Footprint-Based Retrieval

Barry Smyth & Elizabeth McKenna

Department of Computer Science
University College Dublin
Belfield, Dublin 4, IRELAND

Barry.Smyth@ucd.ie
Elizabeth.McKenna@ucd.ie

Abstract. The success of a case-based reasoning system depends critically on the performance of the retrieval algorithm used and, specifically, on its efficiency, competence, and quality characteristics. In this paper we describe a novel retrieval technique that is guided by a model of case competence and that, as a result, benefits from superior efficiency, competence and quality features.

1 Introduction

Case-based reasoning (CBR) systems solve new problems by retrieving and adapting the solutions to previously solved problems that have been stored in a case-base. The performance of a case-based reasoner can be measured according to three criteria: 1) Efficiency – the average problem solving time; 2) Competence – the range of target problems that can be successfully solved; 3) Quality – the average quality of a proposed solution. Recently, researchers have begun to investigate methods for explicitly modelling these criteria in real systems. Their aim is twofold. On the one hand, it is important to develop predictive performance models to facilitate evaluation and comparative studies (see for eg., [7,12]). However, in addition, the models can also be used to drive the development of new techniques and algorithms within the CBR problem solving cycle. For example, a variety of different efficiency and competence models have been used recently to guide the growth of case-bases during learning and case-base maintenance [8, 11, 12].

In this paper we return to the classic problem of case retrieval and propose a novel retrieval method, *footprint-based retrieval*, which is guided by a model of case competence. The next section surveys related work on retrieval. Section 3 describes the model of case competence that forms the basis of our new retrieval method, which is discussed in Section 4. Finally, before concluding, Section 5 describes a comprehensive set of experiments to evaluate the performance of the new algorithm.

2 Related Work

The retrieval process has always received the lion's share of interest from the CBR community. All CBR systems have at least a retrieval component, and the success of a given system depends critically on the efficient retrieval of the right case at the right time. Every retrieval method is the combination of two procedures; a similarity assessment procedure to determine the similarity between a given case and target problem, and a procedure for searching the case memory in order to locate the most similar case. Research on the former topic has focussed on developing efficient and accurate similarity assessment procedures capable of evaluating not just the similarity of a case but also other criteria such as its adaptability (see for eg., [5, 10]).

In this paper we focus on the search procedure. Research in this area has been concerned with reducing the search needed to locate the best case without degrading competence or quality [2, 3, 4, 6, 9, 1, 15]. The simplest approach to retrieval is an exhaustive search of the case-base, but this is rarely viable for large case-bases. Thus the basic research goal is to develop a strategy that avoids the need to examine every case. Most approaches achieve this by processing the raw case data in order to produce an optimised memory structure that facilitates a directed search procedure.

One approach is to build a decision-tree over the case data (see for eg., [14]). Each node and branch of the tree represents a particular attribute-value combination, and cases with a given set of attribute-values are stored at the leaf nodes. Case retrieval is implemented as a directed search through the decision tree. These approaches are efficient but may not be appropriate for case-bases with incomplete case descriptions, or where the relative importance of individual case features can change.

Spreading activation methods (eg., [2]) represent case memory as an interconnected network of nodes capturing case attribute-value combinations. Activation spreads from target attribute-value nodes across the network to cause the activation of case nodes representing similar cases to the target. The approaches are efficient and flexible enough to handle incomplete case descriptions, however there can be a significant knowledge-engineering cost associated with constructing the activation network. Furthermore the spreading-activation algorithm requires specific knowledge to guide the spread of activation throughout the network. Related network-based retrieval methods are proposed by Lenz [6] and Wolverton & Hayes-Roth [15].

Perhaps the simplest approach to controlling retrieval cost is to employ an exhaustive search on a *reduced* case-base. This strategy is common in the pattern-recognition community to improve the performance of nearest-neighbour techniques by editing training data to remove unnecessary examples (eg., [1, 4]). Many editing strategies have been successfully developed, often maintaining retrieval competence with an edited case-base that is significantly smaller than the full case-base.

With all of the above methods there is an inherent risk in not examining every case during retrieval. The optimal case may be missed, which at best can mean sub-optimal problem solving, but at worst can result in a problem solving failure. In this paper we propose a novel retrieval method based on the idea of searching an edited subset of the entire case-base. The key innovation is that retrieval is based on two searches of two separate edited subsets. The first search identifies a reference case that is similar to the target problem. This case acts as an index into the complete case-base, and the second search locates the best available case in the region of the reference case. In

this sense our method is related to the "Fish-and-Shrink" strategy [9] where cases are linked according to specific *aspect* similarities. However, our method is unique in its use of an explicit competence model to guide the selection of a non-arbitrary case.

