Competence Guided Casebase Maintenance for Compositional Adaptation Applications

Ditty Mathew and Sutanu Chakraborti

Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai - 600036
{ditty,sutanuc}@cse.iitm.ac.in

Abstract. A competence guided casebase maintenance algorithm retains a case in the casebase if it is useful to solve many problems and ensures that the casebase is highly competent in the global sense. In this paper, we address the compositional adaptation process (of which single case adaptation is a special case) during casebase maintenance by proposing a case competence model for which we propose a measure called retention score to estimate the retention quality of a case. We also propose a revised algorithm based on the retention score to estimate the competent subset of the casebase. We used regression datasets to test the effectiveness of the competent subset obtained from the proposed model. We also applied this model in a tutoring application and analyzed the competent subset of concepts in tutoring resources. Empirical results show that the proposed model is effective and overcomes the limitation of footprint based competence model in compositional adaptation applications.

Keywords: Casebase Maintenance, Case Competence, Footprint based Competence model, Compositional Adaptation

1 Introduction

Case Based Reasoning(CBR) systems solve new problems by retrieving similar past problems from a casebase and adapting their solutions. The adaptation process can be done in two ways - single case adaptation and compositional adaptation. In single case adaptation, the solution of a single case can be adapted to solve the target problem whereas in compositional adaptation the solutions from multiple cases are combined to produce a new composite solution [16]. Casebase Maintenance is a branch of CBR, which aims at looking into the quality of cases that should be retained in the casebase; the goal is often to maintain a compressed casebase that can solve new problems effectively [12]. We need to ensure that the cases in the compressed casebase would be able to be retrieved and adapted for a wide range of problems in the casebase. Thus, the competence of a casebase can be determined by the ability of the cases in the casebase to solve a large number of problems. A competence guided casebase maintenance algorithm retains a case in the casebase if it is useful to solve many problems and ensures that the casebase is highly competent in the global sense [13]. For this, it is important to mark the cases that are involved in both the single case and compositional adaptation process in the past so that we can use this knowledge to measure coverage.

Footprint-based retrieval [15] is an efficient retrieval approach in CBR, which guides the search procedure using a case competence model [14]. This approach identifies a compact competent subset of the casebase called footprint set, using the case competence model. However, the competence model used in the footprint-based approach covers only the situation where a single case is adapted to solve a problem. It turns out that many CBR applications require compositional adaptation for their adaptation process. In such cases, the dependency between the cases has to be taken into consideration when we estimate the competence of each case in the casebase. To the best of our knowledge, no previous work has attempted to address the maintenance of casebase which requires compositional adaptation. So, we are motivated by the research question, *"How can we model a competence guided casebase maintenance model where the adaptation process involves compositional adaptation?"*

In this paper, we propose a new competence model which can be applied in an application that involves compositional adaptation (of which the single case adaptation is a special case). This model is based on a measure called retention score which estimates the retention quality of a case in the casebase. We also propose a revised approach to identify the footprint set where compositional adaptation is required. Section 2 reviews the literature on case competence model and footprint-based approach in particular. Section 3 summarizes the research in compositional adaptation applications and illustrates the flaw of footprint-based approach when used compositional adaptation applications. Our approach to measure the retention quality and the revised footprint approach are described in Section 4 using examples based on synthetic casebases. Section 5 presents the empirical results obtained on synthetically generated datasets. In Section 6, we demonstrate the proposed approach in a tutoring application and show the importance of the retention score measure with the support of experimental results.

2 Footprint-based Approach

In Case Based Reasoning, the impact of utility depends on the size and growth of the casebase. Since efficiency (and on occasions effectiveness) is adversely affected in the presence of large number of not-so-useful cases, it is desirable to weed out such cases. Markovitch and Scott [5] have characterized an information filtering approach based on selective utilization and selective retention strategies to deal with the utility problem. This selective utilization and selective retention strategies ensure that stored knowledge is genuinely useful, and the performance will not be affected by the deletion of any information. In [13], Smyth and Keane introduced a case competence model to guide the learning and deletion of cases. The competence of a CBR system is the range of target problems that the given system can solve. The global competence of a system also relies on the local problem-solving properties such as the coverage and reachability of each case. For the purpose of defining these properties, Smyth and McKenna [15] defined a relation *solves* between a case c and a target problem t as c solves t ($solves(c, t)$) and this relation is defined as in Def 1.

