Building Compact Competent Case-Bases

Barry Smyth & Elizabeth McKenna

Department of Computer Science, University College Dublin,
Belfield, Dublin 4, IRELAND
{Barry.Smyth, Elizabeth.McKenna@ucd.ie}

Abstract. Case-based reasoning systems solve problems by reusing a corpus of previous problem solving experience stored as a case-base of individual problem solving cases. In this paper we describe a new technique for constructing compact competent case-bases. The technique is novel in its use of an explicit model of case competence. This allows cases to be selected on the basis of their individual competence contributions. An experimental study shows how this technique compares favorably to more traditional strategies across a range of standard data-sets.

1 Introduction

Case-based reasoning (CBR) solves problems by reusing the solutions to similar problems stored as cases in a case-base [9]. Two important factors contribute to the performance of a CBR system. First there is *competence*, that is the range of target problems that can be successfully solved. Second, there is *efficiency*, the computational cost of solving a set of target problems.

Competence and efficiency both depend critically on the cases stored in the case-base. Small case-bases offer potential efficiency benefits, but suffer from reduced coverage of the target problem space, and therefore from limited competence. Conversely, large case-bases are more competent, but also more susceptible to the utility problem and its efficiency issues [see eg., 6, 11, 12, 14]. Very briefly, the utility problem occurs when new cases degrade rather than improve efficiency. For example, 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.

A key performance goal for any CBR system has to be the maintenance of a case-base that is optimal with respect to competence and efficiency, which in turn means maximising coverage while minimising case-base size. There are two basic ways of working towards this goal. The most common approach is to employ a case deletion strategy, as part of the run-time learning process, in order to ensure that all cases learned increase competence and efficiency. Recent research has suggested a number of successful deletion policies for machine learning systems [see eg., 11, 12], and more recently, a set of novel policies designed specifically for CBR systems [16].

Deletion works well by drawing on valuable statistical run-time performance data, but its starting point is an initial case-base that may be far from optimal. A second (complimentary) approach is to tackle the construction of the initial case-base itself. Instead of building a case-base from all available training instances we select only those that are likely to contribute to performance. This ensures that the initial case-base is near-optimal from the start. This process is referred to as *editing* the training data, and in this paper we present a new editing technique designed specifically for CBR systems.

Section 2 focuses on related editing work from the machine learning and pattern recognition literature that can be adapted for CBR. These techniques lack an explicit model of case competence, which, we argue, limits their effectiveness in a CBR setting. Section 3 addresses this issue by describing a competence model that can be used during case-base editing. Finally, Section 4 describes a comprehensive evaluation of the new approach.

2 Related Work

Related work on pruning a set of training examples to produce a compact competent edited set comes from the pattern recognition and machine learning community through studies of nearest-neighbour (NN) and instance-based learning (IBL) methods. In general, nearest neighbour methods are used in classification problems, regression tasks, and for case retrieval. Training examples are represented as points in an n-dimensional feature space and are associated with a known solution class (classification problems) or continuous solution value (regression tasks) or even a structured solution representation (case-based reasoning). New target instances (with unknown solutions) are solved by locating their nearest-neighbour (or k nearest neighbours) within the feature space [see eg., 1, 5, 7, 8, 18].

Since the 1960's researchers have proposed a variety of *editing* strategies to reduce the need to store all of the training examples. For instance many strategies selectively add training examples to an edited training set until such time as consistency over the original training set is reached; that is, until the edited set can be used to correctly solve all of the examples in the original training set [5, 7, 8, 18, 20, 21, 22].

Cases used in CBR systems are similar to the training examples used in classification systems and hence many of the same ideas about editing training data can be transferred to a CBR setting. The central message in this paper is that the successful editing of training data benefits from an explicit competence model in order to guide the editing process. Previous NN and IBL research reflects this, but the available models were designed for classification domains and not for case-based reasoning. We argue the need for a new competence model designed for the specific requirements of a case-based reasoner.

