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.