IMPROVING CASE RETRIEVAL PERFORMANCE THROUGH THE USE OF CLUSTERING TECHNIQUES

Paulo Tomé
Department of Computing, Polytechnic of Viseu, Viseu, Portugal

Ernesto Costa
Department of Computing, University of Coimbra, Coimbra, Portugal

Luís Amaral
Department of Information Systems, University of Minho, Guimarães, Portugal

Keywords: Case-Based Reasoning, Clustering Techniques, Case-Base Maintenance.

Abstract: The performance of Case-Based Reasoning (CBR) systems is highly depend on the performance of the retrieval phase. Usually, if the case memory has a large number of cases the system turn to be very slow. Several mechanisms have been proposed in order to prevent a full search of the case memory during the retrieval phase. In this work we propose a clustering technique applied to the memory of cases. But this strategy is applied to an intermediate level of information that defines paths to the cases. Algorithms to the retrieval and retention phase are also presented.

1 INTRODUCTION

CBR is a method (Watson, 1999) that allows to solve problems based in previous resolved ones (Kolodner, 1993; Mantaras et al., 2006). Every CBR system comprises a retrieval phase, a re-using phase and a retention phase (Aamodt and Plaza, 1994). These phases have different impacts on the performance of the CBR. According with Smyth and McKenna (Smyth and McKenna, 1999), the performance of a CBR system can be measured according to three criteria:

- Effiency - the average problem solving time;
- Competence - the range of target problems that can be successfully solved;
- Quality - the average quality of a proposed solution.

As mentioned by Smyth and McKenna (Smyth and McKenna, 1999), the retrieval process deserved always highest interest of the CBR research community because it has an high influence on the CBR system performance.

The retrieval process involves the combination of two procedures: similarity evaluation and searching in the memory case. The first procedure judges the similarity of the current problem with the ones previously resolved. If the case memory have a large number of cases, the number of similarity evaluations is large representing a computational burden.

Some authors tried to improve the retrieval phase performance using case memory structures. The aim of these structures is to organize the case memory in way that enable a fast case access avoiding the similarity evaluation of all cases in memory. There are several examples of case memory structures proposals, for example, Kolodner (Kolodner, 1993) identifies four ways of organizing the case memory. Following those proposals Schaaf (Schaaf, 1996) organizes cases in a network by considering cases as a polyhedron with a face for each aspect. And Wolverton (Wolverton, 1994) organizes the case memory in a semantic network. Within that semantic network, small subgraphs of nodes and links which represent aggregate concepts are explicitly grouped together as conceptual graphs. More recently Yang and Wu (Yang and Wu, 2000) split the case into a set of clus-

Tomé P., Costa E. and Amaral L. (2008).
IMPROVING CASE RETRIEVAL PERFORMANCE THROUGH THE USE OF CLUSTERING TECHNIQUES.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 450-454
DOI: 10.5220/0001687704500454

ters distributed by different sites/machines.

Furthermore, the structure of the case memory are directly related with the case base maintenance (CBM) methods (Wilson and Leake, 2001). The objective of the CBM is maintaining consistency, preserving competence and controlling case-base growth.

The previous presented approaches does not address the problem of missing case features. This problem is currently addressed in CBR research field (GU and Aamodt, 2005; Gu and Aamodt, 2006).

We propose the use of clustering techniques to improve the performance of the CBR during the retrieval phase. However, we use different approach of Yang and Wu (Yang and Wu, 2000). We do not split the case memory into distinct locations instead we use clusters of links to cases. Besides that, our proposal also deals with cases with missing features. In Section 2 we review some clustering concepts essential to the understanding of our proposal shown in section 3. In section 3 we present also the results of the application of our proposal to a CBR system with 915 cases.

2 THE CLUSTERING TECHNIQUE

Clustering techniques organizes data into groups that are meaningful, useful or both (Tan et al., 2006). One group of data is called a cluster, while the entire collection of clusters is commonly referred to as a clustering.