3 A Model of Case Competence

Competence is all about the number and type of target problems that a given system can solve. This will depend on a number of factors including statistical properties of the case-base and problem-space, and the proficiency of the retrieval, adaptation and solution evaluation components of the CBR system in question. The competence model described in this section is based on a similar model first introduced by Smyth & McKenna [12]. The present model introduces a number of important modifications to increase the effectiveness and general applicability of the model.

In this paper, the crucial feature of our competence model is that it constructs a subset of the case-base called the *footprint set*, which provides the same coverage as the case-base as a whole. This footprint set, and its relationship to the complete case-base, is central to our new retrieval algorithm.

3.1 Coverage & Reachability

Consider a set of cases, C, and a space of target problems, T. A case, $c \in C$, can be used to solve a target, $t \in T$, if and only if two conditions hold. First, the case must be retrieved for the target, and second it must be possible to adapt its solution so that it solves the target problem. Competence is therefore reduced if adaptable cases fail to be retrieved or if non-adaptable cases are retrieved. We can model these relationships according to the definitions shown in Def. 1 – 3.

Def 1: RetrievalSpace(t)={$c \in C$: c is retrieved for t}

Def 2: AdaptationSpace(t)={$c \in C$: c can be adapted for t}

Def 3: Solves(c,t) iff $c \in$ [RetrievalSpace(t) $\cap$ AdaptationSpace(t)]

Two important competence properties are the *coverage set* and the *reachability set*. The coverage set of a *case* is the set of all *target problems* that this case can solve. Conversely, the reachability set of a *target problem* is the set of all *cases* that can be used to solve it. By using the case-base as a representative of the target problem space it is possible to estimate these sets as shown in Def. 4 & 5 (see also [11]).

Def 4: CoverageSet(c)={$c' \in C$:Solves(c,c')}

Def 5: ReachabilitySet(c)={$c' \in C$:Solves(c',c)}

Furthermore, we use the term *related set* to refer to the set produced from the union of a case's coverage and reachability sets.

Def 6: RelatedSet(c)= CoverageSet(c) $\cup$ ReachabilitySet(c)

3.2 Relative Coverage

The size of the coverage set of a case is only a measure of its local competence. For instance, case coverage sets can overlap to limit the competence contributions of individual cases, or they may be isolated and exaggerate individual contributions [11, 12]. It is actually possible to have a case with a large coverage set that makes little or no contribution to global competence simply because its contribution is subsumed by the local competences of other cases. At the other extreme, there may be cases with relatively small contributions to make, but these contributions may nonetheless be crucial if there are no competing cases.

Def 7:
$$RelativeCoverage(c) = \sum_{c' \in CoverageSet(c)} \frac{1}{|ReachabilitySet(c')|}$$

For a true picture of competence, a measure of the coverage of a case, relative to other nearby cases, is needed. For this reason we define a measure called *relative coverage* (RC), which estimates the unique competence contribution of an individual case, c, as a function of the size of the case's coverage set (see Definition 7). Essentially, relative coverage weights the contribution of each covered case by the degree to which these cases are themselves covered. It is based on the idea that if a case c' is covered by n other cases then each of the n cases will receive a contribution of 1/n from c' to their relative coverage measures.

The importance of relative coverage is that it provides a mechanism for ordering cases according to their individual, global, competence contributions. In section 3.4 we will see how this allows us to represent the competence of a complete case-base in terms of a subset of cases called the *footprint set*.

3.3 Competence Groups

As a case-base grows clusters of cases tend to form distinct regions of competence. We can model these regions as *competence groups* (see Figure 1). A competence group is a collection of related cases, which together make a collectively independent contribution to overall case-base competence.

Def 8: For c1, c2 $\in$ C, SharedCoverage(c1, c2)
iff [RelatedSet(c1) $\cap$ RelatedSet(c2)]

The key idea underlying the definition of a competence group is that of *shared coverage* (see Definition 8). Two cases exhibit shared coverage if their related sets

overlap. This is seen as an indication that the cases in question make a shared competence contribution, and as such belong to a given competence group.

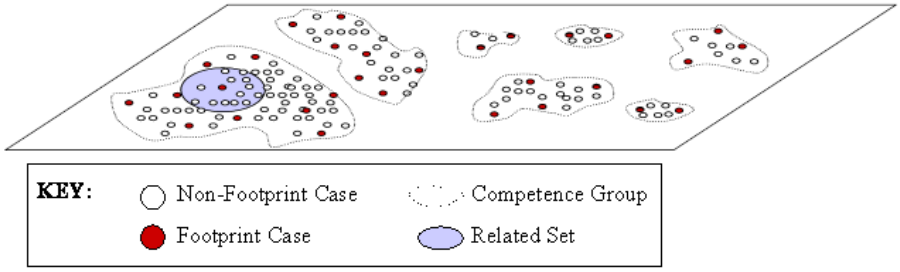

Figure 1. The formation of competence groups and the footprint set in a case-base.

