

## A Constructive System Forgetting as part of Case-Based Reasoning

Abdulmajid H. Mohamed<sup>1</sup>, Ilham Salem Ben-salem<sup>2</sup>, Hanan Ettaher Dagez<sup>3</sup>

<sup>1</sup>Sebha University, Faculty of Science,  
PO Box 187758, Central Post office, Sebha, Libya

<sup>2</sup>Department of Computer Science, Tripoli University,  
Alfornaj street, Tripoli, Libya

<sup>3</sup>Faculty of Information Technology, Tripoli University,  
Alfornaj street, Tripoli, Libya

### Abstract:

The human behavior has always been very influential in systems engineering. In fact, AI methods and techniques are largely influenced by the human behavior in the form of mental and mechanical capabilities. The human learning, persevering and then recalling knowledge is the focal point in artificial intelligence research. But less attention is paid to forgetting as one of the characteristics that have a very positive role in human intelligence. This paper seeks to integrate data forgetting as part of the behavior of case-based reasoning systems. The aim is to improve the performance of CBR systems by filtering out irrelevant cases as part of the machine behavior in the form of CBR systems. The paper presents a prototypical implementation of the of a forgettable CBR system that provides course recommendations as part of student registration system. Experimental work has been carried out using historical data of postgraduate students in the Computer science department, Tripoli University, Libya.

Keywords: Knowledge management, CBR, machine forgetting, recommendation systems.

### 1. Introduction

The case-based reasoning (CBR) is introduced as an artificial reasoning mechanism. It is originated from the idea that human beings tend to recall their previous experience to solve new similar issues. According to Aamodt and Plaza [1], CBR is a technique used “to solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation”. Currently, CBR is widely used to build knowledge management systems, such as recommendation systems and other decision-support systems, however, the efficiency of these systems tend to degrade due to part of captured knowledge become irrelevant. Therefore, the knowledge accumulated in the case base must be updated in order to keep pace with the active appropriateness of captured knowledge over time. For this reason, there must be a continual maintenance of knowledge captured by these systems. knowledge evolution issue has not been addressed so far [2].

Though forgetting is socially regarded as a bad habit, but it also plays a very constructive role in human behavior [3], likewise, it is expected to play a very constructive role in supporting the behavior of knowledge-based systems as well. This research aims to apply the concept of machine forgetting within the case-based reasoning process. This is applied as part of the academic recommendation system for graduate students in the Department of Computer Science, Faculty of Science, , University of Tripoli. The main benefit of implementing this forgettable CBR system is to improve the performance and competency of the knowledge base (case base) attached to recommendation systems.

### 2. Case Based Reasoning

The concept of case-based reasoning is first emerged in the work of Schank and Abelson in 1977[4]. It is strongly coupled with AI, and “it is regarded as a subfield of machine learning” [1]. CBR has been successfully exploited in many AI applications especially diagnostic applications where historical

experience is intensively reused. Within CBR systems, the 'case' represents the basic building block of these systems. Each 'case' represents an experience of a previously solved problem [5]. All Cases is stored in a 'cases base' where different types of transactions related to adding, retrieving and updating cases can be performed [6]. In terms of data representation, each case is described by a set of attributes such as problem description, solution, characteristics, owner, etc [7]. In the course of searching for solutions to similar problems, users define attributes of the new problem to be solved. These attributes are used to retrieve the most similar cases from the 'case base'. Retrieved cases can be reused as is, or modified to suit the slightly different context of the new problem. According to Tiwana [8], as more new cases are added, the case-based reasoning becomes increasingly powerful and accurate. Fig 1 illustrates a general CBR cycle[9] the figure illustrates the four tasks that exist as a part of all CBR systems, this includes tasks RETRIEVE, REUSE, REVISE and RETAIN. CBR systems can be used in many decision-support applications. A typical application of CBR applications is the recommendation system.