2.1 Condensed Nearest Neighbour Methods

A common approach for editing training data in NN and IBL methods is the condensed nearest neighbour method (CNN) shown in Algorithm 1. CNN produces an edited set of examples (the e-set) that is consistent with the original unedited training data (the o-set) [5, 8].

```
O-SET ← Original training examples
E-SET ← {}
CHANGES ← true

While CHANGES Do
          CHANGES ← false
          For each case C∈O-Set Do
            If E-SET cannot solve C Then
              CHANGES ← true
              Add C to E-SET
              Remove C from O-Set
            EndIf
          EndFor
EndWhile
```

Algorithm 1. Condensed Nearest-Neighbour Algorithm

CNN makes multiple passes through the training data in order to satisfy the consistency criterion. In classification problems a single pass is not sufficient as the addition of a new example to the e-set may prevent an example from being solved, even though it was previously solved by the smaller e-set. IB2, a common instance-based learning approach to editing, employs a version of CNN that just makes one pass through the training data and hence does not guarantee consistency [1].

The CNN has inspired a range of variations on its editing theme [1, 4, 5, 7, 18] but this represents just one half of the editing story. A second strategy was inspired by the work of Wilson [22]. While CNN filters correctly classified cases, so-called "Wilson editing" filters incorrectly classified cases. As with the seminal work of Hart [8], Wilson editing has inspired many follow-up studies [see 3, 10, 13, 19]. A full review of this large body of editing work is beyond the scope of this paper and the interested reader is referred to the references provided.

2.2 Competence Models

The CNN method suffers from two important shortcomings. First, the quality of the edited set depends on the order in which training examples are considered. Different orderings can result in different size edited sets with different competence characteristics.

A second problem is that the CNN approach adopts a naïve competence model to guide the selection of training examples. Its strategy is to add an example, e, only if it cannot be solved by the edited set built so far – by definition such an example will make a positive competence contribution. However, this is only true in the context of the edited set that has been built so far. In reality after more examples are added, it may turn out that the example, e, does not make any significant competence contribution because it is covered by later examples. In classification problems one approach is to select boundary examples for the edited set as these provide necessary and sufficient class descriptions. CNN as it stands tends to select such boundary examples but also contaminates the edited set with redundant interior examples [5, 7, 18] – it should be noted that alternative approaches, which focus on the selection of non-boundary (interior) cases or the generation of classification prototypes, do also exist (eg, [4, 21, 22]).

To address these issues the reduced NN (RNN) algorithm processes the final CNN edited set to delete such redundant examples. Briefly, if the removal of an example has no effect on consistency it is permanently deleted [7].

An alternative strategy is to order examples before CNN processing. One successful ordering policy for classification problems is to use the distance between an example and its nearest unlike neighbour (NUN). The NUN concept is based on the idea that training examples with different classes lie close to each other only if they reside at or near the boundaries of their respective classes; such examples have small NUN distances. By sorting examples in ascending order of NUN distance we can ensure that boundary examples are presented to CNN before interior examples and in this way increase the chances that interior examples will not be added to the final edited set [5, 18].

The NUN concept is a competence model for classification problems. It predicts that the competence of an individual example is inversely proportional to its NUN distance and as such provides a means of ordering training examples by their competence contributions.

2.3 Editing Case-Bases

The question we are interested in is how can CNN type techniques be best used in a CBR setting? In a more general sense however we are interested in how existing editing approaches from the classification community can be married with case-based deletion policies to produce a CBR-centric hybrid editing strategy.

Clearly the CNN concept is appropriate for CBR systems, but of course on its own it will produce sub-optimal case-bases that are order dependent and that include redundant cases. In the previous section we described how the NUN concept provided insight into the competence of training examples within classification problems. An analogous competence model is needed for case-based reasoning.

While conventional nearest-neighbour methods (or more correctly nearest-neighbour classifier rules) are often used in CBR systems, there are often a number of distinctions worth noting [9]. Firstly, cases are often represented using rich symbolic descriptions, and the methods used to retrieve similar cases are correspondingly more elaborate. Secondly, and most importantly, the concept of a correct solution can be very different from the atomic solution classes found in classification systems, where there are a small number of possible solution classes and correctness is a simple equality test. For example, in case-based planning or design problems, solutions are composite objects and the concept of correctness usually refers to a proposed target solution that is functionally or behaviourally equivalent to the true target solution (eg., [9, 15]).