Shared coverage provides a way of linking related cases together. Formally, a competence group is a maximal collection of cases exhibiting shared coverage (see Definition 9). Thus, each case in a competence group must share coverage with some other case in the group (this is the first half of the equation). In addition, the group must be maximal in the sense that there are no other cases in the case-base that share coverage with any group member (this is the second half of the equation).

Def 9: For $G = \{c1,...,cn\} \subseteq C$,

$$\text{CompetenceGroup}(G) \text{ iff } \forall ci \in G, \exists cj \in G-\{ci\}: \text{SharedCoverage}(ci,cj) \wedge \forall cj \in C-G, \neg\exists cl \in G: \text{SharedCoverage}(cj,cl)$$

3.4 The Footprint Set

The footprint set of a case-base is a subset of the case-base that covers all of the cases in the case-base[1] – it is related to the concept of a minimal consistent subset in classification research (see for eg., [3]). By definition, each competence group makes a unique contribution to the competence of the case-base. Therefore, each competence group must be represented in the footprint set. However, not all of the cases in a given competence group are included in this subset (see Figure 1). For example auxiliary cases make no competence contributions [11] – an auxiliary case is a case whose coverage set is completely subsumed by the coverage set of another case.

The construction of the footprint set is carried out at the group level. For each group we compute its *group footprint*, that is, the subset of group cases that collectively cover the entire group. The algorithm in Figure 2 is used for identifying the group footprint cases; it is a simple modification of the CNN/IB2 algorithms (see for eg., [1, 4]). The first step is to sort the cases in descending order of the relative coverage values; this means that cases with large competence contributions are added before cases with smaller contributions, and thus helps to keep the footprint size to a

[1] The footprint concept used here should not to be confused with Veloso's concept of footprint similarity (see [13]). Our present notion is based on the footprint concept introduced in [11].

minimum. The group footprint is then constructed by considering each case in turn, and adding it to the group footprint only if the current footprint does not already cover it. Note that a number of passes over the group cases may be necessary if new additions to the footprint can prevent previously covered cases from being solved, as does occur in many classification problem domains. Finally, the overall footprint set is the union of all of the cases in the individual group footprints.

```
Group ← Original group cases sorted according to their RC value.
FP    ← {}
CHANGES ← true

While CHANGES Do
    CHANGES ← false
    For each case C∈Group Do
           If FP cannot solve C Then
                  CHANGES ← true
                  Add C to FP
                  Remove C from Group
           EndIf
    EndFor
EndWhile
```

Figure 2. The Group Footprint Algorithm.

Note that the relationship between the footprint set and the complete case-base is preserved. By using the related set of a footprint case we can associate it with a set of similar cases from the complete case-base. This critical link between the footprint set and the case-base is a key element in our new retrieval algorithm.

```
Target ← Current target problem
CB ← Original case-base
FP ← Footprint Set

Stage 1
    ReferenceCase ← a case in FP that is closest to the Target.

Stage 2
    RelatedSet ← RelatedSet(ReferenceCase)
    BaseCase ← a case in RelatedSet that is closest to Target
```

Figure 3. The Competence-Guided Retrieval Procedure.

4 Footprint-Based Retrieval

The objective in this paper is to present footprint-based retrieval, a simple but novel approach to case retrieval that is comprised of two separate stages. Stage one is

designed to focus the search in the local region of the case-base that contains the target problem. Stage two then locates the nearest case to the target in this region. The key innovation of the approach stems from its direct use of a model of case competence to guide the retrieval process. The basic algorithm is shown in Figure 3 and the retrieval process is illustrated in Figure 4.

4.1 Stage 1: Retrieving from the Footprint Set

The footprint set is typically much smaller than the full case-base, and this means that the process of searching the footprint set is much less expensive than searching the entire case-base. During the first stage of retrieval the target problem is compared to each case in the footprint, in order to locate the case that best matches the target. This case is termed the *reference case*.

The reference case provides important clues concerning the ultimate solvability of the target problem, even at this early stage of problem solving. For example, the reference case may be able to solve the target problem as it stands, and further retrieval work may not be needed, especially if an optimal case is not required. However, the real importance of the reference case is that it provides an index into the full case-base for the next stage of retrieval.

4.2 Stage 2: Retrieving from the Related Set

The reference case may, or may not, be able to solve the current target problem, it may even be the closest case to the target in the entire case-base – however, this cannot be guaranteed. The objective of the next stage of retrieval is to compare the target problem to other (non-footprint) cases in the case-base in order to locate the most similar case in the entire case-base. The footprint case selected in stage one acts as a reference point for this next stage of retrieval. Only the cases nearby to the reference case need to be compared to the target.