Def 1. $solves(c, t)$ iff c is retrieved and c can be adapted for t

Using this relation the competence properties such as coverage and reachability of each individual case is defined as in the Def 2 and 3 respectively.

Def 2. $Coverage(c) = \{c' \in \mathbb{C} : solves(c, c')\}$

Def 3. $Reachability(c) = \{c' \in \mathbb{C} : solves(c', c)\}$

Global competence of a casebase is a function of how the local competences of the cases interact when they are combined. When there is any overlap between the coverage of cases in the casebase, its individual contribution may not contribute globally [14]. The unique competence contribution of an individual case to solve a target problem depends on the presence of alternate solutions for the target problem. Smyth and McKnenna [15] defined a measure called relative coverage based on the idea that if a case c can be solved by n other cases then each of the n cases will get a contribution of $1/n$ from c to their relative coverage measures. Thus, relative coverage provides a mechanism to estimate the contribution of each case to global competence.

$$RelativeCoverage(c) = \sum_{c' \in Coverage(c)} \frac{1}{|Reachability(c')|} \tag{1}$$

The maintenance strategy of the casebase becomes more and more critical in real-world situations. Competence directed casebase maintenance should delete irrelevant cases that guide the casebase to maximizes its competence [13]. Smyth and McKenna [15] estimated the set of cases that is to be retained using the relative coverage measure where the final set (i.e. footprint set) contains cases with large competence contributions and this set covers the rest of the cases in the casebase. For the construction of footprint set, first the cases are sorted in the descending order of the relative coverage values and then each case is added to the footprint set in this order only if the current footprint does not already cover it. As the cases are sorted based on the relative coverage, the larger competent cases will get added before the smaller competent cases and thus keep the footprint size to a minimum. The retrieval strategy based on this footprint set is not only a simple and novel approach but also it directs the use of competence model to guide the retrieval process. However, the relation *solves* considers only a single case for adaptation while estimating the footprint set.

3 Compositional Adaptation

In compositional adaptation, solutions from multiple similar cases are combined to obtain a new solution for a query problem. For example, in a regression setting where the data instances are the cases, the solutions from the k-nearest neighbor cases can be adapted to predict the target value of the corresponding case [10]. In the Airquap CBR system for predicting pollution levels, the solution to the target problem is the mean value of the solutions of the most similar cases [3]. Arshadi et. al [1] proposed an approach for designing a tutoring library by applying compositional adaptation. This method identifies the books or parts of the book for the user's search topic in the library by combining the solutions of the similar past requests by other users. Atzmueller et. al [2] examined the compositional case adaptation approach in the multiple disorder situation during medical diagnosis. The proposed approach identifies the solution based on the solutions of the k most similar cases. In [8], Muller et. al attempted the compositional adaptation of cooking recipes by decomposing the cooking recipe cases into reusable streams [9]. The adaptation process compensates the deficiencies of the retrieved recipe by replacing the retrieved one with the streams of appropriate cooking recipes.

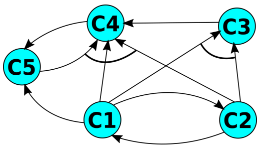

Fig. 1: An example of casebase where compositional adaptation is involved

We illustrate the drawback of the competence model in footprint-based approach when the adaptation process involves compositional adaptation. Fig 1 shows a network of cases where each node represents cases and an edge from one case (say c_1) to other case (say c_2) indicates that the case c_1 can be retrieved and its solution can be adapted to solve c_2. As per the definition of *solves* in Def 1, the edge $c_1 \to c_2$ implies c_1 *solves* c_2. The arc (*AND* arc) between the edges represents compositional adaptation. For example, the arc between the edges $c_1 \to c_3$ and $c_2 \to c_3$ in the network indicates that the composite solution of the case c_1 and c_2 can solve the problem c_3. It is to be noted that neither case c_1 nor c_2 can solve c_3 in isolation. The footprint-based approach discussed in Section 2 cannot have the *AND* arcs between incoming edges, and outputs a footprint set $\{c_1\}$ corresponding to this network. Though Smyth et. al [15] proposed the footprint approach such that the footprint set covers the entire casebase, the footprint set identified for the casebase in Fig 1 solves all the cases in the network only when compositional adaptation is not taken into consideration. For example, case c_3 cannot be solved by this footprint set as the case c_3 needs case c_2 which is not present in the footprint set, apart from c_1 to solve it. The current competence model has to be enhanced to include compositional adaptation.

