often do you like to have friends in your room? | Never: 0<br>Seldom: 20<br>Occasionally: 70<br>Frequently: 100 |
| In your free time, what are you most likely doing? | Binge-watching netflix: 50<br>On some sort of outdoor adventure: 50<br>Online gaming: 50<br>Putting together a kickball game: 50<br>Reading or studying in a quiet spot: 50<br>Relaxing with friends: 50 |
| Do you study best: (assigning them all at 50) | During the day: 50<br>In quiet: 50 |

| | |
|---|---|
| | In the evening: 50<br>With music: 50<br>After midnight: 50 |

# Online Learning with Reoccurring Drifts: The Perspective of Case-Based Reasoning

Marie Al-Ghossein[1⋆], Pierre-Alexandre Murena[1,2⋆], Antoine Cornuéjols[2], and Talel Abdessalem[1,3]

[1] LTCI - Télécom ParisTech, Paris, France
[2] UMR MIA518 - AgroParisTech INRA, Paris, France
[3] UMI CNRS IPAL NUS

**Abstract.** While machine learning usually focuses on learning one single concept from a batch of instances, the development of new media for data acquisition has led to the emergence of data streams. In such streams, the data distribution can change over time, and in particular previous states can reoccur. Handling such re-occurrences requires to manage a memory of past states. In this paper, we show that a parallel can be drawn between this task and the framework of case-based reasoning. Based on this parallel, we propose a general methodology and apply it to the problem of online topic modeling.

**Keywords:** Incremental learning · Concept Drift · Topic Modeling.

## 1 Introduction

The recent emergence of new sources of data related to the Internet of Things, online platforms or social media, among others, has led to a change in the way data is acquired in machine learning. Traditionally, all learning data is available in a batch and points are supposed to be drawn independently from a single distribution. In these new environments, data arrive in the form of streams that have to be handled online at high rates. One of the challenges raised by data stream mining, among others, concerns *concept drift*, i.e., change in the distributions [28].

Depending on the problem, several types of drifts can be observed [13]. *Abrupt* drifts correspond to a sudden change of distributions, switching from one distribution to another. On the contrary, *incremental* drifts correspond to slight evolutions of the distribution at each time step. *Reoccurring* drifts are particularly frequent and model the reoccurrence of past states (either cyclic or episodic). In general, algorithmic methods for data stream mining are biased toward one class of drifts and behave poorly on other categories. Active methods, such as ADWIN [2], focus on detecting drifts and adapt their models only when a drift is detected: thus, they are more adapted to abrupt drifts. On the contrary, passive methods adapt their model at each step, no matter if a drift actually happened

⋆ Both authors contributed equally.

or not: these methods are particularly efficient for incremental drifts. Dealing with reoccurring drifts requires a bit more adaptation and especially the use of a memory to evaluate the relatedness of the current observation with the past.

Such a use of memory is highly similar to the problems encountered in the domain of Case-based Reasoning (CBR). In particular, the four steps of CBR are observed in memory management for stream mining. Retrieval is implied in the process of detecting similar states in the past (*did the drift lead to a previously encountered distribution?*); Reuse brings a solution to the current case based on the retrieved cases; Revision exploits information of the new case to adapt current cases; Retention evaluates if the new case has to be kept in memory [6]. Exploiting this analogy, we propose to discuss in this paper the correlations between memory-based systems for data stream mining and CBR. We illustrate our reflection with the application of online topic modeling, which consists in evaluating topics of texts in streams [23].

The remainder of this paper is organized as follows. In Section 2, we present a general introduction to online learning as well as a couple of works that already exploit the similarity of online learning and CBR. In Section 3, we present our idea and discuss how each step of CBR can be adapted as a step in online learning. In Section 4, we propose the application of online topic modeling and illustrate our idea with some examples. Finally, we conclude our paper in Section 5 with a discussion of the potential implications and perspectives.

## 2 Related Work

**Online learning and memory.** Unlike batch learning, data stream mining requires to use a dynamic memory to store and forget data or concepts. [13] makes the distinction between short term memory, which is filled with current data, and long term memory which stores generalization of data, hence models.

Short term memory captures the current state of the stream at the time of the observation. A first family of methods stores the most recent data in a window, the length of which can be either fixed [28] or variable [12], [21]. Such methods are by nature inefficient for reoccurring concept drifts, since they are motivated by the assumption that most recent data are the best representations of current state. Other solutions are employed to overcome the main drawbacks of windowing strategies. Instance weighting strategies, for instance, keep all data in memory but the examples are weighted depending on their age in the stream [20].

Management of long term memory is the main concern of ensemble approaches. Online ensemble methods are based on the use of a pool of models that can be used and combined in order to describe current observations. Three strategies can be employed: training the models in advance and combining them dynamically [27], [28], continuously updating the models [10], or adding/removing (activating/desactivating) models [26].

**Recurring concept drifts.** The question of reoccurring drifts is essential in applications where seasonal effects can be observed or where the environment

can oscillate between several states. Various algorithms have been designed to tackle this issue.

The first method that was explicitly designed for reoccurring drifts (working with categorical attributes) is FLORA3 [28], an evolution of the original window-based FLORA method. When a drift is detected, FLORA3 inspects a pool of saved models instead of relearning a brand new model from scratch. The reuse procedure can be decomposed into three steps: Finding the optimal model (i.e., the model which makes the best predictions on the current data), update the chosen model (in order to make it consistent with current state) ,and comparing the updated version of the model to its memorized version. As an alternative to FLORA3, SPLICE-2 [15] offers another adaptation to recurring concept drifts on categorical features. The algorithm considers batches of data on which the concept is supposed to be stable. These batches are then clustered together, based on a notion of context similarity. In [29], the past history is modeled by a Markov chain and the future state is predicted according to the computed transition matrix.

Ensemble approaches are ideal for recurring drifts. For instance, Ensemble Building (EB) [24] aims to combine multiple classifiers with weights depending on their scores. If none of the known classifiers have good prediction rate on the currently observed chunk, a new classifier is trained and added to the pool. In a slightly different way, [11] chooses current models from a pool of previously learned model. The models are stored in memory, as well as their associated referee. In [17], the traces of past relevant concepts are stored in the pool of base-learners. These base-learners are learned each time that no existing classifier is a good predictor on the current window of examples. A diversity criterion on the pool of base-learners guarantees that the pool is both diverse and not cluttered.

The approach of [18] is very similar but exploits an idea that is close to CBR: Batch examples are selected by the algorithm and transformed to *conceptual vectors*. These vectors are then clustered together and a new classifier is learned for each cluster. Finally, the more generic algorithm Learn$^{++}$.NSE [9] is also perfectly tuned for recurring drifts: The algorithm is based on a passive incremental approach and proposes a weighted majority vote on a pool of classifiers.

**CBR and online learning.** Interestingly enough, the similarities between the main questions of CBR and online learning have not been exploited much. Apart from the ensemble techniques mentioned above which are implicitly related to CBR (in particular [18]), some methods use CBR in an explicit way. In [25], all new observations are directly stored in memory but, depending on their relevance to the context, they can be deactivated or reactivated. It is shown that this strategy improves the robustness of lazy learning algorithms to concept drift. CBR is used in the context of spam classification with concept drift [8]: The case base is filled with a vector representation of emails and managed using a Case Base Editing strategy [7] which removes both noisy and redundant cases. This case base editing strategy is also used by [22]. The problem of instance-based learning has also been expressed in the context of data streams [1]: The proposed

method updates the case base at each detection of a drift, implying the removal of a large number of cases.

## 3 Drift Adaptation seen as a CBR Problem

In this section, we present an interpretation of online learning in terms of case-based reasoning. The presented notions are given at an abstract level: an applied example is proposed in the next section.

### 3.1 General Process

In a context of stream mining, it is not possible to have a full CBR process at each step. The methodology we propose allies the performance qualities of active methods for stream mining and the use of memory, which is typical of CBR.

The data stream is analyzed by a drift detection algorithm (for instance ADWIN [2]) on the base of a *score*. The purpose of this algorithm is to detect when the data distribution changed and when an adaptation is needed. Since a drift is necessarily detected with some delay, a drift detection comes with a batch of instances $\mathcal{D}$ generated by the new distribution. The score is computed based on a representation model of the data. It can correspond to the error rate of the model or to its likelihood for instance. In the following, we will denote by $score(\mathcal{D}, \mathcal{M})$ the score of data $\mathcal{D}$ relative to the model $\mathcal{M}$.

Instead of relearning the model from the batch selected by ADWIN, we propose to select the model from a case base and to adapt it in order to fit the new data. This use of case base is ideal for dealing with recurring concept drifts, as suggested by the state of the art.

### 3.2 Case Representation

One of the central questions of CBR concerns the management of the case base and the representation of cases. In the context of online learning, we propose the following storage process. A case corresponds to a data point, after or before any transformation process. As suggested by [18], the points are then grouped into clusters corresponding to concepts. Each of the clusters is associated to a unique decision model which can be either discriminative (e.g., a classifier in supervised setting) or generative (e.g., a probability distribution in unsupervised setting).

In a perspective of reusing previously solved cases to address new questions, this representation consists of a factorized representation of problems: the solution (here the decision model) is shared by several cases.

### 3.3 Case Retrieval

When a drift is detected, the first question is how to associate the batch of points to a corresponding group of cases. Using the representation we proposed,

the relatedness of a batch to any case inside a cluster can be measured by its relatedness to its associated model. As a good candidate for this measure, we propose to use the score function.

The optimal cluster of cases is chosen to be the cluster such that the associated model maximizes $score(\mathcal{D}, \mathcal{M})$. Note that, especially for the first drifts, none of the learned models might describe well the observed data. In order to discard incorrect models, a threshold can be given for the score, under which no cases are selected. In the scope of this paper, we will ignore this problem.

### 3.4 Case Reuse

The retrieved cases do not necessarily correspond exactly to the current distribution of data. In order to cope with this problem, the decision model in use is retrained on a specific batch of data. This batch contains the points in the case cluster and the points in batch $\mathcal{D}$. This reused model thus incorporates both knowledge from the past and from current data. The model is taken as the reference model for the next observations, until a new drift is detected.

### 3.5 Case Revision

In the time interval between two drifts, we propose a case revision based on two aspects. On the one hand, the description model is updated online for each new observation, using a stochastic optimization scheme [5]. On the other hand, the most relevant data instances are kept in a short-term memory, in order to feed the case in the retainment phase. The relevance of an instance is evaluated with the score function, for the current model. These two actions are complementary: the model update is important in order to keep the decision model up-to-date, while the data selection contributes to an optimal case design.

### 3.6 Case Retainment

When a drift is detected, the model has to be saved in the case base. Two possibilities appear: either to re-write the selected case or to create a new case. This decision is motivated by the impact of creating a new model onto the global case base. If $(\mathcal{M}^{old}, \mathcal{D}^{old})$ designates the previous model and the cases associated to it, and $(\mathcal{M}^{new}, \mathcal{D}^{new})$ designates the current model and the data stored in short-term memory, one possibility to discriminate the two options is to compare $score(\mathcal{D}^{old}, \mathcal{M}^{new})$ and $score(\mathcal{D}^{old}, \mathcal{M}^{old})$. If the first score is higher, the new model is better at describing data from previous case model and thus the model has to be overridden. Otherwise, the previous model was satisfactory and the new model is relevant only for the new cases. Thus a new model has to be created and is associated to the instances in short-term memory.

In the case where the previous model is overridden, the cases stored in short-term memory are added to the case cluster of the model. In simple applications, where the number of cases per cluster is limited, only the cases with higher score are kept.

## 4 Application: Online Topic Modeling with AWILDA

In this section, we propose an application of the described methodology in the case of online topic modeling.

### 4.1 Presentation of the Problem

Topic modeling is a machine learning technique that processes documents as bag-of-words and represents them as vectors of topics. One of the most prominent methods used for topic modeling is Latent Dirichlet Allocation (LDA) [4], where documents are modeled as mixtures over latent topics, and each topic is characterized by a distribution over words.

Taking into account the order in which documents are generated is important since it allows one to track the evolution of topics over time. Variants of LDA have been proposed to incorporate temporal dynamics into the topic model [3]. However, these variants are not able to handle streams of documents arriving in real-time: They require the whole documents to be accessible in order to infer the corresponding model.

AWILDA [23] is designed to process documents arriving in a stream, one by one, and combines online LDA [16] and ADaptive Windowing (ADWIN) [2], a technique for drift detection. The main idea is that a change in the distribution generating the documents will result in a drift in the likelihood of the LDA model currently used. This change is detected in AWILDA using an ADWIN component that processes the series of likelihoods and detects when the LDA model is no longer adapted to the recently received documents. When this is the case, ADWIN returns the sub-window of documents corresponding to the new distribution. These documents are used to retrain the new topic model. Further details about AWILDA can be found in [23].

To the best of our knowledge, there is no previous work handling reoccurring drifts that are present in topic modeling. Such scenarios could easily occur in the news domain for example, where we observe unexpected events related to specific topics that recurrently appear over time and affect the distribution of words and topics in documents. In the following, we present the variant of AWILDA that is able to adapt to reoccurring drifts and that we explore in this work, followed by experiments demonstrating our approach.

### 4.2 AWILDA with Reoccurring Drifts

Textual content written by individuals and shared online on several platforms (e.g., tweets, news, reviews) is usually affected by their specific context that is in turn influenced by real-life events. It is essential to account for changes happening in the distribution of topics and words in order to improve document modeling. While AWILDA retrains a model at each detected drift, it cannot leverage previous learned information about a concept when it reappears due to its possible recurrence. We propose to store learned models that are no longer adapted to the current context and reuse them later when they are valid again.

In terms of the methodology described in Section 3, this problem can be described as follows. Each point corresponds to a document (described as a bag-of-words) and documents arrive sequentially as a stream. The task we address here is a modeling task: The purpose is to identify a good model that fits the data in real time. As a consequence, the model used to select the cases to cluster corresponds to the LDA model itself. The score function that we use is the log-likelihood, which measures the probability of observed documents to be generated by the model.

### 4.3 Experimental Results

In this subsection, we present the experiments we conducted on two datasets from different domains in order to demonstrate our approach.

**Datasets.** The first dataset gathers hotel reviews posted on TripAdvisor [14] and is denoted by *trip*. The dataset comprises approximatively 200k reviews published from October 2001 to November 2009 and related to hotels located in ten different cities. We expect to observe a recurrence of concepts in this type of dataset due to the seasonality effect that influences the behavior of tourists and the hotel aspects they attach importance to. The second dataset contains a collection of 9k news articles published in German on several news portals, during the month of February 2016 (*plista dataset* [19]) and is denoted by *news*. Documents from both datasets have been preprocessed by mainly removing stop words, removing words occurring once, and stemming remaining words.
**Evaluation.** The main goal in topic modeling is to maximize the likelihood on unseen documents. In order to evaluate the topic model, we measure, for each received document, the perplexity which is defined in [4]. Perplexity measures the capacity of the model to generalize to new data. A lower value of perplexity indicates a better generalization capacity.
**Methods.** We compare our approach, denoted by CB-AWILDA, to AWILDA [23] that is able to analyze documents arriving in a stream. AWILDA is better suited to handle abrupt drifts: The model is retrained for each detected drift using the documents corresponding to the new distribution. AWILDA and CB-AWILDA are considered to be receiving a stream of document in real-time and to process documents sequentially. We use the first 20% of the document stream to initialize the models and we measure perplexity for all the documents received afterwards. We report the results obtained by fixing the number of topics to 5, and the minimum number of cases to 2.
**Results.** Figure 1 shows the perplexity measured on the document streams of *trip* and *news* for AWILDA and CB-AWILDA. The performance of both methods at the beginning of the process is relatively similar. This is expected since the learning process is the same before any drift is detected. As more documents are received, CB-AWILDA outperforms AWILDA for the task of document modeling. For each detected drift, AWILDA is retrained using the documents related to the new distribution. This is thus pushing the model to forget previously learned information that may be valid in the future. On the

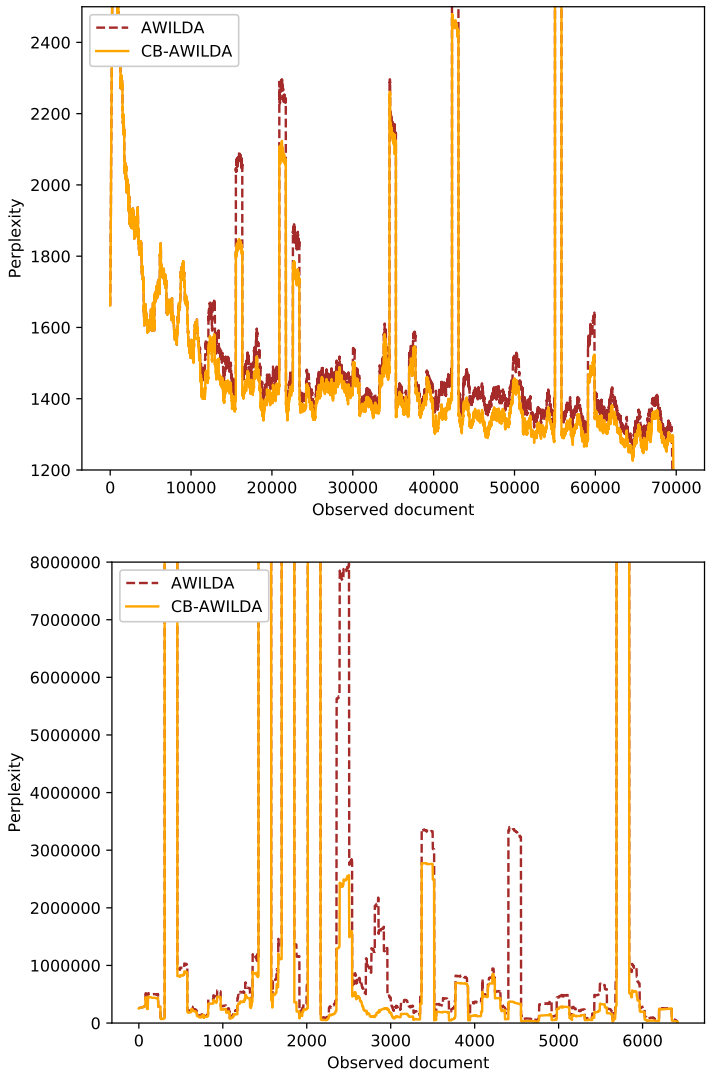



**Fig. 1.** Evaluation of AWILDA and CB-AWILDA for the task of document stream modeling on the *trip* (first figure) and the *news* (second figure) datasets.

other hand, CB-AWILDA leverages previously seen documents that correspond to the current distribution and uses them in the learning process. CB-AWILDA is therefore more adapted to the documents that are currently being received, which results in a better performance in terms of perplexity.

## 5 Conclusion

In this paper, we address the problem of online learning with reoccurring drifts and we formulate its solution in the context of the case-based reasoning framework. Observations are represented as cases and are grouped into clusters corresponding to concepts and associated to an adapted model. When a drift is detected, we retrieve the case related to the concept that is currently being observed. Retrieved cases are used to update the decision model and can be updated or overridden if necessary. We propose an application of our approach for online topic modeling. We show that taking into account reoccurring drifts improve the task of document modeling.

Future work includes the application of the proposed solution to other tasks. In particular, online recommender systems could benefit from such an approach in order to adapt to reoccurring drifts appearing on the user and item level due to seasonal and unexpected events.

## References

1. Beringer, J., Hüllermeier, E.: Efficient instance-based learning on data streams. Intelligent Data Analysis **11**(6), 627–650 (2007)
2. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. pp. 443–448. SIAM (2007)
3. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning. pp. 113–120. ACM (2006)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
5. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
6. De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., T COX, M., Forbus, K., et al.: Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review **20**(3), 215–240 (2005)
7. Delany, S.J., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In: European Conference on Case-Based Reasoning. pp. 128–141. Springer (2004)
8. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowledge-based systems **18**(4-5), 187–195 (2005)
9. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks **22**(10), 1517–1531 (2011)

10. Fern, A., Givan, R.: Online ensemble learning: An empirical study. Machine Learning **53**(1-2), 71–109 (2003)
11. Gama, J., Kosina, P.: Tracking recurring concepts with meta-learners. In: Portuguese Conference on Artificial Intelligence. pp. 423–434. Springer (2009)
12. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Brazilian symposium on artificial intelligence. pp. 286–295. Springer (2004)
13. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 44 (2014)
14. Ganesan, K., Zhai, C.: Opinion-based entity ranking. Information retrieval **15**(2), 116–150 (2012)
15. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. Machine learning **32**(2), 101–126 (1998)
16. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent dirichlet allocation. In: advances in neural information processing systems. pp. 856–864 (2010)
17. Jaber, G., Cornuéjols, A., Tarroux, P.: A new on-line learning method for coping with recurring concepts: the adacc system. In: International Conference on Neural Information Processing. pp. 595–604. Springer (2013)
18. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. Knowledge and Information Systems **22**(3), 371–391 (2010)
19. Kille, B., Hopfgartner, F., Brodt, T., Heintz, T.: The plista dataset. In: Proceedings of the 2013 International News Recommender Systems Workshop and Challenge. pp. 16–23. ACM (2013)
20. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intelligent data analysis **8**(3), 281–300 (2004)
21. Kuncheva, L.I., Žliobaitė, I.: On the window size for classification in changing environments. Intelligent Data Analysis **13**(6), 861–872 (2009)
22. Lu, N., Lu, J., Zhang, G., De Mantaras, R.L.: A concept drift-tolerant case-base editing technique. Artificial Intelligence **230**, 108–133 (2016)
23. Murena, P.A., Al Ghossein, M., Abdessalem, T., Cornuéjols, A.: Adaptive window strategy for topic modeling in document streams, accepted in International Joint Conference on Neural Networks 2018
24. Ramamurthy, S., Bhatnagar, R.: Tracking recurrent concept drift in streaming data using ensemble classifiers. In: Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on. pp. 404–409. IEEE (2007)
25. Salganicoff, M.: Tolerating concept and sampling shift in lazy learning using prediction error context switching. In: Lazy learning, pp. 133–155. Springer (1997)
26. Street, W.N., Kim, Y.: A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 377–382. ACM (2001)
27. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In: Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on. pp. 679–684. IEEE (2006)
28. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine learning **23**(1), 69–101 (1996)
29. Yang, Y., Wu, X., Zhu, X.: Mining in anticipation for concept change: Proactive-reactive prediction in data streams. Data mining and knowledge discovery **13**(3), 261–289 (2006)

# First steps toward finding relevant pathology-gene pairs using analogy

Marie-Dominique Devignes[1], Yohann Fransot[1], Yves Lepage[2], Jean Lieber[1], Emmanuel Nauer[1], and Malika Smaïl-Tabbone[1]

[1]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
firstname.lastname@loria.fr
[2]IPS, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
yves.lepage@waseda.jp

**Abstract.** This paper presents a first study to infer pathology-gene relation instances using analogy. The meaning of this relation between a pathology $\mathtt{P}$ and a gene $\mathtt{G}$ is "A mutation of $\mathtt{G}$ in a person can cause the appearance of pathology $\mathtt{P}$ for this person." In this work, a pathology is represented by a set of classes from HPO, the Human Phenotype Ontology, whereas a gene is represented similarly, but using GO, the Gene Ontology. Some $(\mathtt{P}, \mathtt{G})$ instances of the pathology-gene relations are known and the idea is to use analogical reasoning to infer new relations. The schema of the inference is as follows: if a target pathology $\mathtt{P}$ is in analogy with three other pathologies $\mathtt{P}_A$, $\mathtt{P}_B$ and $\mathtt{P}_C$ for which associated genes $\mathtt{G}_A$, $\mathtt{G}_B$ and $\mathtt{G}_C$ are known, then it is plausible that the gene $\mathtt{G}$, to be associated with $\mathtt{P}$, is in analogy with $\mathtt{G}_A$, $\mathtt{G}_B$, $\mathtt{G}_C$. This idea has proven to be fruitful in other domains, such as machine translation.
The preliminary question explored in this paper is the following: given four pathologies $\mathtt{P}_A, \mathtt{P}_B, \mathtt{P}_C$ and $\mathtt{P}_D$ in analogy and for which the associated genes $\mathtt{G}_A$, $\mathtt{G}_B$, $\mathtt{G}_C$ and $\mathtt{G}_D$ are known, are thess genes in analogy, or, at least, in approximate analogy?
Results of a large scale analysis ($4,000$ $(\mathtt{P}, \mathtt{G})$ pairs) reveal that the quadruples of genes associated with quadruples of pathologies in analogy do not display statistically different analogical dissimilarity values than randomly selected quadruples of genes. Nevertheless very low analogical dissimilarity values are found in a small subset of gene quadruples that are specifically associated with pathologies in analogy. Analysis of these quadruples may allow us to learn more sophisticated analogical relations on genes in order to improve the recovery of pathology-gene pairs using analogy.

**Keywords:** analogy, gene-pathology relation, ontology, annotation

## 1 Introduction

Transposing proportional analogies from a domain to another one is a general principle in problem solving. It has been applied to several problems in natural language processing, like grapheme-phoneme transcription (i.e., pronounciation) [7], morphological analysis [3], syntactic analysis [1] or machine translation [4]. A *proportional analogy* is

a quaternary relation between four objects $A$, $B$, $C$ and $D$ denoted by $A : B :: C : D$. It is read: "$A$ is to $B$ as $C$ is to $D$". As for the special case of translation, the approach is based on a set of cases, a case being an ordered pair $(S, S^{\neg})$ where $S$ is a sentence in a first language and $S^{\neg}$is a translation of $S$ in a second language. The principle consists in, given a sentence $D$ in the first language, (1) finding three cases $(A, A^{\neg})$, $(B, B^{\neg})$, $(C, C^{\neg})$ verifying $A : B :: C : D$ in the first language and (2) solving the *analogical equation* $A^{\neg}$: $B^{\neg}$:: $C^{\neg}$: $x$ in the second language. The following example illustrates this idea, for translation from French into English:

$$
\begin{aligned}
(A, A^{\neg}) &= (\textit{Tu évites de danser le tango ?}, && \textit{Do you avoid dancing tango?})\\
(B, B^{\neg}) &= (\textit{J'évite de manger du melon.}, && \textit{I avoid eating melon.})\\
(C, C^{\neg}) &= (\textit{Tu aimes danser le tango ?}, && \textit{Do you like dancing tango?})\\
D &= \textit{J'aime manger du melon.} && \text{(target problem)}
\end{aligned}
$$

As the relation $A : B :: C : D$ holds, the three cases are returned and it is inferred that a candidate translation $D^{\neg}$of $D$ is a solution of the analogical equation $A^{\neg}$: $B^{\neg}$:: $C^{\neg}$: $x$. In this example, the solution $x =$ *I like eating melon.* is a correct translation $D^{\neg}$of $D$. In [5], this approach has been reformulated in case-based reasoning terms and some extensions are proposed.

Now, the question is whether this approach can apply to different data than language data. This paper examines this issue in the context of pathology-gene pairs $(\mathtt{P}, \mathtt{G})$: roughly said, such a case means that the gene $\mathtt{G}$ plays an important role in the pathology $\mathtt{P}$. Thus the idea is to use proportional analogies to mine a pathology-gene base $\mathcal{B}$, in order to find hypotheses of new pairs $(\mathtt{P}, \mathtt{G})$, according to the following inference rule:

$$
\frac{\begin{array}{c}\mathtt{P}_A : \mathtt{P}_B :: \mathtt{P}_C : \mathtt{P}_D \quad \text{in the pathology space}\\ (\mathtt{P}_A, \mathtt{G}_A), (\mathtt{P}_B, \mathtt{G}_B), (\mathtt{P}_C, \mathtt{G}_C) \in \mathcal{B}\\ x \text{ is a solution of } \mathtt{G}_A : \mathtt{G}_B :: \mathtt{G}_C : x \quad \text{in the gene space}\end{array}}{\text{It is plausible that } (\mathtt{P}_D, x) \text{ is a relevant pathology-gene pair}} \qquad (1)
$$

To address this issue, some notions related to proportional analogy are necessary: they are introduced in Section 2. It is also necessary to have some biological notions about pathologies and genes, in particular about their representations: the goal of Section 3 is to introduce these notions. Based on all these notions, the first approach to examine this inference in this application domain is detailed in Section 4 and the implementation principles are presented in Section 5. Section 6 presents the first results and interpretations. Finally, Section 7 concludes and proposes several research directions..

## 2 Proportional Analogy : Definitions

*Basic Definitions.* Let $\mathcal{U}$ be a set. A *proportional analogy* on $\mathcal{U}$ is a quaternary relation on members of $\mathcal{U}$ that is usually denoted as $A : B :: C : D$. It is read: "$A$ is to $B$ as $C$ is to $D$". In this expression, $A : B$ and $C : D$ are called *ratios*, and the binary relation :: is called *conformity*. A proportional analogy (PA) generally satisfies the following postulates: for any $(A, B, C, D) \in \mathcal{U}^4$,

– $A : B :: A : B$ is always true (reflexivity of conformity);
– if $A : B :: C : D$ then $C : D :: A : B$ (symmetry of conformity);
– if $A : B :: C : D$ then $A : C :: B : D$ (exchange of the means).

Other properties, like the one called *exchange of the extremes*, can be deduced from the previous postulates: if $A : B :: C : D$ then $D : B :: C : A$. Indeed, for one given analogy, there exist seven other equivalent forms.

The reflexivity of conformity gives birth to many analogies $A : B :: A : B$, equivalent to $A : A :: B : B$, by exchange of the means. Such analogies are poorly informative in that they give no information about $A$ and $B$ or about their relation. In the sequel, we call them *flat analogies*.

*Boolean Representations.* Given $\mathcal{U} = \mathbb{B} = \{0, 1\}$ the set of Booleans (where 0 and 1 represent `false` and `true`, respectively), the proportional analogy defined by $A : B :: C : D$ if $B - A = D - C$ (where these differences belong to $\{-1, 0, 1\}$) satisfies the PA postulates. There are six patterns $ABCD$ satisfying this relation: 0000, 1111, 0011, 1100, 0101 and 1010.