During the construction of our competence model the related set of each case, c, is computed. The cases in this set are precisely those cases that are nearby to c. Therefore, during the second stage of retrieval each of the cases in the reference case's related set is compared to the target problem. Again, as in stage one, this is an inexpensive procedure, compared to searching the entire case-base, since each related set contains only a very small subset of the entire case-base.

4.3 Discussion

One way to think about the proposed retrieval method is as a single search through one subset of the entire case-base. In this sense the new technique looks very similar to other methods such as CNN retrieval. However, there is one important difference. The subset used by a technique such as CNN is computed once, at training time, without reference to a specific target problem – essentially an eager learning technique. However, the new method combines cases from a similar once-off subset of the entire case-base with additional cases that have been chosen with respect to a

particular target problem – the additional cases are chosen according to a lazy learning policy.

This turns out to be an essential feature. It allows the proposed retrieval approach to adapt its search space to the characteristics of an individual target problem. This in turn greatly improves the competence and quality of the retrieved cases.

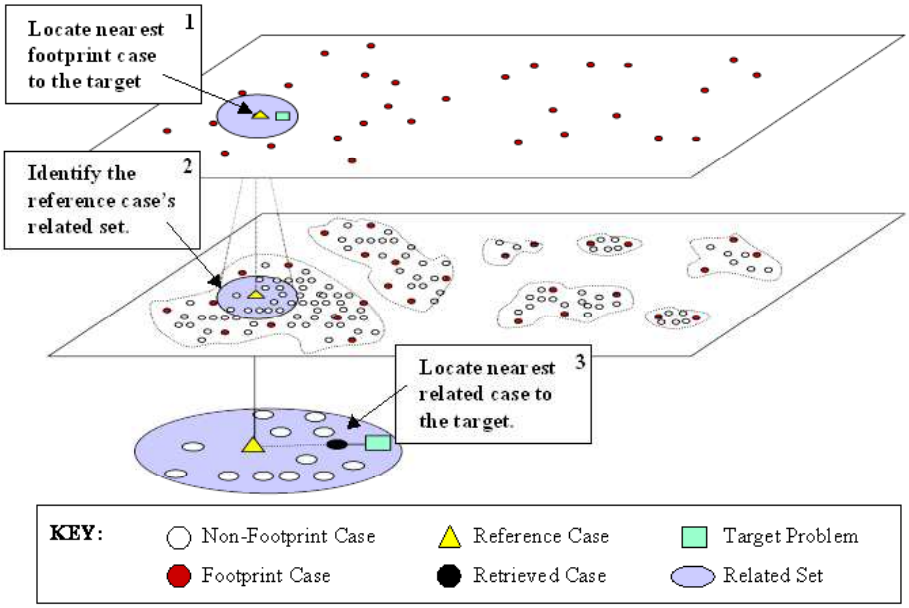

Figure 4. Footprint-based retrieval is a two-stage retrieval process. First, the footprint case that is nearest to the target problem is identified. Second, the case from its related set that is nearest to the target is identified and returned as the base case.

5 Experimental Studies

The footprint-based retrieval technique has been described, which, we claim, benefits from improved efficiency, competence, and quality characteristics. In this section we validate these claims with a comprehensive experimental study.

5.1 Experimental Setup

Altogether four different retrieval methods are evaluated; all use a standard weighted-sum similarity metric. The first (Standard) is the brute force, nearest-neighbour method where the target case is compared to every case in the case-base and the most similar case is retrieved. The second method (CNN) is a standard way to reduce retrieval time by producing an edited set of cases using the standard CNN approach.

The third method (FP) is analogous to the CNN method, except that the footprint set of cases is used as the edited set – this is equivalent to running stage one of footprint-based retrieval only. Finally, the fourth technique (FPRS) is the full footprint-based approach. Note that by comparing the FP and FPRS results in the following sections we can evaluate the contributions of each retrieval stage separately.

Two standard data-sets are used in the study. The first is a case-base of 1400 cases from the Travel domain. Each case describes a vacation package using a range of continuous and discrete features such as: type of vacation; number of people; length of stay; type of accommodation, etc. The case-base is publicly available from the AI-CBR case-base archive (see http://www.ai-cbr.org). The second data-set contains 500 cases from the Property domain, each describing the residential property conditions in a particular region of Boston. This data-set is publicly available from the UCI repository (see http://www.ics.uci.edu/~mlearn/MLRepository.html)

These data-sets are processed to produce a range of different case-base sizes and target problem sets. In the Travel domain we produce 10 case-base sizes, ranging from 100 cases to 1000 cases, with accompanying target problem sets of 400 cases. For the Property domain we produce 6 case-base sizes from 50 to 300 cases, and 200 target problems. In each domain, for each case-base size n, we produce 100 different random case-bases and target problem sets to give 1000 test case-bases for the Travel domain and 600 test case-bases for the Property domain. There is never any direct overlap between a case-base and its associated target problem set.

