Overview of Maintenance for Case based Reasoning Systems

Abir Smiti
LARODEC
University of Tunis
Institut Supérieur de Gestion
Tunis, Tunisia

Zied Elouedi
LARODEC
University of Tunis
Institut Supérieur de Gestion
Tunis, Tunisia

ABSTRACT
The success of a Case Based Reasoning (CBR) system depends on the quality of case data and the speed of the retrieval process that can be expensive in time especially when the number of cases gets large. To guarantee this quality, maintenance the contents of a case base becomes necessarily. As a result, the research area of Case Base Maintenance (CBM) has drawn more and more attention to CBR systems.
This paper provides a snapshot of the state of the art, reviewing some important methods of maintaining case based reasoning. We introduce a framework for distinguishing these methods and compare and analyze them. In addition, this paper also presents simulations on data sets from U.C.I repository to show the effectiveness of some CBM methods taking into account the accuracy, the size and the retrieval time of case bases. Our simulation results which are obtained by compared well known reduction techniques show that these CBM methods have good storage reduction ratios, satisfying classification accuracies and short retrieval time.

General Terms
Machine Intelligence, Case Based Reasoning, Algorithms.

Keywords
Case based reasoning, Case base maintenance, evaluating case base, Case base partitioning, Clustering, Selection method, Case base optimization.

1. INTRODUCTION
One of the principal goals of Artificial Intelligence (AI) is to conceive systems able to reproduce human reasoning. Case Based Reasoning (CBR) [1, 2] is a variety of reasoning by analogy. It is a methodology to model the human way in reasoning and thinking. It offers a technique based on reusing past problem solving experiences to find solutions for future problems: A new problem is solved by retrieving one or more previously experienced cases, reusing the case in one way or another, revising the solution based on reusing a previous case, and retaining the new experience by incorporating it into the existing case base (CB).

CBR is able to find a solution to a problem by employing its luggage of knowledge or experiences which are presented in form of cases. Typically, the case is represented as a pair "problem" and "solution". Cases are grouped in a case base. Each case describes one particular situation and all cases are independent from each other.

To solve the problems, CBR system calls the past cases, it reminds to the similar situations already met. Then, it compares them with the current situation to build a new solution which, in turn, will be added to the case base.

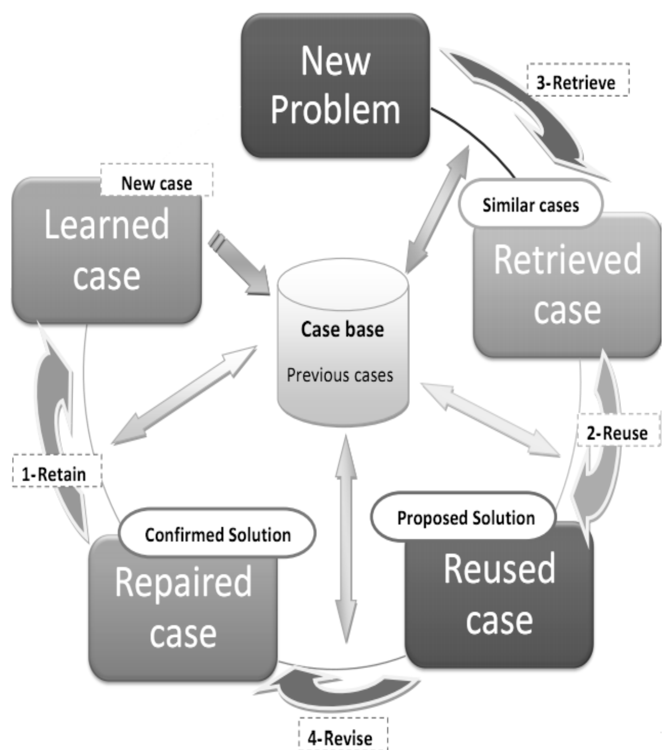

Fig. 1 CBR cycle

As mentioned, a general CBR cycle may be described by four top-level steps (see Fig 1):

1. RETRIEVE the most similar cases: During this process, the CB reasoner searches the database to find the most approximate case to the current situation.

2. REUSE the cases to attempt to solve the problem: This process includes using the retrieved case and adapting it to the new situation. At the end of this process, the reasoner might propose a solution.

3. REVISE the proposed solution if necessary: Since the proposed solution could be inadequate, this process can correct the first proposed solution.

4. RETAIN the new solution as a part of a new case.

This process enables CBR to learn and create a new solution and a new case that should be added to the case base [1].