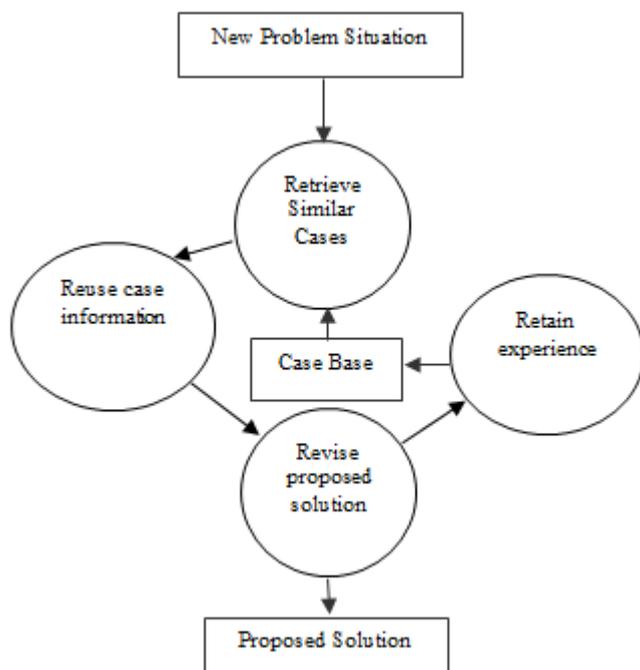


Fig. 1: General CBR Program Cycle

### 3. Students Recommendation systems

A recommendation system is a specific type of intelligent systems, which exploits historical user ratings on items and/or auxiliary information to make recommendations on items to the users [10]. It refers to the use of domain-specific applications to

provide inquirers with information and advice to help them decide what to choose from huge alternative information. In terms of students' recommendation systems, they can be used to help students to choose the most relevant courses among the courses offered in each calendar year. Selection of the most relevant system advises is made based on the reasoning through the performance of historical cases.

As any decision support system, the three most prominent reasoning methodologies available to create these systems are rule-based, case-based and model-based reasoning [11]. But due to the complex rules that govern most decision support systems, we believe that the flexibility of CBR systems is more suitable to develop student's recommendation systems. A typical software solution for automating academic advising might include a rule-based expert system. But, the sheer amount of knowledge required would make it extremely difficult to express in sequential rules [12].

### 4. Machine forgetting

Basically forgetting could be defined as a mental state whereby humans fail to recall what has been previously known or resided in human memory [13]. But, as Gregory Bateson, the British social scientists, once said "You can't live without an eraser", in other words, human forgetting tend to play a very constructive role in human lives. Systems such as CBR systems also need to incorporate mechanisms for knowledge forgetting whereby irrelevant 'cases' are dismantled, otherwise, the effectiveness and the performance of these systems could be compromised. Because as much as reusing historical cases would help to deliver cost-saving, time-saving and reliability, but eventually there is no guarantee that the advices concluded by a CBR system would always be relevant. In this regard, [3] argues that a CBR system has to be forgetful in the sense that irrelevant cases should be constructively forgotten. This constructive forgetting is aimed to facilitate the inhibition of reusing irrelevant data. This can be accommodated by Markovitch and Scott' process model where parts of the organised representation (i.e. of experience) are rearranged and dismantled [14]. In order to incorporate this feature as part of CBR systems, various strategies for determining which cases to remove are proposed. The simplest forgetting strategy is to pick a random case to

remove, but the chances of removing successful cases are very high [15]. Another metric which is widely adopted, is how frequently a case is reused. However, these deletion policies could lead to negative consequences such as the accident deletion of critical cases. Meanwhile, the deletion of critical cases can significantly reduce the competence of a CBR system, rendering certain classes of target problems permanently unsolvable [16]. Further details about artificial forgetting can be explored in ([17],[18],[2],[19]).