5.2 Efficiency

The first experiment is concerned with evaluating the efficiency of the retrieval algorithms over a range of case-base sizes. Efficiency is measured as the inverse of the number of cases examined during retrieval. This is a fair measure as all four algorithms perform a simple search through a set of cases using the same similarity operator.

Method: Each case-base of size n is tested with respect to its target problem set and the average retrieval cost for the set of targets is computed. This cost is then averaged over the 100 case-bases of size n. This produces an average retrieval cost per target problem for each case-base size.

Results: These retrieval efficiency results are shown in Figure 5(a) - (d) as plots of efficiency (inverse number of cases examined during each retrieval) versus case-base size, for the Travel domain and the Property domain, respectively. Figures 4 (a) and (b) show the mean retrieval cost for each of the four retrieval algorithms as case-base size increases, however since the CNN, FP, and FPRS curves are difficult to distinguish, Figures 5(c) and (d) show additional detail over a restricted efficiency range. As expected, the Standard retrieval method performs poorly, while the three edited retrieval methods perform much better. Note that the FP curves also show how small the footprint set is with respect to overall case-base size.

Discussion: For small and medium case-base sizes both of the footprint-based methods (FP and FPRS) out-perform the CNN approach. However, eventually the FPRS method may become marginally less efficient than the CNN and FP methods; this is seen after the 600 case mark in the Travel domain (see Figure 5(c)), but is not

evident in the Property domain.. The reason for this is the second stage of retrieval in the FPRS method. CNN and FP do not incur this extra cost. For small and medium sized case-bases the related sets of cases are small and so this extra cost does not register as significant. However, as the case-base grows, so too does the average size of a related set, and therefore, so too does the cost of this second phase of retrieval. This second stage cost remains low however, and, we argue, is comfortably offset by the significant benefits for FPRS when it comes to competence and quality, as we shall see in the following sections.

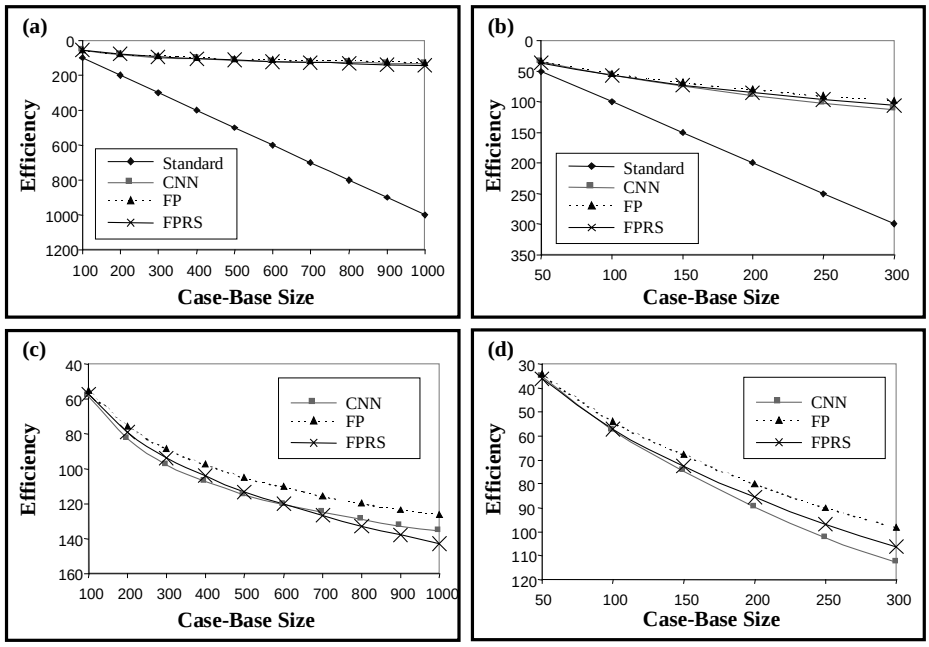

Figure 5. Efficiency vs. Case-Base Size for the Travel (a & c) and Property (b & d) domains respectively. Graphs (c & d) show additional detail for the CNN, FP, and FPRS results.

5.3 Competence

There is typically a tradeoff between the efficiency and competence of a retrieval technique. In particular, since the footprint and CNN methods do not examine every case in the case-base, it is possible that important cases are missed during retrieval thereby limiting the overall problem solving competence. In this experiment we look at how each retrieval method performs in terms of competence, where competence is defined to be the percentage of target problems that can be successfully solved by a given retrieval algorithm.