In $\mathcal{U} = \mathbb{B}^n$, the relation defined by a component by component analogy — i.e., $A : B :: C : D$ if $a_i : b_i :: c_i : d_i$ for every $i \in \{1, 2, \ldots, n\}$ — also satisfies this postulates (where, e.g., $A = (a_1, a_2, \ldots, a_n)$). For example,

$$(0, 1, 1, 0, 0) : (1, 1, 0, 1, 0) :: (0, 0, 1, 0, 0) : (1, 0, 0, 1, 0) \tag{2}$$

An $n$-tuple of Booleans $T = (T_1, T_2, \ldots, T_n)$ can be encoded by the set $\widehat{T}$ of its indices with the value 1. For example, if $T = (0, 1, 1, 0, 1, 0)$, then $\widehat{T} = \{2, 3, 5\}$. Now, given $T$, $U \in \mathbb{B}^n$, let $\texttt{key}(T, U) = (\widehat{T} \setminus \widehat{U}, \widehat{U} \setminus \widehat{T})$. In fact $\texttt{key}(T, U)$ encodes the changes from $T$ to $U$. It can be shown that the proportional analogy on $\mathcal{U} = \mathbb{B}^n$ defined above can be characterized by:

$$A : B :: C : D \quad \text{iff} \quad \texttt{key}(A, B) = \texttt{key}(C, D) \tag{3}$$

This can be verified on the example (2) as:

$$\texttt{key}(A, B) = \texttt{key}(C, D) = (\{3\}, \{1, 4\})$$

When, in the data, the $n$-tuples of Booleans are sparse (i.e., they contain a majority of 0), the interest in this characterization is algorithmic: the size necessary to encode $\texttt{key}(A, B)$ is much smaller than $n$.

*Analogical Dissimilarity.* If four objects $(A, B, C, D) \in \mathcal{U}^4$ are *not* in proportional analogy, a question that can be raised is "How far are these objects from forming an analogy?" An analogical dissimilarity (AD) [6] is a function $\texttt{AD} : (A, B, C, D) \mapsto [0, +\infty[$ satisfying the following postulates: for any $(A, B, C, D, E, F) \in \mathcal{U}^6$,

– $\texttt{AD}(A, B, C, D) = 0$ iff $A : B :: C : D$ (consistency with analogy);
– $\texttt{AD}(A, B, C, D) = \texttt{AD}(C, D, B, A)$ (symmetry, reflecting symmetry of conformity);
– $\texttt{AD}(A, B, C, D) = \texttt{AD}(A, C, B, D)$ (central permutation, i.e., exchange of the means);

- $\texttt{AD}(A, B, E, F) \leq \texttt{AD}(A, B, C, D) + \texttt{AD}(C, D, E, F)$ (triangle inequality);
- If $A \neq B$ then $\texttt{AD}(A, B, C, D) \neq \texttt{AD}(B, A, C, D)$.

In $\mathcal{U} = \mathbb{B}$, AD defined by $\texttt{AD}(A, B, C, D) = |(B - A) - (D - C)| \in \{0, 1, 2\}$ for $A, B, C, D \in \mathbb{B}$ satisfies the AD postulates.

In $\mathcal{U} = \mathbb{B}^n$, AD defined by $\texttt{AD}(A, B, C, D) = \sum_{i=1}^{n} \texttt{AD}(A_i, B_i, C_i, D_i)$ also satisfies the AD postulates. For example,

$$\texttt{AD}((0,0,0,0,1),(1,0,1,1,0),(0,1,0,1,0),(1,0,0,0,1)) = 0 + 1 + 1 + 2 + 2 = 6$$

## 3 Pathology-Gene Relations

A gene is a sequence of nucleotides along a segment of DNA that encodes instructions for RNA synthesis, which, when translated into protein, leads to the expression of phenotypes. The genes are thus the basic physical units of heredity. Phenotypes such as pathologies (or diseases) are associated to genes in some public databases. The most famous one is the OMIM database which focuses on human hereditary diseases (`http://omim.org/`). OMIM provides pathology-gene relations that were carefuly curated and documented w.r.t. the literature. Today, a large number of diseases lack responsible gene(s) hence the numerous gene prioritization methods [2].

On one hand, genes are annotated with their known functions (in fact the functions are accomplished by the proteins produced by the genes). Such functions are taken from GO (Gene Ontology), an ontology encompassing thousands of terms (or classes) mainly linked with subclass-of relations (`http://www.geneontology.org/`). The GO is stuctured as an r-DAG (rooted directed acyclic graph). Some of the gene-GO term relations are based on experimental evidence or published papers whereas others are simply inferred from known relationships combined with gene sequence similarity for instance. The gene-GO term relations are indeed qualified with evidence codes. Manually-assigned evidence codes fall into four general categories: experimental (such as EXP: inferred from experiment), computational analysis (e.g., ISS: inferred from sequence or structural similarity), author statements (for instance TAS: traceable author statement), and curatorial statements (for example IC: inferred by curator). Only one evidence code (IEA: inferred from electronic annotation) is not assigned by a curator. Such gene annotations in many species are available in a public database, AMIGO (`http://amigo.geneontology.org/amigo`).

On the other hand, diseases are annotated with their known associated phenotypes (a.k.a. symptoms) taken from HPO (Human Phenotype Ontology). Similarly to genes and GO-terms, HPO is structured as a r-DAG and disease-HPO term (or class) relations are qualified with evidence codes (e.g. PCS for published clinical study, ICE for individual clinical experience, ITM for inferred from text mining, TAS and IEA having the same meaning as for gene-GO relations) depending on the origin of the relationship. Disease annotations are also stored in the OMIM database.

Table 1 shows an example of disease-gene relationship along with their respective annotations.

| Pathologie or Gene | Name | Abbr. | HPO or GO term |
|---|---|---|---|
| Pathologie | Cardiomyopathy dilated 1I | CMD1I | Reduced systolic function (TAS)<br>Congestive heart failure (TAS) |
| Gene | Desmin | DES | Muscle contraction (TAS)<br>Regulation of heart contraction (TAS)<br>Structural constituent of cytoskeleton (TAS)<br>Intermediate filament (IEA) |

Table 1: Example of $(\mathtt{P}, \mathtt{G})$ pair with associated HPO and GO annotations. Evidence codes are between parentheses.

In this work, a pathology $\mathtt{P}$ is represented as a tuple of Booleans in the following way. Let $m$ be the number of classes of HPO and $\{\mathtt{CP}_1, \mathtt{CP}_2, \ldots, \mathtt{CP}_m\}$ be the set of these classes. $\mathtt{P}$ is described by the $m$-tuple $(p_1, p_2, \ldots, p_m) \in \mathbb{B}^m$ such that $p_i = 1$ iff $\mathtt{P}$ is described by $\mathtt{CP}_i$ (for each $i \in \{1, 2, \ldots, m\}$). It must be noted that if a class $\mathtt{CP}_i$ is a subclass of $\mathtt{CP}_j$ (either directly or by transitivity of the subclass-of relation), then a pathology $\mathtt{P}$ described by $\mathtt{CP}_i$ is also described by $\mathtt{CP}_j$: if $p_i = 1$ then $p_j = 1$ (this is mere application of the deductive closure based on the subclass-of relation). The tuple $(p_1, p_2, \ldots, p_n)$ is sparse: only a small proportion of the classes of HPO are used to describe each single pathology.

The representation of a gene $\mathtt{G}$ is done in a similar way by a tuple $(g_1, g_2, \ldots, g_n) \in \mathbb{B}^n$, where $n$ in the number of classes in GO.

## 4 Proposed Approach

The goal of the proposed approach is to examine whether the inference rule (1) that associates to a pathology $\mathtt{P}_D$ a gene $\mathtt{G}_D$ gives a relevant pathology-gene pair $(\mathtt{P}_D, \mathtt{G}_D)$, at least with a reasonable frequency. If, e.g., the answer was that it holds $10\%$ of the time, it would still be interesting in a knowledge discovery perspective: providing an expert with original hypotheses with such a proportion of correctness still remains interesting. Having this is mind, two experiments were conducted.

*First experiment.* The set of quadruples $(\mathtt{P}_A, \mathtt{P}_B, \mathtt{P}_C, \mathtt{P}_D)$ of pathologies (in the chosen representation formalism) such that $\mathtt{P}_A : \mathtt{P}_B :: \mathtt{P}_C : \mathtt{P}_D$ is computed. Only non flat analogies were kept. A gene is associated to each pathology: $\mathtt{G}_A$, $\mathtt{G}_B$, $\mathtt{G}_C$ and $\mathtt{G}_D$. The experiment is designed to meet two objectives:

- find out the proportion of quadruples of genes (related to analogies on pathologies) which are in analogy;
- the situation where $\mathtt{P}_A : \mathtt{P}_B :: \mathtt{P}_C : \mathtt{P}_D$ holds but $\mathtt{G}_A : \mathtt{G}_B :: \mathtt{G}_C : \mathtt{G}_D$ does not hold, may still be interesting if the analogical dissimilarity between these genes is low (i.e., it is close to an exact analogy). Thus, the second objective is to compare the distribution of $\mathtt{AD}(\mathtt{G}_A, \mathtt{G}_B, \mathtt{G}_C, \mathtt{G}_D)$ provided that the corresponding pathologies are

in analogy with the distribution of $\mathtt{AD}(\mathtt{G}_A, \mathtt{G}_B, \mathtt{G}_C, \mathtt{G}_D)$ in general. In particular if the average of the first distribution is significatively lower than the average of the second distribution, this would mean that analogies between genes are somehow connected with analogies between pathologies.

*Second Experiment.* The second experiment examines the inference the other way round: from genes to pathologies, i.e., the following inference

$$\frac{\begin{array}{c}\mathtt{G}_A : \mathtt{G}_B :: \mathtt{G}_C : \mathtt{G}_D \quad \text{in the gene space} \\ (\mathtt{P}_A, \mathtt{G}_A), (\mathtt{P}_B, \mathtt{G}_B), (\mathtt{P}_C, \mathtt{G}_C) \in \mathcal{B} \\ x \text{ is a solution of } \quad \mathtt{P}_A : \mathtt{P}_B :: \mathtt{P}_C : x \quad \text{in the pathology space}\end{array}}{\text{It is plausible that } (\mathtt{P}_D, x) \text{ is a relevant pathology-gene pair}} \tag{4}$$

The same objectives as in the first experiment are pursued in the reverse direction.

## 5 Implementation Principles

The main steps of the approach are schematized on Figure 1. All of the data has been loaded into a database and further expanded within the database by (i) deductive closure on HPO and GO annotations and (ii) computation of keys for each pathology (respectively gene) pair.

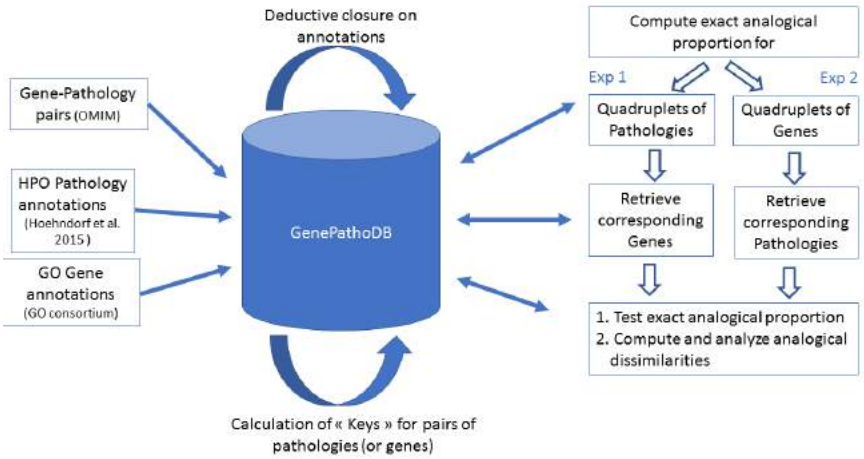



Fig. 1: Outline of the general approach for finding relevant pathology-gene pairs using analogy.

The main algorithmic difficulty was to efficiently find those quadruples $(\mathtt{P}_A, \mathtt{P}_B, \mathtt{P}_C, \mathtt{P}_D)$ of pathologies that are in analogy: the naïve algorithm is in $O(n^4)$,

where $n$ is the number of pathologies for which a gene is known. Since $n \simeq 5000$, this solution is intractable.

The idea is to use the characterization (3) of analogies based on keys. First, the data on pathologies and genes where stored in tables. A table for pathology pairs $(\mathtt{P}_A, \mathtt{P}_B)$ with their keys $\mathtt{key}(\mathtt{P}_A, \mathtt{P}_B)$ was built. A query on this table with a `GROUP BY` clause on these keys is executed. When there are several lines in a group, they correspond to two pairs $(\mathtt{P}_A, \mathtt{P}_B)$ and $(\mathtt{P}_C, \mathtt{P}_D)$ such that $\mathtt{P}_A : \mathtt{P}_B :: \mathtt{P}_C : \mathtt{P}_D$. The flat analogies are subsequently removed.

Other algorithmic difficulties were overcome in a similar way.

## 6 First Results

The results of the two experiments described in Section 4 are presented below.

For the first experiment, The number of quadruples of pathologies that are non flat analogies is $3,501$ in the chosen representation. Unfortunately, there is no quadruple of genes (related to the retrieved quadruples of pathologies) which are in analogy.

As for the second objective, a computation of $\mathtt{AD}(\mathtt{G}_A, \mathtt{G}_B, \mathtt{G}_C, \mathtt{G}_D)$ for about $3,500$ quadruples of genes chosen at random has given the following result:

$$\text{mean} = 240 \qquad \text{median} = 222 \qquad \text{standard deviation} = 111$$

For the $3,501$ quadruples of genes associated to quadruples of pathologies that are in non flat analogies, the computation of their analogical dissimilarity has given a distribution with the following features:

$$\text{mean} = 247 \qquad \text{median} = 230 \qquad \text{standard deviation} = 112$$

These basic statistics are quite similar, suggesting that in average the distributions are similar between randomly selected quadruples of genes and quadruples of genes associated with quadruples of pathologies in analogy. However close inspection of the distribution histogram reveals that a small subset of these gene quadruples (associated with pathologies in analogy) display very low values of analogical dissimilarity not reached by randomly selected gene quadruples. These quadruples are currently under investigation by experts.

The results in the second experiment, which takes the inference rule the other way round (from gene to pathology), are similar to those in the first experiment.

## 7 Conclusion and Future Work

This paper examined the following hypothesis: "If four pathologies are in analogy, is it plausible that the corresponding genes are in analogy?" With our databases, under the representation choices of pathologies and genes we adopted and within the simple proportional analogy and analogical dissimilarity framework presented in this paper, the answer is negative for the great majority of pathologies in analogy.

The results we obtained on a large scale analysis ($4,000$ $(\mathtt{P}, \mathtt{G})$ pairs) reveal that the quadruples of genes associated with quadruples of pathologies in analogy do not

display statistically different analogical dissimilarity values compared to randomly selected quadruples of genes. Nevertheless very low analogical dissimilarity values are found in a small subset of gene quadruples that are specifically associated with pathologies in analogy. Analysis of these quadruples may allow us to learn more sophisticated analogical relations on genes in order to improve the recovery of pathology-gene pairs using analogy.

Therefore, an idea for future work would be to find analogical relations on pathologies and on genes so that the inference (1) from pathologies to genes or the inference (4) from genes to pathologies work better. A way to do it would be to keep the analogical relation between pathologies defined in this paper and *learn* the analogical relation between genes. More precisely, let $\mathcal{A}_\varphi$ be an analogical proportion between genes parameterized by $\rho$ (that can be, e.g., a tuple of integers). The training set would be quadruples of genes $(\mathtt{G}_A, \mathtt{G}_B, \mathtt{G}_C, \mathtt{G}_D)$ that correspond to pathologies that are in analogy. The objective of the learning method would be to find $\rho$ such that an important proportion of the quadruples in the training set are in analogy according to $\mathcal{A}_\varphi$. Once this learning process is achieved, it is hoped that the analogical inference described by (1) with the classical analogy relation between pathologies and the analogy relation $\mathcal{A}_\varphi$ between genes provides an efficient way to mine pathology-gene pairs.

## References


1. Ando, S.I., Lepage, Y.: Linguistic structure analysis by analogy: Its efficiency. In: Proceedings of NLPRS-97, Phuket (December 1997) 401–406
2. Driel, M.A.V., Brunner, H.G.: Bioinformatics methods for identifying candidate disease genes. Hum. Genomics (2006) 429–432
3. Lavallée, J.F., Langlais, P.: Unsupervised morphological analysis by formal analogy. In: Lecture Notes in Computer Science. (2010) 8 pages
4. Lepage, Y., Denoual, É.: Purest ever example-based machine translation: detailed presentation and assessment. Machine Translation **19** (2005) 251–282
5. Lepage, Y., Lieber, J.: Case-based translation: First steps from a knowledge-light approach based on analogy to a knowledge-intensive one. In: Proceedings of the Computational Analogy Workshop at the 25th International Conference on Case-Based Reasoning (ICCBR-18), Stockholm, Sweden (August 2018)
6. Miclet, L., Bayoudh, S., Delhay, A.: Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. Journal of Artificial Intelligence Research **32** (2008) 793–824
7. Pirelli, V., Federici, S.: "Derivational" paradigms in morphonology. In: Proceedings of COLING-94. Volume I., Kyōto (August 1994) 234–240

# Towards Distributed K-NN Similarity for Scalable Case Retrieval

Shaibal Barua[1][0000-0002-7305-7169], Shahina Begum[1][0000-0002-1212-7637] and Mobyen Uddin Ahmed[1][0000-0003-3802-4721]

[1] School of Innovation, Design and Engineering, Mälardalen University, SE-72123 Västerås, Sweden
{shaibal.barua, shahina.begum, mobyen.ahmed}@mdh.se

**Abstract.** In Big data era, the demand of processing large amount of data posing several challenges. One biggest challenge is that it is no longer possible to process the data in a single machine. Similar challenges can be assumed for case-based reasoning (CBR) approach, where the size of a case library is increasing and constructed using heterogenous data sources. To deal with the challenges of big data in CBR, a distributed CBR system can be developed, where case libraries or cases are distributed over clusters. MapReduce programming framework has the facilities of parallel processing massive amount of data through a distributed system. This paper proposes a scalable case-representation and retrieval approach using distributed k-NN similarity. The proposed approach is considered to be developed using MapReduce programming framework, where cases are distributed in many clusters.

**Keywords:** Case representation, k-NN similarity, Scalable Case Retrieval, distributed CBR, big data.

## 1 Introduction

Big data is an ever-increasing field in computer science that can be used for a multitude of goals. Both new and old technologies are used to manage and process the magnitude of data these systems have. The concept of case-based reasoning (CBR) is that it uses past experiences to solve a new problem. The past experiences are stored as cases into a CBR system. Case representation is a very important issue as the efficiency of a CBR system and case retrieval process are depended on it. Thus, cases must be formatted in such a way that they resemble the problems that is intended to be solved. The challenge with the proliferation of data generation is that faster and better data management and processing is required; a CBR system should be developed that can handle the V's [1] of big data. Challenge arises with the growth of data in the case library, which means a single machine can no longer process or hold all the data. Also,

in big data paradigm, data is no longer collected from a single source, rather heterogenous and multivariate data are used to build a case library. These diverse data sources simulate a big data environment where scalability is an important issue. One strategy to deal with the problems is to partition the case library over clusters and use distributed CBR. In recent years, MapReduce, which is a programming paradigm for big data, gained popularity because of massive parallel processing capability over large distributed system using commodity hardware [2]. However, data structure for case representation needs to be designed to adapt with MapReduce. Another issue with big data is the demand of big data processing speed is very high while the amount of data that requires processing is constantly increasing.

One major steps in CBR is the retrieval of the most similar cases from a case library and usually is computationally the most demanding one with the increase of the size of a case library. Thus, the key factor for a successful CBR application in big data paradigm depends on efficient and fast case retrieval method. Over the years different methods have been proposed for efficient case library management and case retrieval, however they have been usually limited by available technology to a single machine implementation. Recent advances in network computing resources have opened new possibilities for improvement of the CBR methods and extending their use to big data problems with the use of distributed case retrieval methods. CBR uses similarity matching to retrieve most similar cases from the case library and k-NN is the most common and powerful method that has been using for CBR system. However, k-NN search and similarity matching is sequential in nature and thus become more time consuming in larger case libraries [3].

MapReduce is a software framework meant to enable writing programs that take advantage of large clusters of nodes to process large amounts of data [4]. MapReduce is based on the 'map' and 'reduce' functions commonly found in functional programming. It works by using a 'map' function to process a key/value pair which then generates a set of intermediate key/value pairs [5]. These intermediate key/value pairs are then used in a 'reduce' function that merges all the values associated with their key. This is what allows the process to be run in parallel. Each node processes its own part of the map function on the data it receives. The results are then distributed again to apply the reduce function. So, as its name implies the reduce function is always performed after the map function. In MapReduce paradigm, similarity matching can be implemented by different techniques such as numerical matching attribute-value pair, rule-based matching, and workflow-based matching [6].

This paper proposes attribute-value pair case representation approach using MapReduce for scalable and distributed CBR system. It is assumed that case library can be consisted of data coming from heterogenous sources. These diverse data sources simulate a big data environment where scalability is important. In addition, a case retrieval approach from such case representation requires parallel or distributed similarity search algorithm. Hence, a distributed k-NN similarity function of scalable case retrieval is proposed.

The rest of the paper is organized as follows: section 2 presents the background and related work, section 3 presents proposed method for case representation and case retrieval. Finally, chapter 4 summarizes the work.

## 2 Background and Related Work

According to the literature, the structuring of case representation methods can be grouped into two categories such as hierarchical representation and vector representation. Hierarchical representation is designed to have multiple layers within them to provide structure of the multitude of data being stored. Case representation in a hierarchical layer is presented in [7-9]. In [9] tree of nodes approach is proposed with two layers and each node in the tree either contains data or is a category. It can be scaled up to accommodate any number of layers beyond that and the versatility of this particular representation has been shown with an example. Assali et al. [10], present a similar case representation for ontology-based CBR systems, where cases are broken down into fundamental data points to identify important information. Attributes are referred as either complex or simple, and complex attribute can be broken down to its corresponding simple attributes. Object-oriented approach has been proposed in [11, 12] for the hierarchical case representation, which practices inheritance for objects in the cases. The vector representation is much simpler than the hierarchical representation. It is a flat representation consisting of a pair of vectors, where the first part of the pair contains data describing the case and the later part contains the solution data. This makes the representation itself minimal but external knowledge is required to understand the data structure. More on vector representation can be found in [13-15]. Another important issue of case representation is the structure of the case library. The two most common structures are a flat library using indexing and a hierarchical library [16, 17]. Both of these two approaches use some sort of indexing to place cases in the case library and the indexing can be a solution type or a weighting algorithm. Moreover, these approaches can reduce the case retrieval process while accounting for the size of the case library. Typically, markup language such as XML is used to create hierarchical library so that library structures are human readable [18]. This allows the cases to be organized after some sort of metric or weights.

Distributed and scalable information retrieval has been an active research area, where the popularity of MapReduce is increasing. Ahuja et al. [19], propose a faster data retrieval using MapReduce based on scatter and gather processing technique. The proposed technique is demonstrated with a geographically distributed data. In [20], proposes an agent-based distributed CBR tool for medical prognosis, where data is gathered from multiple sources such as different doctors or clinical units. Prasad et al. [21] has proposed a multiagent system for cooperative retrieval and cases are represented as a composition, where subcases are distributed across different agents. The authors have argued that summation of the best case may not be the result of good overall cases and hence proposes a negotiation driven case retrieval approach to dynamically resolve the problem. Other applications are also discussed on multiagent CBR systems [22-26].The two distributed memory models and case retrieval approaches such as distributed case library and composition of distributed cases are introduced in [27].

In recent years improvements on k-NN algorithm of distributed similarity matching have been proposed in several studies. Most of the proposed methods are developed using MapReduce programming framework. Rheinländer and Leser [28] have proposed a scalable similarity search approach to operate into the main memory and multicore

systems. In [29] performance of different MapReduce-based approximate k-NN similarity join approaches are investigated. Based on the experimental results the authors suggest to use Locality Sensitive Hashing approach. In [30], a novel MapReduce based algorithm, which is an index structure called similarity join tree (SJT) is proposed for distributed and scalable similarity search on partition data, where data are partitioned according to data distribution and distributes similar data into same group.

## 3 Method and Approach

In CBR, the case representation is an extremely important issue, as the cases must be formed such that they are suited for the problem the system that seeks to solve. Often problems differ in its nature and a generic system should have ways to try and solve problems that do not register a one-to-one match. Thus, the case representation requires having to somehow map problems together to create a solution. On the other hand, during the case retrieval in a CBR system, it seeks to match the most similar cases and to do that the target case is compared with every case in the case library. In a distributed environment, where case library can be generated from heterogenous data, cases can be compared more than once due to different manifestations of a case and a case is consisted of multiple attributes or features. For MapReduce based distributed environment, cases can be distributed on overlapping clusters, which is another reason of a case is compared more than once with the target case. Hence, the efficiency of the case retrieval mechanism can be reduced due to duplicate similarity matching. This section presents a distributed case representation and corresponding case retrieval mechanism using the MapReduce programming framework.

### 3.1 Approach for Case Representation

The data structure and case representation for the distributed case library is shown in Fig 1.

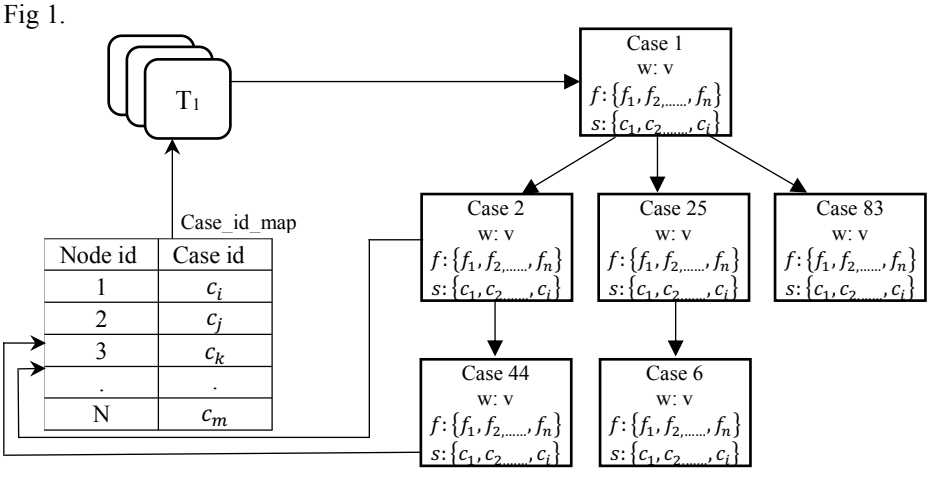



**Fig. 1.** Distributed case representation architecture.

The proposed case representation is a hierarchical structure with a tree indexing which is mapped with a Case_id_map data structure. The Case_id_map is a <key, value> data structure consists of a node id and a case id. Each case is consisted of four main parameters: case id e.g., case 1, some weight value *w*, features *f*, and a list of most similar cases *s*. Number of items in the list *s* can be limited by predefined constrain. This allows to create a case using markup language [31, 32] and each can be stored in clusters. The weight value can be a function that indicates the most used case or the important case in the case library. Thus, with some constrain the case retrieval can reduce the search space and enhance the efficiency of a distributed CBR system. An example of a case using XML and JSON is shown in Fig 2.

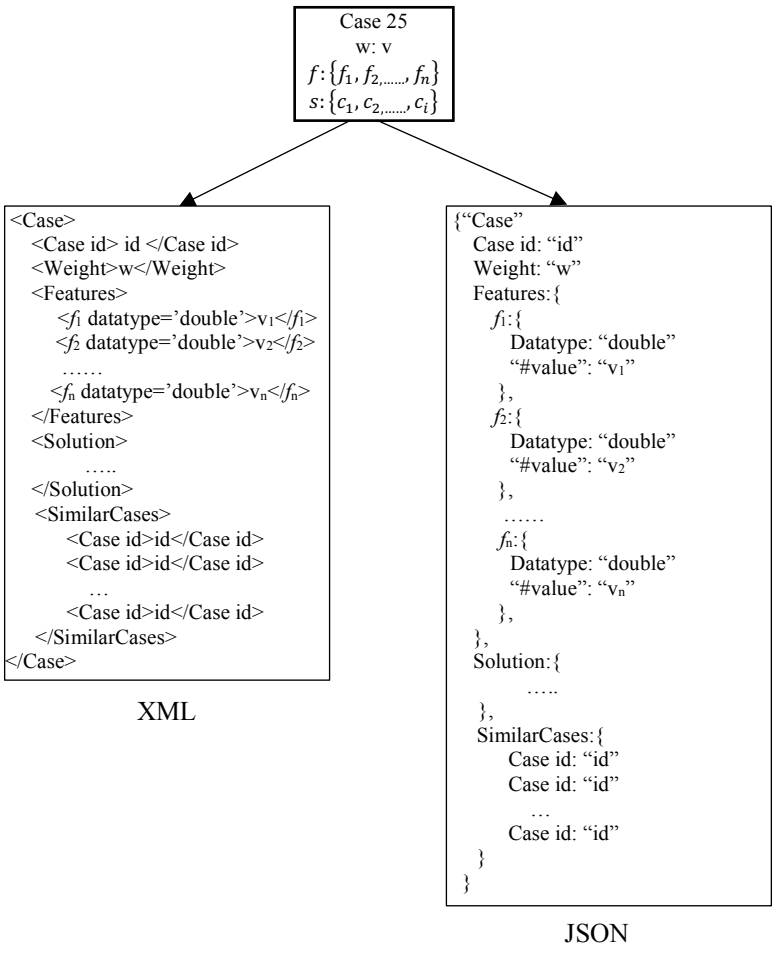



**Fig. 2.** A generic representation of a case using XML and JSON.

A new case can be initiated in any node, say $T_1$, as shown in Fig 1. This node will be considered as the master node in the cluster and rest of the nodes are client nodes.

The master node will communicate with the client nodes by sending the new case and similarity function and retrieve the k-most similar cases from each client nodes. Each node may have different similarity function depending of the case and the data type, hence it requires to send the similarity function along with the case. A typical representation is shown in Fig 3.

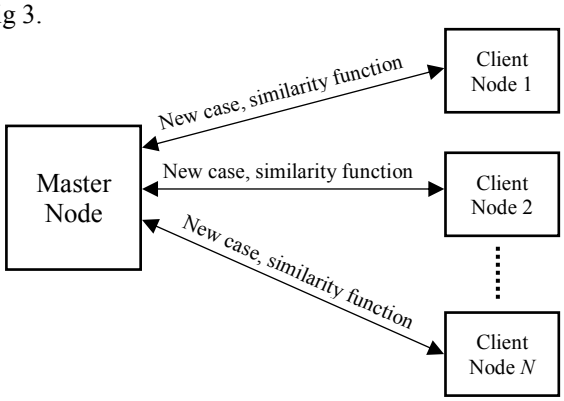



**Fig. 3.** Communications among the nodes.

### 3.2 Proposed Case retrieval Algorithm

In a multi-cluster distributed environment cases are distributed among the clusters and for fault handling often redundant copies are made across the clusters. Hence, during case retrieval, there is a chance that similar cases are compared redundantly, which reduce the efficiency. As MapReduce based approach is proposed for case representation, the efficiency of case retrieval can be speed up using the MapReduce framework. Also, big data platform like Apache Spark can even use in-memory computation for faster execution.