### 5. The realization of the CBR systems forgetting

There are many motivational causes for realizing the system forgetting. It is well-known that CBR systems are very vulnerable to the Utility and Swamping problems. The Swamping problem is caused by increasing size of the case base. As the size of the case base increases, searching through the case base and retrieval become very costly in terms of systems performance. On the other hand, the Utility problem is caused by not all the historical cases is always relevant, and irrelevant cases might be retrieved as a result.

For this reason, the system forgetting part is mainly for preparing the case base by eliminating the cases that could badly affect both quality of the recommendations and the system performance. The following steps represent the stages of realising the system intentional forgetting. The following subsections represent the steps of realizing the forgettable CBR system.

#### 5.1 Step 1: Setup initial case set

This first step is to filter out only the cases related to students who passed all the courses of the program. These cases would then represent the active set of cases.

#### 5.2 Step 2: Cases clustering

At this step, clustering algorithms are used to make cases clustering based on the range of similarity. DBSCAN algorithm has been utilised to form the cases clustering. This clustering algorithm arranges the case clusters based on two parameters:  $\epsilon$  and  $MinPts$ . For a point  $p$ , the  $\epsilon$ -neighborhood of  $p$  is the set of all the points around  $p$  within distance  $\epsilon$ . If the number of points in the  $\epsilon$ -neighborhood of  $p$  is no smaller than  $MinPts$ , then all the points in this set, together with  $p$ , belong to the same cluster [20]. The output

of this process is the clustering map that categories the historical cases into internal and outlier cases as shown in Fig. 2.

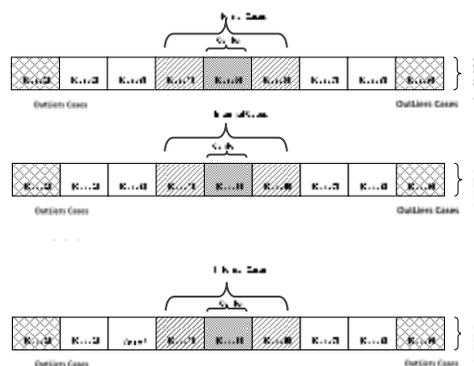


Fig. 2: Case base clustering and the formation of internal and outlier cases.

#### 5.3 Step 3: Selecting internal cases

Internal cases represent the cases which are very similar, and because of the similarity factor, eliminating some of these cases would not affect the quality of output recommendations. This is unlike the random elimination strategy where valuable and unique cases could be discarded. In other words, there is always a representative to eliminated cases from the crowded case base. Even one representative case would provide a relevant recommendation for respective students.

#### 5.4 Step 4: Selecting outlier's cases

At this stage, unique or dissimilar cases are identified. The data values of those cases vary compared to most of the stored cases, and its elimination cannot be compromised. These cases are identified based on the Interquartile Rang:

$$IQR=(Q3-Q1)/2$$

Where: Q1=25% of the sorted cases

Q2=50% of the sorted cases

Q3=75% of the sorted cases

#### 5.5 Step 5: Case-base updating

At this stage, all internal and outlier cases are saved in a new case base. This represents the active case base after eliminating the forgettable cases. Outlier cases are not eliminated fully, but all outlier cases that represent courses with grades which are greater than or equal to 70% are retained as part of the newly formed case set.

## 6. Experimental work and evaluation

The proposed approach is aimed to improve the CBR system performance, memory utilisation, and improving the efficiency of the system. It is estimated that as a result of case base sterilization, the forgettable CBR (FCBR) should result in a reduced size of the input case base. This of course would lead to a better retrieval time and reduced size of the case base. Two different clustering algorithms namely DBSCAN and K-MEANS were used; the aim is to explore the efficiency of each method as a clustering algorithm, which is set as a sub goal. The prototypical implementation of the proposed approach is evaluated based of the standard methodology from Information Retrieval. We specifically employed the followings factors: *Precision*, *Recall* and *F-measure* (combination of Precision/Recall).