CBR has been used to create several applications in a wide range of domains including medicine, law, management, financial, electronic commerce, customer support, software engineering, etc. For instance, in the e-commerce fields, CBR has been used as an assistant in e-commerce stores and as a reasoning agent for online technical support, as well as an intelligent assistant for sale support or for e-commerce travel agents. It uses cases to describe commodities on sale and identifies the case configuration that meets the customers' requirements [3].

Like the majority of systems that are built to work for long periods of time and are developed to deal with large amounts of information and cases, CBR suffers from a large storage requirement and a slow query execution time concerning case-research phase. To guarantee the system's quality, maintenance of CBR system becomes necessarily. As a result, there has been a significant increase in the research area of Case Base Maintenance. Its objective is to guarantee a good operating in time of an information processing system and to facilitate future reasoning for a particular set of performance objectives [4].

Various case base maintenance policies have been proposed to maintain the Case Base (CB). Most existing works on CBM are based on updating a case base, adding or deleting cases to optimize and reduce the case base. These policies of CBM involve different operations [7]: out dated, redundant or inconsistent cases may be deleted; groups of cases may be merged to eliminate redundancy and improve reasoning power; cases may be re-described to repair incoherencies. However, many evaluations vaguely compare the CBM algorithms and limit there experimentation on reduction size and the accuracy, while the system's quality of the case base is typically ignored.

The propose of this paper is to present a general view of the approaches concerning CBM; it provides a snapshot of the state of the art, presenting some important methods of maintaining case based reasoning. We introduce a framework for distinguishing these methods and compare them and in order to evaluate the performance rate of some CBM policy, we test on real databases obtained from the U.C.I. repository [29], taking into account the accuracy, the size and the retrieval time of case bases. Our paper is organized as follows: In section 2, criteria for evaluating case base will be approached. Section 3 reviews some existing CBM policies. Section 4 presents and analyzes experimental results carried out on data sets from the U.C.I. repository [29]. Finally, Section 5 ends this work.

2. Criteria for evaluating case base

To know if the CB is able to give a correct result or not, how do we evaluate the quality of CB? To answer this question, several measures were proposed in order to carry out an evaluation concerning case-base.

In fact, we noted that the performance and the competence of case base have been given much attention in the literature. These measures are an essential tool for use in all stages of system development. They are particularly important during system maintenance, where knowledge is added, deleted and modified to effect system adaptation and improvement.

In this Section, we describe concrete performance and competence measures that implement quality measures for case base maintenance [4-7, 9]:

- Competence measured by the range of problems that can be satisfactorily solved. There is a strong relationship between the competence of a CBR system and the cases in its case-base, however, the precise nature of this relationship is not clear [10-12]. The most recently explicit algorithmic model of competence for case-based reasoning systems was suggested by Smyth and McKenna [12], they defined two key fundamental concepts which are coverage and reachability.
    - Coverage is an important competence property. Coverage of a case is the set of target problems that it can be used to solve. The overall coverage of a case base in relation to a set of queries is the total number of covered queries divided by the total number of queries in the query set.
    - Reachability is an important competence property. Reachability of a target problem is the set of cases that can be used to provide a solution for the target.

In order to have a case base with good competence, its coverage ratio must be high and its reachability rate must be low.

- Performance is the answer time that is necessary to compute a solution for case targets. This measure is bound directly to adaptation and result costs. Performance depends critically on the accuracy and the cases stored in the case base:
    - Accuracy takes into account the correctness of the actual solution of the retrieved most similar case given a query. We say, a case within the case base classifies a query, if the problem description of this case is most similar to the query. Similarly, a case correctly classifies a query, if the case classifies the query and the solution of the case actually solves the query. The accuracy of a case base in relation to a set of queries is the number of correctly classified queries divided by the total number of queries in the query set.
    - Storage Space takes into account that retrieval speed is related to costs and hence, customers want quick answers to their problems. In turn, retrieval time is related to storage space. Hence, we define the performance measure for storage space, which is the total number of cases in the case base.

Many CBR systems use retrieval methods whose efficiency is related to the case base size, and under these conditions the addition of redundant cases serves only to degrade efficiency by increasing retrieval time. There are several strategies in the literature dedicated to the study of these criteria [6, 8].

3. CASE BASE MAINTENANCE POLICIES

The objective of CBM approaches is reducing case research time: this will be done on different types of strategies. There are several ways of categorizing existing case base maintenance approaches. In [4], the maintenance policies were categorized in terms of data collections which explain how they gather data relevant to maintenance, triggering which decide when to trigger maintenance, the types of maintenance operations available and how selected maintenance operations are executed. Moreover, Pan et al. [13] classified CBM policies in search direction, order sensibility and evaluation criteria. Other research mainly relied on the deletion and the revision of irrelevant and redundant cases.