As a result, CBR competence is different from competence in classification problems where boundary training examples can offer complete class coverage. Cases do tend to be clustered according to gross solution classes. However, the ability of a boundary case to solve an interior case is entirely dependent on the potential for solution adaptation and the availability of limited adaptation knowledge. Thus, the distinction between boundary and interior cases is no longer well-defined.

An implication of this argument is that the NUN distance metric may not be an appropriate competence model for CBR applications. A new competence model, designed specifically for CBR, is needed.

3 Modelling Case Competence

The idea that one can accurately model the competence of a case-base is a powerful one. In fact it has lead to a number of important developments in CBR in recent times, most notably in case deletion [16] and case-base visualisation and authoring support [17]. In this section we will argue that similar competence models can also be used to guide the construction of a case-base. This model differs from the model introduced by Smyth & Keane [16] in that it provides the sort of fine-grained competence measures that are appropriate for a CNN-type editing approach. In contrast the work of Smyth & Keane focused on a coarse-grained competence model capable of highlighting broad competence distinctions between cases, but incapable of making the find-graining distinctions that are important here. We will describe a new metric for measuring the relative competence of an individual case, and present this as a mechanism for ordering cases prior to case-base construction (editing).

3.1 A Review of Case Competence

When we talk about the competence of a case we are referring to its ability to solve certain target problems. 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 [15]. We can model these relationships according to the definitions shown in Def. 1 – 3.

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

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

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

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 be used to solve. Conversely, the reachability set of a *target problem* is the set of all *cases* that can be used to solve it.

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

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

If we could specify these two sets for every case in the case-base, and all possible target problems, then we would have a complete picture of the competence of a CBR system. Unfortunately, this is not feasible. First, due to the sheer size of the target problem space, computing these sets for every case and target problem is intractable. Second, even if we could enumerate every possible problem that the system might be used to solve, it is next to impossible to identify the subset of problems the system would actually encounter. Clearly, the best we can do is to find some approximation to these sets by making some reasonable, simplifying assumption.

So, to characterise the competence of a case-base in a tractable fashion we make the following **Representativeness Assumption**:

The case-base is a representative sample of the target problem space.

To put it another way, this assumption proposes that we use the cases in the case-base as proxies for the target problems the system is expected to solve. This assumption may seem like a large step, as it proposes that the case-base is representative of all future problems encountered by the system. It could be argued that we are assuming that all the problems faced by the system are already solved and in the case-base. We think that this greatly overstates the reality of the situation and underestimates the contribution that adaptation knowledge can play in modifying cases to meet target problems. Furthermore, we would argue that the representativeness assumption is one currently made, albeit implicitly, by CBR researchers; for if a case-base were not representative of the target problems to be solved then the system could not be forwarded as a valid solution to the task requirements. In short, if CBR system builders are not making these assumptions then they are constructing case-bases designed *not* to solve problems in the task domain. Of course implicitly this assumption is made by all inductive learners, which rely on a representative set of example instances to guide their particular problem solving task.

Armed with the representativeness assumption, we can now provide tractable definitions for coverage (Def. 6) and reachability (Def. 7):

Def 6: CoverageSet($c \in C$)=$\{c' \in C$:Solves(c,c')$\}$

Def 7: ReachabilitySet($c \in C$)=$\{c' \in C$:Solves(c',c)$\}$

Intuitively, the relative sizes of these sets seem to capture the relative competence of different cases. For example, cases with large coverage sets seem important because they can solve many other problems and therefore should solve many of the future target problems. Conversely, cases with small reachability sets seem important because they must represent regions of the target problem space that are difficult to solve (regions with a rich solution topology that require more cases for sufficient coverage). Unfortunately an accurate measure of true case competence is more complex than this. Overlapping sets between different cases can reduce or exaggerate the relative competence of an individual case (see also [16, 17]).

