NTNU
Kunnskap for en bedre verden

Department of Computer Science

TDT4173 - Assignment 1

Topic Name

Case Based Reasoning

Group:
Group Case_Based_Reasoning_Group 7

Authors:
Piri Babayev (pirib)
Danilas Miscenko (danilasm)
Aleksander Simmersholm (alekssim)

October 15, 2020

Abstract

This method paper introduces and gives a brief overview of Case-Based Reasoning. The first section starts with the introduction of the CBR's core concepts, going over its paradigm and functionalities. We introduce the core parts of what goes into a CBR system, and further this explanation through definitions and specific constructs designed for CBR in mind.

The second section delves deeper into the core phases of the typical CBR cycle, and their respective functions and roles as components of the CBR system as a whole. It also includes a simple example for how CBR can be used, its versatility as a tool and which medium it operates on.

In the third part, the focus is set on the Generalized CBR model as formalised by Armin Stahl in his paper "Learning Similarity Measures: A Formal View Based on a Generalized CBR model". The paper generalizes the classic CBR model in a way that makes room for more modern uses and approaches of CBR systems. It does also go over its inherent effect from generalizing the model on similarity measures.

Last but not least, the fourth part presents and briefly examines three recent papers that showcase the versatility and use case of the generalized CBR model discussed in the previous section. The paper is then concluded and further work is outlined.

Table of Contents

List of Figures

List of Tables

1 Introduction

Case-Base Reasoning stands out in the crowd of Machine Learning approaches by opening up and presenting all its core concepts as a whole. It is a method that reasons its way to a proposed solution by analyzing and using the cases of previous experiences. One could immediately draw parallels with our own human way of solving problems, and this is not a far reach as CBR indeed has its roots from the field of Cognitive Science [7, p. 533]. One of the most intuitive approaches to arriving at the solution is recalling relevant past knowledge and experience - having had an unforgettable experience with spoiled dairy product once, one would hardly dare to drink milk past its expiration date again.

Before explaining CBR further, a few aspects of interest have to be pointed out. CBR is a learning paradigm rather than a set of specific methods - it presents and describes an overall approach to problem solving, and leaves the architect with the freedom to choose how to implement its components [5, p. 230]. Specific methods and algorithms used within CBR vary to a great extend depending on the domain it is deployed in, meaning that a variety of methods borrowed from other machine learning topics might be used when implementing CBR. That leads us to the second aspect - CBR can be employed to tackle situations of very different nature, from reasoning about legal cases based on the previous experience with rulings [5, p. 240], or in an application to help people address lower back pain [6].

To state a very high-level description of CBR would be to start off by presenting the *case base* - this is the memory repository where "experiences" or more formally, *cases*, are stored. Any time a new problem needs addressing, CBR tries to retrieve cases with similar problems, which might in best cases point to a solution fit to address the initial problem. After doing so, the system would learn from this experience by updating the case base accordingly with the newly solved case and adapt its new knowledge for future cases.

CBR is what one would refer to as a *lazy learning* method, meaning that it generalizes beyond the training data only when a query has been sent to the system. Therefore, a CBR system waits for a problem to be able to decide how to solve it. Another vital detail is the fact that CBR responds to a problem by examining similar stored cases and ignores the ones that are not. Furthermore, these cases have no restraints on their representation and are usually chosen to fit the problem domain - images, sounds or text, everything is on the table for the CBR. [5, p. 240]

The aforementioned flexibility of CBR is the source of many complications that arise when designing such a system - questions about the instance representation; the approach for similarity measures and retrieval; query and case comparison; adaptation of the new cases etc. are central in the analysis of this approach.

The next section dissects CBR into its basic components, explain how they all fit together and which processes outline CBR systems ability in problem solving tasks. Then the paper will discuss the Generalised CBR model that expands on the classical model, giving it more flexibility for more modern approaches. Furthermore, a few examples utilising this approach will be presented. Last but not least, some pointers for future work will be given together with the conclusion.

2 Foundations

To delve deeper into depths of thrilling opportunities CBR offers one needs to get better acquainted with it first. It can be daunting to tackle a system of such proportions, hence this section will going through the CBR cycle to get a better intuition of the process and to understand how each element fits within the system. The CBR cycle model by Agnar Aamodt and Enric Plaza [fig 1] [1] presented below will be used as reference going forward, as it models the most classic CBR cycle.