In this paper, we classify CBM algorithms in three classes: one class, following a partitioning policy that builds an elaborate case base structure and maintains it continuously. Second class, following selection based data reduction methods that start with an empty set, select a subset of instances from the original set and add it into the new one. The third class, following a deletion policy based on cases' competence to optimize the case base.

These policies involve different operations: out dated, redundant or inconsistent cases may be deleted; groups of cases may be merged to eliminate redundancy and improve reasoning power; cases may be re-described to repair incoherencies [7].

In the next Subsections, we provide an overview of some of the important approaches to CBM.

3.1 Case base partitioning strategies

The partitioning policy creates a collection of distributed case bases, where each element in the distributed case base structure is one cluster created as a result of the clustering process. From each cluster we build a representative case, which takes a subset of the attributes. Therefore, the attribute with rich information content are selected, and may possess more potential to cover a wider CB structure. These policies allow the addition and deletion of cases in each small CB, without using the whole base in the same time. We may cite some works:

Shiu et al. [14] proposed a case-base maintenance methodology based on the idea of transferring knowledge between knowledge containers in a case-based reasoning (CBR) system. A machine-learning technique, fuzzy decision-tree induction, is used to transform the case knowledge to adaptation knowledge. By learning the more sophisticated fuzzy adaptation knowledge, many of the redundant cases can be removed. This approach is particularly useful when the case base consists of a large number of redundant cases and the retrieval efficiency becomes a real concern of the user. The method of maintaining a case base from scratch consists of four steps: First, an approach to learning feature weights automatically is used to evaluate the importance of different features in a given case base. Second, clustering of cases is carried out to identify different concepts in the case base using the acquired feature-weights knowledge. Third, adaptation rules are mined for each concept using fuzzy decision trees. Fourth, a selection strategy based on the concepts of case coverage and reachability is used to select representative cases. The proposed methodology is particularly useful when a case base has a lot of redundancy that is not caused simply by repeated cases but rather by the interaction among features. This type of redundancy will seriously affect the quality of the problem-solving ability of a CBR system. By learning the feature weights of the cases, this type of redundancy can be illuminated [16]. One of the drawbacks of this approach is the complexity issues. In fact, the time and space complexity of this method is solely dependent on the complexity of generating the fuzzy adaptation rules and selecting the representative cases, and that give a high complexity.

Also, for the branch of CBM partitioning, there is the method proposed by Cao et al. [15]. This methodology is mainly based on the idea that a large case library can be transformed to a small case library together with a group of adaptation rules, which take the form of fuzzy rules generated by the rough set technique. This approach is the same of Fuzzy decision tree approach [16], just the adaptation rules will be mined for each concept using fuzzy-rough approach. By applying such a fuzzy-rough learning algorithm to the adaptation mining phase, the complexity of case base maintenance is reduced, and the adaptation knowledge is more compact and effective compared to the maintenance results of using fuzzy ID3.

In opposite, these two methods suffer from some drawbacks; we can mention the problem of incremental cases: when a new case arrives, we should rebuild the whole strategies again.

We can mention, in the branch of case base partitioning, the COID method: Clustering, Outliers and Internal cases Detection, proposed in [24]. It uses the clustering technique to create small case bases which are easy to treat and to maintain each one individually. Then, it applies outliers and internal cases detection methods, for each partition, to reduce the size. This method aims at selecting cases which influence the quality of the case base, from each cluster. Thus, in this method, the clustering ensures that each case base is small and it is easier to maintain each one individually. For each small group, the cases of type outliers and the cases which are near to the center of the group are kept and the rest of cases is removed. An extension of COID is presented by WCOID: a Weighting, Clustering, Outliers and Internal case Detection [25] that adds feature weights to achieve a higher reduction and better competence case bases.

In [17], the method partitions cases into clusters where the cases in the same cluster are more similar than cases in other clusters. Clusters can be converted to new case bases, which are smaller in size and when stored distributed, can entail simpler maintenance operations. The contents of the new case bases are more focused and easier to retrieve and update. Clustering technique is applicable to CBR because each element in a case base is represented as an individual, and there is a strong notion of a distance between different cases. The density-based is used as a clustering method. After the partition of a large case base, a domain expert can build some smaller case bases on the basis of clustering result. Each cluster has a case base name and a list of keywords. The case name is the description of the case base. It is a set of the most frequently used words by the cases in the case base. There is a set of attributes associated with the case base. They are all the attributes that are associated with the cases in the case base. This method is simple and easy to run because it decomposes the large case base into small groups of closely related cases. Since the size of a cluster is relatively small, any simple CBR retrieved method can be used.