The proposed approach is inspired by the work of Rheinländer and Leser [28], where k-NN similarity search algorithm is proposed for large a string sequence. Here, for a new target case, at first the algorithm is searching at the root node. The search will traverse only the nodes that satisfies some threshold of weight $w$ for each case such that distance of the target case and the selected case of that node is greater than or equal to the threshold value. This will filter the top most $k$ cases and during the traversal for each node corresponding most similar cases of that selected case will be skipped, which reduces the search space. However, the distance function is actually a pair-wise similarity computation, which in general is a Cartesian product of entities or cases. In big data scenario, the time complexity of Cartesian product is very high. To reduce the search space similar cases or class of cases are grouped into clusters. Thus, in every cluster similarity matching is performed on pairs of cases presented within the cluster.

As shown in Fig 3, the master node will retrieve $k$ most similar cases from each client nodes. Therefore, the master node should sort out these selected cases and save the final $k$ most similar cases and it also needs to apply the weight function based on the selected cases. As mentioned, the task at master node is to create and save the new case for the future work. The transformation and an XML representation of the combination of all selected cases is shown in Fig 4.

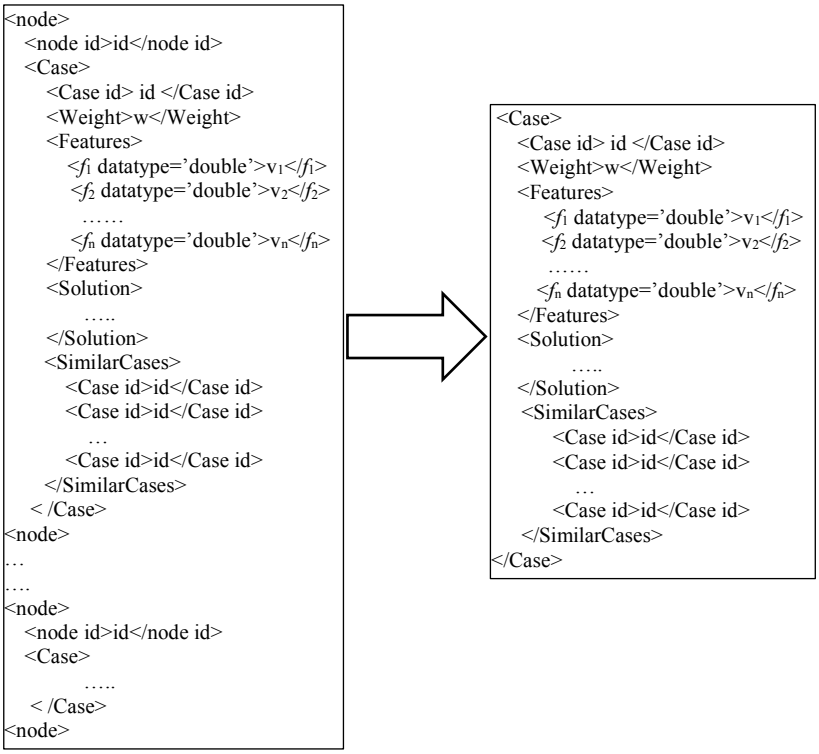



**Fig. 4.** Transformation and creating new case structure using selected cases.

## 4 Conclusion and Feature Work

The CBR systems need to be enhanced and/or scaled up to be adapted to the V's of big data. This paper proposes a distributed and scalable approach for case representation and case retrieval on MapReduce programming framework. In CBR, the cases can be represented in different ways such as object-oriented or tree representations [33]. These are traditional ways of case representation that are based on known concepts, which represents an action and a case is defined by the consequence of that action. Moreover cases can be divided into major categories such as traditional and ontological methods [34]. The traditional representations lack cohesive information of the case library, while ontological representations have ways to relate cases together. Both of these categories have advantages and disadvantages for example, ontological representations create an overhead in maintaining the relationships between cases. Cases can include references to similar cases which then can be used to create clusters of similar cases. However, if a case is removed or edited or all of its references are updated then problem can arise in maintaining the case library. Although the traditional representations have very little overhead, it cannot contain information about the case for example, variables and the solution. These are the challenges of case representation for scalable and distributed CBR and an optimal solution is required that can enhance the efficiency of scalable

case retrieval. On the other hand, simple search using a traditional index structure may return too many results or no result. The k-NN similarity is the most common algorithm applied in CBR systems for case retrieval. However, with big data k-NN similarity matching become tedious process. The result of this method also depends on data quality and data sources which is a challenge to confirm in big data. For heterogenous data, data size grows and the relationship between these various sources of data, and the need for similarity matching is also increasing. MapReduce is widely adopted technology for big data and is considered as an efficient technology for scalable and distributed systems. In a distributed environment using MapReduce, centralize indexing for similarity searching cannot be used and modification and adaption is required. The proposed case representation uses a data structure that is scalable and can be distributed over many clusters using MapReduce. Moreover, the k-NN similarity search can be implemented on big data framework such as Spark that can support in memory and parallel computation. Thus, it can provide both efficient and fast computation on multi-cluster or cloud environment. In future, the approach will be evaluated based on the performance of storing data and the global cost of retrieving cases from distributed case libraries. With increasing size of the case libraries two factors such as load factor and replication factor can be measured to indicate the performance of storing data. CPU cost of computing distance or similarity of the cases and the communication cost among the nodes for case retrieval will be considered measuring the global cost. Last but not least, the proposed approach can be applicable in textual-CBR system, multi-layer complex product system, application such as educational hypermedia [35], etc.

**Acknowledgement**

The authors would like to acknowledge the contributions of Carl Larsson and Kostadin Rajkovic, two thesis students who are working within the scope of this research.

**References**

1. Fan, J., F. Han, and H. Liu, *Challenges of Big Data analysis*. National Science Review, 2014. **1**(2): p. 293-314.
2. Dean, J. and S. Ghemawat, *MapReduce: simplified data processing on large clusters.* Commun. ACM, 2008. **51**(1): p. 107-113.
3. Zhong, S., X. Xie, and L. Lin, *Two-layer random forests model for case reuse in case-based reasoning.* Expert Systems with Applications, 2015. **42**(24): p. 9412-9425.
4. Dean, J. and S. Ghemawat, *MapReduce: a flexible data processing tool.* Commun. ACM, 2010. **53**(1): p. 72-77.
5. Srinivasa, K.G. and A.K. Muppalla, *Getting Started with Hadoop*, in *Guide to High Performance Distributed Computing: Case Studies with Hadoop, Scalding and Spark*. 2015, Springer International Publishing: Cham. p. 33-72.
6. Köpcke, H. and E. Rahm, *Frameworks for entity matching: A comparison.* Data Knowl. Eng., 2010. **69**(2): p. 197-210.
7. Zhang, H. and G. Dai, *Research on traffic decision making method based on image analysis case based reasoning.* Optik, 2018. **158**: p. 908-914.

8. Liu, L., et al. *Research on case representation of case-based reasoning approaches for electric power engineering design*. in *Power System Technology, 1998. Proceedings. POWERCON '98. 1998 International Conference on*. 1998.
9. Zhou, M., et al., *Representing and matching simulation cases: A case-based reasoning approach.* Computers & Industrial Engineering, 2010. **59**(1): p. 115-125.
10. Abou Assali, A., D. Lenne, and B. Debray. *Case Retrieval in Ontology-Based CBR Systems*. 2009. Berlin, Heidelberg: Springer Berlin Heidelberg.
11. Bergmann, R. and A. Stahl. *Similarity measures for object-oriented case representations*. 1998. Berlin, Heidelberg: Springer Berlin Heidelberg.
12. Quan, Q., Z. Rui, and C. Hong-Yi. *Object-oriented Case Representation and Its Application in IDS*. in *2009 Eighth IEEE/ACIS International Conference on Computer and Information Science*. 2009.
13. Wang, X. and J. Dong. *Fuzzy based similarity adjustment of case retrieval process in CBR system for BOF oxygen volume control*. in *2013 Sixth International Conference on Advanced Computational Intelligence (ICACI)*. 2013.
14. Guo, S., T. Li, and K. Zhou. *An Improved Case Retrieval Method For the Production Manufacturing Process of Aluminum Electrolysis*. in *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*. 2017.
15. Hongru, L., H. Xiaochen, and Z. Yubin. *Research on Case Representation Method Based on Image Information*. in *2013 Sixth International Symposium on Computational Intelligence and Design*. 2013.
16. Kwong, C.K., G.F. Smith, and W.S. Lau, *Application of case based reasoning injection moulding.* Journal of Materials Processing Technology, 1997. **63**(1): p. 463-467.
17. Shang, F., et al. *Research on Scalable Case Representation and Its Retrieval Based on Description Logic*. in *2009 Second International Symposium on Knowledge Acquisition and Modeling*. 2009.
18. Lodhi, I.Y., et al. *Optimizing retrieval process and using neural networks for adaptation process in case based reasoning systems*. in *7th International Multi Topic Conference, 2003. INMIC 2003.* 2003.
19. Ahuja, R., et al., *FAST DATA RETRIEVAL USING MAP REDUCE: A CASE STUDY.* Journal of High Performance Computing, 2010. **1**(1): p. 01-05.
20. Pla, A., et al., *eXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis.* Decision Support Systems, 2013. **54**(3): p. 1499-1510.
21. Prasad, M.V.N., V.R. Lesser, and S.E. Lander, *Retrieval and Reasoning in Distributed Case Bases1.* Journal of Visual Communication and Image Representation, 1996. **7**(1): p. 74-87.
22. Myllymäki, P. and H. Tirri. *Massively parallel case-based reasoning with probabilistic similarity metrics*. 1994. Berlin, Heidelberg: Springer Berlin Heidelberg.
23. Plaza, E., J.L. Arcos, and F. Martín. *Cooperative Case-based Reasoning*. 1997. Berlin, Heidelberg: Springer Berlin Heidelberg.
24. Tran, H.M. and J. Schönwälder, *DisCaRia-Distributed Case-Based Reasoning System for Fault Management.* IEEE Transactions on Network and Service Management, 2015. **12**(4): p. 540-553.

25. Plaza, E. and L. McGinty, *Distributed case-based reasoning.* The Knowledge Engineering Review, 2006. **20**(3): p. 261-265.
26. Rishi, O., R. Govil, and M. Sinha, *Distributed case based reasoning for intelligent tutoring system: an agent based student modeling paradigm.* environment, 2007. **2**: p. 9.
27. Lenz, M. *Preparing Case Retrieval Nets for Distributed Processing*. in *Proc. of the Workshop on Concurrency, Specification and Programming (CS&P-96), Humboldt University, Berlin*. 1996.
28. Rheinländer, A. and U. Leser. *Scalable Sequence Similarity Search and Join in Main Memory on Multi-cores*. 2012. Berlin, Heidelberg: Springer Berlin Heidelberg.
29. Čech, P., et al. *Comparing MapReduce-Based k-NN Similarity Joins on Hadoop for High-Dimensional Data*. 2017. Cham: Springer International Publishing.
30. Liu, W., Y. Shen, and P. Wang, *An efficient MapReduce algorithm for similarity join in metric spaces.* The Journal of Supercomputing, 2016. **72**(3): p. 1179-1200.
31. Asemota, E., et al., *DEFINING CASE BASED REASONING CASES WITH XML.* 2007.
32. Hayes, C., P. Cunningham, and M. Doyle. *Distributed cbr using xml*. in *Proceedings of the KI-98 Workshop on Intelligent Systems and Electronic Commerce*. 1998.
33. Michael, M.R. and O.W. Rosina, *Case-Based Reasoning A Textbook*. First ed. 2013: Springer-Verlag Berlin Heidelberg. 546.
34. El-Sappagh, S.H. and M. Elmogy, *Case based reasoning: case representation methodologies.* International Journal of Advanced Computer Science & Applications, 2015. **1**(6): p. 192-208.
35. Chorfi, H. and M. Jemni. *XML Based CBR for Adaptive Educational Hypermedia*. in *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)*. 2006.

# Knowledge-Based Systems in Computational Design and Media (KBS-CDM)

Viktor Eisenstadt[1]

[1]University of Hildesheim, Institute of Computer Science
Intelligent Information Systems Lab (IIS)
Samelsonplatz 1, 31141 Hildesheim, Germany
viktor.eisenstadt@uni-hildesheim.de

## 1 Preface

Case-based reasoning as well as other knowledge-based methods and approaches have a rich tradition in design and related creative domains. Systems developed for such domains usually deal with knowledge-intensive design and media data and complex algorithms of different types for reasoning during configuration and decision support tasks. Retrieval and retention of design solutions are among the core functionalities of such approaches and have a long history in a multitude of research projects as well.

The ICCBR workshop *Knowledge-Based Systems in Computational Design and Media* (KBS-CDM) is intended to continue this tradition of CBR in creative and interaction-based domains in order to explore the newest directions and trends of this research field. Therefore, the main aim of the workshop is to track and examine the current trends in CBR as well as other knowledge-based methods and approaches in research and development areas of design, media, and creativity.

## 2 Organizers

**Viktor Eisenstadt** – (Chair) *University of Hildesheim*
**Klaus-Dieter Althoff** – (Chair) *University of Hildesheim, DFKI*
**Ashok Goel** – *Georgia Institute of Technology*
**Christopher McComb** – *Pennsylvania State University*
**Christoph Langenhan** – *Technical University of Munich*
**Seong-Ki Lee** – *Technical University of Munich*

# Reasoning about Time in Case-Based Reasoning (RATIC)

## Workshop at the Twenty-Sixth International Conference on Case-Based Reasoning (ICCBR 2018)

**Stockholm, Sweden July 2018**

Odd Erik Gundersen[1] and Miltos Petridis[2]

[1]Norwegian University of Science and Technology (NTNU), Norway
odderik@ntnu.no
[2]Middlesex University London, UK
m.petridis@mdx.ac.uk

## 1 Preface

The workshop is dedicated to time in case-based reasoning and how time is dealt with in all aspects of it. The literature on case-based reasoning (CBR) that takes time into account is broad. Still, there are aspects that have not been given much consideration. Reasoning about time drives the complexity of AI systems, but with the increasing amount of streaming and event-based data, this complexity has to be dealt with, also in CBR. The aim of this series of workshops is to refocus the CBR community's attention to temporal reasoning, as the focus has moved away in past years, even though the number of temporal CBR applications is increasing. Several open problems exist in temporal CBR, and these contain among others temporal revise and CBR on data streams.

Four previous workshops on applying case-based reasoning to temporal data have been organised at ICCBR. This workshop is a continuation - in spirit - to the workshops on applying CBR to time-series prediction that was organized in 2003 and 2004, and it is a direct descendant of the RATIC 2014 and RATIC 2016 workshops.

Following from an open call for submissions, and a peer-review process by the programme committee, the workshop has two papers that were selected for publication.

In Textual Summarisation of time series using Case-Based Reasoning: A case study, Dubey et al. propose an end-to-end Case-Based Reasoning (CBR) approach for generating the textual summaries of time series data in weather domain. This approach is intended to assist the user in decision making to generate the summary of a given time series. Empirical results presented show the effectiveness of the approach in generating suggestions very close to human-authored summaries in a vast majority of cases.

Petridis et al. propose suitable workflow similarity metrics for developing efficient performance measures for the rail industry in Predictive Process Mining Using a Hybrid

CBR Approach for the Rail Transport Industry. The approach proposed uses extensive business process workflow pattern analysis based on Case-based Reasoning. An evaluation of the approach and a number of modelling experiments are presented, that show that the approach can provide a sound basis for the effective and useful analysis of operational sensor data from train Journeys.

The goal for this workshop is to emphasize the need for the CBR community to investigate problems related to temporal reasoning, as we firmly believe that reasoning about time is a central challenge in CBR and that it deserves more attention from the broader community. The papers published in this workshop proceeding show that temporal case-based reasoning still has a broad set of challenges that need further investigation in a variety of domains. We look forward to see final results of these initial developments in future ICCBR conferences.

Stockholm, Sweden
July 2018

Odd Erik Gundersen
Miltos Petridis

### 1.1 Co-Organisers

- Odd Erik Gundersen, Norwegian University of Science and Technology (NTNU), Norway
- Miltos Petridis, Middlesex University London, UK
- Bjorn Magnus Mathiesen, Norwegian University of Science and Technology (NTNU), Norway

### 1.2 Program Committee members

- Stelios Kapetanakis, Univeristy of Brighton
- Mirjam Minor, Johann Wolfgang Goethe-Universitet Frankfurt
- Stefania Montani, Università del Piemonte Orientale A. Avogadro

# Textual Summarization of Time Series using Case-based Reasoning: A Case Study

Neha Dubey, Sutanu Chakraborti, Deepak Khemani

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

**Abstract.** In this paper, we propose an end-to-end Case-Based Reasoning (CBR) approach for generating the textual summaries of time series data in the weather domain. Our approach is intended to assist the user in decision making to generate the summary of a given time series. For that, the approach first generates appropriate abstractions of a time series and then generates the textual summary for each of the abstractions. Empirical results show the effectiveness of the approach in generating suggestions very close to human-authored summaries in the majority of the cases.

**Keywords:** Time Series, Natural Language Generation, CBR.

## 1 Introduction

Data-to-text generation is a subfield of Natural Language Generation(NLG), which deals with generating the textual descriptions of non-linguistic data sources such as time series, and event logs. The subtasks in an NLG system involves selecting the relevant information from input data and organizing it coherently (content selection), making relevant choices to express the relevant information (micro planning), and rendering the information in the output text (surface realization) [6]. In this work, we focus on summarization of time series using text in weather domain. Textual summarization of a time series requires an understanding of the time series in the context of the underlying process that generates the time series as well as the end user's requirements. For example, an expert might identify significant patterns in a time series based on her experience and provide an explanation for the same, which a non-expert might find hard to arrive at, by visual inspection alone. Furthermore, depending on the end user, the same time series can have different summaries, for example, summary for a sports news channel is different from one meant for farmers, and both are different from the forecast meant for trekkers in mountains.

Content selection in time series summarization involve finding the appropriate level of abstraction for a time series, which includes selecting change points in a time series. These change points along with the trend information can summarize the time series data. Once this is done, the later stages of an NLG system can choose appropriate linguistic realization to describe the information. One

significant issue in doing so is that of lexical choice. For example, an increasing trend in a time series can be described using *increasing*, *gradually increasing*, or *rising*. This, in particular, makes the evaluation of an NLG system hard as the there can be multiple correct textual summaries for a given time series.

In this paper, we propose an end-to-end Case-Based Reasoning (CBR) approach to assist the user in generating the textual summary of a time series in the weather domain. CBR works by recalling past experiences and is based on the premise that similar problems recur and have similar solutions. In our case, the experiences in the case library are the time series and their corresponding textual summaries. The incoming new time series is matched against the cases in the case library to retrieve relevant similar cases and the textual summary of the relevant cases is reused to generate the summary for the new time series.

In the past, researchers have proposed other approaches for generating weather forecast text summary for a time series. These systems can be mainly categorized in two ways: knowledge-rich (top-down) [11], and knowledge-light (bottom-up or data-driven) [3, 4]. While it is expensive and time-consuming to acquire knowledge in top-down system, knowledge-light systems need large and reliable parallel corpora of input and output text. The system by Sripada et al. [11] is a top-down system that generates the forecast text by following the pipeline architecture of an NLG system, which includes all the subtasks in NLG. Most of the other systems that generate weather forecast text are not complete NLG systems as these systems focus only on micro-planning and realization and not on content selection [3, 4].

To the best of our knowledge, ours is the first attempt to propose an end-to-end CBR system that performs all subtasks in an NLG pipeline from content selection to text realization. In addition to that, the proposed system generates multiple summaries rather than just one to account for the fact that there can be multiple correct summaries for a given time series. This is consistent with our goal of assisting a human user, who can select one of the outputs and, if necessary post-edit to produce the final summary.

## 2 Our Approach

The architecture of our system follows the typical pipeline architecture of an NLG system: Content Generation ($CBR_1$) and Text Generation ($CBR_2$). The first component decides the abstraction level of a wind time series, which is used to generate the intermediate representation of that time series and the second component generates the textual description of the given intermediate representation. Figure 1 shows the system's architecture. Two components each of which is a CBR system are illustrated in Figures 2 and 3, respectively.

**Time Series Summarization** Weather data consists of day-wise wind time series and the corresponding textual summaries. The time series is a multivariate time series as it involves two channels, namely, wind speed and wind direction. A sample time series in weather domain is shown as input in Figure 1. The human-written summary for the given time series is "S 02-06 increasing 16-20", which

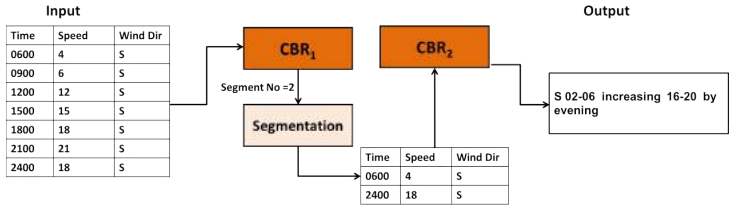



Fig. 1: Architecture for text generation system

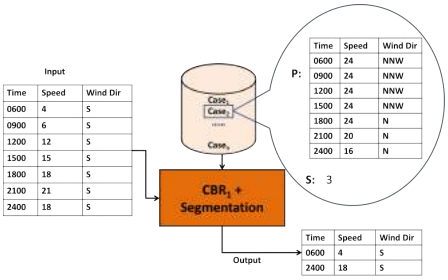



Fig. 2: $CBR_1$ Case representation
Problem: Wind time series; Solution: Number of segments

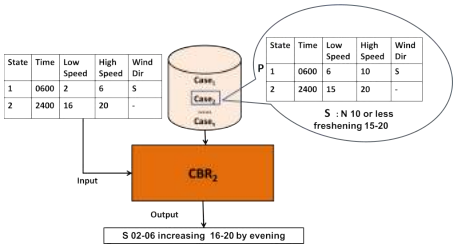



Fig. 3: $CBR_2$ Case representation
Problem: Intermediate representation of the wind time series; Solution:Weather forecast text

describes the change in wind speed as increasing from 4 (02-06) knots to 18 (16-20) knots. These weather forecasts are produced to assist offshore oil company staff in making right decisions for the tasks that depend upon the weather, for example, to carry out the supply boat operation.

Content generation from time series involves a trade-off between minimizing the number of change points reported and maximizing the faithfulness of its approximation. The error in approximation can be reduced by increasing the length of the summary (by increasing the number of change points reported). In the proposed system, we predict the number of change points in a time series using $CBR_1$ and then generate the approximation to the time series that minimizes the approximation error. To approximate a given time series, we use optimal segmentation algorithm [2]. Segmentation is the process of approximating a time series with straight line segments. For example, Figure 4 shows the segmentation of a time series with five segments. Given the number of segments, the optimal segmentation algorithm globally minimizes the error of approximation.

In the weather domain, a calm day with fewer fluctuations may need a lower number of segments than a bad weather day. This is because the more the weather is volatile, the more is the number of segments that will be abstracted out. For example, the text in Figure 1 has two wind states (S, 02-06) and (-, 16-20). The count of wind states (as the number of segments) is used as input to the segmentation algorithm to generate the intermediate representation.

### 2.1 Generating Intermediate Representations

In the weather domain, based on the observation that days with similar weather conditions have similar forecast text and the similar level of abstraction in time

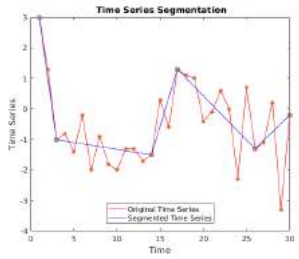



Fig. 4: Segmentation example

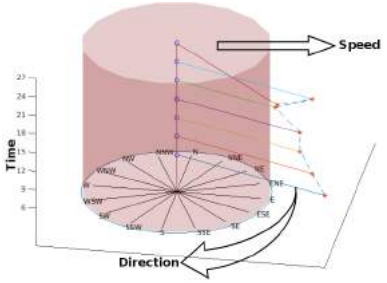



Fig. 5: Wind vector representation of wind time series

series, we hypothesize that *similar wind time series have the similar number of segments in the forecast text*. Thus, the case used in $CBR_1$ has wind time series as the problem component and number of segments as the solution component. The representation of wind time series is the wind vector (wind speed as magnitude and wind direction as direction of the wind vector) representation as shown in Figure 5. This choice of representation of time series allows us to consider two univariate time series: wind speed and wind direction into a single time series with a wind vector at each time stamp.

**Retrieval** In an NLG system, for a time series, the relevance of a case to a query depends upon the end user requirements. For example, a wind time series on a calm day can have similar behaviour in terms of patterns on a bad weather day, however, both have entirely different interpretations with respect to the end user. Thus, a case is relevant to a query if it has the patterns similar to that in the query time series and the similar error tolerated in approximating a case and the query. To retrieve the cases with patterns similar to the query, we use the dynamic time warping (DTW) [8] on wind vector time series. In order to retrieve the cases with the similar error of approximation, we define a distance measure called $Error_{distance}(query, case)$ between a query and a case.

DTW aligns two time series by scaling/shrinking on time axis. In our case, since the time series have wind vector representation, the equation of DTW similarity for two time series $T_1$ and $T_2$ can be modified as :

$$DTW(i,j) = \min\{DTW(i-1,j), DTW(i,j-1), DTW(i-1,j-1)\} + vectordist(i,j) \quad (1)$$

where $DTW(1,1)$=$vectordistance(1,1)$ is the distance between first point of both time series, and $i$ and $j$ are the time indices in time series $T_1$ and $T_2$, respectively. The local distance measure $vectordist$ in our case is defined as $vectordist(i,j) = \sum s^2_{T_{1i}} + s^2_{T_{2i}} \leftarrow 2\,\emptyset\, cos(d_{T_{1i}} \leftarrow d_{T_{2i}})$, where $s_{T_{1i}}$ and $d_{T_{1i}}$ denote speed and direction at time $i$ of time series $T_1$. Thus, the $DTW$ distance between a query and a case is $dist_{dtw}(query, case) = DTW(n,n)$,where $n$ is the length of time series.

$Error_{distance}(query, case)$ denotes that a case is relevant to a query if both case and query have the similar error of approximation for the given choice of segmentation. To define $Error_{distance}(query, case)$ we define two terms: the error of approximation of a query, i.e., $Error_{query}$, and error of approximation of a case, i.e., $Error_{case}$. $Error_{query}$ with respect to a case is the error between query time series and its segmented intermediate representation when the query is seg-

mented with the segment number as in the case. $Error_{case}$ is the error between the problem component of a case (wind time series) and its intermediate representation as used in generating the textual summary of the case. Thus, $Error_{case}$ gives the error as tolerated by a human forecaster while writing the textual summary of a time series. To get the intermediate representation for a time series corresponding to the forecast summary, we reconstruct a time series from the text by using text to time series mapping, where each entry in the text is mapped to some entry in the time series. For example, the forecast text in Figure 1 contains two entries (02-06, S), and (-,16-20) corresponding to the times 6:00, and 24:00 in the time series, respectively. The rest of the time series values between 6:00 and 24:00 are obtained by interpolating between these two values in the text. Now, we define the $Error_{distance}(query, case)$ as:

**if** $Error_{case} < Error_{query}$(Case is relevant to query) **then**
$Error_{distance}(query, case) = \alpha\, \emptyset\, |Error_{query} \leftarrow Error_{case}|$
**else**
$Error_{distance}(query, case) = \exp(\beta\, \emptyset\, |Error_{query} \leftarrow Error_{case}|)$
**end if**
The parameter $\alpha, \beta$ are set using cross validation.

Finally, the total distance between a case and query can be defined as

$$dist(query, case) = dist_{dtw}(query, case) + Error_{distance}(query, case) \quad (2)$$

The corresponding similarity can be defined as$1/(1 + dist(query, case))$. At the end, the cases with maximum similarity with the query are retrieved.

**Multiple Representations of a Query** Typically, an expert forms an "overall" view of a time series using her experiences in the process of generating content for it. These views are often tacit and can vary across experts. Further, each of the intermediate abstractions of a time series can be used to generate a textual summary. Therefore, in the proposed approach, the $CBR_1$ system generates multiple appropriate abstractions of a time series, each with a certain confidence value. Next, the $CBR_2$ generates the textual summary for each of these abstractions of a time series.

**Confidence Scores** Let the retrieved cases for a query be the set $C_1, C_2, C_3, .., C_m$ with the corresponding segment counts as $k_1, k_2, k_3, ..., k_m; \forall k_i < K$, where $K$ is the maximum segment count possible for a time series. Let the similarity of retrieved cases in the set with be $s_1, s_2, s_3, ..., s_m$, then the confidence associated with each representation of query becomes the weighted mean of the segment counts in the retrieved cases, i.e., for $k = 1, 2, 3, ...K$

$$confidence(k) = \frac{\sum_{i=1}^{m} s_i * I\{k_i = k\}}{\sum_{i=1}^{m} s_i} \quad (3)$$

where $I\{k_i = k\}$ is an indicator function, which is 1 when $k_i$ is equal to $k$, else 0. For each segment count $k$, for which the confidence score is greater than a threshold value, we generate the intermediate representation. For that, we use optimal segmentation algorithm by Bellman [2]. The algorithm takes the input as the number of segments and outputs the approximated (segmented) time series which has the least possible error of approximation. Let the set of such query

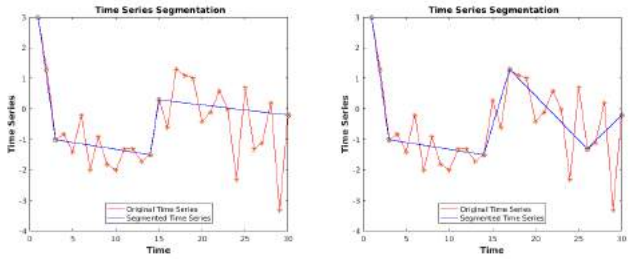



Fig. 6: Time series representations with multiple levels of abstractions

representations be $Q_{=}\{Q_k\}$, where $k < K$. For illustration, Figure 6 shows time series representations for a given time series, each with four and five segments.