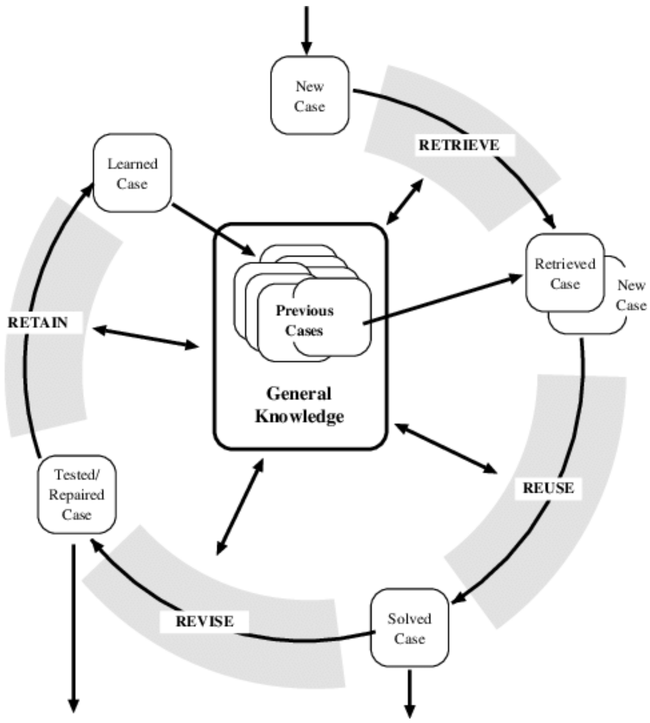

Figure 1: CBR Cycle (Image courtesy of Agnar Aamodt and Enric Plaza)

The main focus here will be put on the four processes shown in the model: **Retrieve, Reuse, Revise and Retain.**

2.1 Retrieve

When a new problem, also known as a query, is introduced to the system, the system is expected to find an example that is the most analogous to the posed query, for example recommend a similar book or suggest the correct spelling.

To do so, CBR utilizes its case base by looking up a previous case (or cases) that is the most similar to the problem at hand. Similarity by itself is a very common topic, but no general definition of similarity exists to be used in every application. It is the domain and usage that forms the definition of similarity between two instances in the implementation of a CBR system. One could argue that two items are similar if they share some attributes, but their similarity is also heavily dependent on the context they're evaluated in. A good method that is commonly used is to use some algorithm to cluster the cases together based on their attributes, then when a new case arrives, the k-Nearest Neighbours algorithm is used to find the nearest neighbours to the new case, the idea being that if the case is similar to some other case that has a solution, then it is likely that that solution will work for the case in question. This notion is the inductive bias in CBR - when a new case is classified, it will be most similar to the ones that this classification was derived from. [5, p. 234]

One of the principles used for calculating similarities is the Local-Global principle. This principle is usually adequate enough for most applications. The principle says: "There are elementary (local or atomic) description elements (attributes). Each object or concept A is (globally) described by some construction operator C from the local elements:

$$A = C(A_i | i \in I)$$

where I denotes some index set of the atomic elements A_i (attributes)" [7, p. 96].

By using this principle one can define and compare cases. By assigning a weight to each attribute, a similarity measure of the objects based on the important attributes can be acquired:

$$sim(q, c) = \sum_{i=1}^{i=n} \omega_i sim_i(a_i, b_i)$$

q being the query, a_i being an attribute of q, c being some case from the case base and b_i being an attribute of c. It is also important to note that the sum of all weights $(\omega_1, \omega_2 ... \omega_n)$ is equal to 1.

2.2 Reuse

After retrieving a relevant case from the case base, the system tries to apply its solution to the current task at hand, or "Reuse" it. If an exact problem has been tackled before, the system simply reuses the old solution, but it is not often that the case base has a solution that matches the problem exactly. In this case the old knowledge can be adapted to become useful, either by generalising it, or appending it with information from other solutions. The exact steps taken heavily depend on the context and case representation (discussed later in this chapter).