3.2 Selection based data reduction methods

Selection method aims to reduce a dataset by selecting representatives from the training dataset. From selection based data reduction methods, we can mention:

CNN (Condensed Nearest Neighbor Rule) proposed by Hart [21], it is a redundancy technique that incrementally builds an edited case base from scratch. Cases are added to a new case base, and removed from the original one, if and only; if it cannot be correctly classified by the edited case base build so far. CNN makes multiple passes through the original case base until no more additions can be made. Actually, CNN suffers from serious problems: It is sensitive to noise, so it can view the noisy cases as important exceptions and give an unsatisfying result.

The Reduced Nearest Neighbor Rule (RNN) [23] starts from using the whole training set as the initial reduced set: S = T, and removes each instance from S if such a removal does not cause any other instances in T to be misclassified by the instances remaining in S. The process repeats until no further reduction can be achieved. However, the iterative process is very time-consuming if the original training set is large and it is computationally more expensive than Harts condensed NN rule (CNN).

The Selective Nearest Neighbor Rule (SNN) devised in [22], improves the CNN and RNN by ensuring that a minimal consistent subset is found. This selection method is based on the following rule: all instances in the training set must be closer to an instance in the selective set than any instance of a different class found in the training set. The algorithm for SNN is more complex than most other reduction algorithms, and the learning time is significantly greater. Besides, it is sensitive to noise, though it will tend to sacrifice storage more than accuracy when noise is present.

The Edited Nearest Neighbor Rules (ENN) [31] removes all instances which have been misclassified by the k-NN rule from the training set. ENN keeps all the internal instances but deletes the border instances as well as the noisy instances, unlike the CNN algorithm.

Repeated Edited Nearest Neighbor (RENN) [32] applies the ENN algorithm repeatedly until all remaining instances have a majority of their neighbors with the same class, which continues to widen the gap between classes and smooth the decision boundary of ENN. The ANN (All k-NN) algorithm [32] is similar with the iterative ENN with the only exception that the value k is increased after each iteration.

Aha et al [26], proposed a series of Instance Base Learning algorithms: IBL. IB1 is just simple 1NN (one Nearest Neighbor). IB2 begins with an empty training set and it adds each case if it is not classified correctly by the instances already in the training set. The drawback of IB2 is that it is very sensitive to noisy cases. IB3 uses the classification accuracy to decide whether to add new training cases into a case base if the case is classify correctly at a satisfactory level.

3.3 Case base optimization strategies

Many researchers have addressed the problem of CBM optimization. From a given case base, these strategies value cases according to criteria in order to be able to suppress and bring the case base to a specific number of cases. The evaluation criteria, like competence and performance, have been used in different methods.

3.3.1 Traditional deletion methods

Some researchers advocate a random deletion policy: A random item is removed from the knowledge-base once the knowledge-base size exceeds some predefined limitations [18]. This is an easy-to-use policy, it is very easy to implement, it can work very well and can often be as effective as more principled and expensive methods. On the other hand, it does not give convincing and satisfying results concerning the optimization of the case base size.

Another approach, ironically policy, depends on the frequency of each case. It is a slightly more complicated method. However, this approach cannot avoid deleting important cases. In other words, some cases those are very good for reuse, are possibly deleted [19]. Like random deletion policy, ironically policy is simple but it does not give a satisfying result. In fact, ironically policy degrades the competence of the case base more than the random deletion policy. The problem with both of these approaches is that important cases can be deleted by mistake.

Utility deletion UD is based on Minton's utility metric which chooses a case item for deletion by estimating its performance benefits. This utility deletion method removes case items with negative utility (see Eq. 1).

Utility = [ApplicationFreq * AvgSavings] – MatchCost (Eq.1)

Where ApplicationFreq is the number of times the case has been retrieved, AvgSavings is how much time you save if you have that case and MatchCost is the cost to compute similarity. The utility problem manifests itself as a trade-off between the solution quality associated with large CBs and the efficiency problem of working with a large CB. System efficiency is measured by taking the mean time to solve a target problem; note that the decreasing solution times correspond to an increase in efficiency. The solution quality is bound to the percentage of good answers, provided by the system. Solution quality increases with CB size [20].