### 2.2 $CBR_2$: Text Generation

Once we have the intermediate representations for a time series, we generate the forecast text corresponding to each representation using a similar approach as in [1]. The case library used in this step as shown in Figure 3, with the intermediate representation of time series as the problem side of a case and the textual summary as the solution side. Thus, for each of the query representations in $Q_{=}\{Q_k\}$, where $k < K$, the approach finds the cases which have exact patterns in speed and direction as in the text. The patterns for speed are *Increasing, Decreasing, and Stable*, and for direction *Backing(anticlockwise) and Veering(clockwise)*. For example, for query representation $Q_2$ with two segments, we retrieve those cases where the number of segments in the text is two. Next, among these cases, the cases are retrieved in which the patterns in speed and direction are same as the patterns in the query representation $Q_2$. Finally, among the retrieved cases with exact patterns, cases with the closest distance to the query are retrieved. The distance is the average vector distance between the query representation $Q_k$ and a case $C$. This distance can be defined as $dist(Q_k, C) = \sum_{i=1}^{k} vectordist(Q_{ki}, C_i)/k$, where $i$ an $k$ are the time index and the length of the time series $Q_k$ and $C$, respectively. Let the set of retrieved cases for each query representation in $Q_k$ be the set $C_k = \{C_1, C_2, ..., C_m\}$ and the corresponding similarity with the query representations be $\{sim_1, sim_2, ..., sim_m\}$, where $sim_j = 1/(1 + dist(Q_k, C_j))$. Finally, all the cases retrieved for all query representations can be denoted as $Candidate\_Cases = \bigcup_{i=1}^{k} C_k$, where $k < K$.

**Ranking of candidate cases** The final ranking of the cases is done by using confidence score for the query representation for which the case is retrieved and the similarity of the case with that query representation. Thus, the final relevance score of a case $C_j \in Candidate\_Cases$ with respect to a query is $confidence(k) * sim_j$.

**Text generation** Among the top relevant cases in the ranked list of candidate cases $Candidate\_Cases$, the solution component of the cases, i.e., the textual summary is reused to generate the final textual summary for each query representation. Since the patterns in the case and query are same, we replace the

corresponding speed and directions in the retrieved text to generate the summary for the query representation. Figure 3 for $CBR_2$ shows the example of the text reuse as the text in the case *N 10 or less freshening 15-20 by evening* is reused to generate the summary of the incoming time series as *S 02-06 freshening 16-20 by evening* by replacing (N, 10 or less), (-,15-20) with (S, 02-06), and (-,16-20), respectively.

## 3 Experiments

We have used the SUMTIME-MAUSAM parallel corpus [11] of 1045 numerical weather data and human written forecasts where we take only wind time series. All results are obtained by performing 3-fold cross-validation with 90-10 split.
**Evaluation Measure** *1. Modiffed Edit Distance* Ideally, the evaluation of an NLG system is either human evaluation in the form of ratings given by humans for text quality or task-based evaluation(extrinsic) when the system is deployed in the real world. However, human-based evaluation is costly and not always feasible in absence of experts. Therefore, we measure the performance of our system by measuring how semantically close the generated text is to the actual human-authored summary by using modified edit distance measure. Metric-based evaluations (NIST, ROGUE) are not appropriate for our approach since we have only one reference text for a generated text.

The Levenshtein distance between two texts, $Text1$ and $Text2$ with lengths $n$ and $m$, respectively can be defined as [5]:

$$L(i,j) = \min\{L(i \leftarrow 1, j) + 1, L(i, j \leftarrow 1) + 1, L(i \leftarrow 1, j \leftarrow 1) + c_I(a_i, b_j)\} \quad (4)$$

where $0 \leq i \leq n$, $0 \leq j \leq m$, and the Indicator function $c_I(a_i, b_j)$ can be defined as $c(a_i, b_j) = \{1 - sim(a_i, b_j)$ where $sim(a_i, b_j)$ is the similarity between words $a_i$ and $b_j$ in $Text1$ and $Text2$, respectively.

We modify the above measure in our case, instead of words in the text, we take semantic units in the text like speed, direction, patterns in speed and direction, time phrase. Here, we use the pre-processed parsed text available with the parallel corpus [11] to remove any other information present. Next, the similarity between these semantic units can be defined as follows:
*Similarity between two values of wind speed and direction* In weather forecasts, forecasters write ranges instead of numbers from time series, for example, "20-25" instead of 22 knots for a value of speed. Since our system selects points from time series, it generates the single number for the value of the speed. Therefore, we calculate the similarity between the single value of speed with the range in reference text as the $1 -$ distance from mid-value in the range. For direction, if it is similar to the ground truth text, the value of similarity is 1 or else it is 0.
*Pattern Similarity* Patterns in speed and direction can be summarized using similar words, for example, a rising pattern in a time series can be described using *Increasing, Rising, Freshening*. These choices are author dependent [7]. For illustration, we use synonym similarity values as shown in Table 1. These values, however, should be given by domain experts.

*Time phrase Similarity* For time phrase matching, we use the Jaccard similarity (common words divided by total words) between the generated time phrase and the time phrase present in ground truth text. We believe that this case can be further improved by knowing how authors use a time phrase, for example, in some cases, "0900" is used as "morning" while in some cases as "midday". Using the above similarities, the edit distance between the generated text and the ground truth text can be calculated using equation 4 where $a_i$ and $b_j$ are semantic units in generated text and the ground truth text, respectively. Next, the similarity between the generated text and ground truth text can be defined as $sim(text_{generated}, text_{groundtruth}) = 1 - L(i,j)/max(n,m)$ where $n$ and $m$ are numbers of semantic units in generated text and in the ground truth text, respectively and $0 \leq i \leq n$, $0 \leq j \leq m$.
*2. Relevance Factor: Evaluation of generated multiple output texts* We define the relevance of a generated text as follows:

**if** $sim(text_{generated}, text_{groundtruth}) > Threshold$ **then**
  $relevance(text_{generated}) = 1$
**else**
  $relevance(text_{generated} = 0$
**end if**

For a query $Q$, the system generates a ranked list of texts, If any of the generated texts in the ranked list is relevant with respect to the ground truth, we say the ranked list is relevant. Over a set of queries, the Relevance Factor can be defined as the average number of the queries for which the $Ranked_{list}$ is relevant.
*3. Mean Average Precision (MAP): Evaluation of generated multiple output texts* The average precision for a single query $Q$ is the mean of the precision obtained in the $Ranked_{list}$. The mean average precision for a set of queries is the mean of the average precision scores of each query.
**Experiment Design** Within our knowledge, there is no existing work for direct comparison with our system. The earlier approach by Sripada et al. [10] evaluated using post edit data, where forecasters edited the generated text. The counts of the edits were used for measuring the performance of the system. Therefore, we compare our system with the following configurations: First, we change the similarity measure in the first module, i.e., $CBR_1$ to generate the intermediate representation of a query time series as shown in Figure 2. The evaluation measure for $CBR_1$ is the accuracy of correctly predicting the number of segments. Second, while keeping the configuration of $CBR_1$ fixed, we change the output configuration of second CBR, i.e., $CBR_2$: First, when the system generates one textual summary; Second, when the system generates a ranked list of summaries. To generate the single output for a query time series $Q$, we take the estimated segment count $k$ by $CBR_1$ and segment the query time series, resulting in an intermediate representation $Q_k$. Now, input to the second system is $Q_k$, which outputs the textual summary for $Q_k$. To generate the multiple outputs of a query time series $Q$, we generate different segmented version of $Q$, $\{Q_k\}$, where $k < K$, and the confidence value for each representation is provided by $CBR_1$ using equation 3. Next, $CBR_2$ generates the textual summaries for each of the representations. The evaluation measures used here are

Table 1: Synonyms Similarity

| Word1\Word 2 | Increasing | Rising | Freshening |
|---|---|---|---|
| Increasing | 1 | 0.8 | 0.8 |
| Rising | | 1 | 0.8 |
| Freshening | | | 1 |

Table 2: Result for various configurations of the CBR system

| Sl No. | CBR 1 Configurations | Accuracy of Segment Prediction (%) | CBR 2 Configurations | | | |
|---|---|---|---|---|---|---|
| | | | Single Output Text | | Multiple Output Text | |
| | Similarity | | MAP Score | Relevance Factor | MAP Score | Relevance Factor |
| 1 | DTW | 55.91 | 0.59 | 0.59 | 0.55 | 0.80 |
| 2 | DTW +Error Distance | 60.57 | 0.61 | 0.61 | 0.62 | 0.80 |

Mean Average Precision (MAP) and Relevance factor. The results of both the configurations with a relevance threshold of 0.4 are shown in Table 2.

## 4 Results and Discussion

We analyzed our generated textual summary with respect to the ground truth textual summary based on the level of abstraction for a time series, i.e., number of segments and the similarity with the ground truth summary.
*Case 1 (System works as expected): When the estimated level of abstraction for a time series is same as of ground truth and the generated text is similar to the ground truth summary.* Since near synonym choices used in the text such as *Backing* and *Becoming* are mostly author dependent, we accommodate this in our evaluation measure by taking synonyms similarity.
*Case 2: When the estimated level of abstraction for a time series is same as of ground truth and the generated text is not similar to the ground truth summary.* For example, if the actual text is *SE 25-30 backing SE-ESE 20-25* and the generated text is *SSE 25.0 easing later to 23.0.* In the summary, the verb type is determined by the change in wind speed, and direction, i.e., a speed verb is chosen if the change in speed is more significant than the change in direction and vice versa. We notice that humans elide verb information and their text is shorter. Therefore, the correct level of abstraction, i.e., number of segments does not guarantee the high similarity of generated text with the ground truth text.
*Case 3: When the estimated level of abstraction for a time series is not same as of ground truth and the generated text is similar to the ground truth summary.* This case is interesting, because, even though the intermediate content selection is not correct, the final text is similar to the actual text. For example, if the actual text is *S-SW 18-22* and the generated text is *SW 18.0 increasing SSW 21.0*, when we looked at the time series, we found that the change in speed, which is increasing from 18 to 21 in this case, is ignored by forecasters as they wrote only ranges (*18-22,S-SW*). However, in the multiple text configuration of our system, a forecaster can choose the summary, which she prefers.
*Case 4: When the estimated level of abstraction for a time series is not same as of ground truth and the generated text is similar to the ground truth summary.* In this case, our system does not perform as expected. For example, if the actual text is *SW 30-35 rising 38-42 by afternoon/evening and later veering W'LY 25-30* and the final text is *SW 31.0 veering W 26.0 in the evening*, we lose most of the trend information of a time series in the generated text. We suspect that these cases are harder cases and need more domain knowledge. In future, we plan to store such cases as the exceptional cases so that the system can flag a warning message for an external review if any of the exceptional case is reused.

Further, we made some general observations for time series summarization:
1. The data analysis techniques to summarize time series need to be adapted according to the domain and end-user requirements. For example, in the medical domain, artefacts, anomalous spikes are more important, while in the weather domain, trends are more important. This knowledge can be integrated into various forms: for example, we learn the number of segments using available data, or we can use distributional measures like word2vec and wordnet on a large parallel corpus to get the synonyms similarity to evaluate the system.
2. Adaptation of the CBR system can be further improved. For example, cases can be stored as segment wise and for a query time series, multiple cases can be retrieved for each segment in the query. Next, the retrieved segments of these cases can be reused to generate the textual summary.

## 5 Related Work

An earlier approach by Sripada [11] to generate the textual summary of time series in weather domain is a rule-based approach. The approach uses error thresholds provided by experts to find the appropriate abstraction of a time series (content selection). Next, to choose the appropriate words and phrases for the textual summary, the system uses rules induced from corpora and as provided by experts. The other approach by Sowdaboina [9] is a purely data-driven approach to select the content for a summary. It uses a neural network to select the representative points from a time series. The approach, however, considers only one channel to select the points and does not generate the final textual summary. The other text generation systems are MOUNTAIN [4] and (Probabilistic Context-Free Grammar) PCFG based system [3], and the CBR system by Adeyanju [1]. These systems do not perform content selection and focus on other NLG subtasks like microplanning and realization. The input to these systems is the time series reversed engineered from its textual summary, i.e., same as the problem component of a case in $CBR_2$.

## 6 Conclusion

In this paper, we presented an end-to-end CBR system for generating the textual summaries of time series in the weather domain. First, the system generates appropriate abstractions of a given time series with certain confidences for various abstractions. Later, it generates the textual descriptions of these abstractions. In future, we plan to analyze the $CBR_2$ in isolation with $CBR_1$ to specifically study the cases where a correct abstraction of a time series does not necessarily result in the generated textual summary similar to ground truth summary. Further, we plan to store some specific cases with poor alignment as exceptional cases in the case base. That means, in our case, a small perturbation in time series gives rise to an entirely different textual summary. This, in turn, has the cognitive appeal as humans also have few specific experiences combined with the abstract experiences formed over time.

## References


1. Adeyanju, I.: Generating Weather Forecast Texts with Case based Reasoning. International Journal of Computer Applications **45**(10), 35–40 (2012)
2. Bellman, R.: On the Approximation of Curves by line segments using dynamic programming. Communications of the ACM **4**(6), 284 (1961)
3. Belz, A.: Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models. Natural Language Engineering **14**(04), 431–455 (2008)
4. Langner, B., Black, A.W.: MOUNTAIN: A Translation-based Approach to Natural Language Generation for Dialog Systems
5. Miura, N., Takagi, T.: Wsl: Sentence similarity using semantic distance between words. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 128–131 (2015)
6. Reiter, E.: An architecture for data-to-text systems. In: Proceedings of the Eleventh European Workshop on Natural Language Generation. pp. 97–104. Association for Computational Linguistics (2007)
7. Reiter, E., Sripada, S., Hunter, J., Yu, J., Davy, I.: Choosing words in computer-generated weather forecasts. Artificial Intelligence **167**(1-2), 137–169 (2005)
8. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978). https://doi.org/10.1109/TASSP.1978.1163055
9. Sowdaboina, P.K.V., Chakraborti, S., Sripada, S.: Learning to summarize time series data. In: International Conference on Intelligent Text Processing & Computational Linguistics. pp. 515–528 (2014)
10. Sripada, S., Reiter, E., Hunter, J., Yu, J.: Segmenting time series for weather forecasting. Applications and Innovations in Intelligent Systems X pp. 105–118 (2002)
11. Sripada, S., Reiter, E., Hunter, J., Yu, J.: Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201 (2002)

# Predictive Process Mining Using a Hybrid CBR Approach for the Rail Transport Industry

Eleftherios Bandis[1], Miltos Petridis[1], Stelios Kapetanakis[2]

[1]Department of Computer Science, Middlesex University London, The Burroughs, Hendon, London NW4 4BT
{M.Petridis, E.Bandis}@mdx.ac.uk
[2] School of Computing, Engineering and Mathematics, University of Brighton, Moulsecoomb Campus, Brighton NB2 4GJ, UK
S.Kapetanakis@brighton.ac.uk

**Abstract.** Rail transportation improvements have always been considered of high impact to society due to their tangible improvement to quality of life in modern cities. Both public and private companies are highly concerned on how travel patterns, vehicle-passenger behaviours and other relevant phenomena such as weather affect their performance since usually any travel network can be remarkably expensive to build and swiftly saturated after its public release. We propose suitable workflow similarity metrics for developing efficient performance measures for the rail industry using extensive business process workflow pattern analysis based on Case-based Reasoning. We use meta-heuristic features and extend our similarity measures to capture relevant-to-the-industry granular features and apply this work to an industrial case study. Preliminary results of this work are promising since they perform well with the complexity of the problem and can scale on demand while providing an efficient predictive accuracy. Several modelling experiments are presented, that show that the approach proposed here can provide a sound basis for the effective and useful analysis of operational sensor data from train Journeys.

**Keywords:** Case Based Reasoning, Process Mining, Business Process Workflows, Workflow Monitoring, Temporal Reasoning

## 1 Introduction

The rail transportation industry has experienced substantial growth over the last decade in terms of operational method advancement (wayside detectors, wheel profile monitors, extended sensor network), processes, software and hardware equipment (Rail Defect Test Facility, Asset Health Strategic Initiative, and others). The modernisation of the industry has led to increasing usage of computer systems for logistics, tactical, planning, performance and maintenance reasons. Therefore, significant amounts of data are continuously accumulated by many stakeholders (e.g. private rail companies and government entities). The Remote Condition Monitoring (RCM) is an example to those which allows trains to enrich both the volume and the quality of their information systems and be able to monitor their process workflows.

In the UK, railway transportation across the country is operated by multiple private organizations usually called "Rail Operators" (ROs). ROs own the trains and manage all services whereas the whole rail network infrastructure is managed by a national infrastructure provider -Network Rail (NR)- [2]. Each RO may have a bespoke business model to represent their processes and specific services running on routes. However, any operational model must be approved from NR to form a unified and functional timetable. The unified timetable shows the routes, timeframes and other relevant information that should be followed and respected by all operators. ROs must comply to the timetable timeframes to avoid disruption to other companies' routes and to sustain the desirable level of performance.

Rail industry can experience severe reduction in performance when it comes to unexpected disruptions in service. Such disruptions are experienced by the public as delays, since a "delay" in service is a well understood term across all relevant stakeholders. Any cause of delay may be attributed to a train malfunction, temporary crew shortage and other reasons. In many cases the reasons behind a delay are difficult to identify, as it may have several contributing factors and can have a cascading effect, triggering further delays or cancellations, etc.

To be able to identify the reasons behind delays, we propose process mining techniques based on workflows to assist in deviation measurements from scheduled processes (i.e. timetable routes) against the workflows logged by the information system. Such an approach can enable process managers to identify patterns and possible bottlenecks within workflow processes. To achieve that, workflow executions should be associated with the expected business process instances (i.e. timetable). However, this has proven to be a complicated task as several bottlenecks exist within the Railway system [1].

We introduce a multi-level Case-based Reasoning (CBR) approach to achieve workflow alignment between monitoring data and business processes by considering the railway domain unique characteristics and challenges as described above. The work presented here follows on from earlier work [1]. This paper is based on the original work presented there using CBR to integrate data from timetabling, geographic and RCM data, but provides an evaluation based on several experiments done using predictive machine learning techniques including CBR to validate the effectiveness of the approach to provide real predictive insights into rail route operation as business workflows.

The rest of this paper is organised as follows: Section 2 presents the relevant literature in terms of CBR, Workflows, Process mining and hybrid models, Section 3 formulates our proposed methodology for effective process mining of rail routes a workflows, Section 4 shows our preliminary evaluation results, Section 5 presents a study based on a set of experiments and examples of analysis that has been conducted on workflow information provided by the approach and finally, we discuss our overall findings and our future research steps.

## 2 Related Work

Modern organisations use business process workflows to coordinate their processes, tasks, roles and synchronise their resources with the aim to improve efficiency, efficacy and profitability. Business process workflow management differs across organizations. The size, sector and strategic orientation of an organization play a key role on how they adopt, analyse and practice Business workflows [8]. A common taxonomy includes the phases of: Design, Implementation, Enactment, Monitoring and Evaluation as the workflow life cycle in Business process management [8] [9] [10]. Among those the Monitoring phase enables the supervising of business processes in terms of management (e.g. performance, accuracy) and organization (e.g. utilization of resources, length of activities etc.) [10]. Therefore, the Monitoring phase is a crucial ,indicating to process managers what amendments are required to improve their processes.

CBR is an approach based on the assumption that: "problems tend to (re)-occur". Thus, problems occurred in the past tend to re-appear in the future in a similar form. Respectively, any solutions that managed to solve previous problems may be recycled to solve currently experienced problems [11].

A requirement for CBR to work is the availability of cases. Cases are usually stored in a Case base along with their associated solutions. Based on this knowledge, CBR can produce a solution for a new problem by following the CBR process cycle defined in [12]. The four main (R) phases of CBR are: Retrieve, Reuse, Revise, Retain.

Workflow experts can use various methods to evaluate their processes, however, large or extended volumes of data can make the analysis of event logs extremely difficult. Process Mining (PM) is the technique used to extract knowledge and insights by discovering and analysing processes from event logs [13] [14]. By applying process mining, domain experts can use the derived information as feedback to design new processes or revise and enact predefined ones [15].

In the literature, several algorithmic techniques have been introduced to solve the process mining problem. Algorithms like Alpha miner and alpha+ have been used extensively but other heuristics, genetic and fuzzy algorithms have also been applied [4] [3]. Each algorithm has its limitations on a different aspect of the process discovery (e.g. fitness, simplicity and precision), and they may be unfit to areas where uncertainty, inconsistency and fuzziness is present, therefore a CBR approach may be appropriate.

Several related researches attempts to address problems around workflows. Van der Aalst et al. [16] proposed an approach that compares process models. This approach shows how the degree of similarity between process models can be measured. Also, it is being considered the fact that distinct parts of a process might have "stronger" notion than others. The results are presented on a Petri nets structure.

Dijkman et al. [17] attempts to rank business process models according to their similarities. Four distinct types of graph matching algorithms were compared to solve the similarity search problem. The produced results by the algorithms were based on a trade-off between computational complexity inherited from graph matching and the comparison accuracy. Weber et al. [18] presents a tool that is based on conversational case-based reasoning which complements an adaptive workflow management system. The tool provides knowledge to a management system and enables the adaptiveness of

predefined workflow models based on confronted circumstances. Workflow management systems produce more accurate results over time since it builds experience on the knowledge gained previously. Minor et al. [19] presented a CBR approach that allows the reuse of previous adaptations of workflow instances on the ongoing ones.

We used a graph-based system to retrieve previous cases of adaptations for each part of the workflow structure. Therefore, previous modification that occurred on a similar case can be evaluated to be applied again. Kapetanakis et al. [7] [20] provided explanations to the intelligent monitoring of business process workflows. This approach showed how a similarity measure between workflow instances can be establish considering intervals and temporal relationships using CBR. The fundamental assumption in this approach is that a workflow structure is not met during execution. Therefore, the workflow instances are identical but not same. Consequently, workflow instances marked as problematic, that seem to be similar with other instances, they probably share the same problems and require similar solutions.

## 3 A CBR approach for aligning workflow executions

CBR has been shown effective in monitoring workflow instances under uncertainty [6] [7]. Utilising CBR's fundamental principle of "similar problems have usually similar solutions" we investigated several rail data instances to model route cases.

CBR retrieves past solutions from a case-base matching workflow instances to route-processes. In our industrial scenarios a workflow is a route stored in event log sequences and a business process is the scheduled route as planned and showed on a public timetable. In our CBR model, we treat routes as cases and their related business processes as solutions for those cases. Based on temporal and spatial data our case representation is formulated as in Figure 1.

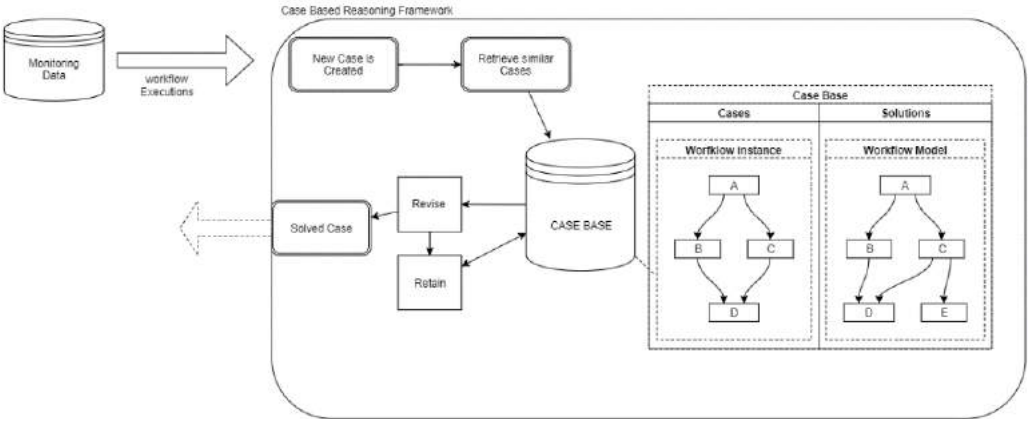



**Fig. 1.** Case representation as generated from temporal/ spatial workflow data

Delay patterns are often related to location and time e.g. rail platforms during peak hours can be overcrowded and this may lead to delays. Another common example are

busy junctions during certain hours causing overheads to any related services. Therefore, it is presumed that same or similar routes share similar bottlenecks (delays).

This section will present our case representation, the formulated similarity metrics for our investigated domain and their formal relevance to temporal logic.

### 3.1 Case Representation

A workflow process consists of multiple activities. Activities involve tasks such as "start of a journey", "departure from a station", "arrive on a station" or "end of a journey". The tasks contain multi-perspective information such as:

1. Time-related information: The start and the end of each activity is marked with a timestamp. The duration of an activity is also given.
2. Location: The station of which the activity takes place
3. Relationships: One activity holds which activity follows as well as the time duration between them

General information about the workflow is also available such as: the total duration of all activities, the train unit responsible to undertake all the workflow activities, the day of the week the workflow took place and the workflow start and end time.

When CBR is adopted to provide solutions, new cases are created enclosing workflow data within cases. Therefore, a new query case will have the following structure:

{ $\mathrm{UnitNumber}_q$, $\mathit{StartDay}_q$, JourneyTimeq, $\mathrm{StartTime}_q$, $\mathrm{EndTime}_q$, $\mathit{StationList}_q$, $\mathit{ActivityList}_q$, }

And for each activity:

{$\mathrm{StationName}_q$, $\mathrm{StopDuration}_q$, $\mathrm{NextStation}_q$, $\mathrm{TimeUntilNextStation}_q$}

### 3.2 General Time Theory

Our workflow data follow a sequential temporal vs. spatial pattern since they represent a variety of activities (as presented in Section 3.1) over time. To represent their sequence in a formal way we use the General Time Theory (GTT) [5] [1]. The general time theory takes both points and intervals as primitive.

A time element t is called an interval if Dur(t) > 0; otherwise, t is called a point

In our graph representation each node represents a station whereas any edge represents the duration from station A to station B. A GTT workflow representation allows for a unified log interpretation which in conjunction with the multi-level similarity representation (Section 3.3) presents a foundation for adequate CBR workflow cases [1].

### 3.3 Similarity Functions

We define a set of multi – level similarity functions relevant to the problem domain. Since elements of temporal information are present throughout a log journey, a GTT representation as shown in Section 3.2 allows for vectorised workflow mapping. Similarity measures are split into two levels (Levels 1 & 2) based on the workflow structure.

Level 1: Identifies relevant timestamps from workflow data. For example, Let case 1, $C_1$ and case 2, $C_2$ as workflow representations and $C_{1L}$, $C_{2L'}$ their respective list of stations. For $C_1$ and $C_2$ if Start date is the same (Binary equal) && Start time relies within $\gamma$ mins fluctuation && $C_{1L}$ is like $C_{2L'}$ based on an μ string threshold.

$$distance\ (C_1, C_2) = |\ StartTimeC_1 = StartTimeC_2 \leq \gamma| * w_1 + |\ EndTimeC_1 - EndTimeC_2 =< \gamma\ | * w_2 + |\ StationListC_1 - StationListC_2\ | * w_3$$ (equation 1)

Where $w_1$, $w_2$, $w_3$ are empirically (expert-based) derived domain constants and

$$w_1 + w_2 = w_3$$ (equation 2)

Upon successful relevance on similarity 1, a Level 2 similarity can be defined as:

$p_1$: create relationships => {[$S_1$, Dur($S_1$), Dur($S_2$), Meets $S_2$] ...}

(equation 3)

Where $S_1$ is a starting point, Dur($S_1$) is the time spent on the station, Dur($S_2$) the time till the next station, and Meets $S_2$ the station that follows. A Level 2 similarity is based on equation 3 quadruplets as:

$$distance\ (C_1, C_2) = |\ [S_1, Dur_{S1}, Dur_{s2}, S_2]C_1 - [sS_1, Dur_{S1}, Dur_{s2}, S_2]C_2\ | * w_1 + |\ StartDayOnlyC_1 = StartDayOnlyC_2\ | * w_2 + |UN_1 = UN_2| * w_3$$

(equation 4)

Where $UN_1$ and $UN_2$ are actor related identification numbers

## 4 Experiments & Results

Our case representation, as presented in Section 3, allows for a rigid problem definition. A key challenge presented from the application domain is the lack of "solutions" due to the following reasons:

1. Constant changes at business process level. A Rail timetable changes every 6 months (seasonal). Variations in any normal operation can vary among a week before the actual service, a few days in advance, or even hours. In several cases any of the above amendments could be chained e.g. a disruptive change a few days between the DTR – LBG route may also be changed hours before the actual service. This raised significantly the fuzziness within the data.
2. Incomplete data. Our provided data although very rich in volume had substantial degree of repetition and severe incompleteness at cases.
3. Our data corpus was coming from variant datasets that posed heavy uncertainty due to their technical compatibility and inconsistencies.

To overcome the above challenges, we had several sessions with senior business process experts, analysts and industry engineers that elaborated extensively in several cases vs. right solution matches. With their help a case base was formulated containing several past workflow executions and their solutions as corresponding process models.

The CBR cycle was modified in the following way to suit the domain:

1. Retrieve: Similarity functions were defined based on the 2-level similarity model (section 3) based on multiple perspectives (such as time, resources, order flow, relations between activities, etc.)

2. Revise: when a queried case couldn't converge in finding a similar case with > 60% relevance, the case was tagged as a "newly" encountered pattern. A low similarity score on cases indicated an incomplete or an "just in time" (JIT) amended services which had no previous process model.
3. Retain: Indicated no modifications to existing cases. After every "new" case encounter it updated the case base and pushed any case with similarity lower than 60% to a new case base for further investigation.

For our evaluation we used an (hourly) workflow data sample of 238MB, achieving a performance accuracy of 76% on cases vs. ranked business process from industry experts with a 10-fold validation on 30%(test)-70%(training) split.

Motivated by the success of the initial experiment we used the trained case-base on a substantially larger dataset of 480GB with (3 months) of workflow data. The ranked case-base performed adequately to similar routes however a substantial number of "newly" seen cases emerged which could not be adequately attributed.

To benchmark our approach a probabilistic approach was adopted, comparing any new investigated case with possible nearest neighbour "paths". An example may be appropriate to explain the concept: Let's assume an imaginary path of letter-labelled stations: A, B, C, D, E and F. Our case base may contain ABC and ACF. If ABDE comes in the 2-level similarity cannot produce convincing results whereas a probabilistic approach can rank ABDE as an AB variant with 50% probability and ignore ACT since the probability to fit, there is less than 33%.

The probabilistic approach seemed to work better than CBR on the large dataset due to sheer volume characteristics which were very hard to address. On a variant experiment using 2 different datasets (one ranked by experts and the other unknown) our methodology did achieve similar performance results: 70% accuracy on the ranked dataset and substantially lower (not decent enough to present) to the other. By applying a similar probabilistic approach as to the one above the gained results seemed better compared to a CBR approach.