4 Approach

In this section, we present a case competence model which covers the compositional adaptation (CA) process. Compositional adaptation composes a new solution by combining the solutions of multiple cases; cases which are used for adapting the new solution form an *AND* relation. It is possible to have multiple adapted solutions (either single case or compositional) for a target problem. These multiple solutions for a target problem shape an *OR* relation. The *AND* relation implies all the cases that are part of this relation are required to adapt a new solution and the *OR* relation indicates any of the cases can solve the target problem. The casebase is comprised of *AND-OR* relations between cases (or a disjunction over conjunctions). We assume that the compositional adaptation operator is a disjunction over conjunctions.

We define the relation $solves_{CA}$ in the context of compositional adaptation corresponding to the relation *solves* in Def 1. For a casebase $\mathbb{C}$, $solves_{CA}$ is defined in Def 4.

Def 4. A set of cases $\mathbb{C}' \subset \mathbb{C}$ solves$_{CA}$ a target problem t if and only if all the cases in $\mathbb{C}'$ are retrievable for t and the solutions of the cases in $\mathbb{C}'$ can be adapted to solve t.

For example, in Fig 1 the combined solution of the cases c_1 and c_2 solves the problem c_3 i.e,. $solves_{CA}(\mathbb{C}', c_3)$ where $\mathbb{C}' = \{c_1, c_2\}$. As compared to the Smyth's competence model [14] which considers c_1 solving c_3 independent of c_2, here we need to model the fact that the cases c_1 and c_2 cannot individually solve the target problem. We exploit $solves_{CA}$ in the competence model and redefine the $coverage_{CA}$ and $reachability_{CA}$ as in Def 5 and Def 6. The Coverage$_{CA}$ is defined for a set of cases and $Reachability_{CA}$ is defined for each case. Each element in Reachability$_{CA}$ of a case c is a set of cases which can be used for either single case or compositional adaptation to solve the target c.

Def 5. Coverage$_{CA}(\mathbb{C}' \subset \mathbb{C}) = \{c \in \mathbb{C} : solves_{CA}(\mathbb{C}', c)\}$

Def 6. Reachability$_{CA}(c) = \{\mathbb{C}' \subset \mathbb{C} : solves_{CA}(\mathbb{C}', c)\}$

For example, in Fig 1 $Coverage_{CA}(c_1, c_2) = \{c_3\}$ and $Reachability_{CA}(c_4) = \{\{c_1, c_2, c_5\}, \{c_3\}\}$. The dependency between the cases in solving the problems has to be considered when we estimate the competence of each case in the casebase. Finally, this should reflect in the footprint set.

We propose a measure called *retention score* which orders the cases by considering compositional adaptation based on the extent to which a case is to be retained in the casebase. This measure quantifies the competence of a case in the casebase. Then, we propose a modified algorithm of Smyth's footprint [15] identification called footprint$_{CA}$ algorithm which identifies the footprint$_{CA}$ which reflects compositional adaptation.

4.1 Retention Score

The retention score is a measure which quantifies the importance of a case in terms of whether it is required to be retained in the casebase or not. To illustrate the idea of retention score, consider the graphs constructed out of synthetic casebases in Fig 2 and Fig 3. In the first one, the cases c_1 and c_2 are essential to retain as both are required to cover the other cases c_3 and c_4. However, in the second one the case c_1 requires c_2 to solve c_3, and both c_2 and c_5 to solve c_4. The factors that determine the retention quality of a case are the range of problems that it solves and the number of cases that are required to solve those problems. In a casebase, we would like to retain fewer good retention quality cases that cover more useful cases. To estimate the retention score, we define two terms - covered cases and support cases.

The covered cases of a case c ($CoveredCases(c)$) include all the cases that c can be used to solve either on its on, or in conjunction with other cases. For example, $CoveredCases(c_1)$ in the network shown in Fig 2 is $\{c_3, c_4\}$.