Two types of clustering can be considered: partional and hierarchical. A clustering is hierarchical if we permit clusters to have subclusters. In partional clustering the data is divided into non-overlapping clusters. Tan et al. (Tan et al., 2006) identified five types of clusters: well-separated, prototype-based, graph-based, density based and shared-based. In our work we will consider prototype-based type. A set of cases is grouped into a cluster with one representative element. Then, in the retrieval phase the number of similarity evaluations is reduced considerably.

There are several techniques to split the data, but k-means and k-medoid are two of the most prominent techniques associated to prototype-based techniques (Tan et al., 2006). K-means defines a prototype in terms of a centroid, while k-medoid defines defines a prototype in terms of a medoid. The medoid is one element of the cluster while the centroid is the mean of the cluster. In our work we use the k-medoid technique. So each cluster is represented by the most representative case among all cases in the group. There are also different proposals to measure similarity between data: the Euclidean and cosine distance are the most used similarity measures. The similarity measure is used whenever a new case has to be added to the case memory. Naturally the updated cluster need to update its prototype.

3 THE APPLICATION OF CLUSTERING TECHNIQUES TO THE CBR RETRIEVAL PHASE

Our proposal, shown in figure 1, has two levels of information. The first level is formed by a set of links to the case memory and the second level is the case memory database. The case links are paths to cases memory. And the clustering technique is applied to case links information. The first level of information requires a low amount of storage space however decreases the waiting time of the retrieval process. We do not considered the division of the database case memory because it is useful to access a case from different ways. The figure 1 illustrates the storage scheme and we can see that groups of clusters are the interface between CBR process and the database of cases.

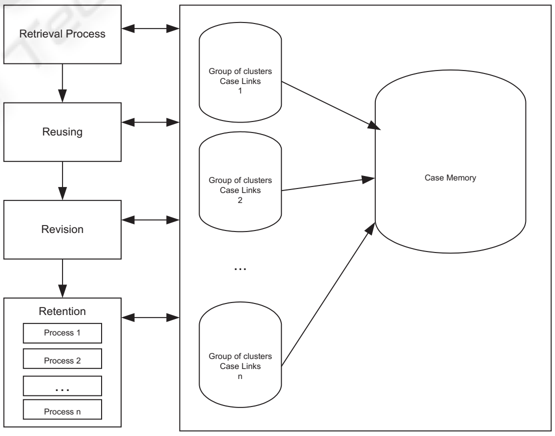
Retrieval Process
Reusing
Revision
Retention
Process 1
Process 2
...
Process n
Group of clusters Case Links 1
Group of clusters Case Links 2
...
Group of clusters Case Links n
Case Memory

Figure 1: CBR system structure.

Each group has clusters of links to cases. And each cluster, as shown in table 1, has a reference to the medoid of the cluster and links to a set of cases that constitute the cluster.

Each Group of clusters is identified by a binary array codification. The binary codification scheme follows the proposal of Kolodner, table 2, who defines that a case is formed by a Problem and by a Solution. And the Problem consists in Objective and a set

Table 1: Cluster definition.

$Clus = < LMed, SCl > \; Med = link\_to\_case \; SCl = \{link\_to\_case\}$
Where:
Clus - Cluster
LMed - Link to medoid
SCl - Set of Cluster link;
C - Characteristic

Table 2: Case Structure.

$Case = < P, S >$
$P = < O, Cs >$
$Cs = \{C\}$
Where:
P - Problem
O - Objective
Cs - Set of characteristics;
C - Characteristic

Table 3: Case Example.

$Cas_1 = < P, S >$
$P = < O1, Cs >$
$Cs = < C_1, C_2, C_3 >$

of Characteristics. Table 3 shows a case with three Characteristics. So each position of the binary array is associated to a particular feature of a case, where 1 (0) indicates the availability (non-availibility) of the feature.

The first positions on the right side of the array are used to represent objectives. The remaining positions are used to represent characteristics. For example, using sixteen bits with the division illustrated in table 4, the case Cas_1, shown in table 2, addresses the group with the following binary array 0000000001110001.

Table 4: Addressing Group Cluster.