### 6.1 Case base volume

This factor calculates the percentage of case base reduction after executing the forgetting procedure. The case base volume is calculated based on the following formula:

$$\text{volume} = \frac{\text{the no. of post\_forgetting cases}}{\text{the no. of original cases}}$$

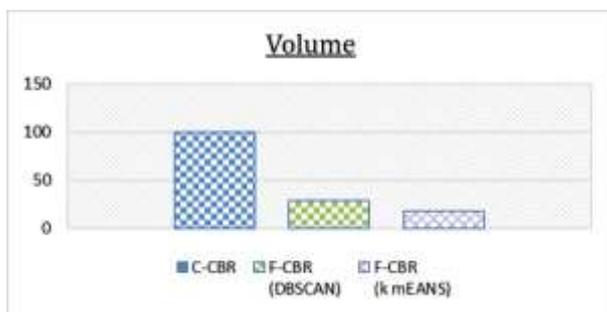


Fig.3: Percentage of case base volume reduction

Fig. 3 shows the percentage of case base reduction as a result of practicing the forgettable CBR. Notice that the scale of the case base size reduction is higher using K-Means as a clustering algorithm. The K-means algorithm reduced the case base size to 17%, while the DBSCAN algorithm reduced the base size of the case to 28%, which is considered a good performance for forgetfulness in terms of size.

### 6.2 Accuracy

In order to evaluate the accuracy of the proposed approach, Leave one out [21] method is applied to generate advices for 5, 6 and 7 courses. Fig. 4 shows

that the accuracy of the forgettable CBR approach (i.e. FCBR) is more than the conventional CBR. Though the accuracy of FCBR is even more compared to the clustering made by DBSCAN algorithm.

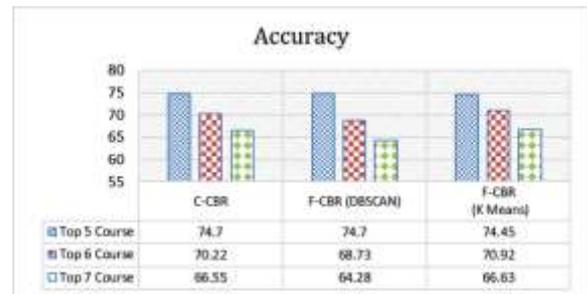


Fig.4: The Percentage of the levels accuracy between conventional and forgettable CBR systems.

## 7. Conclusion

This paper presents a prototypical implementation of a forgettable case-based reasoning system. The aim is to incorporate the characteristic of knowledge forgetting as a strategy for maintaining the case based attached to CBR systems. This approach is realised in the form of a student courses recommendation system. Initial cases were populated from the historical data of students' registration system maintained by the Computer science department, Tripoli university-Libya. Compared to conventional CBR systems, the results of the prototype shown positive results in terms of both the volume of the case base and the accuracy of retrieved cases.

## References

1. Aamodt, A., Plaza, E. (1994), Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AICom - Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59.
2. Mohamed, A. H. (2006) Managing Evolution in Software-Engineering Knowledge Management Systems, 1<sup>st</sup> IEEE International Conference on Digital Information Management (ICDIM 2006), Bangalore, India
3. Mohamed, A. H. (2010) Complementing the CBR by constructive forgetting, 2<sup>nd</sup> Libyan-Tunisian workshop on new directions in advanced computing, 10-12 May, 2010, Beni Walid, Libya.