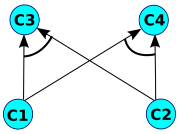

Fig. 2: Synthetic network 1

The support cases of a case c_i to solve the problem c_j ($SupportCases(c_i, c_j)$) is the set of cases that the case c_i requires to solve c_j. For example, in Fig 3 the $SupportCases(c_1, c_3)$ is $\{c_2\}$ and the $SupportCases(c_1, c_4)$ is $\{c_2, c_5\}$.

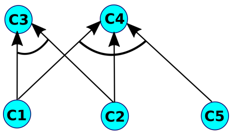

Fig. 3: Synthetic network 2

The proposed measure for retention score is based on these two sets and it is based on the idea that ***a case has high retention score if it can solve several cases that have high retention score with as few cases that have high retention score***. More precisely, the retention score of a case is high if there are more covered cases that have high retention score with less number of support cases that have high retention score. Using this idea we came across the recursive formulation as given in Equation 2.

$$RetentionScore_{k+1}(c) = \sum_{c_i \in CoveredCases(c)} \frac{RetentionScore_k(c_i)}{\sum\limits_{c_j \in SupportCases(c,c_i)} RetentionScore_k(c_j) + 1} \quad (2)$$

where $RetentionScore_{k+1}(c)$ is the retention score of a case c at $k+1^{th}$ iteration. Each covered case contributes to the estimation of the retention score based on its retention score and the retention score of the support cases that solve this covered case. The addition of 1 in the denominator is to handle the situation when a case does not need any support case to solve the corresponding covered case. For the first iteration of the retention score estimation, the retention score of a case c can be estimated as,

$$RetentionScore_0(c) = \sum_{c_i \in CoveredCases(c)} \frac{\frac{1}{1+|\{\mathbb{C}' \,:\, \mathbb{C}' \in Reachability_{CA}(c_i) \text{ and } c \notin \mathbb{C}'\}|}}{1 + |SupportCases(c, c_i)|} \quad (3)$$

The numerator part for each covered case c_i in Equation 3 captures the individual contribution of c in solving c_i. The contribution of c in solving c_i is high if c is involved in all the solutions of c_i. Thus, the individual contribution of c to solve c_i decreases with increase in the number of alternate solutions which do not contain c. The denominator of Equation 3 ensures that the retention score increases with decrease in the number of support cases that c requires to solve

c_i and vice versa. The addition of 1 in the denominator handles the situation when there are no supporting cases.

The retention score recursively measures the global competence of each case in the casebase. The recursive formulation of retention score captures the transitive solving property of cases. For example, if a case c_1 solves c_2 and c_2 solves c_3, then c_1's contribution in solving c_3 will also be captured. But, relative coverage measure used in the footprint-based approach [15] cannot reveal the transitive coverage of a case. The relative coverage measure express only the individual contribution of each case irrespective of the requirements of other cases in solving a target problem.

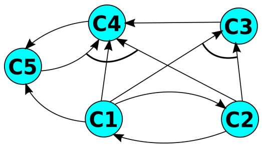

Fig. 4: A sample Casebase graph

Case (c)	c_1	c_2	c_3	c_4	c_5
RetentionScore(c)	2	1.75	1.29	1.23	1
RelativeCoverage(c)	2.25	1.75	0.25	0.5	0.25

Table 1: RetentionScore and RelativeCoverage of cases

In Fig 4, the graph of the casebase example has been reproduced from Fig 1. The retention score of the cases in the network become stable after 15 iterations. The scores are given in Table 1. We normalize the retention score values to range 1 to 2 after each iterations. The ordering based on retention score is obtained as c_1, c_2, c_3, c_4, c_5. The case c_1 secured highest retention score as it covers two cases without any supporting cases and two other cases with support cases. Though c_3 needs no support cases, its score is less due to the lack of its coverage. However, the case c_4 has even lesser score than c_3 although it covers same number of covered cases with no support cases. This is because, the covered case of c_4 i.e., c_5 can be alternatively solved by c_1 which has more coverage. However, the relative coverage values of the cases shown in Table 1 shows that the values are estimated only based on the participation of solving a target case. For example, the case c_5 secures a relative coverage value as it helps in solving c_4 irrespective of the requirement of c_1 and c_2 in solving c_4. This notion has been captured by the retention score.