C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	O_4	O_3	O_2	O_1

However to deal with missing characteristics the cases belong to more than one group. All combination of the available characteristics and objective define different groups. In table 5 it is presented the combinations of characteristics and objective for the case Cas_1. The case Cas_1 is associated to seven groups (figure 2), in each group clustering might be achieved with a distinct number of clusters.

Table 5: Combinations example to Cas_1 case.

C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	O_4	O_3	O_2	O_1	**Id**
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	Co_1
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	Co_2
0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	Co_3
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	Co_4
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	Co_5
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	Co_6
0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	Co_7

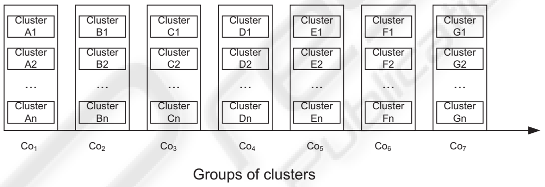

Figure 2: Example of a database of group of case links.

The Retrieval process was modified to enable the adoption of the principles previously described. Table 6 shows the steps of the algorithm when a new problem is presented: 1)- identification of the clustering group 2)-identification of the cluster within the group using a similarity measure; 3)- finally the case is compared with all cases in the cluster. In the second step the **Default Difference** measure strategy (Bogaerts and Leake, 2004) is applied only to the medoid of the clusters. In the third step, different similarity evaluations are used.

The retention process algorithm, shown in table 7, was also redefined. This process was parallelized, e.g. the retention in each Group of cluster is implemented by different program processes. The process begins with the determination of all possible combinations between the available characteristics and the objective of the case. Then for each combination, the case is inserted in respective Group of clusters. This insertion process is parallelized. Each insertion in a Group determines: 1)-evaluation of the similarity with the cluster medoids 2)- identification of the cluster to insert the case, 3) actualization of the medoid of the cluster where the case was assigned. The similarity measure strategy used in step two is also **Default Difference**. In a group a new cluster is created whenever a binary similarity evaluation results in a zero.

The medoid is computed using the use of the expression

$$pos\_med = \sum_{i=1}^{nfeatures} pos(value\_of\_feature(i)) * weight(feature(i)) \quad (1)$$

where $pos(value\_of\_feature(i))$ is the posi-

Table 6: Retrieval Algorithm.

```
/* ————————————————————————————————-
Cas is the case for which it is search a solution
Prop_Cas is the proposed case
————————————————————————————————-*/
procedure retrieval(Cas in Case, Prop_cas out Case)
Clusgroup ClusterGroup;
Clus Cluster;
Sim Similarity;
Sim_a Similarity;
begin
    Clusgroup ← Determine_cluster_group(Cas.Obj,Cas.Cars);
    Clus ← 0;
    Sim ← 0;
    For each Cluster in Clusgroup do
    begin
        Sim_a ← Similarity(Cas, Cluster(i).medoid);
        if Sim_a > Sim then
        begin
            Sim ← Sim_a;
            Clus ← Cluster(i);
        end;
    end;
    Sim ← 0;
    For each Case in Clus do
    begin
        Sim_a ← Similarity(Cas, case(i));
        if Sim_a > Sim then
        begin
            Sim ← Sim_a;
            Prop_cas ← Case(i);
        end;
    end;
end;
```

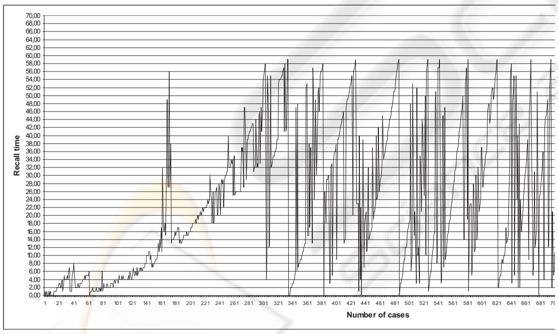

Figure 3: Waiting time before clustering technics application.

tion of feature in a ordered set of values and $weight(feature(i))$ is the weight of the $feature(i)$.

We compare the described strategy with a flat memory case for a CBR system with 915 cases. As we can see in figure 3, the system had bad performance after the insertion of three hundred cases in the case memory.