In summary, these traditional deletion policies can have disastrous results for case based reasoning. The deletion of critical cases can significantly reduce the competence of a CBR system, rendering certain classes of target problems permanently unsolvable. Thus, various approaches have been designed to address this problem where authors consider a competence-preserving approach to case deletion.

3.3.2 CBM methods based on cases' competence

Several approaches are focused on preserving competence of the case memory through case deletion, where competence is defined in Section 2. We can mention competence preserving deletion [30]. In this method, Smyth and McKenna defined several performance measures by which one can judge the effectiveness of a CB, such as: coverage and reachability (see Section 2). Their method categorized the cases according to their competence, three categories of cases are considered:

- Pivotal case: If it is reachable by no other case but itself and if its deletion directly reduces the competence of system. Pivotal cases are generally outliers.
- Support cases: They exist in groups. Each support case provides similar coverage to others in group. Deletion of any case in support group does not reduce competence. Deletion of all in group equivalent to deleting pivot
- Auxiliary case: If its coverage set is a subset of the coverage of one of its reachable cases, it does not affect competence at all; its deletion makes no difference.

This reduction technique Footprint deletes auxiliary problems first, then supports problems, finally pivotal problems. This approach is better than traditional deletion policies for preserving competence. However, the competence of case base is not always guaranteed to be preserved [6].

Smyth and McKenna created the deletion FUD, which is the hybrid strategy between footprint deletion and utility deletion. First, the footprint method is used to select candidates for deletion. If there is only one such candidate then it is deleted. If there are a number of candidates, therefore rather than selecting the one with the least coverage or the largest reachability set, the candidate with the lowest utility is chosen.

Yang and Zhu [33] proposed a method that builds up an edited case base by adding cases incrementally to the set. They substantiate their algorithm with a theoretical analysis showing that it provides a well defined lower-bound on coverage.

Salamo and Golobardes [27] proposed the Accuracy-Classification Case Memory (ACCM) and Negative Accuracy-Classification Case Memory (NACCM) where the foundations of these approaches are the Rough Sets Theory. They used the coverage concept which is computed as follows:

Let $T = \{t_1; t_2; \ldots; t_n\}$ be a training set of instances, $\forall t_i \in T$:

$$\text{Coverage}(t_i) = \text{AccurCoef}(t_i) \oplus \text{ClassCoef}(t_i) \qquad \text{(Eq.2)}$$

Where AccurCoef measure explains if an instance t is an internal region or an outlier region, ClassCoef measure expresses the percentage of cases which can be correctly classified in T, and the $\oplus$ operator is the logical sum of both measures.

The main idea of ACCM reduction technique is firstly, to maintain all the cases that are outliers, cases with Coverage =1.0 value, are not removed. This assumption is based on the fact that if a case is isolated, there is no other case that can solve it. Secondly, to select cases that are nearest to the outliers and other cases nearby can be used to solve it because their coverage is higher. NACCM reduction technique is based on ACCM, doing the complementary process. The motivation for this technique is to select a wider range of cases than the ACCM technique. The same authors presented the DCBM dynamic model that allows to update the case base dynamically taking information from the learning process [28].

4. EVALUATION OF CASE BASE MAINTENANCE POLICIES

In this Section, we try to show the effectiveness of some CBM methods as well as the performance of the CBR system. The aim of the reduction techniques is to reduce the case base while maintaining as much as possible the performance of the system. Thus we will consider the following principal criteria:

1- Storage reduction (S): This is the rate of the reduction of size. The main objective of training set CBM methods is to reduce storage requirements. The percentage of final case base size (S) shows the percentage of case base maintained from the original training set. S denotes the average storage percentage which is the ratio in percentage included in the initial CB.

2- Accuracy (PCC): This is the concept descriptions' classification accuracy. The PCC rate will be the total number of the correct classified instances divided by the total number of instances tested, and is usually expressed as a percentage.

3- Retrieval time (t): This is the concept described the time which is risen when a case is retrieved, we compute the retrieval time exerted in 1-Nearest Neighbor algorithm.

The improvement of the storage reduction, the classification accuracy and the retrieval time are relevant criteria to judge the performance of CBM methods.

In order to evaluate the performance rate of CBM methods, we test on real databases obtained from the U.C.I. repository [29]. Details of these databases are presented in Table 1.