## 5 Predictive Analysis of the integrated workflow data

Following the work and experiments done above we were able to produce reliable route data for just over one-year operation for the fleet of data of one Rail Operator based in the London area. This data covers the years 2015-2016. First a statistical data analysis and visualisation was conducted that allowed railway engineers and planners to get useful insights into the operation. Statistical data were provided for routes at various times of day, different days of the week focusing on specific known "trouble spots".

Additionally, a tool was developed that allowed to use the original RCM data to visualise any one given journey on a route. The RO engineers and planners expressed their satisfaction with this system. This analysis was used to design a radically different timetable that went into effect in May 2018.

### 5.1 Data Analysis Experiments

Following from the original analysis of the data which was mainly statistical and visual, the authors embarked to the second phase of the project that aims to use machine learning techniques to provide additional insights into the workflow data representing the rail routes. In the first instance, the experiments concentrate on supervised learning techniques concentrating on numerical prediction and classification.

A specific known problematic route was selected. One year's worth of journeys were elected from the dataset created as explained above. The structure of the data is as shown in section 3.1 above, containing *trainHeadCode, start-time, Day, {inter-station travel times}, {station dwell Times}, arrival-time.* The attributes in {} were for each of the 14 stops (no dwell time for the start and end station, or inter-station travel for the destination station). The dataset for the route had 6562 cases (all one-way travel)

A series of analysis experiments were conducted aiming at numerical prediction and then after binning of values on standard deviation multiples classification. The experiments were conducted using the IBM SPSS Modeler 18.0 tool. For the purposes of this data, specific information on routes and stations have been anonymised to protect commercial sensitivities.

### 5.2 Numerical prediction experiments

A first set of experiments has been set to estimate how early data in a journey can be used predict the time of arrival of the service. As such, we used specific known problem spots early in the selected route (3 segments that intersect other lines and 3 dwell times in the corresponding starting stations) to predict the overall duration of the journey. The day of the week is also used as a predictor. The results of this can be seen in table 1.

| | **Generalised Linear Model** | **Regression** | **ANN (MLP)** |
|---|---|---|---|
| **Minimum Error** | -533.512 | -525.792 | -344.543 |
| **Maximum Error** | 1974.25 | 1966.404 | 1997.251 |
| **Mean Error** | 0.0 | 0.0 | -0.231 |
| **Mean Absolute Error** | 72.929 | 73.386 | 71.065 |
| **Standard Deviation** | 105.81 | 106.178 | 104.845 |
| **Linear Correlation** | 0.802 | 0.8 | 0.807 |
| **Occurrences** | 6,562 | 6,562 | 6,562 |

**Table 1.** Numerical experiment results

Table 1 shows that the Neural network approach (Multi-layer perceptron -MLP) slightly improves on the other models. It must be stated that Regression does not consider the day of the week attribute though. An error in 70-80 seconds is quite an appropriate level of accuracy, especially as a serious delay is defined as 3 minutes or more.

### 5.3 Classification experiments

In addition to the numerical prediction above, classification algorithms were used after binning was applied to the predicted attribute (total Journey travel), the width of bins aligned to standard deviations for each We used an averaged 5-way folding 80%-20% cross validation to evaluate the model accuracy. The results of this are presented in Table 2. It can be seen there that Logistic regression performs best. However, simple observation on the results shows that KNN picks up more late trains, but also suffers by more "false positives".

| | Log Regression | | C5.1 | | ANN (MLP) | | C&R Tree | | KNN | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct** | 1124 | 86.26% | 1104 | 84.72% | 1117 | 85.71% | 1101 | 84.48% | 1107 | 84.96% |
| **Wrong** | 179 | 13.74% | 199 | 15.28% | 186 | 14.29% | 202 | 15.52% | 196 | 15.04% |
| **Total** | 1304 | | 1304 | | 1304 | | 1304 | | 1304 | |

**Table 2.** Classification experiment results

There is room for more such experimental work, trying different attribute selections, guided by the rail experts. This work is currently under way.

## 6 Conclusions

This work presents transport friendly approach to break down the complexity of temporal spatial data and attempt to identify workflow patterns and trends over time. We propose a new multi-level similarity approach that can elicit meta-heuristic features and can assist in capturing relevant-granular features. We presented some preliminary results from our work on real industry case study, although our results were affected from the investigated dataset(s) bias, limited ranking and very large volume and variety. In our future work we plan to improve substantially our model towards automatic detection of workflow differences, mine patterns efficiently and work on ways to tackle large data volumes and dataset discrepancies and ill-balanced data. We will also work on establishing benchmark tools to enhance the accuracy, precision and recall of our proposed methodology while also combining the current analysis with more supervised and unsupervised machine learning algorithms to produce a more rounded view of the knowledge encapsulated in the data. This will be done by analysing the interaction between more than one routes and concentrating on known "trouble spots" in the network.

## References


1. Bandis, E, Kapetanakis, S, Petridis, M, Fish, A. (2017) Effective Similarity Measures for Process Mining Using CBR on Rail Transport Industry, in Proceedings of the 22nd UKCBR workshop, Cambridge UK, December 2017
2. Network Rail, https://www.networkrail.co.uk/who-we-are/about-us/, last accessed 28/10/2017.

3. Van der Aalst, W. M. P., de Medeiros, A. K. A., & Weijters, A. J. M. M. (2005). Genetic process mining. Applications and Theory of Petri Nets.
4. Tiwari, A., Turner, C. J., & Majeed, B. (2008). A review of business process mining: State-of-the-art and future trends. Business Process Management Journal, 14(1), 5–22
5. Ma, J., Knight, B.: A General Temporal Theory, the Comp Journal, 37(2), 114-123 (1994).
6. Kapetanakis, S., Petridis, M.: Evaluating a Case-Based Reasoning Architecture for the Intelligent Monitoring of Business Workflows, in Successful Case-based Reasoning Applications-2, S. Montani and L.C. Jain, Editors. 2014, Springer Berlin Heidelberg. p. 43-54.
7. 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)
8. Van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1-12. Springer, Heidelberg (2003)
9. Zur Muehlen, M.: Workflow-Based Process Controlling: Foundation, Design and Application of Workflow-driven Process Information Systems. Logos (2004)
10. Reijers, H.A.: Design and Control of Workflow Processes: Business Process Management for the Service Industry. Springer, Heidelberg (2003)
11. Leake, D. (1997). Case Based Reasoning. Experiences, Lessons and Future Directions. AAAI Press. MIT Press, USA, 1997.
12. Aamodt, A., Plaza, E. (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(i), 1994.
13. Van der Aalst. (2011) Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin, 2011.
14. Van der Aalst, W. M. P., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., & Weijters, A. J. M. M. (2003). Workflow mining: A survey of issues and approaches. Data & Knowledge Engineering, 47, 237–267.
15. Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer Verlag, Berlin (ISBN 978-3-642-19344-6).
16. 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)
17. Dijkman, R.M., Dumas, M., Garcia-Banuelos, L. Graph matching algorithms for business process model similarity search. In U. Dayal, J. Eder (Eds.), Proc. of the 7th Int. conference on business process management. (LNCS, Vol. 5701, pp. 48-63). Berlin: Springer. (2009)
18. Weber, B., Wild, W. and Breu, R. "CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning", in Proceedings of ECCBR04, Advances in Case-Based Reasoning, LNCS, Vol. 3155, 434-448, Springer (2004)
19. 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)
20. Kapetanakis, S., Petridis, Ma, J., Bacon, L.: 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 5th International Workshop on explanation- Aware Computing Exact (ECAI 2010), Lisbon, Portugal (2010)

# *Emojinating:* Representing Concepts Using Emoji

João Miguel Cunha, Pedro Martins, Penousal Machado

CISUC, Department of Informatics Engineering, University of Coimbra
{jmacunha,pjmm,machado}@dei.uc.pt

**Abstract.** Emoji system does not currently cover all possible concepts. In this paper, we present the platform *Emojinating*, which has the purpose of fostering creativity and aiding in ideation processes. It lets the user introduce a concept and automatically represents it, by searching for existing emoji and generating novel ones. The system combines the exploration of semantic networks with visual blending, and integrates data from EmojiNet, ConceptNet and Twemoji. To evaluate the system in terms of production efficiency and output quality, we produced emoji for a set of 1509 nouns from the *New General Service List*. The results show a coverage of 75% of the list.

**Keywords:** Computational Creativity, Computational Generation, Concept Representation, Computational Design, Visual Representation, Emoji

## 1 Introduction

Emoji are often associated with the meaning "picture-word", as *e* can be translated to "picture", *mo* to "writing" and *ji* to "character"[1]. Their increasing importance is well documented by statistical data (e.g. [17]) and some authors even discuss a possible shift towards a more visual language [23, 12]. The integration of emoji in written language can be easily observed in the growing number of emoji-related tools and features – e.g. search-by-emoji[2], and the Emoji Replacement and Prediction features implemented in iOS 10[3]. These features explore the relation between concepts and their representation in emoji.

Despite the constant addition of new emoji, there are still a large number of concepts that do not have a representation. Several attempts have been made to complement emoji lexicon, some of which resulted in new emoji being officially added to Unicode Standard. The nature and goals of such attempts are not always the same. Some examples are: to propose culture-specific emoji[4]; to

[1] unicode.org/reports/tr51/proposed.html, retr. 2018
[2] blogs.bing.com/search/2014/10/27/do-you-speak-emoji-bing-does, retr. 2018.
[3] macrumors.com/how-to/ios-10-messages-emoji/, retr. 2018.
[4] finland.fi/emoji/, retr. 2018

increase the scope of a certain trait (e.g. curly hair[5]); to help abuse victims communicate[6]; or even to just propose "missing emoji" (e.g. *condom*[7] and *taco*[8]).

In 2015, the Unicode Consortium decided to add "skin tone" modifiers (characters that could modify other emoji) to Unicode core specifications. One year later, the ZWJ (Zero-Width-Joiner) mechanism was also implemented – an invisible character to denote the combination between two characters [1]. This development meant that new emoji could be created by combining others.

By having these combination mechanisms as inspiration and following the idea presented in [9], we believe that the connection between the pictorial character of emoji and its associated semantic knowledge can be explored in the generation of visual representations for concepts. In this paper, we present *Emojinating* – a tool which allows the user to introduce a concept and automatically presents emoji that represent it. Three resources are used: Twemoji[9], EmojiNet [34] and ConceptNet [31]. By combining semantic network exploration with visual blending, it not only searches for existing emoji but also produces new ones. There is great potential for its usage in brainstorming activities, leading to creativity stimulation and ideation fostering. The system behind *Emojinating* was thoroughly described in [10]. For this reason, we will not go into much detail, but will instead focus on the analysis of the generation and representation for single-word concepts – in [10] only double-word ones were addressed.

The remainder of this paper is organised as follows: section 2 summarises the related work; section 3 describes our approach; section 4 analyses the results obtained in the representation of 1509 nouns from the *New General Service List* [4]; and section 5 presents our conclusions and directions for future work.

## 2 Related Work

Previous research on emoji can be divided into five main topics: Meaning, Sentiment, Interpretation, Role in communication, Similarity, and Generation. Concerning emoji meaning, word embedding techniques are normally used with different data sources (e.g. [13, 3, 16]). Emoji sentiment is often calculated from the sentiment of the text in which they occur (e.g. [25]) and has been used to study the intentions for using emoji [21]. Miller et al. [24] described how the interpretation of meaning and sentiment of emoji change within and across-platforms, and Rodrigues et al. [30] studied interpretation differences between users and developers. Research on the role of emoji in written communication addresses several topics: e.g. redundancy and part-of-speech category [14], emoji function [15], effect on reading time [19], emoji as semantic primes [33], among

[5] adage.com/article/digital/dovelaunchescurlyhairedemojisaddressvoid/301203/, retr. 2018
[6] webcollection.se/bris/abusedemojis/, retr. 2018
[7] businessinsider.com/durexs-condom-emoji-for-safe-sex-2015-11, retr. 2018
[8] tacobell.com/stories/Tacoemoji, retr. 2018
[9] github.com/twitter/twemoji, retr. 2018

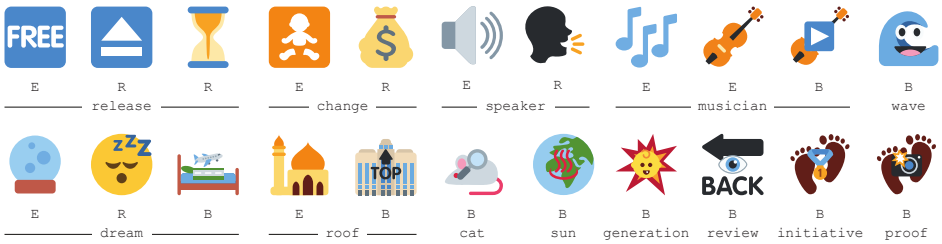



**Fig. 1.** Examples of retrieved emoji: existing (E), related (R) and blended (B)

others [22, 7, 20]. In terms of similarity between emoji, Ai [2] semantically measured emoji similarity. Other authors used emoji vector embeddings to identify clusters of similarity [16, 3]. Pohl et al. [27] organised emoji in a relatedness-hierarchy. Wijeratne et al. [35] built a dataset of human-annotated semantic similarity scores – EmoSim508.

Literature is scarce on emoji generation and most work uses Generative Adversarial Networks to replicate existing emoji, e.g. [28, 29]. The quality of the results is significantly lower when compared to the one of official emoji.

### 2.1 Variation and blending

Several applications allow some degree of variation in emoji (or equivalent graph-icon), e.g. Windows Live Messenger[10] enabled the creation of emoticons through image uploading and Slack[11] currently has the same feature. Moreover, there are other applications that consist in face-related customisation, e.g. Bitmoji[12]. These examples and the emoji proposals (presented in the introduction section) show that there is potential in emoji combination and generation. It is our belief that visual blending can be used to represent novel concepts.

Current computational approaches to visual blending can be divided into two groups according to the type of rendering used: (i) picture or photorealistic rendering; and (ii) non-photorealistic (e.g. pictograms or icons). Examples of the first group are: Steinbrück [32] who combined image processing techniques with semantic knowledge gathering to produce images in which elements are replaced with similar-shaped ones (e.g. round medical tablets are transformed into globes); and Vismantic [36] – a semi-automatic system that produces visual compositions for specific meanings (e.g. *Electricity is green* is represented as the fusion between an electric light bulb and green leaves).

On the other hand, a categorisation can also be done in terms of where the blending process occurs: some interpret or visualise previously produced conceptual blends, e.g. Pereira and Cardoso [26] experimented with conceptual

[10] news.microsoft.com/2003/06/18/msn-messenger-6-allows-im-lovers-to-express-themselves-with-style/,retr. 2018
[11] get.slack.help/hc/en-us/articles/206870177-Create-custom-emoji, retr. 2018
[12] bitmoji.com, retr. 2018

blends produced for the input spaces *house* and *boat*; others use blending only at the visual level, e.g. Correia et al. [6] generated faces out of existing ones by recombining face parts; and in others, which can be called hybrid, the blending process starts at the conceptual level and only ends at the visual level, e.g. Cunha et al. [8] generated visual conceptual blends for the concepts *pig, angel* and *cactus*. In addition, some authors combine entire signs, e.g. [5], while others combine parts, e.g. the blend of Pokémon[13].

The project most similar to ours is Emojimoji[14], an emoji generator which randomly merges two emoji. However, none of abovementioned work addresses our main subject – using existing emoji and associated semantic knowledge for developing a tool to aid in ideation.

## 3 The approach

Before emoji, emoticons were used to express emotions in Computer-Mediated Communication. One of their advantages is the potential for customisation and variation. Whereas emoji are inserted as a whole in the text, emoticons are the result of a combination of individual components [15] – e.g. ":" + ")" = ":)". The changeable parts not only allow a high degree of visual variability but also the exchange of a component leads to a change in the meaning. This is one of the reasons why they are still being used as alternative to emoji [18]. We follow a similar approach in the generation of novel emoji, having the modifier and ZWJ mechanisms as inspiration. By taking advantage of the emoji connection between pictorial representation and associated semantic knowledge, we aim to develop a computer-aiding tool for creativity fostering and icon design.

*Emojinating* has two main functionalities: (i) search for existing emoji and (ii) generation of new ones. In order to implement these two functionalities, we combined data from the following online resources: Twitter's Twemoji 2.3 – a dataset of fully scalable vector graphics with 2661 emoji; EmojiNet – a machine readable sense inventory for emoji built through the aggregation of emoji explanations from multiple sources [34], containing 2389 emoji; and ConceptNet – a semantic network originated from the project Open Mind Common Sense [31], which we use to obtain concepts related to the one introduced by the user.

Twitter's Twemoji dataset, despite allowing an easy blending process due to the layered structure of the vector images, does not have any semantic data associated. For this reason, EmojiNet was used. We extract, for each emoji, the *name*, *definition*, *keywords*, *senses* and *unicode* from EmojiNet, which are used as criteria in the search for emoji. These are afterwards matched with the images from Twemoji and used to retrieve emoji related to the user-introduced concept.

### 3.1 How it works

The system searches for existing emoji semantically related to the introduced concept (T1) and complements this search with a visual blending process which

[13] pokemon.alexonsager.net, retr. 2018
[14] emblemmatic.org/emojimoji, retr. 2018

generates new emoji (T2). After gathering the emoji, it presents them to the user. The blending process is useful in cases when there is no existing emoji that matches the concept but also to suggest possible alternatives. The system output is a variable number of visual representations for the introduced concept, composed of existing (E) emoji, related (R) emoji and generated blends (B). The system makes use of three main components:

1. **Concept Extender (CE)**: based on a given concept, uses ConceptNet to search for related concepts;
2. **Emoji Searcher (ES)**: searches for existing emoji that are semantically related to a given word, using semantic knowledge provided by EmojiNet;
3. **Emoji Blender (EB)**: receives two emoji as input and returns a list of possible blends.

In this paper, we decided to only address single-word concepts. The blends for single-word concepts are generated using double-word related concepts. The different components are used in the gathering and production of emoji. For the retrieval of existing emoji the ES component is used. In the gathering of related emoji, CE and ES are used. The search is currently being conducted for two levels: directly related concepts (1st), and second degree concepts – i.e. indirectly related (2nd). Regarding emoji blending, the system firstly collects related concepts (using CE), then searches for existing emoji for the concepts (using ES) and finally blends them (using EB).

Knowledge from the different resources is used to generate novel representations. One example is the blend for *generation* (Fig. 1). Firstly, CE is used to retrieve the related concept *baby boom*. Then, semantic knowledge associated with emoji is used by ES to obtain matching emoji: the *baby* (from the name) and the *collision* emoji (from the keyword "boom"). Finally, the blending process makes use of attribute-based and positioning knowledge, which is retrieved from existing emoji (i.e. the *baby* emoji is placed according to the position of the *collision* emoji).

### 3.2 Interface

The aim of the *Emojinating* platform is to allow the user to input a concept and receive emoji that represent it. As such, the interface has two main areas: the *search area* and the *results area*. The *search area* contains a search field in which the user writes words to search. After conducting the search and generation of emoji, the results are presented to the user in the *results area*. This area is divided into four sections: (i) the generated blends section which shows the blended emoji; (ii) the existing emoji section which shows emoji retrieved from the search for the introduced word(s); (iii) the related emoji (1st level) section which shows emoji for directly related concepts to the one introduced; and (iv) the related emoji (2nd level) section which shows emoji for indirectly related concepts (directly related to related concepts).

The user is able to download any emoji by clicking on it. Despite being a simple interface, we consider that it serves its purpose as it allows the input to be given and presents the results in a perceptible way.

| | R 1st | R 2nd | B | or RB | nor RB |
|---|---|---|---|---|---|
| **E** | 927 | 853 | 683 | 707 | 921 | 6 |
| $\overline{\mathbf{E}}$ | 582 | 414 | 529 | 336 | 578 | 4 |
| | 1509 | 1267 | 1212 | 1043 | 1499 | 10 |

**Table 1.** Number of nouns with each type of emoji – related (R) 1st and 2nd level, blended ($B$), either R or B, and neither R or B – and the presence of existing emoji ($E$). The number of emoji considered in $R$ does not include the ones that also exist in $E$.

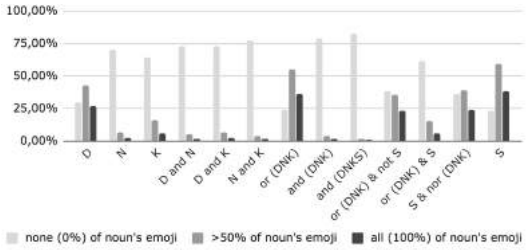



**Fig. 2.** Percentage of nouns in relation to the percentage of noun's existing emoji in terms of semantic information source, e.g. 29,45% of the nouns with existing emoji have none of their emoji (0%) with "definition" as source of their semantic information (first bar on the left). Sources are: Definition, Name, Keywords and Senses.

## 4 Results and discussion

In this section, we present and discuss the experimental results. We begin by describing the setup of a test for the assessment of the system's quality in terms of gathering existing emoji and generation of new ones. Afterwards we present and analyse the results.

In order to evaluate the system, we used a list of concepts, containing concepts with and without official emoji representation. The list selected was the *New General Service List* [4] as it consists of a core vocabulary of 2801 words for second language learners. As most emoji represent nouns, we decided to apply this restriction to the list. Using RiTa[15] library (part-of-speech tagging function), the list was reduced to 1509 nouns. The system was used to produce emoji for each concept of the list and output was analysed in terms of production and quality. Given the large number of nouns, we are still conducting user evaluation on output quality, and at this stage we tested the full set with two users.

### 4.1 Analysing the production of emoji

In general, the system is able to produce emoji that reflect the meaning of the noun, both related (e.g. change) and blended (e.g. generation) – see Fig. 1. From the 1509 input names the system is only unable to produce emoji for 4 nouns (protein, incentive, immigrant and refugee), as observed in Table 1. It produces existing emoji for 927 nouns, 1st level related emoji for 1267 nouns, 2nd level related emoji for 1212 nouns and blends for 1043 nouns.

The most significant source of semantic information is *senses*, with 59.44% of nouns have the majority of their emoji related to senses, 38.3% have all the emoji (100%), and only 23.3% have none of the emoji (0%) related to senses (Fig. 2).

[15] rednoise.org/rita/

(a)

| | **1** | **2** | **3** | **4** | **5** | |
|---|---|---|---|---|---|---|
| **Related** | 692 | 253 | 287 | 215 | 31 | 1478 |
| **Blended** | 668 | 187 | 108 | 74 | 6 | 1043 |

(b)

| | | **Gs** | **G$\overline{\text{s}}$** | **$\overline{\text{G}}$** | |
|---|---|---|---|---|---|
| **E** | *good* | 112 | 675 | 4 | 791 |
| | *bad* | 65 | 69 | 2 | 136 |
| **$\overline{\text{E}}$** | | 288 | 290 | 4 | 582 |
| | | 465 | 1034 | 10 | 1509 |

**Table 2.** ***(a)*** Quality of Related and Blend emoji – (1) none represents the noun, (2) bad, (3) neutral, (4) good and (5) obvious, expressed in number of nouns. ***(b)*** Usage of generated emoji (related and blends) vs presence of existing emoji ($E$), expressed in number of nouns. Nouns with existing emoji were divided into: *good* (at least one existing emoji represents the noun) and *bad* (no existing emoji represents the noun). It shows the number of nouns in which one of the generated emoji was selected to represent the noun ($s$); and in which none of the generated emoji was selected ($\overline{s}$).

It is also important to notice the value of *definition*, with 43.04% of nouns with the majority of their emoji related to *definition*, 26.86% with all the emoji, and 26.86% with none of the emoji. These two sources highly contrast with the rest, as well as, with combinations among them.

### 4.2 Analysing the quality of generated emoji

The system's ability to retrieve related emoji and produce blends does not mean that produced emoji correctly represent the concept. We firstly evaluated the results from gathering of related emoji and blending of new ones, associating an integer from 1 (does not represent the noun) to 5 (represents in a obvious way). This value concerns the best exemplar (if such exists). The obtained results can be seen in Table 2 (side a).

On the other hand, some of the sources used to retrieve existing emoji are not official but result from user attribution (e.g. senses). For this reason, there is no guarantee that they represent the concept well. To evaluate the quality of the existing emoji we attributed a binary value corresponding to wether it represents (*good*) or not (*bad*) the concept. Afterwards, we identified if at least one of the generated emoji (related or blended) can be selected to represent the noun (S) – i.e. it is as good or better than the existing emoji.

From this analysis it is possible to divide the nouns into several groups (see Table 2, side b):

1. **Gs & E** – a generated emoji was selected to represent the noun (S) despite the presence of existing emoji (E). Three situations occur: **(a)** *Good E* but the generated ones are even better. This is the best case scenario and had an incidence of 112 out of 921 emoji with Existing and Generated emoji (12.16%), which we consider a good result – e.g. *musician*, *release*, *roof* and *wave* in Fig. 1; **(b)** *Bad E* and the generated ones are better. We do not

consider the results for this group very good as the generated were only selected in 65 from a total of 134 nouns with generated and bad existing emoji. One reason for this may be the abstract nature of nouns; **(c)** *Equally good*. The generated emoji are as good as the existing emoji. This is often related to different meanings for the same noun – e.g. *change* and *speaker* in Fig. 1;

2. **Gs & $\overline{\text{E}}$:** There are no existing emoji and the system is able to generate emoji that represent the noun well – e.g. *initiative*, *proof*, *generation* and *review* in Fig. 1;
3. **G$\overline{\text{s}}$ & E:** The system is not able to generate anything better than the existing emoji. Two situations occur: **(a)** *Good E.* This is the case with most incidence (675 nouns). This is easy to justify as some nouns have officially associated existing emoji (we did not determine which and we consider it as future work) – e.g. *cat* and *sun* in Fig. 1. The fact that the generated were not selected, does not mean their quality is not good – it was just not enough to surpass the existing emoji. This may be due to the metaphoric quality of generated emoji; **(b)** *Bad E.* Despite the bad quality of existing emoji, the generated ones are not considered better. One reason for this may be the abstract nature of the nouns;
4. **G$\overline{\text{s}}$ & $\overline{\text{E}}$**: the system does not produce anything good enough to represent the noun. This is the worst situation.

Despite stating that the number of nouns with existing emoji is 927 (Table 1), the number of nouns well-represented with existing emoji is only 791 (Table 2, side b). The number of nouns for which the system is not able to produce an adequate emoji is 365 ( (*bad*E G$\overline{\text{s}}$) + ($\overline{\text{E}}$ G$\overline{\text{s}}$) + (*bad*E $\overline{\text{G}}$) + ($\overline{\text{E}}$ $\overline{\text{G}}$) ). This means that the system is able to present the user with representative emoji for 1144 nouns out of 1509 (an increase of 44.63% when compared to the initially well-represented 791 nouns) – see examples in Fig. 1. It is important to bear in mind that the initial number of well-represented nouns would be even lower if we did not consider the emoji retrieved using non-official semantic knowledge (gathered from EmojiNet). Moreover, some of the nouns are abstract and thus highly difficult to represent (e.g. *everyone*). Other nouns do not have any representative emoji, despite having a great number of retrieved ones.

## 5 Conclusion and future work

We presented and described *Emojinating* – a platform which searches for existing emoji and automatically generates new ones, based on a user-introduced word. It combines Semantic Network exploration with visual blending. In order to assess the system's quality in terms of production and output, we produced representations for 1509 nouns from the New General Service List. The system was able to produce emoji for the majority of the nouns, achieving novelty and good quality of representation.

We consider that there is a large range of possible applications for the system, e.g. aiding in ideation, helping in icon design (generated representations should

not be seen as final result as adjustment may be necessary, e.g. legibility issues) or even providing resources for information visualisation (as described in [11]).

Future enhancements include: (i) extending the evaluation to double-word concepts, (ii) increasing the number of evaluators, (iii) studying the relation between nature of nouns and system's performance, and (iv) distinguishing between official emoji and user-associated ones.

**Link** *Emojinating* will be available at `http://rebrand.ly/emojinatingICCBR`.

**Acknowledgements** This research is partially funded by: Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grant SFRH/BD/120905/2016. This work includes data from ConceptNet 5, which was compiled by the Commonsense Computing Initiative. ConceptNet 5 is freely available under the Creative Commons Attribution-ShareAlike license (CC BY SA 4.0) from http://conceptnet.io. The included data was created by contributors to Commonsense Computing projects, contributors to Wikimedia projects, Games with a Purpose, Princeton University's WordNet, DBPedia, OpenCyc, and Umbel.