2.3 Revise

As such, the CBR cycle already produced a potential solution, but without the next two steps the system will never improve its performance. The Revise step is first in line to facilitate this development - having arrived at a potential solution, CBR needs the proposal to be evaluated. It can be done either by a domain expert, in a simulated environment or even in the real world. Naturally, the proposed solution might not be adequate and CBR might need another go at the case base, or if possible get the error to be pinpointed by the user. In either way the main objective here is to evaluate the usefulness of the proposed solution, rework it on demand and prepare it for the next step.

2.4 Retain

The main objective of the Retain phase is to update the case base with the new case by adapting the new solution and ensure that it can be reused in the next cycle. The notion of adaptation here is similar to the one in the Reuse phase and, depending on the specifics, the new solution can be either integrated into the most similar old one, generalised to indicate the notion of a common solution with or provide more detail for a pre-existing case etc. Good implementation of the Retain phase will also minimize the number of stored cases, improving the speed of future retrieval of cases and decreasing the need for heavy processing of the Revise phase.

2.5 Tying it all together

As an example, let's take the situation that is favored by many authors of the field. Consider the following situation: a car rolls into a service centre in need of repair because of its headlights not functioning properly. To do so, the technician rushes to the best expert in the field, that is, the CBR system in-house. First, the problem needs to be introduced - perhaps, by filling up a standardized form or questionnaire that could help the system narrow down the search. After the

CBR system outputs potential cases, it is all up to the expert to analyze the output and look up the cases that could help the one at hand. It might look for the cars of the same brand and model, or similar problems - the significance of each attribute heavily depends on the problem that needs to be solved. As such, color of the car would hardly be of importance, while a problem with back lights could be quite helpful. After receiving the feedback from the technicians about the success of the solved case the system could use the newly gained knowledge to generalize from "back light not working" and "headlight not working" to "signalling lights not working" and as such adapt the new solution for future use.

2.5.1 Case representation

Introduction has already mentioned a few contrasting problems tackled by CBR implementation - this versatility can be achieved by adapting its core components for the task at hand. A good starting point is the representation of a case. A CBR system that focuses on classification of images would benefit from a representation that includes pictures, an e-commerce product recommending system would fare much better if it had some attribute-value type, for example an object-oriented representation. Other types include text, speech, sensor data and conversational representations, or indeed any combination of them. It is important to note that one involves domain experts when designing a CBR system, to better tailor it to the needs of the end user. [7, p. 108] The design of the case representation needs to be done carefully since this decision affects all other components of the system.

2.5.2 Vocabulary

The components of the case base, similarity measures and adaptation process have to communicate with each other at the various stages of the cycle. Retrieving a set of relevant cases depends on the context of the problem, the adaptation process depends on the representation, etc. Therefore the system is in need of a medium to be able to do so. This medium is termed as a *Vocabulary* and it has the responsibility of regulating what the system can reason about. If there is no concept of "engine" (however this concept is defined, encoded and used), the components of the system will have a hard time trying to find a solution for something that does not exist (in their understanding). In a nutshell, the Vocabulary is the domain of all possible discussions within the CBR system. One can think of this as all possible appropriate conversation topics at a social gathering. A given concept can have several interpretations in the Vocabulary that can be used for different tasks. [7, p. 35]

2.5.3 Learning

Given that this paper is discussing CBR in the context of Machine Learning, what does exactly constitute as learning in a CBR system? An imminent answer comes when one conveniently breaks the discussed cycle into two main components, each with their own main task - Reasoning and Learning. Retrieve and Reuse belong to the former, while the latter one incorporates Revise and Retain. Intuitively, the second half contributes to learning by integrating new experiences and increasing the knowledge base. After (quite) a few iterations CBR will have a larger collection of cases that it can utilize to perform better.

The performance of the system, on the other hand, depends not only on the amount of knowledge it has, but also on how it can utilize it. There are many aspects as to how this can be achieved by the system - the Vocabulary needs to facilitate the overall reasoning process, case representation to describe the elements of the system, adaptation to properly evolve the knowledge base, similarity to compare and select the most suitable cases etc.