Table 1. Description of databases

Dataset	Ref.	#instances	#attributes
Breast-W	BW	698	9
Ecoli	EC	336	7
Ionosphere	IO	351	34
Iris	IR	150	4
Sonar	SO	208	60
Vehicle	VE	846	18
Yeast	YT	1484	8

We run the Clustering Outliers Internal cases Detection COID [24], Weighting Clustering Outliers Internal cases Detection WCOID [25], Condensed Nearest Neighbor algorithm CNN [21], Reduced Nearest Neighbor RNN technique [23], Edited Nearest Neighbor ENN [31] and Instance Based learning IBL schemes [26] on the previous data sets. (See Fig.1, Fig.2 and Fig.3)

Fig. 2 shows the results for CBM methods, it compares average storage size percentages.

Fig. 3 shows the average classification accuracy in percent of the testing data based on cross-validation of the previous algorithms.

Fig.4 shows retrieval time's result for the previous methods, in Seconds.

From these figures, it can be clearly seen that there is CBM methods are more efficient than the other ones by achieving a better cases reduction rate in some datasets, and other methods provide better accuracy values than others.

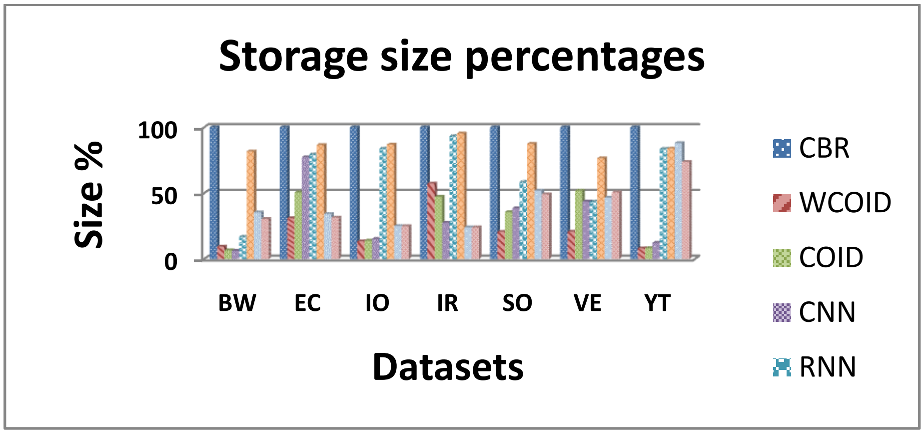

Fig. 2 Comparison of storage size

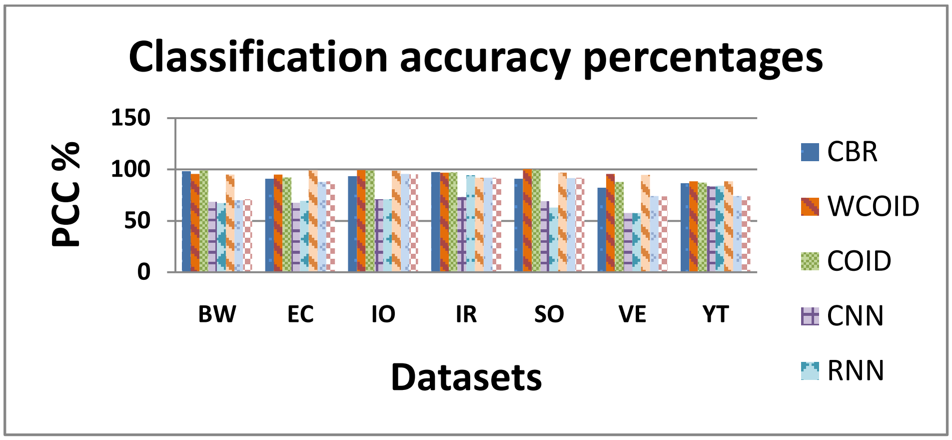

Fig. 3 Comparison of PCC

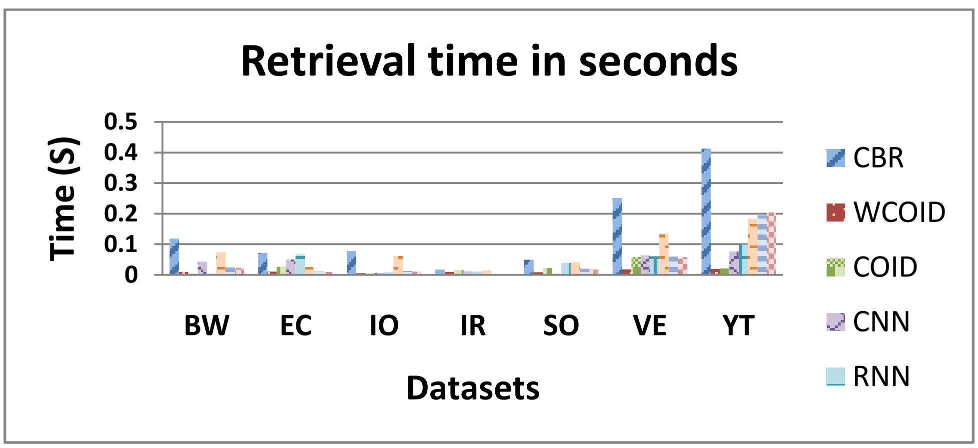

Fig. 4 Comparison of retrieval time

From Fig. 1, we observe that sizes are roughly reduced by more than half, by applying COID, WCOID, IB3, RNN methods on the different datasets, comparing to initial sizes of CBR which contains all instances. Moreover, for "Vehicle" dataset, RNN keeps about 43.63% of the data instances, and that is a huge difference comparing to CBR with 100%.

We observe, also, that the reduction rate obtained using IB3 method is better than the one provided by ENN and CNN policies in the datasets "Iris" and "Ecoli". In addition, WCOID and COID have the greatest data reduction in the most datasets comparing to the other techniques, particularly for "Yeast" dataset, they retain over 10% of the data instances, whereas the other techniques keep more than 50% of the original data.

From Fig.2, the prediction accuracy makes the some observations, where the accuracies provided by COID, WCOID, ENN and IB3 methods show slightly better accuracy values. Sometimes, they are even better than that of CBR which retains all instances, especially for "Sonar" dataset where WCOID, COID and ENN reaches more than 97% PCC, while just 90% for CBR.

As shown in Fig.4, results presented by the reduction techniques studied are better to those given by CBR with the advantages that these methods reduce retrieval time, since case bases have been shorted. For example, for the dataset "Breast-W", since IB2, IB3 and CNN keep approximately 40% of instances, the retrieval time is about 10 times better than the CBR.