4. Roger C. Schank and Robert P. Abelson (1977), "Scripts, Plans, Goals and Understanding." Yale University, New Jersey, 1977
5. Rahman, H., Barua, S., Ahmed, M. U., Begum, S., & Hök, B. (2016). A Case-Based Classification for Drivers' Alcohol Detection Using Physiological Signals. In The 3rd EAI International Conference on IoT Technologies for HealthCare HealthyIoT'16, 18 Oct 2016, Västerås, Sweden
6. [Mohamed, A. H. (2008) Case-Based Reasoning PC purchasing e-Advisor, National conference for information technology and communication (ITC'08), Tripoli, Libya.
7. Mohamed, A. H., Lee, S. P., Salim, S. S. (2004) An Ontology-based Knowledge Model for Software Experience Knowledge, *Journal of Knowledge Management Practice*, Volume 5, May 2004, WWW URL: [www.tlinc.com/jkmpv5.htm](http://www.tlinc.com/jkmpv5.htm)
8. Tiwana, A. (2000) Knowledge Management Toolkit, the: Practical Techniques for Building a Knowledge Management System, Upper Saddle River, NJ :Prentice Hall PTR, ISBN 0-13-012853-8, pp.224.
9. Althoff, K-D (2001) Case-based reasoning. In Chang, SK (ed.) Handbook on Software Engineering and Knowledge Engineering, vol. 1 Fundamentals. World Scientific, pp. 549–588.
10. Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69, 29-39.
11. Rintala, L., Leikola, M., Sauer, C., Aromaa, J., Roth-Berghofer, T., Forsén, O., & Lundström, M. (2017). Designing gold extraction processes: Performance study of a case-based reasoning system. *Minerals Engineering*, 109, 42-53.
12. Sandvig, J., & Burke, R. (2005) Aacorn: A CBR recommender for academic advising. *Technical Report, TR05-015, DePaul University*.
13. Mehrizi, M.H.R, Niri, M. B., Rahimi, M., Yarmand, H.(2009) Knowledge Active Forgetting: The Forgotten Side of Knowledge Management, PICMET 2009 Proceedings, August 2-6, Portland, Oregon USA.
14. Markovitch, S. & Scott, P. D. 'The role of Forgetting in Learning', proc. Of the fifth Int. conf. On Machine Learning, Ann Arbor, MI:Morgan Kaufmann,1988
15. Kira, Z. and Arkin, R.C. (2004) "Forgetting Bad Behavior: Memory Management for Case-based Navigation", *Proc. IROS-2004*, Sendai, JP, 2004.
16. B. Smyth, M.T. Keane (1995) Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems, in: 14th International Joint Conference on Artificial Intelligence, 1995, pp. 377–382
17. Lee, S. P., Mohamed, A. H. and Salim, S. S. (2001) Towards an Intelligent Organisational Memory System, 2001, *Knowledge Management International Conference and Exhibition (KMICE 2001)*, Langkawi, Malaysia.
18. Mohamed, A. H., Lee, S. P., Salim, S. S. (2005) LiSER: An Organisational Memory subsystem to support Organisational Learning in software development organisations, *International Arab Journal of Information Technology (IAJIT)*, vol.2, no.1, 2005.
19. Mohamed, A. H. (2008) Capturing Software-Engineering Tacit Knowledge, *European computing conference (ECC'08)*, Malta.
20. Tran, T. N., Drab, K., & Daszykowski, M. (2013) Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, 120, 92-96.
21. Bogaerts, S. and Leake, D. (2004) "IUCBRF: A Framework for Rapid and Modular Case-Based Reasoning System Development," 2004.



Abdulmajid H. Mohamed received the MSc in Information systems from Leeds university, UK in 1993 and the PhD degree in software engineering from the University of Malaya, Malaysia, in 2004. Since 2007 he become an academic staff in the department of computer science, Sebha university. But since 2013 he is on secondment to the Libyan Students Bureau, Malaysia. Meanwhile, he is active

in supervising PhD students in UPM University, Malaysia. His research interests include software knowledge management, human-computer interaction, ICT & Islam and requirements engineering.



Ilham Salem Ben-salem is an MSc student in the department of computer science, Tripoli University. Her research interests include Web programming, e-learning and case-based reasoning systems.



Hanan Ettaher Dagez received the MSc in software engineering from the University of Malaya, Malaysia, in 1999 and the PhD degree in E-learning from the University of Malaya in 2010. She joined the faculty of information technology, Tripoli University, Libya in 2013 as an academic staff. Currently she is associated professor and the head of Information System Department at the faculty of Information technology, Tripoli University.