3 Improvements of the CBR method

While the original model by Agnar Aamodt and Enric Plaza [1] is simple and retains clarity when describing the typical CBR cycle in general terms, it is quite limited when it comes to describing current research issues and popular approaches used in modern day CBR application. For example, when discussing similarity measures, even though it is a crucial component of CBR systems, clear methodologies for defining these measures do not yet exist. Most similarity measures derive from the context in which the application is to propose a solution for. Multiple approaches exist as seen in the previous section, but most can be boiled down to general distance metrics solutions. Of course, this is an oversimplification as more complex similarity measures can be considered with domain knowledge, or machine learning approaches, but to keep the justification frank, there exists a need for clearer goal-oriented approaches when it comes to similarity measures.

In [8], Stahl proposes a more formal generalization derived from the classic CBR model that was discussed in the previous sections. This can be seen as merely the modernisation of the classic CBR model as it attains the usage of more modern approaches that have become prevalent in most CBR systems to-date. Unfortunately, for the sake of generalization, the model might look a tad bit complex next to what is already mentioned, but we will try our best to keep each method cycle frank and to the point.

Stahl proposes a generalized CBR model, making it easier to orient clearer definitions of modern methods. Furthermore, Stahl commends the classical figure for its clarity but also points out that the classical approach creates limitations for more recent research since the time the classical model was proposed. This makes it harder to create clear definitions of methods used in CBR systems today.

The classical CBR model accounts for a typical problem-solving situation: It is given a problem, then proposes the solution(s). This creates the structure of cases that consist of a problem part, and a corresponding solution part that successfully solved the problem in the past. It falls under the typical paradigm of CBR systems where “Similar problems have similar solutions”, given the assumption that the system already possesses some case knowledge in the specific domain. This can be naïvely summarized as:

A function that indexes a list of solutions that might apply to the current query problem based on the similarity of past problems, resulting in an action of useful information retrieval.

However, nowadays slightly different approaches are used where this classical approach, which consist of a problem part and a solution, might not work.

Another limitation of the classic CBR cycle is that it does not take into consideration issues and topics of recent research. Some of these are

- **Dialog strategies**, which is the topic on how to obtain efficient and accurate queries as input to the system to better its efficiency in providing more precise solutions.
- **Explanation**, where the CBR system explains the underlying reasons of its proposed solutions.
- **Feedback**, where CBR systems incorporates a feedback-loop to better its understanding on cases and refining solutions to problems.

The classical CBR cycle consists of the four basic steps: Retrieve, Reuse, Revise and Retain, and to make it easier to compare the CBR generalization to the classical CBR model the new function cycle will be organised under the already established traditional CBR cycle [Table 1].

Retrieval	• Situation-Characterization, Query. • Case, Case Characterization, Case Lesson, Case Space. • Similarity Measure. • Retrieval Function.
Reuse	• Adaptation Function. • Output Function, Output Space .
Revise	• Output Processing Function. • Feedback Function.
Retain	• Learning Functions

Table 1: Function cycle categorized by their equivalent phases in classical CBR phases

Finally, we arrive at the Formal Generalization of the Classical CBR Cycle [Figure 2]. It presents a more formal way for analysing CBR functionality in more detail.

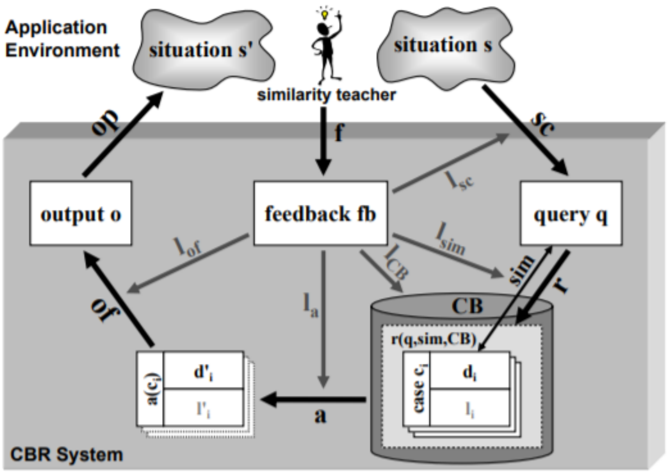

Figure 2: The Generalized CBR Cycle by Armin Stahl

The CBR system is to provide the necessary information by generating a corresponding output solution (o).