Method: Each case-base of size n is tested with respect to its associated set of target problems, and the competence of each retrieval method over these target problems is computed (that is, the percentage of target problems that can be correctly

solved using each retrieval method). This cost is averaged for each of the 100 case-bases of size n to compute a mean competence for each case-base size and retrieval method.

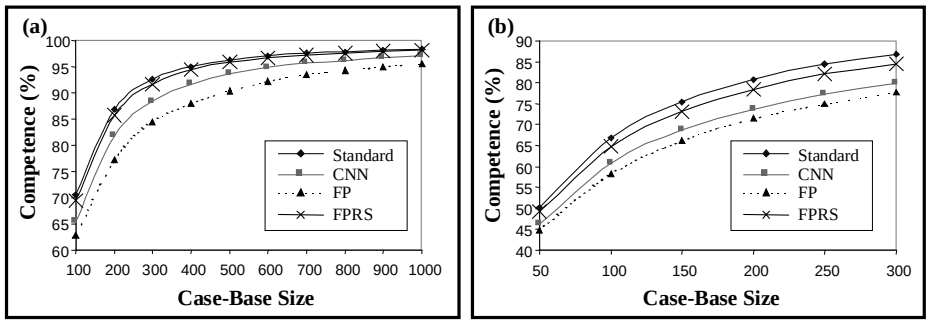

Figure 6. Competence *vs.* Case-Base Size for (a) the Travel, and (b) the Property domains.

Results: The results are displayed as graphs of competence versus case-base size for each domain in Figure 6(a) and (b). Each graph shows the results for the four different retrieval methods. The trade-off between retrieval efficiency and retrieval competence now becomes clearer. By definition, the Standard retrieval method defines the optimal competence for each case-base size in the sense that it guarantees the retrieval of the nearest case to a given target problem, which for our purposes is assumed to be the correct case. This assumption is typical in most traditional CBR systems but it may not hold in other domains such as classification problems – future work will focus on this issue further. In this experiment the important thing to note is the difference between each of the edited retrieval methods and the Standard method. It is clear that in both domains, and for every case-base size, the FPRS method outperforms the CNN and FP methods. In fact the FPRS method exhibits competence characteristics that are nearly identical to the optimal Standard method results. For example, Figure 6(b) shows that for the Property domain the competence of the Standard method, at the 300 case mark, is 86%. Compare this to competence of 84.5% for the FPRS method but only 77% and 79% for the FP and CNN methods respectively.

Discussion: The reason for the improved competence of the FPRS method is its second, target-specific retrieval stage. During this stage the FPRS method searches a small but dense set of cases from the original case-base in the region of the target problem, and thus benefits from the additional detail of the full case-base in the vicinity of the target problem. The CNN and FP methods derive no such benefit from their single stage search since their edited sets lack the detail of the original case-base in the region of the target problem.

5.4 Quality

In many problem solving settings (for example classification problems) the notion of solution quality is not meaningful – the concept of quality is implemented as the

accuracy of class prediction, and a solution class is either correct or it is not. However, in other domains and tasks quality is vitally important. There may be a wide range of correct solutions to a problem that lie on a quality continuum. Different retrieval algorithms can have similar competence characteristics but very different quality characteristics – they may facilitate the solution of the same range of problems, but the quality of their proposed solutions may differ significantly. In general, solution quality is a function of the distance between the target and the retrieved case. As this distance increases, the amount of adaptation needed also tends to increase, and as adaptation work increases, solution quality tends to degrade. This correlation between similarity distance, adaptation effort, and ultimate solution quality is typical in many CBR systems. Of course pairing similarity and quality in this way does simplify the quality issue, but we believe that it is nonetheless a valid and useful pairing, one that allows us to at least begin to understand the implications that footprint-based retrieval may have for solution quality. The question to be answered in this experiment then is exactly how close do the CNN, FP, and FPRS methods get to the optimal quality level of the Standard approach?

Method: As in the earlier experiments, each case-base of size n is tested with respect to its target problem set. This time the average distance between target and retrieved case is computed. This distance is then averaged over the 100 case-bases of size n to produce a mean distance per target problem for each case-base size. The inverse of this average distance is used as a quality measure.