3.2 Relative Coverage

Previous work on the competence of cases has ignored ways of measuring the *precise* competence contributions of *individual* cases. For example, Smyth & Keane [15] present a number of competence categories to permit a coarse-grained competence assessment. Alternatively Smyth & McKenna [17] focus on the competence of groups of cases. We are interested in developing a more fine-grained measure that is similar in spirit to efficiency models such as the utility metric [11, 12].

To measure the competence of an individual case one must take into account the local coverage of the case as well as the degree to which this coverage is duplicated by nearby cases. To do this 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 Def. 8).

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

Def 8:

Some of the cases covered by c will also be covered by other cases, thereby reducing c's unique competence. For this reason, the relative coverage measure 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.

Figure 1 displays a number of cases and their relative coverage values. Case A makes an isolated competence contribution that is not duplicated by any other cases. Its coverage and reachability sets contain just a single case (case A itself) and so its relative coverage value is 1; case A is a pivotal case according to the competence categories of Smyth & Keane [16]. Case B makes the largest local competence contribution (its coverage set contains 3 cases, B, C and D) but this contribution is diluted because other cases also cover C and D. The relative coverage of B is 11/6 (that is 1+1/2+1/3). B is also a pivotal case but using relative coverage we can see that it makes a larger competence contribution than A; previously such fine-grained competence distinctions were not possible. Cases C and D make no unique competence contribution as they only duplicate part of the existing coverage offered by B. Consequently, C and D have relative coverage values of 5/6 and 1/3 respectively; they are both auxiliary cases according to the competence categories of Smyth & Keane [16].

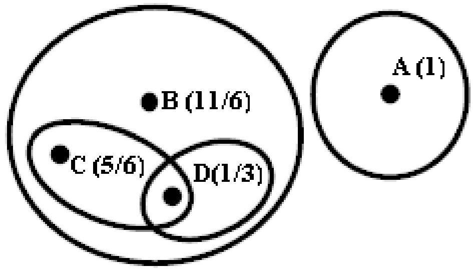

Fig. 1. Relative coverage values for cases. Each ellipse denotes the coverage set of its corresponding case and each RC value is shown in brackets.

3.3 Relative Coverage & CNN

In Section 2 we suggested that the CNN editing procedure could be used to construct compact competent cases-bases once a suitable measure could be found to sort cases by their likely competence contributions. Relative coverage is this measure. Our proposed technique for building case-bases is to use CNN on cases that have first been arranged in descending order of their relative coverage contributions. This will allow competence-rich cases to be selected before less competent cases and thereby maximise the rate at which competence increases during the case-base construction process.

4 Experiments

Our new editing technique is based on a specific model of competence for case-based reasoning. We argue that it has the potential for guiding the construction of smaller case-bases than some existing editing methods without compromising competence, specifically CNN on its own or CNN with NUN distance ordering. In turn we believe that, as an ordering strategy, relative coverage will continue to perform well in traditional classification problems. In this section we validate these claims by comparing the consistency, size, and competence of the case-bases produced using the different editing techniques on a range of standard data-sets.

4.1 Experimental Setup

Three different editing techniques are compared for this experimental study (1) CNN – the standard CNN approach; (2) NUN – CNN with cases ordered according to their NUN distances; (3) RC – CNN with cases ordered according to their relative coverage values.

Four different data-sets are used. Two, Credit (690 cases) and Ionosphere (351 cases), represent classification problems and are available from the UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html) [2]. The other 2 are more traditional CBR data-sets. Property (506 cases) is also from the UCI repository and Travel (700 cases) is available from the AI-CBR Case-Base Archive (www.ai-cbr.org). The important point to note is that Property and Travel are not used as classification data-sets. Instead they are used to build a case-based recommendation system where the objective is to locate a case that is sufficiently similar to a given target problem across a range of solution features. Consequently, the concept of a single solution class is no longer valid in keeping with many CBR applications and domains.

4.2 Consistency Growth

This first experiment is designed to investigate how the consistency of a case-base (that is, competence with respect to the initial training data) varies as more cases are added. We are interested in comparing the rate of increase of consistency for the various editing strategies across the different data-sets.