Situation-Characterization. Query. At the first step, a situation/problem (s) is characterized into a query (q) by comparing it to existing situations in the situation space. (sc)

Case, Case Characterization, Case Lesson, Case Space. In the second step, the query (q) is compared to the cases (c) that exists in the case base (CB) which might contain useful information for the solution of the situation which the query is derived from. It is important to note that cases consist of case lessons and case characterizations. A case might also only consist of characterizations to be used as utility for solutions, as in, they might inherent some sort of information that might be of value to future case estimations.

Similarity Measure. While still in case base, similarity measures are used to estimate the query to a case. This is denoted as sim(q,c).

Retrieval Function. The retrieval function then returns a set of case-bases for the given query according to their similarity values. The retrieval function (r()) can be done in a multitude of ways. Either by simply retrieving the list of cases ordered by their similarity values, or some other arbitrary way. This function is denoted r(q,sim,CB).

Output Function, Output Space The output function (of()) has the ability to transform the output (o) in many ways, from a respective dialogue for the original input situation (s), to further filter and select from the cases left after the adaptation process. This depends on the application's purpose and if there is a further need to elaborate the finished result. One problem of the classic CBR-system that this component might solve is the ability to explain the underlying reasoning for the output solution.

Output Processing Function. The output process function (op()) is an informal process which typically modifies, or creates a situation (s) into the situation space (S). It can have the job of updating the situation accordingly with the new case it just proposed. Of course, if the current situation is still left without a sufficient solution it might just update the current situation case with a new initial situation to run through the cycle again.

Feedback Function. The CBR system receives feedback from the feedback function about its performance and the usefulness of its proposed solution, applying learning strategies to itself and improving its function by updating its knowledge containers.

Learning Functions. The learning functions allow the system to take note from feedback and appropriating itself, modifying case bases, similarity measures and adaptation. It describes the general purpose of learning functions, where the system might improve itself on some matter.

By abstracting from the traditional problem-solution scenario, the generalized model is more suitable to describe more prevalent application scenarios. For example, instead of cases requiring a problem and a solution, cases may only consist of case characterizations which might include complimentary knowledge to a similar case. Moreover, with the introduction of the feedback function and learning functions, the system can use machine learning strategies to better its understanding for similarity between cases and their characterization. With the output function, the model can use dialog strategies and explanations to describe its evaluation of proposed solutions more accurately.

4 Current applications of CBR

The generalized model paves way for introduction of more elaborate ways to utilize the different sections of the CBR model. Two such characteristics comes from the usage of the feedback function and the availability to define more complex similarity measures. The feedback function offers opportunity to enhance the CBR approach during its operation and utilize this feedback to make more precise retrieval of cases, while the more complex similarity measures allows for the implementation of machine learning methods to reach more precise solutions. This chapter will be focusing on a few of the recent applications that have utilised these exact approaches.

4.1 Predicting the Electricity Consumption of Buildings: An Improved CBR Approach

The first example of adapting CBR is in predicting electricity consumption of buildings as presented by Aulon S. and Radu P. [2]. The main objective was to improve effectiveness of CBR in the aforementioned task to better manage thermal energy storage systems and reduce energy consumption. Other data-driven machine learning methods have already been successfully put to use, but they face the issue of data shortage - not all buildings have the required amount of information to power these algorithms. Luckily, this is where a CBR system could excel.

Earlier solutions for the same problem using CBR have been showing undesirably high error rates, which led Aulon S. and Radu P. to adopt some of the more advanced techniques, including the methods of finding the weights used in similarity measurements, as outlined by Stahl. [8] Their implementation cut down the prediction error by half, reached similar performance as Artificial Neural Networks when there was abundance of data, and even performed better when data was sparse.

4.2 A Data-Driven Approach for Determining Weights in Global Similarity Functions

When creating a CBR system, the data used in the case base needs to be specified to the domain of the tasks that the CBR system is supposed to solve. The knowledge on the domain needed for this may however be unavailable in early stages of development. To address this issue, Jaiswal and Bach [4] use the generalized CBR model to propose a method that generates weights for the features of the cases for the initial global similarity measure.