## References

1. Abbing, R.R., Pierrot, P., Snelting, F.: Modifying the universal. Executing Practices p. 33 (2017)
2. Ai, W., Lu, X., Liu, X., Wang, N., Huang, G., Mei, Q.: Untangling emoji popularity through semantic embeddings. In: ICWSM. pp. 2–11 (2017)
3. Barbieri, F., Ronzano, F., Saggion, H.: What does this emoji mean? a vector space skip-gram model for twitter emojis. In: LREC (2016)
4. Browne, C.: A new general service list: The better mousetrap weve been looking for. Vocabulary Learning and Instruction 3(1), 1–10 (2014)
5. Confalonieri, R., Corneli, J., Pease, A., Plaza, E., Schorlemmer, M.: Using argumentation to evaluate concept blends in combinatorial creativity. In: Proc. of the Sixth Int. Conference on Computational Creativity. pp. 174–181 (2015)
6. Correia, J., Martins, T., Martins, P., Machado, P.: X-faces: The exploit is out there. In: Proc. of the Seventh Int. Conference on Computational Creativity (2016)
7. Cramer, H., de Juan, P., Tetreault, J.: Sender-intended functions of emojis in us messaging. In: Proc. of MobileHCI 2016. ACM (2016)
8. Cunha, J.M., Gonçalves, J., Martins, P., Machado, P., Cardoso, A.: A pig, an angel and a cactus walk into a blender: A descriptive approach to visual blending. In: Proc. of the Eighth Int. Conference on Computational Creativity (2017)
9. Cunha, J.M., Martins, P., Cardoso, A., Machado, P.: Generation of concept-representative symbols. In: Workshop Proc. ICCBR 2015. CEUR (2015)
10. Cunha, J.M., Martins, P., Machado, P.: How shell and horn make a unicorn: Experimenting with visual blending in emoji. In: Proc. of the Ninth Int. Conf. on Computational Creativity (ICCC 2018) (to appear)
11. Cunha, J.M., Polisciuc, E., Martins, P., Machado, P.: The many-faced plot:strategy for automatic glyph generation. In: Proc. of the 22st International Conference Information Visualisation (IV), 2018 (to appear) (2018)

12. Danesi, M.: The semiotics of emoji: The rise of visual language in the age of the internet. Bloomsbury Publishing (2017)
13. Dimson, T.: Emojineering part 1: Machine learning for emoji trends (2015)
14. Donato, G., Paggio, P.: Investigating redundancy in emoji use: Study on a twitter based corpus. In: Proc. of WASSA 2017 (2017)
15. Dürscheid, C., Siever, C.M.: Beyond the alphabet–communcataion of emojis. Kurzfassung eines (auf Deutsch) zur Publikation eingereichten Manuskripts (2017)
16. Eisner, B., Rocktäschel, T., Augenstein, I., Bosnjak, M., Riedel, S.: emoji2vec: Learning emoji representations from their description. In: Proc. SocialNLP (2016)
17. Emogi, R.T.: 2015 Emoji Report. `http://cdn.emogi.com/docs/reports/2015_emoji_report.pdf` (2015), [Online; accessed Feb. 2018]
18. Guibon, G., Ochs, M., Bellot, P.: From emojis to sentiment analysis. In: WACAI (2016)
19. Gustafsson, V.: Replacing words with emojis and its effect on reading time. USCCS 2017 (2017)
20. Herring, S., Dainas, A.: nice picture comment! graphicons in facebook comment threads. In: Proc. of the 50th Hawaii Int. Conference on System Sciences (2017)
21. Hu, T., Guo, H., Sun, H., Nguyen, T.v.T., Luo, J.: Spice up your chat: The intentions and sentiment effects of using emoji. arXiv preprint arXiv:1703.02860 (2017)
22. Kelly, R., Watts, L.: Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships. Experiences of Technology Appropriation: Unanticipated Users, Usage, Circumstances, and Design (2015)
23. Lebduska, L.: Emoji, emoji, what for art thou? (2014)
24. Miller, H., Thebault-Spieker, J., Chang, S., Johnson, I., Terveen, L., Hecht, B.: Blissfully happy or ready to fight: Varying interpretations of emoji. Proc. of ICWSM 2016 (2016)
25. Novak, P.K., Smailović, J., Sluban, B., Mozetič, I.: Sentiment of emojis. PloS one 10(12) (2015)
26. Pereira, F.C., Cardoso, A.: The boat-house visual blending experience. In: Proc. of the Symposium for Creativity in Arts and Science of AISB 2002 (2002)
27. Pohl, H., Domin, C., Rohs, M.: Beyond just text: Semantic emoji similarity modeling to support expressive communication. ACM TOCHI-17 24(1), 6 (2017)
28. Puyat, M.: Emotigan: Emoji art using generative adversarial networks (2017), cS229: Machine Learning Course, Stanford University
29. Radpour, D., Bheda, V.: Conditional generative adversarial networks for emoji synthesis with word embedding manipulation (2017)
30. Rodrigues, D., Prada, M., Gaspar, R., Garrido, M.V., Lopes, D.: Lisbon emoji and emoticon database (leed): norms for emoji and emoticons in seven evaluative dimensions. Behavior research methods 50(1), 392–405 (2018)
31. Speer, R., Havasi, C.: Representing general relational knowledge in conceptnet 5. In: LREC. pp. 3679–3686 (2012)
32. Steinbrück, A.: Conceptual blending for the visual domain. Ph.D. thesis, Masters thesis, University of Amsterdam (2013)
33. Wicke, P.: Ideograms as semantic primes: Emoji in computational linguistic creativity (2017)
34. Wijeratne, S., Balasuriya, L., Sheth, A., Doran, D.: Emojinet: An open service and api for emoji sense discovery. In: Proc. of ICWSM-17 (2017)
35. Wijeratne, S., Balasuriya, L., Sheth, A.P., Doran, D.: A semantics-based measure of emoji similarity. In: Proc. of WI-17 (2017)
36. Xiao, P., Linkola, S.: Vismantic: Meaning-making with images. In: Proc. of the 6th Int. Conference on Computational Creativity, ICCC-15 (2015)

# ICCBR Video Competition 2018

Competition at the
Twenty-Sixth International Conference on
Case-Based Reasoning
(ICCBR 2018)

Stockholm, Sweden
July 2018

Editors:
Michael W. Floyd, Knexus Research, National Harbor, MD, USA
Brian Schack, Indiana University, Bloomington, IN, USA

# Preface

Inspired by the precedent-setting video competition sponsored by the Association for the Advancement of Artificial Intelligence, we ventured to undertake our inaugural video competition at the International Conference on Case-Based Reasoning last year. We aspired that the videos (strictly limited to no more than five minutes each) would educate students, highlight interesting research, demonstrate relevant applications, and even entertain. During breaks between the sessions, the discussions of both the competitors and the spectators filled the hallways of the conference venue at the Norwegian University of Science and Technology with a palpable excitement. Even after returning home, participants continued to electronically share the nominees and winning videos with their colleagues around the world.

Propelled by this success, we began in earnest to organize a fitting sequel at the conference this year. Fortunately, the competitors again exceeded our expectations in both technical content and production value. Therefore, over the following pages, we share the abstracts of five peer-reviewed and accepted submissions. Of course, if a picture is worth a thousand words, as the classic idiom asserts, then so much more a video. Thus we urge you to browse the competition website at the following address to watch the videos for yourself. And, keep in mind your favorites because, starting this year, instead of an awards committee, we're opening up the voting to everyone attending the conference in person. Thanks to the conference organizers, program committee, and competitors, we hope that you enjoy the videos as much as we do.

`http://sce.carleton.ca/~mfloyd/ICCBRVC2018/`

July 2018

*Michael Floyd*
*Brian Schack*

# Evolutionary Computations and Case-Based Reasoning: A Brief Survey

Hayley Borck

Adventium Labs,
111 Third Avenue South, Suite 100, Minneapolis, Minnesota, USA

**Abstract.** Genetic Algorithms (GA) and Evolutionary Computations (EC) are effective techniques, for achieving optimal or near optimal solutions which have gained explicit attention of researchers over the last decade, and is still growing each year. Many current research areas in Case-Based Reasoning (CBR) such as case adaptation, feature weight selection, case base creation and maintenance, and case injection can be solved with help from optimization techniques. Evolutionary Computations can also benefit from Case-Based Reasoning through techniques which create or inject individuals in the population. CBR and EC can also be used in conjunction as separate systems each solving a part of a larger problem. This video is a brief survey of EC and CBR hybrid systems which highlights several works in different categories of the types of hybrid system. In addition to the systems highlighted, a more complete set of citations of the body of work is included in the credits.

**Keywords:** Case-Based Reasoning, Evolutionary Computations, Genetic Algorithms, Survey

# Knowledge Tradeoffs in Case-Based Reasoning*

Devi Ganesan and Sutanu Chakraborti

Artificial Intelligence and Databases Lab,
Indian Institute of Technology Madras, Chennai, India

**Abstract.** Case-based reasoning (CBR) is a problem-solving paradigm that re-uses the solution of past experiences to solve new problems. Within a case-based reasoner, the domain knowledge is distributed across four knowledge containers namely Case Base, Vocabulary, Similarity, and Adaptation. It is a known fact in the CBR community that knowledge can be interchanged between containers. However, the explicit interplay between them, and how this interchange is affected by the knowledge richness of the underlying domain is not yet fully understood. We attempt to bridge this gap by proposing footprint size reduction as a measure for quantifying knowledge tradeoffs between containers. In this video, we use entertaining examples to introduce case-based reasoning and footprint set. Further, we motivate the proposed measure derived from footprint set to quantify knowledge tradeoffs between containers. Due to time constraints, the scope of this video does not include the evaluation of the proposed measure.

**Keywords:** Case-Based Reasoning, Knowledge Containers, Footprint Set, Footprint Size Reduction, Knowledge Tradeoffs

* This video accompanies the paper: Ganesan, D., and Chakraborti, S. (2018). An Empirical Study of Knowledge Tradeoffs in Case-Based Reasoning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence.*

# Medical CBR Assistant System: Web-based Collaborative Learning Platform*

Sara Nasiri, Katharina Brenner, Christopher Göbel, Marc Wildermuth, Kevin Klöckner, Oliver Koch, Tenantsa Balaye N'kantio, Johnson Momo Kagho, Francis Kenne Wamba, and Madjid Fathi

Institute of Knowledge Based Systems and Knowledge Management, University of Siegen, Siegen, Germany

**Abstract.** Case-based reasoning methodology is utilized to develop a web-based collaborative learning platform for medical students. The main objective of this platform is learning how to diagnose different diseases-cases in a playful way with different modules e.g., lesson, Medduell and checker. Students can also exchange their knowledge and experiences with the fellow students and their professors which are the domain expert of our CBR system. This platform is developing by informatics and business informatics students involved in the MedAusbild[1] student group project (SS2018) based on the core algorithm and content of DePicT Dementia CLASS. Our main idea is to create a learning assistant system for new medical students at the University of Siegen. Therefore, we have selected CBR methodology for our proposed platform to perform adaptive learning with the positive development-loop between developers/users (students) inside the university to learn effectively, retain the case base efficiently, and update the whole CBR system dynamically. Further work will involve extending the development of the proposed platform in light of current evaluation results and based on the new features and cases for the other diseases.

**Keywords:** Case-Based Reasoning, Medical Education, DePicT Dementia CLASS

* This video accompanies the papers: Nasiri, S., and Fathi, M. (2017). Case Representation and Similarity Assessment in a Recommender System to Support Dementia Caregivers in Geriatric and Palliative Care. In *Proceedings of the Workshop on Process Oriented Case-based Reasoning at the 25th International Conference on Case-Based Reasoning*, 157-166.; Nasiri, S., Klingauf, K., Li, D., Ortmann, J., and Fathi, M. (2017). DePicT Dementia CLASS: Medical CBR Learning Assistant System. In *Proceedings of the Video Competition at the 25th International Conference on Case-Based Reasoning*, 291.

[1] https://www.eti.uni-siegen.de/ws/projekte/medausbild/index.html.en?lang=en

# AROMatiC: AbstRactiOn Model Comparison*

Manuel Striani

Computer Science Department,
University of Torino, Torino, Italy

**Abstract.** Process model comparison can be exploited to assess the quality of organizational procedures, to identify non-conformances with respect to given standards, and to highlight critical situations. Sometimes, however, it is difficult to make sense of large and complex process models, while a more abstract view of the process would be sufficient for the comparison task. In this paper, we show how process traces, abstracted on the basis of domain knowledge, can be provided as an input to process mining, and how abstract models (i.e., models mined from abstracted traces) can then be compared and ranked, by adopting a similarity metric able to take into account penalties collected during the abstraction phase. The overall AROMatiC framework has been tested in the field of stroke management, where we were able to rank abstract process models more similarly to the ordering provided by a domain expert, with respect to what could be obtained when working on non-abstract ones.

**Keywords:** Semantic Process Mining, Knowledge-based Trace Abstraction, Process Model Comparison, Medical Applications, Stroke Management

* This video accompanies the paper: Leonardi, G., Striani, M., Quaglini, S., Cavallini, A., and Montani, S. (2018). From knowledge-based trace abstraction to process model comparison. In *Proceedings of the Workshop on Synergies between CBR and Machine Learning at the 26th International Conference on Case-based Reasoning.*

# C2C Weighting and C2C Trace Retrieval*

Xiaomeng Ye

Indiana University Bloomington,
Bloomington, USA

**Abstract.** In this video, we first talk about feature weighting methods in k-nearest neighbors algorithm, which is based on the assumption that cases of the same class share similar features. We then introduce class-to-class (C2C) weighting, which is based on the assumption that cases of two classes are different in a consistent manner. C2C weighting learns the difference pattern between cases of different classes, and reuses the learned pattern in classification tasks. When using C2C weighting, a comparison between a query and a case not only tells us whether they match or not, but also suggests where else to look if they do not match. A series of such suggestions build up the traces. Using these suggestions, we invent a fast retrieval method called C2C trace retrieval.

**Keywords:** k-Nearest Neighbor, Class-to-Class Weighting, Feature Weighting

* This video accompanies the paper: Ye, X. (2018). The Enemy of My Enemy is My Friend: Class-to-class Weighting in K-Nearest Neighbors Algorithm. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*.

# ICCBR 2018 Doctoral Consortium

Cindy Marling[1] and Antonio A. Sánchez-Ruiz[2]

[1] Ohio University, Athens, Ohio, USA
[2] Universidad Complutense de Madrid, Spain

## Preface

This year marks the tenth anniversary of the ICCBR Doctoral Consortium (DC). The DC was designed to nurture PhD candidates by providing them with opportunities to explore and obtain mutual feedback on their research, future work plans, and career objectives with senior case-based reasoning (CBR) researchers, practitioners and peers. We are proud to carry on the tradition with a cohort of seven doctoral students from five different countries.

PhD candidates who applied to the program submitted summaries of their doctoral research. In their research summaries, they detailed the problems they are addressing, outlined their proposed research plans, and described progress to date. Accepted applicants were paired with mentors, who helped them to refine their research summaries in light of reviewer feedback. The updated research summaries, which appear in this volume, were then orally presented at the ICCBR DC in Stockholm, Sweden, on July 10, 2018.

This year's participants presented a broad array of ongoing CBR research. Marta Caro-Martínez discussed the design of case-based recommender systems with new explanation techniques for explainable artificial intelligence (XAI). Zohreh Dannenhauer presented work on case-based explanation for goal-driven autonomous agents. Amar Jaiswal described case-based treatment recommendation for non-specific musculoskeletal disorders. Maximiliano Miranda explained how CBR can help to create artificial agents that play video games in a more human-like manner. Jakob Michael Schoenborn described the importance of generating trustworthy explanations and the minimal amount of knowledge necessary for a CBR system to do so. Eriya Terada presented a textual CBR approach to interactively assisting users with writing tasks. Anjana Wijekoon discussed reasoning with multi-modal sensor streams for m-Health applications.

We gratefully acknowledge support from the National Science Foundation (NSF) and the Artificial Intelligence Journal (AIJ), which helped to defray the cost of student participation in the DC. We also thank all of the students, mentors, and program committee members who worked so hard to make the DC a success. We would especially like to thank our invited speaker, David Leake, for sharing his insight and invaluable advice with the next generation of CBR researchers. We trust that this tenth annual ICCBR doctoral consortium was of interest and benefit to the student participants and to the CBR research community as a whole, and we look forward to the next ten years of ICCBR DCs.

## Students and *Mentors*

| | |
|---|---|
| Marta Caro-Martínez | Universidad Complutense de Madrid, Spain |
| *Jean Lieber* | *LORIA, France* |
| Zohreh Dannenhauer | Wright State University, USA |
| *Kerstin Bach* | *Norwegian University of Science and Technology, Norway* |
| Amar Jaiswal | Norwegian University of Science and Technology, Norway |
| *Michael Floyd* | *Knexus Research, USA* |
| Maximiliano Miranda | Universidad Complutense de Madrid, Spain |
| *Stelios Kapetanakis* | *University of Brighton, UK* |
| Jakob Michael Schoenborn | University of Hildesheim, Germany |
| *David Leake* | *Indiana University, USA* |
| Eriya Terada | Indiana University, USA |
| *Antonio A. Sánchez-Ruiz* | *Universidad Complutense de Madrid, Spain* |
| Anjana Wijekoon | The Robert Gordon University, UK |
| *Isabelle Bichindaritz* | *State University of New York at Oswego, USA* |

## Program Chairs

| | |
|---|---|
| Cindy Marling | Ohio University, USA |
| Antonio A. Sánchez-Ruiz | Universidad Complutense de Madrid, Spain |

## Program Committee

| | |
|---|---|
| Agnar Aamodt | Norwegian University of Science and Technology, Norway |
| David Aha | Naval Research Laboratory, USA |
| Klaus-Dieter Althoff | DFKI / University of Hildesheim, Germany |
| Kerstin Bach | Norwegian University of Science and Technology, Norway |
| Ralph Bergmann | University of Trier, Germany |
| Isabelle Bichindaritz | State University of New York at Oswego, USA |
| Sarah Jane Delany | Dublin Institute of Technology, Ireland |
| Michael Floyd | Knexus Research, USA |
| Stelios Kapetanakis | University of Brighton, UK |
| David Leake | Indiana University, USA |
| Jean Lieber | LORIA, France |
| Stefania Montani | University of Piemonte Orientale, Italy |
| Santiago Ontanon | Drexel University, USA |
| Luigi Portinale | University of Piemonte Orientale, Italy |
| Nirmalie Wiratunga | The Robert Gordon University, UK |

# Recommender Systems and Explanations Based on Interaction Graphs and *Link Prediction* Techniques*

Marta Caro-Martinez

Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Spain
email: martcaro@ucm.es
Guillermo Jiménez-Díaz - gjimenez@ucm.es (supervisor)
Juan A. Recio-García - jareciog@ucm.es (supervisor)

**Abstract.** This work presents the motivation, research plan, goals and progress to date for my PhD. The main research line of my PhD is the study and design of recommender systems based on interaction graphs and *link prediction* techniques, the way of developing explanations for this type of recommenders and the way of visualizing explanations for getting the best user satisfaction.

## 1 Introduction

Nowadays, the amount of information that we can find on the Internet is immense. Although it provides an staggering number of products and experiences for users, it also can be a problem when users do not know what products are more suitable for their needs. Recommender systems play an important role in the resolution of the information overload problem and provide techniques to suggest interesting products for users [16]. These systems help users to pick new products more efficiently and effectively according their preferences, making easier the classification task and the acquisition of items through new technologies [17]. Nevertheless, many times, recommender systems act as black boxes for users: they recommend a list of personalized items without justifying why these items are the most appropriated for the target user. Therefore, users do not understand recommenders; they do not trust them and intention to reuse them decreases. In recent years, research in explanations for recommender systems has increased with the goal of solving this problem and improving user perception and acceptance of this type of systems [6].

Traditionally, recommender systems are based on previous user interactions or are based on interesting products for similar users to the target user. Another way of tackling the problem lies in representing the user interactions as a user-graph or an item-graph and using these graphs for making recommendations.

* Supported by the UCM (Research Group 921330) and the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R)

This solution presents advantages over traditional methods: it does not require ratings from users of recommended products or additional information on items.

*Link prediction* is a technique used in social network analysis [5] that determinates which new links will appear or disappear between graph nodes using similarity measures [11, 20]. A recommendation can be seen as a *link prediction* problem: given a graph that represents interactions between users and products, we can predict the appearance of a new link between a user node and an item node. There are studies that use *link prediction* techniques as a recommender method [3, 21, 4] and other ones that use them as an explanation method [1]. But there is still more work to do in this field and techniques from social network analysis unexplored.

On the other hand, research in visual explanations is increasing. Visualization facilitates the user understanding about why a system recommends an item. In explanations that we can find in the state-of-the-art of the problem, the most common justifications are described by natural language and textual information. Recently, new studies have begun to try new ways of providing explanations, more visual and interactive [19, 6, 9].

## 2 Research Plan and Goals

The main goal of this thesis is the study and development of explanations in recommender systems based on interaction graphs and methods of social network analysis. Moreover, we will study and implement new different visualization modes of explanations for this type of recommender systems. We consider four goals to develop my PhD thesis:

- **Objective 1**. Study *Explainable Artificial Intelligence (XAI)* and explanations for recommender systems and decision support systems. The final goal of this study is to generate a formalized classification model for explanations. This model can help to design explanations in a better way, making developers take into account all the aspects that the explanation must have for their purposes. The most important contribution of this work is the formalization, that will help to build an ontology for explanations in recommender systems.

- **Objective 2**. Study, design and develop different recommender methods based on interaction graphs and techniques of social network analysis, like *link prediction*. A graph can represent how users interact with items (they buy products, they rank movies, etc.) if we use nodes as users and items and we use links as an interaction. With *link prediction* techniques, we can know if a link will be formed between two nodes. If two nodes (a user node $X$ and an item node $Y$) have a probability of joining together, then this item $Y$ can

be recommended to this user $X$ as an interesting item for $X$.

- **Objective 3**. Explain recommendations of graph-based systems with *link prediction* techniques and more social network analysis techniques. As in other types of recommender systems, including explanations for graph-based recommender systems will improve the user trust, acceptance and satisfaction. The most used techniques for implementing explanations are machine learning and knowledge-based methods [10]. However, recommender systems based on interaction graphs do not provide the information required by these algorithms, like user preferences or background context. We only own information about the user-item interaction on that specific system. *Link prediction* techniques can help us to solve the problem and design and develop good explanations for recommendations based on graphs [1].

- **Objective 4**. Study different visualization modes for explanation systems based on graphs and study which ones offer better results for user satisfaction and usability. We will examine alternatives and new approaches and we will evaluate which ones adapt better to this type of explanations.

## 3 Motivating example

In order to illustrate the work around this thesis proposal, we depict an example of explanations in a recommender system of TV series. It works using the interaction graph formed when a user watch a TV series, with user and TV series as nodes, which are linked if a user $U_i$ has watched the TV series $i_j$. *Link prediction* techniques for recommendation will be applied over the user-user graph, created as a projection of the original graph. In the user-user graph, nodes represent users and two nodes are linked when both users have watched at least one TV series in common. Figure 1 shows an example where we can find the TV series that each user already watched and the corresponding user-user graph.

The recommendations are based on the similarity between nodes, which is calculated using the *link prediction* similarity measure called Common Neighbors (CN): the similarity between two nodes is the number of neighbors that they have in common. Following the previous example, the most similar user to $U_1$ is $U_3$ because she is the user with most common neighbors with $U_1$. Therefore, the system will recommend the TV series "Westworld" to $U_1$ and the explanation provided looks like that "Westworld" is a TV series watched by $U_3$ (and not already watched by $U_1$), who is the most similar user to her.

Regarding the previous example, we could address different ways to visualize the explanation using graphs, analyzing the advantages and drawbacks that should provide. For example, we could display a region of the graph drawn in Figure 1 with visual modifications in the nodes (color, size or shapes) or the links (thickness, length or tags) in order to clarify user $U_1$ that $U_3$ is similar to her, so it leads the recommender system to suggest "Westworld".

| | West World | Stranger Things | Doctor Who |
|---|---|---|---|
| $U_1$ | | X | |
| $U_2$ | | | X |
| $U_3$ | X | | |
| $U_4$ | X | X | |
| $U_5$ | X | X | |
| $U_6$ | X | X | X |

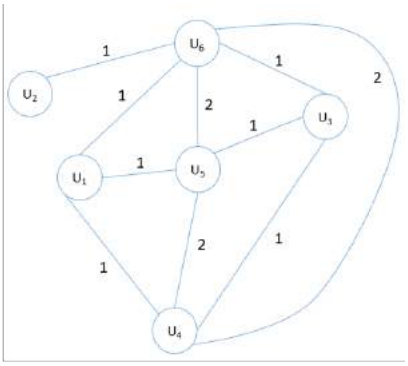



Fig. 1: Example with the TV series that each user already watched and the corresponding user-user graph

## 4 Description of the progress to date

Objectives 1 and 2 have been already started. We started objective 2 two years ago, finishing the work last year. We applied the new recommender methods, outlined in the previous section, to an online judge, a platform where users can submit solutions for programming problems in order to achieve a verdict for this solution. We designed two recommendation approaches using *link prediction* techniques: based on a user-user graph and based on an item-item graph. The results were described in a Master thesis (in Spanish)[12] and we published two research papers [8, 2].

Objective 1 has been developed during this year. It has been focused on the study of the state-of-the-art of the problem [18, 13], on taxonomies of explanations [14, 15] and on studies about new techniques, tools or explanation systems [10, 7]. Currently, we are working on a new classification model of explanations for recommendations that has been submitted to the Workshop on CBR for the explanation of Intelligent Systems at ICCBR18.

## References

1. Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Who to follow and why: link prediction with explanations. In *20th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pages 1266–1275. ACM, 2014.
2. Marta Caro-Martinez and Guillermo Jimenez-Diaz. Similar Users or Similar Items? Comparing Similarity-Based Approaches for Recommender Systems in Online Judges. In *International Conference on Case-Based Reasoning*, pages 92–107. Springer, 2017.
3. Hsinchun Chen, Xin Li, and Zan Huang. Link prediction approach to collaborative filtering. In *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, pages 141–142. IEEE, 2005.

4. Nitin Chiluka, Nazareno Andrade, and Johan Pouwelse. A link prediction approach to recommendations in large-scale user-generated content systems. In *European Conference on Information Retrieval*, pages 189–200. Springer, 2011.
5. Borko Furht. *Handbook of social network technologies and applications*. Springer Science & Business Media, 2010.
6. Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.
7. Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.
8. Guillermo Jimenez-Diaz, Pedro P. Gómez-Martín, Marco A. Gómez-Martín, and Antonio A. Sánchez-Ruiz. Similarity metrics from social network analysis for content recommender systems. *AI Communications*, 30(3-4):223–234, 2017.
9. Béatrice Lamche, Ugur Adıgüzel, and Wolfgang Wörndl. Interactive explanations in mobile shopping recommender systems. In *Joint Workshop on Interfaces and Human Decision Making in Recommender Systems*, page 14, 2014.
10. O-Joun Lee and Jason J Jung. Explainable movie recommendation systems by using story-based similarity. 2018.
11. Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.
12. Marta Caro Martínez and Guillermo Jiménez Díaz. Sistemas de recomendación basados en técnicas de predicción de enlaces para jueces en línea. 2017.
13. David McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, 2005.
14. Ingrid Nunes and Dietmar Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3-5):393–444, 2017.
15. Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, 24(3):555–583, 2012.
16. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender systems handbook*. Springer, 2015.
17. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
18. Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE, 2007.
19. Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56. ACM, 2009.
20. Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.
21. Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):046115, 2007.

# Case-Based Explanation for Goal Monitoring

Zohreh A. Dannenhauer

Wright State University, Dayton OH 45324, USA
alavi.3@wright.edu

## 1 Introduction

Real world autonomous agents will be expected to manage their behavior across many complex situations and to solve severe problems that arise while pursuing their goals. This work considers agents that are operating in partially observable worlds which are changing due to external events. These events are not always visible for these agents and may cause them to fail in achieving their goals and plans. The agent should be able to explain the cause of these changes and adjust their goals and plans to perform competently. Specifically this work addresses the research problem of how to best monitor the world for those changes that affect the agent's goals.

My research focuses on Goal-Driven Autonomous (GDA) Agents [9]. Goal-driven autonomy involves recognizing unexpected or possibly new problems, explaining the causal factors underlying the problems and generating goals to remove the cause of the problems in order to achieve the given task. The GDA agent itself is expected to identify situations in which new goals are to be formulated or current goals changed and abandoned [3, 8]. Identification of these situations is where plan monitors and goal monitors are needed. My previous work showed that plan monitors can enable a planner to respond to changes in the world during plan generation without having to restart planning from the beginning [7]. However, there are some changes that affect the agent's goal which are outside the scope of plan monitors. Hence the need for goal monitors which is the focus of my current work.

We adopt the classic planning formalism where goals are a state or subset of a state that an agent tries to achieve, where any given goal $g_i \subseteq s \in S$. The agent's goal agenda $\hat{G} = \{g_1, ..., g_c, ..., g_n\}$ contains the current goal $g_c$ and any other goals it may pursue. A goal monitor includes two major conditions. First the monitor encapsulates environmental conditions whose change signals the need for goal reconsideration. Second the monitor includes a specification of the response (e.g., goal abandonment) if perceptions detect the first condition.

If the agent formulates a goal on its own, it should have a reason for doing so. These reasons establish the means for monitoring that the goal is still worth achieving. Explanatory goal monitors assume that the goal was formulated in response to a discrepancy between the agent's expectations and observations [1, 6]. Meta-AQUA [5] is a story understanding system that applies the methods of case-based reasoning to understand what event caused the expectation failure. It provides the causal antecedents that cause the failure. These antecedents provide the environmental conditions that must persist for the goal to remain valid. Goal monitors observe these conditions and provide the

response if they change. Goal monitors are created using information from explanation cases in Meta-AQUA.

The Meta-AQUA system will be used to detect and explain the problems that arise while performing tasks. Meta-AQUA works by retrieving *explanation patterns* (XPs) cases [11] and applying them to the given situation to explain the failure in terms of what it occurred. XPs are causal structures that explain a state by presenting the prior events influencing these states [5, 10]. When an anomalous or otherwise interesting state is detected, the system builds an explanation of the event, incorporating it into the preexisting model of the situation [5]. An XP is applicable if its antecedent can be unified with the current situation.

Until now, Meta-AQUA/MIDCA has only been used for fully observable domains. In partially observable domains, the GDA agent might consider multiple hypotheses. If it retrieves two or more possible explanations, each could be equally likely initially. But if new information is observed, the system should increase the probability of one of the explanations. One of the contributions of this work will be adapting the case retrieval mechanism in Meta-AQUA to handle partial matching of explanation patterns in situations where more than one hypothesis may be true. The goal monitors are created for all possible hypotheses, and when the agent observes new information, one or more of the goal monitors may fire which in turn will lead to a possible change in the goals the agent pursuing. Here is an example to make these ideas more concrete.

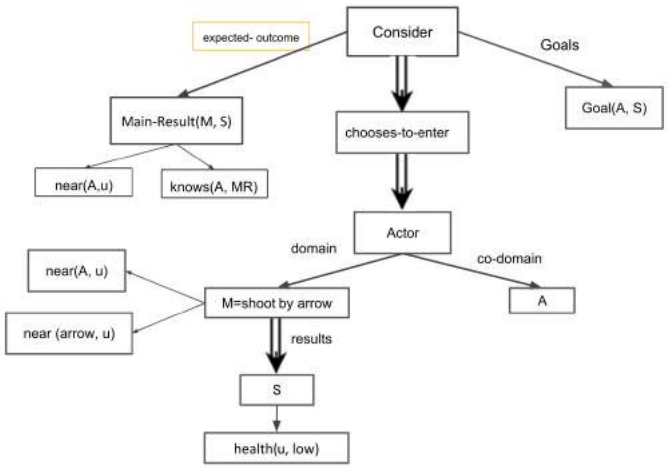



**Fig. 1.** Case representation of XP-RESULTS-FROM-ACTION ←STATE explanation. A is the agent, M the action he chooses to do, and S outcome for A as a result of doing that action. A chooses to perform M knowing the outcome. Double arrows highlight the main element of the casual chain comprising the explanation. $u$ represents Steve and $A$ is the skeleton.

## 2 A Motivating Example in Minecraft

In the Minecraft game[1], the character named Steve explores an infinite 3D virtual world while gathering resources and surviving dangers. Minecraft has become popular evaluation domain for artificial intelligence research because it has some properties of real-world domains: partially observable, 3-dimensional, infinite state space, and real-time. There are different factors that can damage Steve's health like falling in lava, getting shot by a skeleton archer, triggering an arrow trap, and low hunger level.