Method: For each data-set, 3 case-bases (edited sets) are constructed by using each of the editing strategies on the available training cases. As each case is added to a case-base, the consistency of that case-base is measured with respect to the initial training cases; that is, we measure the percentage of training cases that can be solved by the case-base built so far.

Results: This experiment generates 4 consistency graphs (one for each data-set), each containing 3 plots (one per editing strategy). The results are shown in Figures 2(a)-(d) as graphs of percentage consistency versus case-base size as a percentage of overall training set size.

Discussion: In this experiment 100% consistency is achieved by RC with fewer cases (albeit marginally fewer) than with any other editing strategy. Unfortunately, as we shall see in the next experiment, this result does not hold in general. However, aside from the size of the final edited case-bases, we do notice that the graphs indicate that the RC method is selecting more competent cases more quickly that the other strategies. For example, in the Travel domain the consistency of the case-base produced by the RC strategy at the 10% size level is approximately 65% (that is 65% of the training set can be solved by a 10% subset). In contrast, the CNN policy produces a case-base with only 40% consistency, and NUN produces a case-base with only 45% consistency at this 10% size level. Similar results are found in the Property domain. The results on the classification data-sets are not as positive, but still bode well for RC. The RC policy generally out-performs CNN and keeps pace with NUN particularly for small case-base sizes. This leads us to conclude that the relative coverage measure is also a valid measure of competence in traditional classification domains.

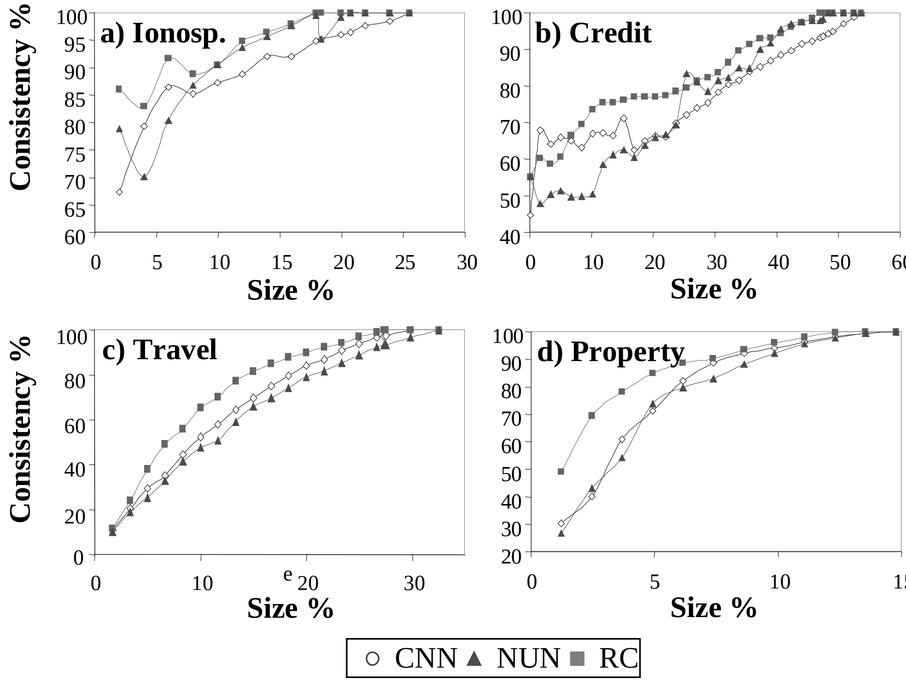

Fig. 2(a)-(d). Case-Base Consistency versus Size.

4.3 Size vs Competence

While consistency is a measure of performance relative to the training set, the true test of editing quality is the competence of the edited set on unseen test data. In this experiment, we compare the sizes of the case-bases to their competence on unseen target problems.

Method: Each editing strategy is used to generate case-bases for the 4 data-sets. However, this time 100 random test problems are removed from the training set before case-base construction. The final size of the case-bases (at the 100% consistency mark) and their competence over the 100 test problems is noted. This process is repeated 100 times, each time with a different set of 100 random test problems, to generate 1200 case-bases.