After the application of the clustering approach the waiting time for each of the 915 cases was less than 1 second. Although, the retention process takes more time than the flat memory case. The figure 4 shows the retention waiting time for all of the 915 cases. Although, the total waiting time is smaller after the application of the clustering techniques as shown in figure 4.

Table 7: Retention Algorithm.

```
/* ————————————————————————-
Cas is the case that will be retained
————————————————————————-*/
procedure retention(Cas in Case)
Combs Combinations;
Combination TCombination;
Clusgroup ClusterGroup;
Clus Cluster;
Sim Similarity;
Sim_a Similarity;
begin
    Combs ← Generate_all_combinations(Cas.Obj,Cas.Cars);
    For each Combination in Combs do
    begin
        Clusgroup ← Determine_clus_group(Combination(i));
        Sim ← 0;
        For each Cluster in Clusgroup do
        begin
            Sim_a ← Similarity(Cas, Cluster(j).medoid);
            if Sim_a > Sim then
            begin
                Sim ← Sim_a;
                Clus ← Clus(i);
            end;
        insert_case_cluster(Clus,Cas);
        recalculate_medoid(Cluster(i));
    end;
end;
```

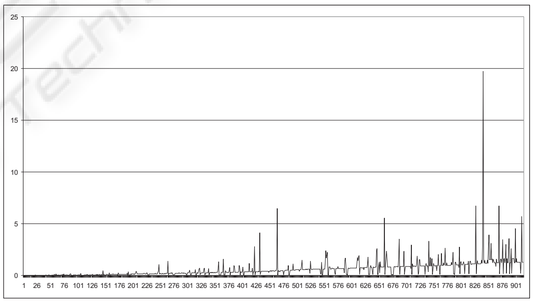

Figure 4: Waiting time after clustering technics application.

4 CONCLUSIONS

This paper presents an approach that can be used to improve the performance of CBR system Retrieval phase. The approach uses clustering techniques to structure the case memory. The use of clustering techniques define a particular case memory structure and consequently algorithms for the retrieval and retention phase are proposed.

The approach was tested in a CBR system with 915 cases. The result show that the overall system performance is improved. It is important to notice that the retrieval waiting time was considerably reduced and the total waiting time (time of retrieval and retention) is also substantially smaller than with a flat case memory organization.

REFERENCES

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations and systems approaches. *AI-Communications*, 7(1):39–52.

Bogaerts, S. and Leake, D. (2004). Facilitating cbr for incompletely-described cases: Distance metrics for partial problem descriptions. In *ECCBR 2004*, pages 62–74.

GU, M. and Aamodt, A. (2005). A knowledge-intensive method for conversational cbr. In *6th International Conference on Case-Based Reasoning*, pages 296–311. Springer-Verlag.

Gu, M. and Aamodt, A. (2006). Dialog learning in conversational cbr. In *19th International FLAIRS Conference*, pages 358–363, Florida, EUA. AAAI Press.

Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers.

Mantaras, R. L., Mcsherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A., and Watson, I. (2006). Retrieval, reuse, revison and retention in case-based reasoning. *The Knowledge Engineering Review*, 20(3):215–240.

Schaaf, J. W. (1996). Fish and shrink. a next step towards efficient case retrieval in large scale case bases. In Smith, I. and Faltings, B., editors, *Advances in Case-Based Reasoning*, pages 362–376. Springer-Verlag.

Smyth, B. and McKenna, E. (1999). Footprint-based retrieval. In *Third International Conference on Case-Based Reasoning*, pages 343–357, Munich, Germany. Springer Verlag.

Tan, P. N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Pearson Education.

Watson, I. (1999). *CBR* is a methodology not a technology. *Knowledge Based Systems Journal*, 12(5-6).

Wilson, D. C. and Leake, D. (2001). Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence*, 17(2):196–213.

Wolverton, M. (1994). *Retrieving Semantically Distant Analogies*. Ph.d thesis, Stanford University.

Yang, Q. and Wu, J. (2000). Keep it simple: A case-base maintenance policy based on clustering and information theory. In *Canadian AI 2000*, pages 102–114. Springer-Verlag.