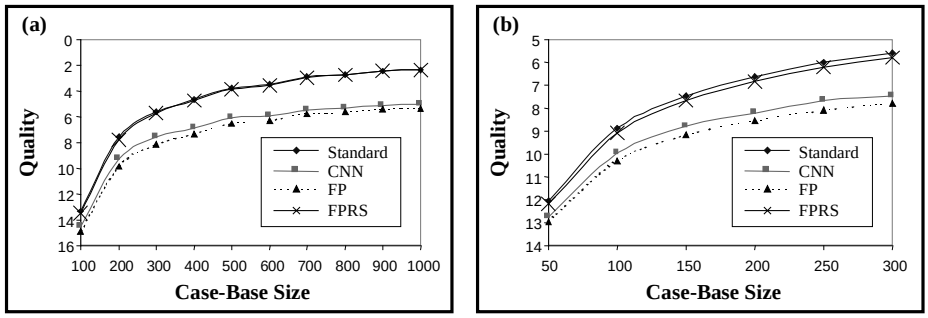

Figure 7. Quality *vs*. Case-Base size for (a) the Travel, and (b) the Property domains.

Results: The results are displayed in Figure 7(a) and (b) as graphs of quality (decreasing average distance) versus case-base size for the Travel and Property domains, respectively. The results show a clear separation of the four algorithms into two groups. The Standard and FPRS methods perform significantly better than the FP and CNN methods. In fact, the FPRS method displays a retrieval quality that is virtually identical to that of the Standard method. For example, Figure 7(a) shows that for the Travel domain the average distance of retrieved case from a target problem, at the 1000 case mark for the Standard and FPRS methods, is 2.33 and 2.37, respectively. This is compared to values of 5.1 and 5.3 for the CNN and FP methods respectively. Thus, the quality of FPRS is more than twice that of a CNN or FP.

Discussion: Clearly, from a quality viewpoint, the FPRS method benefits greatly from its second retrieval stage, a benefit that can be seen directly in the graphs as the

difference between the FPRS and FP quality curves. In fact, it is interesting to note that for both domains, the difference in quality between the FPRS method and the CNN and FP methods is itself increasing with case-base size.

5.5 Optimality

So far we have seen that the FPRS method benefits from superior efficiency, competence and quality characteristic. We have shown that the method approaches the optimal competence and quality characteristics of an exhaustive case-base search, while at the same time benefiting from the efficiency characteristics of edited case-base methods such as CNN. This last experiment is a refinement of the above competence and quality experiments. It is concerned with investigating retrieval optimality, that is, the ability of a retrieval algorithm to select the closest case to a target problem. Obviously, the Standard method will always retrieve this optimal case. The same is not true of CNN since it may not have access to these optimal cases – they may have been dropped during the editing process. The question that remains to be answered is: how often does the FPRS method retrieve the optimal case?

Method: As in the earlier experiments, each case-base of size n is tested with respect to its target problem set. This time we are interested in how often each of FPRS, FP, and CNN select the same case for a target problem as the Standard method.

Results: The results are displayed in Figure 8(a) and (b) as graphs of optimality versus case-base size for the Travel and Property domains, respectively. The results show clearly that the FPRS method is far superior to the CNN and FP methods. In fact in both domains, and for all case-base sizes, FPRS optimality is at least 90%. In contrast, the optimality of the CNN and FP methods decreases with case-base size and fall to as low as 15% for the Travel domain (at 1000 cases) and 35% for the Property domain (at the 300 case mark).

Discussion: Of course the reason for the poor performance of the CNN and FP methods is that they do not have access to an entire case-base, and therefore, they do not always have access to the optimal case. For example, in the Travel domain, the CNN case-base at the 1000 case mark contains an average of 135 cases, that is, 13.5% of the total cases. Therefore, all other things being equal, we can expect the optimality of CNN, at the 1000 case mark, to be as low as 13.5%, a prediction that conforms well to the observed value of 15% optimality (see Figure 8(a)). While the FPRS method does not explicitly eliminate any cases from the case-base, during any given retrieval it is limited to a search of a small subset of these cases; namely, the FP set plus the related set of the reference case. Like the CNN subset, the FPRS subset is small relative to the entire case-base. For example, at the 1000 case mark in the Travel domain, the FPRS subset is about 14% of the case-base, and so one might expect FPRS optimality to be comparably low. However, the results show that the FPRS method has a retrieval optimality of 96% for the 1000 case mark in the Travel domain (see Figure 8(a)). The critical factor is that part of the FPRS subset is target specific. The related set has been chosen with reference to a specific target problem and, therefore, it is likely to contain the optimal case for a given target. This makes all of the difference, and ensures near perfect retrieval optimality for the FPRS method.