Results: For each data-set and editing strategy we compute the mean case-base size and competence over the 100 test runs. The results are shown in Table 1. Each cell in the table holds two values: the mean size (top value) and competence (bottom value) of the case-bases produced by a given editing strategy on a given data-set.

Discussion: RC and NUN produce smaller case-bases than the standard CNN approach for the classification data-sets (Ionosphere & Credit) – NUN case-bases are marginally smaller than the RC case-bases, but to compensate the competence of the

RC case-bases is higher. In fact, with the Credit data-set the RC method produces a case-base with a competence value that is higher than the CNN case-base which is, on average, nearly 50 cases larger.

RC produces significantly smaller case-bases than both of the other editing strategies for the CBR data-sets (Travel & Property). This is because relative coverage is an explicit competence model for CBR while NUN is designed for classification problems. In fact, we notice that in these data-sets the NUN method is performing even worse than CNN – further evidence that the NUN distance concept is not appropriate in a CBR setting.

Dataset/Editing	**CNN**	**NUN**	**RC**
Ionosphere	61.93 85.78	46.39 84.44	49.47 85.3
Credit	344.84 58.85	297.43 58.95	299.19 60.44
Travel	184.28 89.25	196.98 88.72	165.42 86.4
Property	55.19 95.92	57.81 95.53	45.44 94.62

Table 1. A comparison of different editing strategies over the test data-sets in terms of mean case-base size and competence. The upper value in each cell is the average size of the case-bases produces and the lower value is the average competence value.

One of the problems with this experiment is that it is impossible to compare case-bases with different sizes and competence values. For example we've already noted that the RC method produces slightly larger case-bases than NUN in the classification problems, but that these case-bases have better competence values. Conversely, in the CBR data-sets, RC is producing much smaller case-bases, but these case-bases have slightly lower competence values. What do these competence differences mean? Are the competence drops found in the CBR data-sets because the RC method is selecting cases that generalise poorly over the target problems, or are they a natural implication of the smaller case-bases? If we remove cases from the CNN and NUN case-bases (or conversely add cases to the RC case-bases) so that all case-bases are normalised to the same size, how would this change their competence values? These questions are answered in the next experiment.

4.4 Normalising Competence

This experiment compares the competence of the case-bases produced by the different strategies after normalising each with respect to the size of the RC case-bases. The argument could be made that this size-limiting experiment is artificial and that is serves only to hamper the performance of the other algorithms. However we disagree. We are not just interested in the ultimate size and competence of the edited case-base that is produced by a particular editing policy. We are interested in how competence grows as more cases are added. If, for example, the RC policy is seen to more aggressively increase competence than the competing policies then this is an important advantage, particularly if our editing strategies must work within a resource-bounded setting where, for example the maximum size of the edited set is limited.

Method: Each of the CNN and NUN case-bases from the previous experiment are normalised with respect to their corresponding RC case-base by adding or removing cases as appropriate. To ensure fairness cases are added or removed using the appropriate strategy. For example, if a case is removed from a NUN case-base then it will be the last case that was added.

Results: The results are shown in Table 2. Each value is the mean competence of the case-bases produced by each of the editing strategies once they have been normalised to the appropriate RC case-base size.

Discussion: The results are positive. The competence of the RC case-bases is higher than the corresponding case-bases produced by the other strategies after normalisation. This demonstrates that the RC method is selecting cases that are more competent than those selected by any other method, backing up the results found in section 4.2 when consistency was measured. Moreover, the relative coverage measure performs well in both classification and CBR settings, while the NUN method performs relatively poorly in the CBR data-sets. In fact, in Table 2 we see that the normalised competence values for the NUN case-bases are smaller than the competence values for the CNN case-bases, for the CBR data-sets.

Dataset/Editing	CNN	NUN	RC
Ionosphere	84.26	85.23	85.3
Credit	58.36	59.3	60.44
Travel	85.03	83.23	86.4
Property	92.65	91.9	94.62

Table 2. The competence values of all case-bases normalised to the RC case-base size.