Their method uses an algorithm to give each feature a weight based the relevance of that feature to the task. This is achieved by using a set of scoring methods defined in the paper to give each feature a score - the weight of each feature is thus based on these scores. A percentage of features to be used in these scoring methods is also provided, because sometimes not all features presented in the dataset are useful. Jaiswal and Bach found that this method can be used to identify whether all features are needed to carry out the task of the CBR system. By using this method and gradually decreasing the percentage of features, the best initial similarity measure may be found before discussing the details with a domain expert, which can save time and effort.

4.3 Human-machine collaboration in online customer service – a long-term feedback-based approach

Employees in customer service face the challenge of providing correct and reliable responses while being consistent with their solutions within a short time frame. The incoming customer problem is answered by an employee with certain domain knowledge about the topic of the problem. Using a CBR system, retrieval of semantically similar cases is given to the employee to base their solution of past solved customer problems. The problem sent by the customer constitutes a case. When the case is solved, it is sent to the case base. Over time, this case base grows to a large set of cases.

The paper tasks itself on bettering case retrieval in the CBR system by using feedback from the employees on retrieved cases [3]. Understanding semantics is something humans excel at, compared to the machines. Thus, the long-term feedback approach attempts to utilize human knowledge on semantic similarity and couple this feedback with the motivation to better future information retrieval. The feedback is stored and reused to improve the retrieval for all users. With each added solution and accompanying feedback, employees increase the effectiveness of the CBR approach by advanced case retrieval enhanced by the feedback.

To enhance the retrieval of semantically related cases, each incoming customer problem is adapted before the Retrieval phase. Through this, only semantically similar cases are retrieved, rather than cases which are only syntactically similar. The knowledge about the semantic similarity between the cases for the adaptation process is acquired from the feedback on semantic similarity from past employees use of the same approach.

5 Conclusions and Further Work

Case-Based Reasoning is a learning paradigm with an active research community that are to this day making advancements in the field, bettering the overall methodology and efficiency of CBR to further its goals as a solution-driven machine learning process. New and exciting discoveries are made every day with interesting ramifications for the rest of the Artificial Intelligence field. CBR is by itself also a paradigm that can take a very flexible form in it's use cases due to it being so contextually dependent.

The purpose of this paper, in part, was to illustrate how broad the range of topic in CBR truly can be. The Generalised CBR cycle presented in Chapter 3 furthers this notion and gives more opportunities to improve upon the standard model. It did so by further breaking down the structure of CBR. However it is not a panacea for highlighting all the aspects of a CBR application. [8, p. 513] Naturally, two further work pointers can be outlined.

First, Stahl dedicated most of the paper to the advantages that Generalized CBR brings to similarity measures, and so have a good portion of current applications that have been evaluated for chapter 4. Hence, looking into the opportunities that the Generalised CBR cycle provides for the processes of adaptation, evaluation, and any of the four R's would be a good start.

Second, the competence of Generalized CBR could also be scrutinized. The advantage over the standard model is undeniable, but the potential pitfalls are yet to be identified. This also presents an opportunity to use the advanced model as groundwork to improve it further.

References

[1] Plaza E. Aamodt A. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches". In: *Ai Communications* 7 (2001), pp. 39–59.

[2] Radu P. Aulon S. "Predicting the Electricity Consumption of Buildings: An Improved CBR Approach". In: (2016). URL: https://www.researchgate.net/publication/310766399_Predicting_the_Electricity_Consumption_of_Buildings_An_Improved_CBR_Approach.

[3] Roland Graef et al. "Human-machine collaboration in online customer service – a long-term feedback-based approach". In: *Electronic Markets* (May 2020). DOI: 10.1007/s12525-020-00420-9.

[4] A. Jaiswal and K. Bach. "A Data-Driven Approach for Determining Weights in Global Similarity Functions". In: *ICCBR*. 2019.

[5] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[6] Paul Jarle Mork. *selfBACK*. 2020. URL: http://www.selfback.eu/.

[7] Michael M. Richter and Rosina Weber. *Case-Based Reasoning, A Textbook*. Springer-Verlag, 2013.

[8] Armin Stahl. "Learning Similarity Measures: A Formal View Based on a Generalized CBR Model". In: *German Research Center for Artificial Intelligence DFKI GmbH* (2005), pp. 507–521. URL: https://link.springer.com/chapter/10.1007/11536406_39.