4.2 Footprint$_{CA}$ Algorithm

The footprint algorithm proposed by Smyth et. al [15] does not consider compositional adaptation while constructing the footprint set. We modified the Smyth's footprint algorithm to obtain the footprint$_{CA}$ set and the algorithm is described in Algorithm 1. This algorithm estimates the footprint by adding the cases in the decreasing order of retention score if none of the composite solution of a case is present in the footprint set. Thus the cases with high retention quality are added before the cases with less retention quality, and thus help to keep the good quality cases in the footprint set. We preserve the retention score ordering of cases in the final footprint set. In this way, the footprint$_{CA}$ set for the example in Fig 1 is obtained as $\{c_1, c_3\}$. It may be noted that this set can

Algorithm 1: Footprint$_{CA}$ algorithm

```
Input: Cases sorted based on retention score, Output: Footprint$_{CA}$ (FP)
Cases $\leftarrow$ Sorted cases according to their retention score
FP $\leftarrow$ {}
Changes $\leftarrow$ true
while Changes do
    Changes $\leftarrow$ false
    for each $c \in Cases$ do
        if none of the composite solution of $c$ is a subset of FP then
            Changes $\leftarrow$ true
            Add $c$ to FP
```

cover all concepts in the given network whereas the Smyth's footprint set $\{c_1\}$ which is based on relative coverage cannot cover all the cases in the network.

5 Evaluation

We empirically tested the proposed competence model by using synthetic regression datasets.The datasets are generated based on the factors like dimensions, the number of data points, the distance between neighbors, and non-linearity. The generation process of datasets used for analysis are illustrated below.

1. *Synthetic data 1:* $y = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$ +noise
2. *Synthetic data 2:* $y = x_1^4 + x_2^3 + x_3^2 + x_4 + \cos^2(x_5)$ +noise
3. *Synthetic data 3:* $y = \sin(x_1 x_2) + \sqrt{x_3 x_4} + \cos^2(x_5) + x_6 x_7 + x_8 + x_9 + x_{10}$ +noise

The data points across each dimension of all the datasets are sampled uniformly with values between 0 and 10; we added a random gaussian noise with mean 0 and standard deviation 10. The structure of the datasets are - *Synthetic data 1* is linear and high dimensional; *Synthetic data 2* is nonlinear and low dimensional; *Synthetic data 3* is nonlinear and high dimensional.

5.1 Experimental Setup

Each data instance is considered as a case in the casebase and each case is assumed to be solved by the compositional adaptation solution of its k-nearest neighbor cases. Thus the casebase graph contains cases as nodes, and edges from the k-nearest neighbors of each case which are connected to it by an AND arc. Then the footprint$_{CA}$ set is estimated using this graph and is compared with the footprint$_{OR}$[1] set which is obtained from the same graph by removing the composition (AND) condition. The experiments are done with k=1,2 and 4 and by varying the number of instances (casebase size) from 10 to 100. At k=1, the adaptation process uses a single case; multiple cases are used when k> 1.

[1] We refer the Smyth's footprint set [15] as the footprint$_{OR}$ set

5.2 Evaluation Criteria

The analysis of the footprint size is one of the common criteria for evaluation. However, the size of both the footprint sets are not strictly comparable as the footprint$_{CA}$ is expected to have more cases than the footprint$_{OR}$ set due to composition condition in the former set. Fig 5 illustrates that the footprint$_{OR}$ size is less compared to footprint$_{CA}$. The size of footprint$_{OR}$ decreases with increase in the value of k where as the size of footprint$_{CA}$ increases with increase in the value of k. For a high value of k, more cases are involved in compositional adaptation during which the footprint$_{OR}$ size compresses more and thereby loses composition knowledge of adaptation. Hence, we propose two measures to estimate the effectiveness of footprint$_{CA}$ obtained based on the retention score in a compositional adaptation application - casebase coverage and footprint sanity measure. We compare the results obtained over the footprint$_{CA}$ with the footprint$_{OR}$ set computed using the relative coverage measure in the same application.