5. CONCLUSION

In this paper, we have considered case base maintenance as one of the most important issues in current CBR research. We have proposed some approaches of maintenance case bases: methods offer a reducing size, consequently reducing the case retrieval time. Each approach provides a satisfying reduction rate, but sometimes it suffers from some limitations like the expensive to run for large CB and the decrease of competence especially when it exists some noisy cases, since the competence depends on the type of the stored cases.

To finish our CBM study, we ran well-known reduction schemes on the UCI data sets. All the experiments illustrated in Fig 1, Fig 2 and Fig 3, show that is no method works well in all datasets. Actually, the choice of the 'best' algorithm depends on the domain. No one policy can be called the best, rather the knowledge engineer of a CBR system must select the most suitable technique using his knowledge of the domain and the performance necessities of the system, in terms of effectiveness and storage constraints.

We conclude that the CBM policies could be improved in future work by exploring the effect of missing values and uncertain data. In fact, many future researches, in this area, focus on the study of the application of soft computing techniques, like genetic algorithms, fuzzy logic and neural networks for the maintenance of the CBR systems.

6. REFERENCES

[1] A. Aamodt and E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, Artificial Intelligence Communications 7, 7, 1, 39-52 (1994)

[2] D. B. Leake, Case-Based Reasoning: Experiences, Lessons and Future Directions, MIT Press, Cambridge, MA, USA (1996)

[3] Z. Sun and G. Finnie, A Unified Logical Model for CBR-based E-commerce Systems, International Journal of Intelligent Systems, 1-28 (2005)

[4] D. B. Leake and D. C. Wilson, Maintaining Case-Based Reasoners: Dimensions And Directions, Computational Intelligence, 17, 196-213 (2001).

[5] T. Reinartz , I. Iglezakis and T. Roth-Berghofer , On quality measure for case base maintenance. Advances in Case-Based Reasoning, 5th European Workshop, EWCBR 2000, 247-259, (2000).

[6] J. Zhu, Similarity Metrics and Case Base Maintenance, the School of Computing Science, University of British Columbia, Thesis (1998).

[7] B. Smyth and M. T. Keane, Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems, Proceeding of the 14th International Joint Conference on Artificial Intelligent, 377-382 (1995).

[8] 8. S. Minton, Quantitative Results Concerning the Utility of Explanation-based Learning, Artif. Intell., 363-391 (1990).

[9] I. Iglezakis and C. E. Anderson, Towards the use of case properties for maintaining case-based reasoning systems, Proceedings of the Pacific Rim Knowledge Acquisition Workshop (PKAW), 135-146 (2000).