If the agent's (Steve) health decreases, Meta-AQUA finds it anomalous because it is expecting good health (health-value $\geq 20$), but it observes a lower health value. Low health is the consequence of some event that has occurred. Meta-AQUA poses a question *what could have make my health go down?*. It then retrieves XP cases to answer the question using index *low-health*. Those XPs are applicable that their consequences are in the current set of belief. One possible XP is shown in Figure 1. This XP is applicable because $nearby(arrow, u)$ is true in the current state, where $u$ is Steve. The XP shown in Figure 1 hypothesizes that Steve was shot by an arrow and a skeleton caused the shooting (where A is the skeleton and M is the event of shooting an arrow). This XP relates the action (shooting arrow) that the actor performed to the outcome of those actions. The XP asserts that the shot caused the health to decrease.

A new goal is created to counter the antecedents of the XP, which is that there is an actor A near the agent Steve. In this case, it can be blocked by removing the actor of shooting and a goal to destroy the skeleton is generated. The goal monitor for this newly generated goal tracks the condition that the actor A (the skeleton) is $near(A, u)$. If this condition changes in the state, the goal monitor will fire, and the GDA process for goal management will know to reconsider pursuing this goal.

## 3 Agent Design in the MIDCA Architecture

The metacognitive integrated dual-cycle architecture (MIDCA) [2, 10] is a cognitive architecture that models both cognition and metacognition for intelligent agents. It consists of "action-perception" cycles at both the cognitive level and the metacognitive level. In general, a cycle performs problem-solving to achieve its goals and tries to comprehend the resulting actions and those of other agents. The output side of each cycle consists of intention, planning, and action execution, whereas the input side consists of perception, interpretation, and goal evaluation (see Figure 2). The *Interpret* phase in MIDCA is the core of this research. It is implemented as a GDA procedure and analyzes the current state to determine which new goals should be pursued. In our scenario, this is the phase that detects an anomaly and formulates new goals in response. Meta-AQUA performs explanation and goal monitors are created in the Interpret phase.

Meta-AQUA relies on general domain knowledge, a case library of prior explanation schemas and a set of general XPs that are used to characterize useful explanations involving that background knowledge. These knowledge structures are stored in a separate memory sub-system and communicated through socket connections to the rest of MIDCA.

[1] http://www.minecraft.net/

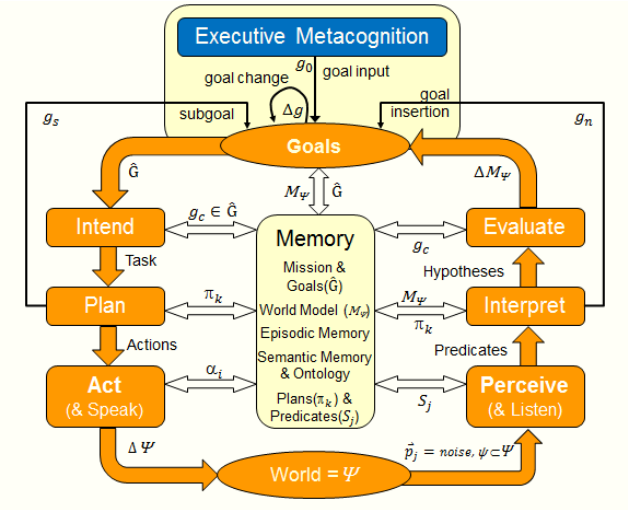



**Fig. 2.** The action-perception cycle in MIDCA[2]. Together Intend, Plan, and Act compose the problem-solving mechanism in the architecture, and Perceive, Interpret, and Evaluation constitute the comprehension mechanism. Case-based explanations are embedded in Interpret.Hypotheses are the explanations that are generated in Interpret. Cases are stored in episodic memory.

## 4 Research Plan / Progress

The research here examines a cognitive mechanism to reason about changes in the environment and allow the agent to change its goal and the associated plan. When the reason for the goal ceases to hold, the agent should also abandon the goal or otherwise change its behavior. I propose goal monitoring as a cognitive process that oversees the continuing benefit of each currently pursued goal and when situations change unexpectedly, decides whether to abandon or change these goals.

This research focuses on the relationship between planning, acting, interpretation, and perception in the MIDCA cognitive architecture. I propose plan monitors and goal monitors as a solution for interleaving these major cognitive processes.

Previously I have studied the problem of how a planner can dynamically adjust its planning search using perception during planning time. Specifically, plan monitors can be used to focus vision to observe the states that form the basis of planning choices. When a feature being monitored changes, the planner will update the plan and alter the planning search. I implemented and evaluated these plan monitors in MIDCA in prior work, demonstrating that there are benefits to adjusting the planning search by taking into account visually detected changes [7].

In extending monitoring capabilities of cognitive agents, I have more recently begun examining the problem of monitoring goals. I categorize goal monitors into two different types based on how the goal is formulated: Operator style and Explanatory style. In Operator style, a set of rules generate goals when their conditions are satisfied in the world. The persistence of these conditions in the future is the target of the goal

monitor. These operator based goal monitors have been implemented and evaluated in the Logistics domain [4].

Currently, I am investigating explanatory style goal monitors, which has been the work I have described in this research summary. Here goals are formulated based on an explanation of a discrepancy. The goal monitors for these goals consider the casual structure of the explanation that was used to generate the goal. To evaluate the performance of MIDCA agent with goal monitors, I will conduct tests in Minecraft domain and compare the result with MIDCA agent with no goal monitors.

## References


1. Cox, M.T.: Perpetual self-aware cognitive agents. AI magazine **28**(1), 32 (2007)
2. Cox, M.T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H.: MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In: AAAI (2016)
3. Cox, M.T., Dannenhauer, D., Kontrakunta, S.: Goal operations for cognitive systems. In: AAAI (2017)
4. Cox, M.T., Dannenhauer, Z.A.: Perceptual goal monitors for cognitive agents in changing environments. Advances in Cognitive Systems (ACS-17) (2017)
5. Cox, M.T., Ram, A.: On the intersection of story understanding and learning. In: Understanding language understanding. pp. 397–433. MIT Press (1999)
6. Dannenhauer, D., Munoz-Avila, H.: Raising expectations in gda agents acting in dynamic environments. In: International Joint Conference on Artificial Intelligence (IJCAI-15) (2015)
7. Dannenhauer, Z., Cox, M.: Rationale-based perceptual monitors. AI Communications Journal **31**(2), 197–212 (2018)
8. Klenk, M., Molineaux, M., Aha, D.W.: Goal-driven autonomy for responding to unexpected events in strategy simulations. Computational Intelligence **29**(2), 187–206 (2013)
9. Munoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., Molineaux, M.: Applying goal driven autonomy to a team shooter game. In: FLAIRS Conference (2010)
10. Paisner, M., Maynord, M., Cox, M.T., Perlis, D.: Goal-driven autonomy in dynamic environments. In: Goal Reasoning: Papers from the ACS Workshop. p. 79. Citeseer (2013)
11. Schank, R.C.: Explanation patterns: Understanding mechanically and creatively. (1986)

# Personalized Treatment Recommendation for non-speciffc Musculoskeletal Disorders in Primary Care using Case-Based Reasoning

Amar Jaiswal

Supervisor: Associate Professor Kerstin Bach[1]
Co-Supervisor: Professor Ottar Vasseljen[2]

[1] Department of Computer Science,
[2] Department of Public Health and Nursing
Norwegian University of Science and Technology, Trondheim, Norway
http://www.idi.ntnu.no, http://www.ntnu.no/ism

## 1 Introduction

Musculoskeletal disorders and their resulting disability are the primary cause of sickness absence within the workforce worldwide if neglected often leads to a long-term or permanent disability [5]. Also, out of five highest occurrences of occupational diseases, musculoskeletal disorders contribute to the most, nearly 40% in Europe[11,6]. A brief description of musculoskeletal disorders could be found in Section 2 of this paper.

The decision making for optimal interventions in primary care for non-specific musculoskeletal disorders are particularly challenging as there is often no specific cause for the patient's condition or pain [9,7,12]. The diagnosis, prognosis, and treatment for the non-specific musculoskeletal disorders are highly patient-specific, which primarily relies on the physiotherapist's previous experiences with or without the support of systematic evidence.

This research focuses on the artificial intelligence (AI) approach of solving the said problem through case-based reasoning (CBR), by assisting physiotherapist in their process of decision making. It will also enable co-decision making, help physiotherapists to share their best practices and introduce novel findings in designing an effective treatment plan for patients.

## 2 Application Domain and Dataset

The application domain of this research is in healthcare, non-specific musculoskeletal disorders in primary care. The musculoskeletal system is made up of the bones of the skeleton, muscles, cartilage, tendons, ligaments, joints, and other connective tissue that supports and binds tissues and organs together[3]. It provides form, support, stability, and movement to the body. It is also responsible for activities like walking, sitting, running, working, etc.

[3] https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0027058/

The term musculoskeletal disorders denote health problems of the musculoskeletal system which includes all forms of ill-health ranging from light, transitory disorders to irreversible, disabling injuries[4]. The typical symptoms of musculoskeletal disorders include pain, weakness, stiffness, joint noises, and decreased range of motion. They are broadly classified as specific and non-specific. The Specific disorders have evident pathology and symptoms. The non-specific disorders are not attributable to a recognizable, known specific pathology, also the symptoms tend to be diffuse and non-anatomical. These symptoms involve pain, discomfort, and numbness without evidence of any discrete pathology.

We use FYSIOPRIM [5] dataset for our research, which is the result of data acquisition done by Department of Public Health and Nursing [6]. The current dataset consists 506 non-specific musculoskeletal disorder patients. Each patient data has 286 features like demographics, symptoms, problems, questionnaires, treatment plans, etc. The capturing of patient details starts with physiotherapist and patient together at primary care; later the patient is asked to fill the rest. There are 356 females and 150 males between the age group of 18 to 88 years. The dataset has 106, 134, 139, and 127 patients with primary complaints about back, neck, shoulder, and widespread (multiple sites) respectively. The dataset is scarce with respect to non-empty features. 60% of the patients (303 of 506) have just 26.76% non-empty features, also, merely 1% i.e. 3 features (age, gender, and primary complaint) have non-empty values for all (506) patients. The dataset suffers from multiple biases majorly self-reporting bias, social desirability bias, and recall bias [2]. Table 1 describes the major feature categories and their admissible feature value ranges. It also illustrates a brief case representation of our case base.

## 3 Research Problem

The primary care physiotherapist examines patients with plenitude variety of symptoms, but most often have only few minutes to decide on the best possible treatment plans [8]. The features of the FYSIOPRIM dataset will form the basis for creating a treatment plan for a patient. The problems of the dataset described in section 2 along with unknown causes of pain, makes it even more difficult for a physiotherapist to create an effective treatment plan [9]. These issues lead to formulate current research problem and motivated to seek a solution based on artificial intelligence (AI) for assisting physiotherapists in the process of their decision making.

Case-based reasoning is a problem-solving paradigm [1], and has an intrinsic commonality the way a physiotherapist suggests and adapts a treatment plan to a new patient, fits well for solving our research problem [3]. This makes the CBR an ideal AI method to be explored for solving the said problems of treating non-specific musculoskeletal disorders in primary care [4]. The current

[4] http://www.who.int/occupational_health/publications/oehmsd3.pdf
[5] http://www.med.uio.no/helsam/english/research/groups/fysioprim/
[6] https://www.ntnu.edu/ism

| Case | | | | |
|---|---|---|---|---|
| **Problem** | | | | |
| **Feature Group** | **Feature** | **Data Type** | **Range** | **Description** |
| Demographics | Age | Numerical | 0 to 110 | Age of the Patient |
| | BMI | Numerical | 1 to 150 | Body Mass Index |
| | Gender | Categorical | 0 to 1 | Sex of the Patient. 0 (female) and 1 (male). |
| | Latency | Ordinal | 1 to 7 | Duration since the patient has contacted PT. |
| | Marital Status | Categorical | 1 to 4 | 1 (married), 2 (divorced), 3 (widow), and 4 (single) |
| Disability and Function | Activity | Ordinal | 1 to 4 | Daily activity level due to pain complaints. 1 (very reduced) to 4 (not reduced) |
| | Physio Diagnosis | Free Text | - | Diagnosis defined by the physiotherapist and not from the referral. |
| | PSFS | Categorical | 0 to 10 | Patient-Specific Functional Scale. Patient in collaboration with physiotherapist defines 3 activities. 0 (No problem to perform activity) to 10 (Not able to perform the activity). |
| Pain Variables | Medication | Categorical | 0 to 1 | Use of medication last week. 0 (no), 1 (yes) |
| | Pain Duration | Ordinal | 1 to 5 | The duration of the pain. 1 (less than 1 month), 2 (1 to 3 months), 3 (3 to 6 months), 4 ( 6 to 12 months), and 5 ( more than 1 year) |
| | Pain Frequency | Categorical | 1 to 4 | The frequency of the pain from daily to less than once per month. 1 (daily), 2 (several days in a week), 3 (once in a week), and 4 (less than once a month). |
| | Pain Site | Numerical | 0 to 10 | Number of pain sites on a body graph. |
| Primary Complaint | Compalint Region | Categorical | 1 to 4 | 1 (Back), 2 (Neck), 3 (Shoulder), and 4 (Widespread). |
| Psychological Factors | Fear Level | Numerical | 0 to 10 | Level of fear the patient thinks that the pain complaints would increase with physical activity. NRS : (0 to 10). |
| | HSCL_10 | Numerical | 1.0 to 4.0 | Mean of Hopkins Symptom Check List-10. |
| | Orebro Sum | Numerical | 0 to 100 | Ørebro Screening questionnaire 10-item total score. Higher the score higher the levels of estimated risk for developing pain-related disability. |
| | PSEQ | Numerical | 0 to 12 | Pain Self Eı cacy Questionnaire-2 item. Higher the score higher the level of self-eı cacy. |
| | Sleep Quality | Ordinal | 1 to 5 | The Quality of the sleep. 1 (Normal Sleep) to 5 (Severe Sleeplessness) |
| | Treatment Belief | Ordinal | 1 to 5 | Level of the belief that physiotherapy will improve patient's current condition /function. 1 (totally agree) to 5 (totally disagree). |
| **Solution** | | | | |
| Treatment Plan | The treatment plan would comprise of treatment advice, exercise, periodic training, therapy, group treatment, joint mobilization, and multiple other treatments for the patient. | | | |

Table 1: Major feature category of FYSIOPRIM dataset.

research investigates how case-based reasoning can be applied in the domain of physiotherapy at primary care. The research focuses on improving treatment recommendation quality along with reduction in physiotherapist's workload and treatment errors for non-specific musculoskeletal disorders in primary care.

### 3.1 Research objectives

- To investigate FYSIOPRIM dataset for designing an effective CBR system.
- To identify effective local and global similarity measures, and seek for the opportunities to improve or design new ones.
- To develop a personalized treatment recommendation system based on CBR.
- To attribute the developed CBR system with co-decision making for physiotherapists and patients in primary care.

### 3.2 Foreseen challenges

- How to deal with the missing feature values?
- How to identify key features from the complete feature space?
- How to define, evaluate and evolve similarity measures?
- How to develop an explainable decision support system for its users?

### 3.3 Expected research outcomes

- Novel similarity measures for features where the target labels are obscure.
- A methodology for designing a CBR system for an intricate domain as MSDs.
- A viable and explainable decision support system for physiotherapists.

## 4 Proposed Research Plan

### 4.1 Literature review and feature analysis

The literature reviews will be an integral part of our entire research process that helps us to position the research work up to the state-of-the-art. Feature analysis and interactions with healthcare professionals through regular meetings are critical to assure comprehension of medical data and mutual understanding of the concepts which are vital to the clinical relevance and success of this research.

### 4.2 Similarity measures, CBR implementation, and publications

Our prime focus is on similarity measures and enhancement along with case modeling for the FYSIOPRIM patient data. Our CBR system design and implementation will be based on myCBR tool [10]. We will explore suitable machine learning techniques for feature selection and adaptation contributing to CBR implementation. Our contribution in the field of CBR would be through multiple publications, demonstrating our reproducible experimental methodologies, results, and feasibility in the domain of healthcare.

### 4.3 Periodic demonstration and improvements to the model

We have engaged 4 physiotherapists who may serve as user group on the front end. There will be a periodic demonstration of work progress to the stakeholders followed by improvements to the developed models.

## 5 Current Progress and Future Plans

The initial FYSIOPRIM dataset is in place, and further data acquisition is currently in progress by the medical team. We are in the first phase of feature analysis for deciding most relevant features to be used as attributes in case modeling. We are in parallel working on the similarity measures for these features.

The target features and solution part of the cases are still in discussion with the medical counterparts of this research. A primitive case model is developed based on an initial understanding of the dataset. The current developed case model and similarity measures are yet to be validated with domain experts for its relevance.

**Our future yearly goals are listed below:**

- **Year 1 :** Baseline implementation of CBR and literature review.
- **Year 2 :** CBR system enhancement with detailed experiments for similarity measures and case base.
- **Year 3 :** Explanation of adaptation results for physiotherapists.
- **Year 4 :** Final CBR system enhancements and thesis compilation.

## References


1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications, 7(1), 39–59 (1994)
2. Althubaiti, A.: Information bias in health research: definition, pitfalls, and adjustment methods. Journal of multidisciplinary healthcare 9, 211 (2016)
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., Szczepanski, T., Aamodt, A., Gundersen, O.E., Mork, P.J.: Case representation and similarity assessment in the selfback decision support system. vol. 1670, pp. 153–154. CEUR-WS.org (2016)
5. Black, C.M.: Sickness absence and musculoskeletal disorders. Rheumatology 51(2), 204–205 (2012)
6. Buckle, P.: Ergonomics and musculoskeletal disorders: overview. Occupational medicine 55(3), 164–167 (2005)
7. Carlson, H., Carlson, N.: An overview of the management of persistent musculoskeletal pain. Therapeutic advances in musculoskeletal disease 3(2), 91–99 (2011)
8. Holzinger, A.: Lecture 7 knowledge and decision: Cognitive science and human–computer interaction. In: Biomedical Informatics. Springer (2014)
9. Malmgren-Olsson, E.B., Armelius, B.Ä.: Non-specific musculoskeletal disorders in patients in primary care: subgroups with different outcome patterns. Physiotherapy Theory and Practice 19(3), 161–173 (2003)
10. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of cbr applications with the open source tool mycbr. In: ECCBR '08: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. pp. 615–629. Springer-Verlag, Berlin, Heidelberg (2008)
11. Steinbuka, I., Clemenceau, A., Venema, A., Heuvel, S., Geuskens, G.: Health and safety at work in Europe (1999-2007). A statistical portrait. European Union; Eurostat (2010)
12. Wiitavaara, B., Fahlström, M., Djupsjöbacka, M.: Prevalence, diagnostics and management of musculoskeletal disorders in primary health care in sweden–an investigation of 2000 randomly selected patient records. Journal of evaluation in clinical practice 23(2) (2017)

# CBR for Imitating the Human Playing Style in Ms. Pac-Man

Maximiliano Miranda

Departamento de Ingeniera del Software e Inteligencia Artificial
Universidad Complutense de Madrid
c/ Profesor Jos Garca Santesmases 9, 28040 Madrid (Spain)
m.miranda@ucm.es

**Abstract.** Imitating video game players is considered one of the most stimulating challenges for the Game AI research community. The goal for a virtual player is not just to beat the game but to show some human-like playing style. We aim to use Case-Based Reasoning and others Machine Learning approaches for creating agents that mimic the playing style of human players.

**Keywords:** Virtual Video Game Player · Human Behavior Imitation · Artificial Intelligence

## 1 Introduction

Researchers studying Artificial Intelligence (AI) who explore agents that mimic human behavior are always looking for problems that are challenging but feasible at the same time, in order to progress their mission of recreating human intelligence in a computer. Imitating video game players has been considered a stimulating challenge for the Game AI research community, and several competitions on developing believable characters have emerged during the last decade [3].

Usually, in games where there are machine-controlled characters (bots) and they play in a more human-like way, human players perceived the game to be less predictable, more replayable, and more challenging than if the bots were hand-coded [6]. Furthermore, in the Digital Game Industry there is a widespread assumption that wherever there is a machine-controlled character, the game experience will benefit from this character to be controlled by the computer in a less "robotic" and more human-like way. For this reason, player modeling in video games has been an increasingly important field of study, not only for academics but for professional developers as well [10].

There is a wide variety of scenarios where these human-like computer bots come into play. They can be used not only to confront the human player, but also to collaborate with him or to illustrate how to succeed in a particular game level to help players who get stuck. It is reasonable to think that these computer-played sequences will be more meaningful if the bot imitates human-like playing style. Another possible application of these "empathetic" bots is to help during

the testing phase in the game production process. Acting as virtual players, these bots could be used to test the game levels, not only checking whether the game crashes or not, but verifying if the game levels have the right difficulty, or finding new ways for solving a puzzle.

Despite the popularity of the Turing test, in the domain of video games there is no formal, rigorous standard for determining how human-like is an artificial agent [2]. Furthermore there is not a clear concept about what a believable AI should achieve, and its expected behavior will vary strongly, depending on what it is supposed to imitate: to emulate the behavior of other players or to create lifelike characters [4]. In the case of our research, we have been focused on emulating the playing style of specific human players, assuming that this emulation models the way the player moves (how it reacts) in the game scene given the current situation of his avatar and the other game entities (items and enemies).

## 2 Research Plan

Our research is focused on simulating the human playing style in video games. Covering distinct imitation methods (direct and indirect), machine learning and other AI paradigms such as artificial neural networks (ANN), case-based reasoning, neuroevolution (NE) or deep learning. Our main goal is to replicate the features that characterize the way humans play in a simulation of this behavior through virtual players that should be able to deceive human judges by making the artificial nature of the virtual player indistinguishable.

### 2.1 Objectives

Our main objective is to investigate methods for human playing style imitation in video games, mainly in action and real time strategy games, and explore how artificial intelligence in virtual players can be modeled in order to obtain sufficiently human-like controllers able to deceive human spectators.

We consider other specific objectives such as:

1. to revise the literature about the approaches followed in the academia and in the video game industry regarding automatic players for video games concerning the playing style imitation.
2. Evaluate the use of AI paradigms for playing style imitation in different test-beds (different video games).
3. To propose a system that allows the imitation of the playing style of a human player in an automatic agent.
4. To implement this system in virtual players of different video games, analyzing the similarity of its style with human players imitated.

To evaluate the performance of the imitative agents, different measurements are proposed: the first based in low level standard measures such as accuracy,

precision and recall [5, 1]. Other evaluating high level parameters that characterized the playing style of different players, being this second approach currently unexplored in previous research. And finally experiments with human judges will be performed for a phenomenological evaluation.

### 2.2 Methodology

Our plan covers, at least, three iterations of the next stages where the main focus is always set in learning and in the development of knowledge.

1. Study the current literature regarding automatic players that imitates the playing style of human players in video games.
2. Study the current literature in the specific chosen domain (testbed video game, and AI paradigm).
3. Proposal of a model that allows the integration of the AI paradigm in an automatic player for the chosen testbed.
4. Test the system with human judges performing phenomenological evaluations.
5. Analysis and discussion of the obtained results. Study the feasibility of submitting the system to specific academic AI competitions.

### 2.3 Planning

- First approach towards research fields, revising the literature regarding automatic players, behaviour imitation and learning by observation.
- First system proposal, develop its implementation and perform the first tests.
- Once we have a general overview on the current research in these topics we will focus on specific fields that are still unexplored or haven't been development in depth
- Redesign of the system taking into account newer approaches: CBR and high level parameters (see Progress to Date section).
- Implementation of the system and tests with human spectators (is our system able to deceived human judges?)
- Publication of the results.
- Refinement of the CBR system (CBR and Reinforcement Learning), look for competitions.
- publication of the final system and the results and conclusions obtained. Final experiments with human judges.

## 3 Progress to Date

We began our research on human playing style imitation with an approach using neuroevolution (NE) in the classic arcade video game Ms. Pac-Man as it is a paradigm that has been used obtaining satisfactory results in other videogames as Super Mario Bros [5]. Ortega *et al.* (2013) performed an extensive study

testing many different machine learning approaches, in which NE and dynamic scripting achieved the best results.

The decision of using Ms. Pac-Man as a testbed environment was mainly due to three factors. Firstly the game presents a discrete state space and the number of actions required by the player is quite limited (continuous movement in only four possible directions). Secondly, we assumed that this game was sufficiently different to already explored platform video games as Super Mario Bros, since the movement of the characters is not limited to one axis (plus jumping) but in a two-dimensional maze, the levels are designed with a labyrinthine structure with linear paths (in which only"fits" one character on the way), and there are persistent enemies that chase the player with a nondeterministic behavior. Finally, we had already worked with this game before [7] and we were familiar with the implementation of this type of bots.

In an earlier work [9] we described an experiment with human judges to determine how easy it is, for human spectators, to distinguish automatic bots from human players in Ms. Pac-Man. This work allowed the judges to address characteristics of the playing style that led their conclusions. We have used some of these characteristics later as we believe they are valid for characterizing different players.

After working with ANNs and NE [8], we began to explore CBR as a new approach towards human playing style imitation in Ms. Pac-Man, as we believed it is a paradigm that has been used with very good results for player modeling in real time scenarios with many agents active in the same field [1]. In addition to CBR, we have continued using ANNs in the same environment to compare results for both paradigms, with CBR being the system that achieved better results. Recently we have conducted experiments that have been documented in a paper submitted for the next ICCBR 2018, where we describe CBR and ANN systems that learn to play Ms. Pac-Man video game from the traces of human players. The performance of the bots using each system is evaluated using both low level standard measures such as accuracy and recall, and high level measures such as *recklessness* (distance to the closest ghost, as it is mapped in our Ms. Pac-Man domain model), *restlessness* (changes of direction), *aggressiveness* (ghosts eaten), *clumsiness* (game steps the player is stuck) and *survival* (lives left). Results suggest that, although there is still a lot of room for improvement, some aspects of the human playing style (these high level parameters) are indeed captured in the cases and used by our CBR bot.

As part of our future work, we would like to create a CBR agent implementing the full CBR cycle. There are several locations in the system were we can incorporate domain knowledge such as the similarity measure or the adaptation strategy. We also need to pay attention to case base indexing and maintenance because the bot needs to play in real time.

Furthermore, we want to address other ML approaches to compare our CBR system, like using a deep architecture and modern optimization techniques.

## References


1. Floyd, M.W., Esfandiari, B., Lam, K.: A case-based reasoning approach to imitating robocup players. In: Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, May 15-17, 2008, Coconut Grove, Florida, USA. pp. 251256 (2008)
2. Gorman, B., Thurau, C., Bauckhage, C., Humphrys, M.: Believability testing and bayesian imitation in interactive computer games. In: SAB (2006)
3. Hingston, P.: A new design for a Turing test for bots. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, CIG 2010, Copenhagen, Denmark, 18-21 August, 2010. pp. 345350 (2010)
4. Livingstone, D.: Turings test and believable AI in games. Computers in Entertainment 4, 6 (2006)
5. Ortega, J., Shaker, N., Togelius, J., Yannakakis, G.N.: Imitating human playing styles in Super Mario Bros. Entertainment Computing 4(2), 93104 (2013)
6. Soni, B., Hingston, P.: Bots trained to play like a human are more fun. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) pp. 363369 (2008)
7. Miranda, M., Peinado, F.: Improving the performance of a computer-controlled player in a maze chase game using evolutionary programming on a finite-state machine. In: Proceedings 2o Congreso de la Sociedad Espaola para las Ciencias del Videojuego, Barcelona, Spain, June 24, 2015. pp. 1323 (2015)
8. Miranda, M., Snchez-Ruiz, A.A., Peinado, F.: A neuroevolution approach to imitating human-like play in Ms. Pac-Man video game. In: Proceedings of the 3rd Congreso de la Sociedad Espaola para las Ciencias del Videojuego, Barcelona, Spain, June 29, 2016. (2016) 113124
9. Miranda, M., Snchez-Ruiz, A.A., Peinado, F.: Pac-Man or Pac-Bot? exploring subjective perception of players humanity in Ms. Pac-Man. In: Proceedings of the 4th Congreso de la Sociedad Espaola para las Ciencias del Videojuego, Barcelona, Spain, June 30, 2017. pp. 163175 (2017)
10. Yannakakis, G.N., Maragoudakis, M.: Player modeling impact on players entertainment in computer games. In: User Modeling 2005, 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005, Proceedings. pp. 7478 (2005)

# Entering a New World: The Minimal Amount of Knowledge to Act as a Trustworthy Adviser Using Case-Based Explanations in a New Domain

Jakob Michael Schoenborn
schoenb@uni-hildesheim.de

Institute of Computer Science, Intelligent Information Systems Lab
University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany
https://www.uni-hildesheim.de

**Abstract.** *During the last years, there are multiple approaches to adapt Case-Based Reasoning to other domains than usually used before. Nevertheless, starting to develop a working Case-Based Reasoning system with the known issues of how to initialize a well-structured knowledge base and especially how to gather the required knowledge seems to be an issue. On top, the user's acceptance of decisions made by artificial intelligence agents is more skeptical than welcoming. Therefore, plausible explanations have to be generated for each decision made so that the user can develop trust in these. The problem is to determine how much knowledge in the given domain is actually needed to act as a trustworthy adviser and in general how to structure the explanation so that it will be accepted by the user. When building up a new explanation-aware CBR system, the process itself of creating this system should be capable of explaining itself. On top, the resulting CBR system should also be able to offer explanations.*

**Keywords:** Doctorial Consortium · Case-Based Explanation · Explainable AI · Knowledge Measurement and Maintenance · Explanation-aware

To enter a new world, making the first step into unknown terrain is never easy. But still, the range of possible domains and application areas where case-based reasoning can be used is huge. Some of those fields are comprehensibly more important to us, e. g. the medical area which is under constant development. To be able to set up a diagnosis with given symptoms as input is the most obvious application for a commonly accepted knowledge-management methodology. In 2016, N. Choudhury and S. Begum presented an overview of CBR-Systems used in the medicine domain, published in the IJACSA[1] [6]. There are multiple other well-researched fields which are particularly pointed out by A. Goel and B. Diaz-Agudo [8]. But there are also many other – lesser researched – fields with rising interest in development, e. g. CBR in real-time strategy games [5, 17], specific

[1] International Journal of Advanced Computer Science and Applications

areas on IT-security [1] and even temperament and mood detection using case-based reasoning [2]. Besides the well-known issue of how to effectively gather and store knowledge in an easily maintainable way [13], the users acceptance of the systems given diagnosis is crucial. In a recent user-study, Binns et al. [4] investigated the acceptance of data-driven decision making in the health assurance domain using case-based reasoning (besides sensitivity-based explanation and other approaches), which led one of the interviewed students to the conclusion: *'If you know it's on the back of an algorithm, it would incentivise people to work out how to game the algorithm, to find out what the algorithm is exactly doing'* – *CS* [4, p. 8]. The claim is not to solve the Turing-test [18], but to move the focus on generating user-acceptable explanations. The amount of papers published, which are dealing with explanation-aware computing, is rising recently [11] – particularly because of the European Union regulation on algorithmic decision-making and a "right to explanation" [9], which has been summarized by Goodman and Flaxman. This regulation forces algorithms to be transparent so that users can follow or at least understand the decision-making process, thus among other, supporting the right to non-discrimination. This forces existing systems to receive an overhaul in their current implementation and possibly to open up new application domains to case-based reasoning or, more precisely, case-based explanation. There are two possible starting points: Either there is an existing algorithmic decision-making process where it's reasoning process has to become visible to the user, or in the domain does not exist any explanation-aware process or system at all.