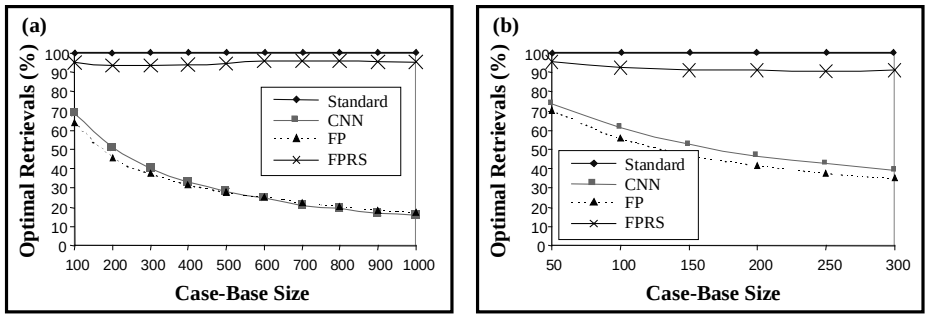

Figure 8. Percentage of Optimal Retrievals *vs*. Case-Base size for (a) the Travel, and (b) the Property domains.

6 Conclusions

In this paper we describe footprint-based retrieval, a novel retrieval approach that uses an explicit model of case competence to guide search. The approach consists of two distinct stages. During stage one, a compact competent subset of the case-base is searched to retrieve a case that is similar to the current target problem. This case acts as a reference point for the full case-base. During stage two the cases nearby to this reference case are searched and the closest one to the target is retrieved. A comprehensive evaluation of the retrieval technique shows that the approach benefits from superior efficiency, competence, and quality characteristics when compared to more traditional retrieval techniques.

The new method relies heavily on the availability of a comprehensive model of case competence and, of course, there is a cost associated with the construction of this model. In fact we can show that the model construction is $O(n^2)$ in the size of the case-base. However, for an existing case-base this can be thought of as an additional once-off setup cost and as such does not contribute an additional runtime expense.

Obviously the success of footprint-based retrieval will depend very much on the properties of the footprint set constructed for a given case-base. In this paper we have described one particular footprint construction algorithm based on CNN. However, other variations are possible and our future research will focus on a complete investigation of alternative approaches to footprint construction.

Our current work continues to investigate the issue of performance modelling in CBR, and we believe that predictive models hold the key to a wide range of open problems. We are currently building a range of applications to demonstrate a variety of different uses for performance models such as our model of competence. For example, we have already applied competence models to the problems of case-base maintenance, case deletion, case-base construction, and authoring support.

References

1. Aha, D.W., Kibler, D., and Albert, M.K.: Instance-Based Learning Algorithms. Machine Learning **6** (1991) 37-66.
2. Brown, M.G.: An Underlying Memory Model to Support Case Retrieval. In: Topics in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 837. Springer-Verlag, Berlin Heidelberg New York (1994) 132-143.
3. Dasarathy, B.V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Press, Los Alamitos, California (1991)
4. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory, **14** (1967) 515-516.
5. Leake, D.B., Kinley, A., and Wilson, D.: Case-Based Similarity Asessment: Estimating Adaptability from Experience. In: Proceedings of the 14th National Conference on Artificial Intelligence. AAAI Press (1997)
6. Lenz, M.: Applying Case Retrieval Nets to Diagnostic Tasks in Technical Domains. In: Smith, I. & Faltings, B. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 1168. Springer-Verlag, Berlin Heidelberg New York (1996) 219-233
7. Lieber, J.: A Criterion of Comparison between two Case-Bases. In: Haton, J-P., Keane, M., and Manago, M. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 984. Springer-Verlag, Berlin Heidelberg New York (1994) 87-100
8. Minton, S.: Qualitative Results Concerning the Utility of Explanation-Based Learning, Artificial Intelligence, **42**(2,3) (1990) 363-391
9. Schaaf, J. W.: Fish and Shrink: A Next Step Towards Efficient Case Retrieval in Large-Scale Case-Bases. In: Smith, I. & Faltings, B. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 1168. Springer-Verlag, Berlin Heidelberg New York (1996) 362-376
10. Smyth, B. and Keane. M. T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. Artificial Intelligence **102** (1998) 249-293
11. Smyth, B. & Keane, M.T.: Remembering to Forget: A Competence Preserving Deletion Policy for Case-Based Reasoning Systems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. Morgan-Kaufmann. (1995) 377-382
12. Smyth, B. & McKenna, E.: Modelling the Competence of Case-Bases. In: Smyth, B. & Cunningham, P. (eds.): .): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 1488. Springer-Verlag, Berlin Heidelberg New York (1998). 208-220
13. Veloso, M. Flexible Strategy Learning: Analogical Replay of Problem Solving Episodes. Proceedings of the 12th National Conference on Artificial Intelligence (1994) 595-600.
14. Wess, S., Althoff, K-D., Derwand, G.: Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning. In: Topics in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 837. Springer-Verlag, Berlin Heidelberg New York (1994) 167 – 181
15. Wolverton, M., and Hayes-Roth, B.: Retrieving Semantically Distant Analogies with Knowledge-Directed Spreading Activation. In: Proceedings of the 12th National Conference on Artificial Intelligence, (1994) 56-61.