5 Conclusions

The ability to edit training data prior to learning has been an important research goal for the machine learning community for many years. We have adapted a traditional editing procedure, CNN, for use with case-based reasoning systems. The central idea behind the adaptation is that effective editing must be based on an accurate model of case competence, so that the competence of a case-base can be optimised with respect to its size. A new editing technique was introduced, based on a novel measure of case competence called relative coverage. This new technique was evaluated with respect to a number of more conventional editing strategies and on a variety of classification and CBR data-sets. The results were positive but tentative. The new method performed well on all data-sets and out-performed all rivals on the CBR data-sets. In general we saw that the relative coverage measure allowed our editing technique to select cases with higher competence contributions than those cases selected by any competing editing strategy.

However, before closing we would like to emphasise that this research represents the tip of the iceberg of case-base editing. Obviously our current experiments need to be extended to include a broader range of traditional editing techniques such as the Wilson-editing approaches [3, 10, 13, 19, 22]. We have described a competence model for CBR that appears to benefit the editing process, and we have integrated this into one particular editing approach. Future work will consider the more general properties of this model with respect to other editing strategies. We believe that, ultimately, the optimal approach to editing case-bases will incorporate a range of ideas from a variety of editing approaches.

References

1. Aha, D. W., Kibler, D., and Albert, M.K.: Instance-Based Learning Algorithms. Machine Learning, **6** (1991) 37-66
2. Blake, C., Keogh, E. & Merz, C.J: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)
3. Broder, A.Z., Bruckstein, A.M., and Koplowitz, J.: On the Performance of Edited Nearest neighbor Rules in High Dimensions. IEEE Transactions on Systems, Man, and Cybernetics, SMC-**15**(1), (1985) 136-139
4. Chang, C.L.: Finding Prototypes for Nearest Neighbor Classifiers. IEEE Transactions on Computers, **2-3**(11), (1974) 1179-1184
5. Dasarathy, B.V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Press, Los Alamitos, California (1991)
6. Francis, A.G. and Ram, A.: A Comparitive Utility Analysis of Case-Based Reasoning and Control Rule Problem Solving. In: Proceedings of the 8th European Conference on Machine Learning (1995)
7. Gates, G.W.: The Reduced Nearest Neighbor Rule. IEEE Transactions on Information Theory, **18**(3) (1972) 431-433
8. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory, **14** (1967) 515-516
9. Kolodner, J. Case-Based Reasoning. Morgan-Kaufmann, San Mateo, California (1993)
10. Koplowitz, J. & Browm, T.A.: On the Relation of Performance to Editing in Nearest-Neighbor Rules. Proceedings of the 4th International Joint Conference on Pattern Recognition, IEEE Computer Society Press (1978) 214-216
11. Marckovitch, S. & Scott, P.D.: Information Filtering: Selection Mechanisms in Learning Systems. Machine Learning, **10** (1993) 113-151
12. Minton, S.: Qualitative results concerning the utility of explanation based learning. Artificial Intelligence, **42**(2-3) (1991) 393-391
13. Penrod, C.S & Wagner, T.J.: Another Look at the Edited Nearest Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics, **SMC-7**(2) (1977) 92-94
14. Smyth, B. & Cunningham, P.: The Utility Problem Analysed: A Case-Based Reasoning Perspective. 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) 392-399
15. Smyth, B. & Keane, M.T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. Artificial Intelligence, **102** . (1998) 249-293

16. 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
17. 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
18. Tomek, I.: Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, **7**(2) (1976) 679-772
19. Wagner, T.J.: Convergence of the Edited Nearest Neighbor. IEEE Transactions on Information Theory, **IT-19**(5) (1973) 696-697
20. Wilson, D. R. & Martinez, T.R.: Instance Pruning Techniques. In: Proceedings of the 14th International Conference on Machine Learning (1997) 404-441
21. Wilson, D. R. & Martinez, T.R.: Reduction Techniques for Exemplar-Based Learning Algorithms. Machine Learning (1998)
22. Wilson, D. L Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics, **2-3** (1972) 408-421
23. Zhang, J.: Selecting Typical Instances in Instance Based Learning. In: Proceedings of the 9th International Conference on Machine Learning (1992)