Given there is such an existing system, it has to be decided in which way the explanation will be provided - depending on the domain, application, and user in question. There are at least three different types of explanations: textual, semantic relations, and graphical representations. The transformation from the given state (i. e. simple debug-messages, verbose mode, ...) to an actual explanation has to be revised by the knowledge engineer inhibiting the technical knowledge and an expert inhibiting the domain knowledge. Given there is a number of experts willing to share their expertise, there needs to be a measurement on when "enough" knowledge is remaining in the system. As a side-effect, it could also lead more experts to willingly share parts of their knowledge without the feeling to be replaced by an AI. The view should not be limited on cases (as evaluated by Leake and Wilson [10] and suggested by Smyth [16]). Instead, additional components which also take effect in generating an explanation (i. e. adaptation rules and the used similarity measure) should be manipulated and then the different output of explanations has to be observed. How does the explanation change and is it still a valid explanation given the current situation? Followed by an evaluation process to see if reducing the number of cases or the number of adaptation rules has a larger effect on the output, the approach is to find the lowest boundary for generating a valid explanation.

Given there is no such system, the problem arises in building a case-based and explanation-aware system with the desirable feature to explain its own building process. A possible starting point is to determine a case structure, using reports of experts with statements on their domain and comparing these to extracted text results - filtering out keywords and use them as a case attribute. If possible, similar domains could also be considered. Another knowledge extraction point could be existing process models: Identifying key process elements and which attributes are used could be used to identify pivot elements. One problem is to build up an initial knowledge structure to provide a very basic explanation. This can be retrieved e. g. through networking/communities, FAQs, Wikis to build up on a basic core functionality. Even if there are networking communities where it might be possible to deploy a web-crawler and build up a knowledge base, legal concerns has to be respected. To be more precise, the General Data Protection Regulation in Europe applicable since May $25^{th}$ 2018, restricts the usage of data gained by mentioned web-crawlers without the creators permission [7].

The advantages of a system which is explaining its own building process is dependent on the domain and the targeted user. In the following, the aircraft domain is the domain to be considered and the system is supposed to support an intern knowledge engineer with the knowledge management when creating a maintenance routine for a new type of air plane. The aircraft domain in general is a very technical domain with a lot of structured information in form of attribute-value pairs, taxonomies and ontologies. The complexity comes with the "*hundreds of components, which consists of dozens of systems, which contains dozen of individual parts, called Line Replacement Units (LRU)*" [14]. Using the correct vocabulary and similarity measures, these information can be stored as cases and thus be used by a CBR-system. Since to generate an explanation there needs to be at least *some* knowledge, rule-based and model-based knowledge which can be retrieved from manuals etc. seems to be a valid starting point as a baseline of a explanation-aware system. Using this way, physically impossible combinations of components can be excluded and a explanation why they are not possible generated. Having a first set of core functionality, the more challenging part has to be considered: When should which knowledge be added to the system and especially: Why? The motivation in general to split up the development of the explanation-aware CBR system can be viewed similar to the motivation of software product line engineering: Reduction of development costs, enhancement of quality and that customers get products adapted to their needs and wishes [12]. In the aircraft domain, a product line can be the start of introducing a new air plane type to the air plane fleet. Since there can not be any practical experience when building a new air plane type, it is crucial for a cost-effective introduction to exclude as many failure risks as possible. This is the entry point for an explanation-aware CBR system. As stated above, the core functionality and knowledge containers need to be expanded so that valid and trustworthy explanations can be offered. Additional sources of knowledge are free texts of aircraft incidents and reports written by maintenance technicians or other staff members. To retrieve the knowledge out of free texts, the

framework FEATURE-TAK has been developed by P. Reuss - a **F**ramework for **E**xtraction, **A**nalysis, and **T**ransformation of **U**nstructu**RE**d - **T**extual **A**ircraft **K**nowledge which combines several methods from natural language processing and CBR [14]. This framework consists of five layers to store domain specific informations like abbreviations and technical phrases which can be accessed by other components i. e. software agents. The workflow is processed by multiple, distributed agents and coordinated by a central supervisor agent. To support the knowledge engineer, eight tasks are completed automatically ranging from phrase and keyword extraction, identifying synonyms and hypernyms to a similarity assessment and sensitivity analysis[2]. The knowledge engineer will then be offered a suggestion to add the retrieved knowledge, but without an explanation why the framework has come to this decision. Either way, the knowledge engineer has to do a consistency check and stores feedback on the process instance. This could be supported by a process on evaluating the current state of knowledge and if this retrieved piece of knowledge has actually a positive effect on the system if stored in the case base. While considering this, the SIAM methodology presented by T. Roth-Berghofer [15] improves the CBR cycle by adding two more steps, review and restore, which are triggered after the retain step. He distinguishes between an application phase (first three R's) and a maintenance phase (retain, review, restore). This is important for the maintenance, because in the original CBR cycle was no way to maintain the knowledge when the environment changes [3,15]. This is especially important for explanations, because they are building up on the current knowledge and it is crucial to be able to review the current state of knowledge (as the added review-step does).

## References


1. Abutair, H., Belghith, A.: *Using Case-Based Reasoning for Phishing Detection.* Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia, Oct 2017.
2. Adebayo, J., Adekoya, A., Ekwonna, C.: *Temperament and Mood Detection Using Case-Based Reasoning.* International Journal of Intelligent Systems and Applications. 6. DOI: 10.5815/ijisa.2014.03.05. 2014.
3. ALthoff, K.-D., Wess, S., Traphöner, R.: *INRECA - A Seamless Integration of Induction and Case-Based Reasoning for Decision Support Tasks*, 1995.
4. Binns, R. et al.: *'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions.* In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 377, 14 pages. DOI: https://doi.org/10.1145/3173574.3173951.2018.
5. Blizzard Entertainment, Google Deepmind: *DeepMind and Blizzard open StarCraft II as an AI research environment.* Online source: https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/. Date accessed: 25 Apr. 2018.
6. Choudhury, N. and Begum, S.: *A Survey on Case-based Reasoning in Medicine.* International Journal of Advanced Computer Science and Applications (ijacsa), 7(8), 2016. http://dx.doi.org/10.14569/IJACSA.2016.070820.


[2] for detailed description of the tasks, refer to [14]

7. European General Data Protection Regulation. Website: https://www.eugdpr.org/. Date accessed: 28 Apr. 2018.
8. Goel, A., Diaz-Agudo, B.: *What's Hot in Case-Based Reasoning.* AAAI Conference on Artificial Intelligence, North America, feb. 2017. Available at: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/15041. Date accessed: 25 Apr. 2018.
9. Goodman, B., Flaxman, S.: *EU regulations on algorithmic decision-making and a "right to explanation".* AI Magazine. 38. 10.1609/aimag.v38i3.2741. 2016.
10. Leake, D., Wilson M.: *How Many Cases Do You Need? Assessing and Predicting Case-Base Coverage.* In: Ram A., Wiratunga N. (eds) Case-Based Reasoning Research and Development. ICCBR 2011. Lecture Notes in Computer Science, vol 6880. Springer, Berlin, Heidelberg, 2011.
11. *IJCAI-17 Workshop on Explainable AI (XAI): Proceedings.* Melbourne, Australia, 20 August 2017. Website: http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai. Date accessed: 30.04.2018.
12. Pohl, K., Bckle, G., and van der Linden, F.: *Software Product Line Engineering: Foundations, Principles and Techniques.* Springer-Verlag, Berlin, Heidelberg. 2005.
13. Reichle M., Bach K., Althoff KD.: *The SEASALT Architecture and Its Realization within the docQuery Project.* In: Mertsching B., Hund M., Aziz Z. (eds) KI 2009: Advances in Artificial Intelligence. KI 2009. Lecture Notes in Computer Science, vol 5803. Springer, Berlin, Heidelberg, 2009.
14. Reuss, P. et al.: *FEATURE-TAK - Framework for Extraction, Analysis, and Transformation of Unstructured Textual Aircraft Knowledge.* In: Case-Based Reasoning Research and Development, Springer International Publishing, Pages 327–341, 2016.
15. R. Roth-Berghofer. *Knowledge Maintenance of Case-Based Reasoning Systems: The Siam Methodology (Dissertations in Artificial Intelligence).* Ios Pr Inc. 2003.
16. Smyth, B.: *Case-base maintenance.* In: Pasqual del Pobil A., Mira J., Ali M. (eds) Tasks and Methods in Applied Artificial Intelligence. IEA/AIE 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1416. Springer, Berlin, Heidelberg. 1998.
17. Stalnaker, R.: *Knowledge, Belief and Counterfactual Reasoning in Games.* In: Arló-Costa H., Hendricks V., van Benthem J. (eds) Readings in Formal Epistemology. Springer Graduate Texts in Philosophy, vol 1. Springer, Cham, 2016.
18. Turing, A.: *Computing Machinery And Intelligence.* Mind, Volume LIX, Issue 236, Pages 433460, 1950.

# The Writer's Mentor

Eriya Terada

Indiana University, Bloomington IN 47405, USA
eterada@indiana.edu

**Abstract.** We propose The Writer's Mentor, an interactive system that aims to guide users write texts suitable for a given domain, based on textual cases. While previous systems achieve this task by suggesting keywords or extracted phrases, The Writer's Mentor suggests topics or asks questions based on the comparison between what the user has already written and its similar counterparts in the case base. This allows the user to stay creative in the writing task while being guided by our system to follow a certain path. In this research summary, we describe our case retrieval strategy for this task, the process of crafting suggestions for the user, and conclude with hypothetical evaluation methods, questions, and future work.

## 1 Introduction

Many works in writing adhere to certain commonalities that vary from domain to domain. For example, writing a review for a restaurant may involve describing the atmosphere at the table and the taste of the dish, while a recipe may start with a list of ingredients and continue with a sequence of actions. While it may seem that there is an infinite amount of ways to write such things, good writing tends to receive high ratings from other people, and this is rather prominent among online reviews.

The existence of these "useful" reviews suggests that while there may not be a rigid set of rules for a well-written review, there are essential components or types of information that people expect to see in other users' reviews. In general, these reviews tend to include both negative and positive sentiments of the target product and the personal experiences that go along with it, whether that be a restaurant, electronic device, or a hotel.

Now suppose that a new user wishes to write an online review for some restaurant or product but does not really know where to start. This writing problem can be reduced to a textual CBR (TCBR) problem if we can treat these useful reviews as cases in a case base and a newly written sentence by the user as a problem.

Given a new sentence, the CBR system can search its case base of previously written reviews for similar sentences and reuse information in the succeeding sentences to generate suggestions or questions to guide the user write the next sentence. Once the user finishes writing the review, it can be retained in the case base, completing the CBR cycle.

We will focus on restaurant reviews for demonstrative purposes, but the general procedure described in this summary should be applicable to other domains as well, given that there is an underlying structure and flow of information. Such domains include stories and instructions on how to do certain tasks.

## 2 Related Work

TCBR has been previously used in the context of assisting users to produce writing. Say Anything creates a story interactively with the user by inserting a relevant sentence from the case base after the user finishes writing his own sentence [7]. [3] have created a system that automatically generates email responses given an email request as input. This approach works fairly well since the context is limited to a financial domain and the email responses appear to be fairly structured. GhostWriter-2.0 suggests parts of sentences taken from previous reviews that could fit well with a new review that a user is currently typing into the system [1]. The Reviewer's Assistant, inspired by the GhostWriter system, suggests features of the product of interest, offering the user a list of choices for what she can write next [2]. [5] used the TCBR framework to retrieve similar cases of transportation incident reports to help explain causes for new incidents.

In contrast to the previous approaches that directly reuse textual cases, we will look at the the task of writing from a higher level and suggest what a user's review is lacking compared to similar reviews in the case base. For example, suppose a useful review for a hotel started off by talking about its dinner service and then went on to describe how good the lobster tasted. Previous approaches will consider the topics "dinner service" and "lobster" as two independent entities, but we will utilize a semantic network in order to find the relationship between these two topics, if any, as a means of crafting a suggestion. Now, if a new user only writes about the "dinner service" in general for that hotel's review, previous systems may simply suggest to write about "lobster" because a previous case had used it. We hope that our system will suggest the user to additionally include a word or two about how a dish tasted, since this user may not have eaten lobster in the first place. Our approach gives more room for the writers to become creative and describe something in their own words, rather than directly reusing stored words or phrases written by previous reviewers.

## 3 Approach

### 3.1 Retrieval of Cases

The general flow of this case retrieval step is shown in Figure 1. Each case in our case base is a previously written online review $r$ consisting of one or more sentences. Suppose that a new user starts writing a new sentence for an online restaurant review, which we shall call query sentence $q$. Given $q$, our system will look in the case base for target sentences $t_1, t_2, .., t_n$ that are most similar to $q$ and also retrieve the reviews $R = \{r_1, r_2, .., r_n\}$ that contain them.

The key factor in this process is determining which target sentences are similar to the query sentence $q$. We are currently considering two ways to do this: one is using the WUP similarity offered by WordNet [4] and the other is to use a cosine similarity of word embeddings, such as those offered by ConceptNet Numberbatch [6].

### 3.2 Abstraction of Cases

The target sentences $t_1, t_2, ..., t_n$ are sorted in order of similarity to the query sentence $q$, and the top $k$ sentences are examined to extract common "trends" that are then used to form suggestions or topics to the user. In doing so, the system will use knowledge between entities in a semantic network such as ConceptNet. We propose two ways of extracting trends between sentences, and both are visualized in Figure 2:

**Common Topics (horizontal search between reviews).** For each target sentence $t_{ij} \in r_i$ we extract the succeeding sentence $t_{ij+1} \in r_i$ if it exists. For each word $w$ in the succeeding sentence, we look up its definition in ConceptNet and look at their immediate *IsA* and *RelatedTo* connections to extract shared concepts among $w$. For example, both *pasta* and *steak* share the *IsA* relation with *food*, while the verb *run* is connected with *walk* via the relation *RelatedTo*.

We create a list of the most commonly shared concepts among the succeeding sentences and suggest them as potential topics for what the user could write next. The topics are ranked by how similar the originating sentences were to $q$.

**Common Relations (vertical search between neighboring sentences).** For each target sentence $t_{ij} \in r_i$ we look at the succeeding sentence $t_{ij+1} \in r_i$ and determine if there are any connections between these two sentences. For example, if the target sentence introduces a dish while the next sentence describes that dish, then we can conclude that the succeeding sentence elaborates on the first. On the other hand, if the target sentence mentions the restaurant itself and the succeeding sentence is about the waitress, we can say that the connection here is not only really about elaboration but that it is about what belongs to a restaurant. This part is yet to be implemented.

### 3.3 Making Suggestions

We propose the theory behind how the system will form useful suggestions or questions for the user. For the time being, we are considering templates such as:

- If the succeeding sentences of the retrieved sentences share topic $A$, the system can say that, "Now might be a good time to talk about $A$".
- If topics in the retrieved sentences share a relation $X$ IsA $Y$ (i.e., $X$ is more specific than $Y$) with their respective succeeding sentences, and the query sentence was talking about $y$, the system can ask, "Can you describe more about $y$?".

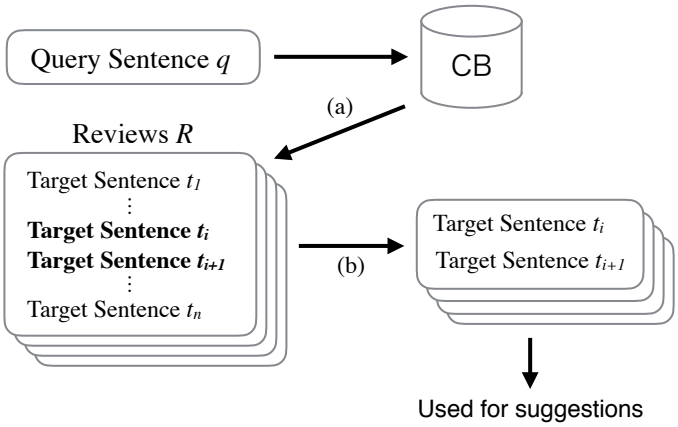
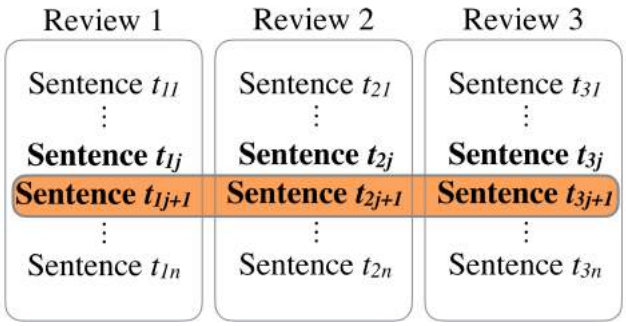
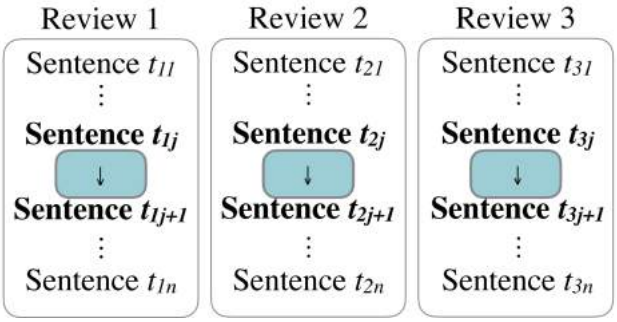



**Fig. 1.** Flow of case retrieval. At (a), the system retrieves reviews that contain sentences similar to the query sentence $q$. Then at (b), the system gathers the similar sentences and their succeeding sentences.




(a) Common Topics




(b) Common Relations

**Fig. 2.** The conceptual differences between Common Topics and Common Relations

The latter case is especially useful if a review talks about things that have hierarchical connections. In the restaurant domain, we can say that *people* go to *eat* at *restaurants*, a place where *waiters serve food* and *drinks*. Additionally, we will avoid suggesting topics that the user has already written.

## 4 Proposed Evaluations

We are currently considering 2 ways of evaluating the system. One is objective, where we try to estimate the usefulness score of an unseen review based on the usefulness scores of reviews in the case base and their common topics and relations. The other is subjective, where we let users use the system and check if they incorporated the suggestions from the system.

## 5 Conclusions and Future Work

In this research summary we have described the overall structure of how a TCBR system can incorporate semantic networks to discover common trends between sentences to assist users write new online reviews. While there are still many tasks to consider, we hope this demonstrates an interesting problem for TCBR.

## References


1. Bridge, D., Healy, P.: Ghostwriter-2.0: Product reviews with case-based support. In: Bramer, M., Petridis, M., Hopgood, A. (eds.) Research and Development in Intelligent Systems XXVII. pp. 467–480. Springer London, London (2011)
2. Dong, R., Schaal, M., OMahony, M., McCarthy, K., Smyth, B.: The reviewer's assistant: Recommending topics to writers by association rule mining and case-base reasoning. In: The 23rd Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2012), Dublin City University, Dublin, Ireland. pp. 17–19 (2012)
3. Lamontagne, L., Lapalme, G.: Textual reuse for email response. Advances in Case-Based Reasoning pp. 242–256 (2004), http://link.springer.com/chapter/10.1007/978-3-540-28631-8_19
4. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11), 39–41 (Nov 1995). https://doi.org/10.1145/219717.219748, http://doi.acm.org/10.1145/219717.219748
5. Sizov, G., Öztürk, P., Štyrák, J.: Acquisition and reuse of reasoning knowledge from textual cases for automated analysis. In: Lamontagne, L., Plaza, E. (eds.) Case-Based Reasoning Research and Development. pp. 465–479. Springer International Publishing, Cham (2014)
6. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972
7. Swanson, R., Gordon, A.S.: Say Anything. ACM Transactions on Interactive Intelligent Systems **2**(3), 1–35 (2012). https://doi.org/10.1145/2362394.2362398, http://dl.acm.org/citation.cfm?doid=2362394.2362398

# Reasoning with Multi-modal Sensor Streams for m-Health Applications

Anjana Wijekoon[1]

School of Computing Science and Digital Media, Robert Gordon University,
Aberdeen AB10 7GJ, Scotland, UK
{a.wijekoon}@rgu.ac.uk

**Abstract.** Musculoskeletal Disorders have a long term impact on individuals as well as on the community. They require self-management, typically in the form of maintaining an active lifestyle that adheres to prescribed exercises regimes. In the recent past m-health applications gained popularity by gamification of physical activity monitoring and has had a positive impact on general health and well-being. However maintaining a regular exercise routine with correct execution needs more sophistication in human movement recognition compared to monitoring ambulatory activities. In this research we propose a digital intervention which can intercept, recognize and evaluate exercises in real-time with a view to supporting exercise self-management plans. We plan to compile a heterogeneous multi-sensor dataset for exercises, then we will improve upon state of the art machine learning models implement reasoning methods to recognise exercises and evaluate performance quality.

**Keywords:** Deep Learning · Privileged Learning · Exercise Recognition · Exercise performance Quality

## 1 Introduction

Maintaining a regular self-managed exercise routine is an essential component when living with Musculoskeletal Disorders(MSDs). Specifically for elderly and people with chronic conditions, it is important to maintain active lifestyle and importantly to adhere to correct execution of exercises. Research on finding technological solutions to support either the prevention or self-management of MSDs has emerged over the last few years. A digital intervention which captures exercises and provides feedback on performance quality at real-time will contribute towards motivating the user to adhere to a regular exercise routine. An effective Digital intervention for self managing MSDs should consist of three main components: intercepting exercises in real-time; recognising exercises; and evaluating performance quality to facilitate personalised feedback generation. In this research we plan to explore each component to design an a optimal digital intervention for self-management of exercises.

Simple sensors on a smart phone are able to identify simple ambulatory activities [11]. Datasets in recent research on HAR [4, 12], Gesture recognition [1],

gym exercises [13] and Activities of Daily Living(ADL) [2] use sensors such as inertial sensors, object sensors, pressure sensors and depth sensors. Exercise is a sequence of independent movements of multiple body parts; specifically exercises recommended for low back pain require capturing greater ground surface compared to ADL or ambulatory activities. Hence a smart watch on the wrist is not able to capture an exercise with the level of precision required. Furthermore a wrist sensor is susceptible to noise (due to high freedom of movement) and could temporarily loose data. Aforementioned datasets do not consider these limitations, rendering them inadequate for this research; and it is evident that exercises require capturing different perspectives from multiple sensors. Accordingly we emphasise the need for a data collection in order to identify which sensors can intercept exercises efficiently.

Exercise recognition can be viewed as a special case of Human Activity Recognition (HAR). Research in HAR involves the use of machine learning methods and more recently, deep learning algorithms to reason with sensor data. Many researches shows that deep learning techniques outperform traditional machine learning techniques that use hand-crafted features [10, 11]. Notably most recent research [8, 5, 17] use combinations of deep learning architectures and yield comparatively improved performance. Sensor fusion has been attempted with deep fusion architectures to classify video [6, 9] and reconstruct video and audio [7] by experimenting fusion in different levels of abstraction. Exercise recognition is not widely seen in literature but we find [13] using traditional methods such as Dynamic Time Warping (DTW) and peak counting. Exercise recognition has not been attempted with heterogeneous data streams and with deep learning techniques to the best of our knowledge. Accordingly we first evaluate aforementioned techniques and learn their transferability to the domain of exercises. Next we select advantages techniques from above experiments to build a comprehensive solution for exercise classification with multi-modal data.

Efficient deployment is an important characteristic in any health care digital interventions with direct impact on user acceptance. In this problem domain, the main restriction is the number of sensors. More sensors force the user in to a more restricted setting hence less efficient. We plan to investigate concept Privileged Learning (PL) [14] in order to improve the deployability of reasoning models. PL mimics how humans learn with a teacher; in HAR we interpret PI as deploying a model with less sensors compared to number of sensors available at training. These techniques should contribute towards building robustness in to models to handle missing modalities in real-time which improves usability and efficient deployment.

Performance quality of an exercise can be defined as how much actual execution deviates from correct execution of the exercise performed under the supervision of a physiotherapist. Measuring the deviation is open to interpretation. Recent research in this area show that quality assessment is modelled as a classification task where classes include many wrong executions and a correct execution [3, 15]. This method can be similar to a rule based system; hence unable to locate a problem in real time. We view quality assessment of an exercise

execution as a similarity comparison task. We plan to employ similarity based methodologies to compare different representation of exercise executions with correct execution to locate problem areas.

## 2 Research Aim

The overall goal is to introduce a sustainable digital intervention for self-managing exercise routines. Following review of literature we identity following research questions to achieve aforementioned goal.

1. How to combine multi-modal data streams to improve exercise recognition?
2. How to maintain performance in the presence of noisy and/or missing sensor modalities?
3. How to analyse exercise performance quality by comparing actual and expected multi-modal sensor data?

We outline four objectives in order to answer each research question. First is to compile a multi-modal sensor dataset in the domain of exercises recommended for low back pain. Secondly we will develop a sensor fusion architecture to recognise the most effective sensors and/or features then combine to improve recognition accuracy. Next we implement methodologies to mitigate noise/absence of modalities in deployment. This would enable the network to learn with all modalities but remain robust even with fewer modalities in real time. Finally we plan to introduce a similarity based architecture for comparing sensor data to generate a quality assessment. The resulting solution should localise performance problem to lower level actions of the exercise.

## 3 Proposed Plan of Research

Comprehensive multi modal sensor data collection for exercises is crucial if we are to address each research question mentioned above. Data collection task will produce a multi-modal sensor dataset for exercise classification and quality assessment. We have selected accelerometers, pressure sensor and a depth sensor to bring together three different modalities of heterogeneous data types; and we have selected seven exercises that are recommended for patients with low back pain.

To achieve Objective 2 first we will investigate how each sensor modality contributes towards accurate classification of exercise movement classes. Next we will explore how a sensor fusion architectures can contribute toward improving previous results. For this we will look at how informed selection of sensors can improve performance. The goal is to create an architecture which will identify the most informative features from different sensors to improve exercise recognition.

Being inspired by the Privileged Learning paradigm [14] we will explore different approaches to address Objective 3. We will investigate how to enforce

robustness in sensor fusion model to handle missing modalities; and we will explore generating synthetic data to represent missing modalities at deployment from available sensors.

To address objective 4 we will define a metric to assess exercise performance quality. Specifically the deviation between expected and actual performance will form the basis for this metric. We plan to investigate methods that learn similarities in spatio-temporal data belonging to one classification class to evaluate quality difference. This will call for similarity measures in different abstraction levels of feature embeddings. In order to locate differences in finer detail we will treat exercises as a sequence of primitive actions. Here the idea is to isolate the differences with respect to a primitive action rather than performing a binary evaluation(correct or incorrect).

## 4 Current Progress

We are at the early stages of data collection task where we compile a multi-modal dataset on exercises for low-back pain. We have identified sensors and exercises we will use is data collection and obtained ethics approval from university ethics committee. We are in the process of recruiting volunteers and collecting data which will continue during the summer of 2018.

A sensor to sensor neural translator for generating missing sensor data was developed and this work is published in ICCBR2018. This work aligns with Objective 3 where we try to minimize number of sensors at deployment for effective deployment. We evaluated this methodology with two datasets (SelfBACK and PAMAP2),both containing ambulatory activities recorded with inertial sensors. Translator method successfully learned dependencies from sensors with different placements and improved k-NN classification accuracy compared to single sensor. These results confirm while we can learn from many sensors, we can re-use these reasoning models in deployment with fewest sensor.

We explored Zero-shot Learning(ZSL) with Matching networks, work presented at The SICSA ReaLX Workshop 2018, where we improved Matching networks[16] to recognise activities never seen during training. We achieve substantially improved performance with modified matching networks compared to original. We will further explore ZSL as it enables a pre-trained network to recognise new activities at deployment. This is desirable when we expand our work from ambulatory activities to exercises where possible number of classes is unmanageably high.

## References


1. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys (CSUR) **46**(3), 33 (2014)
2. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J.d.R., Roggen, D.: The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters **34**(15), 2033–2042 (2013)

3. Chen, P.C., Huang, C.N., Chen, I.C., Chan, C.T.: A rehabilitation exercise assessment system based on wearable sensors for knee osteoarthritis. In: International Conference on Smart Homes and Health Telematics. pp. 267–272. Springer (2013)
4. Chen, Y., Xue, Y.: A deep learning approach to human activity recognition based on single accelerometer. In: Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on. pp. 1488–1492. IEEE (2015)
5. Hammerla, N.Y., Halloran, S., Plötz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1533–1540. AAAI Press (2016)
6. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
7. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 689–696 (2011)
8. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
9. Pigou, L., Van Den Oord, A., Dieleman, S., Van Herreweghe, M., Dambre, J.: Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. International Journal of Computer Vision pp. 1–10 (2015)
10. Ronao, C.A., Cho, S.B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Systems with Applications **59**, 235–244 (2016)
11. Sani, S., Massie, S., Wiratunga, N., Cooper, K.: Learning deep and shallow features for human activity recognition. In: International Conference on Knowledge Science, Engineering and Management. pp. 469–482. Springer (2017)
12. Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T.S., Kjærgaard, M.B., Dey, A., Sonne, T., Jensen, M.M.: Smart devices are different: Assessing and mitigating-mobile sensing heterogeneities for activity recognition. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. pp. 127–140. ACM (2015)
13. Sundholm, M., Cheng, J., Zhou, B., Sethi, A., Lukowicz, P.: Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. pp. 373–382. ACM (2014)
14. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural networks **22**(5), 544–557 (2009)
15. Velloso, E., Bulling, A., Gellersen, H., Ugulino, W., Fuks, H.: Qualitative activity recognition of weight lifting exercises. In: Proceedings of the 4th Augmented Human International Conference. pp. 116–123. ACM (2013)
16. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. pp. 3630–3638 (2016)
17. Yao, S., Hu, S., Zhao, Y., Zhang, A., Abdelzaher, T.: Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In: Proceedings of the 26th International Conference on World Wide Web. pp. 351–360. International World Wide Web Conferences Steering Committee (2017)