[10] P. Rong, Y. Qiang and P. S. Jialin, Mining competent case bases for case-based reasoning, Artif. Intell., 171, 1039-1068 (2007).

[11] S. ck Shiu, Y. Li and X. Z.Wang, Using Fuzzy Integral to Model Case-Base Competence, Workshop on Soft Computing in Case-Based Reasoning, 59-64 (2001).

[12] B. Smyth and E. McKenna, Modeling the Competence of Case-Bases, EWCBR, 208-220 (1998).

[13] R. Pan, Q. Yang and S. J. Pan, Mining competent case bases for case-based reasoning, Artif. Intell., 171, 1039-1068 (2007).

[14] S. C. K. Shiu, D. S. Yeung, C. H. Sun and X. Wang, Maintaining Case-Based Reasoning Systems Using Fuzzy Decision Trees, EWCBR, 285-296 (2000).

[15] G. Cao, Simon C. K. Shiu and Xizhao Wang, A Fuzzy-Rough Approach for Case Base Maintenance, ICCBR, 118-130 (2001).

[16] S. C. K. Shiu, D. S. Yeung, C. H. Sun and X. Wang, Transferring Case Knowledge to Adaptation Knowledge: An Approach for Case-Base Maintenance, Computational Intelligence, 17, 295-314 (2001).

[17] Q. Yang and J. Wu, Keep It Simple: A Case-Base Maintenance Policy Based on Clustering and Information Theory, Canadian Conference on AI, 102-114 (2000).

[18] S. Markovitch and P. D. Scott, The Role of Forgetting in Learning, In Proceedings of the Fifth International Conference on Machine Learning, 459-465 (1988).

[19] S. Minton, Qualitative Results Concerning the Utility of Explanation-Based Learning, Artificial Intelligence, 42, 363-391 (1990).

[20] M. K. Haouchine, B. Chebel-Morello and N. Zerhouni, Competence-Preserving Case-Deletion Strategy for Case-Base Maintenance, Similarity and Knowledge Discovery in Case-Based Reasoning Workshop. 9th European Conference on Case-Based Reasoning ECCBR'08, 171-184 (2008).

[21] C. H. Chou, B. H Kuo and F. Chang, The Generalized Condensed Nearest Neighbor Rule as A Data Reduction Method, Pattern Recognition, International Conference on, IEEE Computer Society, 2, 556-559 (2006).

[22] G. Ritter and H. Woodruff and S. Lowry and T.Isenhour, An algorithm for a selective nearest neighbor decision rule, IEEE Transactions on Information Theory, 21, 665-669 (1975).

[23] W. Gates, The Reduced Nearest Neighbor Rule, IEEE Transactions on Information Theory, 18, 3, 431-433 (1972).

[24] A. Smiti and Z. Elouedi, COID : Maintaining case method based on Clustering, Outliers and Internal Detection, book chapter in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2010, SNPD'10, Springer Berlin / Heidelberg, 295, 39-52 (2010).

[25] A. Smiti and Z. Elouedi, WCOID: Maintaining Case-Based Reasoning systems using Weighting, Clustering, Outliers and Internal cases Detection, International Conference on Intelligent Systems Design and Applications (ISDA), IEEE Computer Society (2011).

[26] D. W. Aha, and D.Kibler, and M. K. Albert, Instance-based learning algorithms, Machine Learning, Springer Netherlands, 6, 37-66 (1991).

[27] M. Salamo and E. Golobardes, Hybrid Deletion Policies for Case Base Maintenance, Proceedings of FLAIRS-2003, AAAI Press, 150-154 (2003).

[28] M. Salamo and E. Golobardes, Dynamic Case Base Maintenance for a Case-Based Reasoning System, IBERAMIA, 93-103 (2004).

[29] A. Asuncion and D.J. Newman, UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, http://www.ics.uci.edu/mlearn (2007).

[30] B. Smyth, Case-Base Maintenance, Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE, London, UK, 507-516 (1998).

[31] D. L. Wilson, Asymptotic Properties of Nearest Neighbor Rules Using Edited Data, IEEE Transactions on Systems Man and Cybernetics, 2, 408-421 (1972).

[32] D. Guan, W. Yuan, Lee, Y. K. Lee, S. Lee, Nearest neighbor editing aided by unlabeled data, Inf. Sci., 179, 13, 2273-2282 (2009).

[33] Q. Yang and J. Zhu, A case addition policy for case-base maintenance. Computational Intelligence Journal, A Special Issue on Case-Base Maintenance. Blackwell Publishers, Boston MA UK, 17(2): 250-262 (2001).