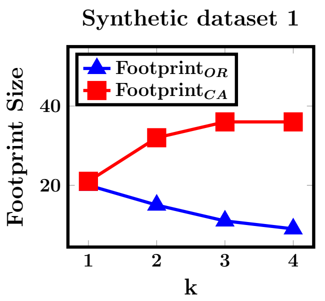

Fig. 5: Footprint size analysis

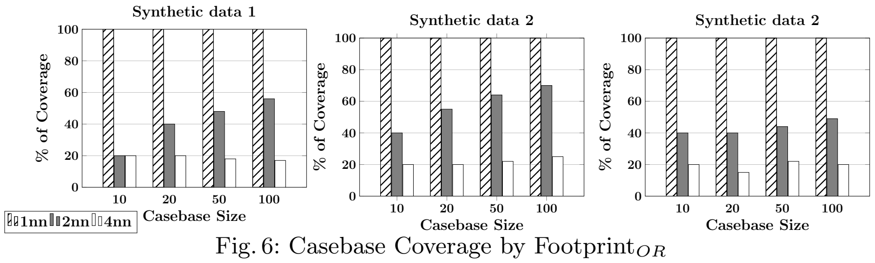

Fig. 6: Casebase Coverage by Footprint$_{OR}$

Casebase Coverage The essential idea of the footprint set is that the footprint cases solve all the cases in the casebase. The casebase coverage of a footprint set fp is measured as follows,

$$\text{Casebase Coverage}(fp) = \frac{|\text{Cases that are solved by } fp|}{\text{Casebase Size}} \tag{4}$$

The main aim of this evaluation measure is to examine the effectiveness of footprint$_{CA}$ and footprint$_{OR}$ in the compositional adaptation application. As footprint$_{CA}$ is formulated for compositional adaptation; this set is expected to cover the entire casebase. However, the usefulness of footprint$_{CA}$ set can be observed by analyzing the casebase coverage of footprint$_{OR}$ set.

We analyzed that footprint$_{CA}$ has full casebase coverage all the dataset. However, the footprint$_{OR}$ set covers the entire casebase only when k=1. The analysis of coverage on footprint set is illustrated in Fig 6. We can observe that the percentage of coverage increases with increase in the number of data points when k=2 in all the datasets. Also, the coverage percentage decreases with increase in the value of k. The reason behind this is that the increase in the number of neighbors decreases the size of footprint set and there by reduces its effectiveness. This indicates the ineffectiveness of the footprint$_{OR}$ set to apply it in compositional adaptation applications.

Sanity Check To measure the sanity of the footprint set, we found a method to identify a set of cases that can cover the entire casebase using a graph-theoretic approach. We estimate the footprint set from the case network that is constructed using the relation $solves_{CA}$. In the same network, if we repeatedly remove the cases that do not solve any other cases until there are no such cases, the final network turns out to be a compressed set of cases that can solve all the cases in the casebase transitively. This final network is called the *kernel* of the case network. The algorithm for computing the kernel is given in Algorithm 2. Though there is no ordering of cases provided within the kernel, the cases in the kernel are the potential cases that can be presented in a footprint set. So, we compare the cases in the footprint set and kernel. The sanity measure is defined as,

$$\text{Sanity rate} = \frac{|\text{footprint cases} \cap \text{kernel cases}|}{|\text{kernel cases}|} \times 100 \tag{5}$$

This idea is adapted from [6] where Masse et. al estimate the grounding kernel of a dictionary graph where the graph is constructed from word definitions. Here the grounding kernel turns out to be the set of words from which the entire dictionary words have been defined.

Algorithm 2: Computing the Kernel of the Case Network

```
Input: Case Network $\mathbb{G}$, Output: Kernel $\mathbb{K}$
$\mathbb{K} \leftarrow \mathbb{G}$ do
    Let $\mathbb{C}$ be the set of cases (vertices) in $\mathbb{K}$
    $\mathbb{U} \leftarrow \{v \in \mathbb{C} : \text{out-degree of v in } \mathbb{K} = 0\}$
    Remove all elements in $\mathbb{U}$ from $\mathbb{K}$
while $\mathbb{U} == \emptyset$;
```

In Fig 7, the sanity rate of footprint$_{CA}$ and footprint$_{OR}$ are compared in all the three datasets for 1nn, 2nn, 4nn and various casebase sizes. We can observe that footprint$_{CA}$ has high sanity rate for all the results with k=2,4, and there is a significant difference in the sanity rate between footprint$_{CA}$ and footprint$_{OR}$ sets. At k=1 (single case adaptation), both the methods are performing similar which indicates that footprint$_{CA}$ is as good as footprint$_{OR}$ in the single case adaptation process.

We also check the sanity of the footprint sets by performing a reconstruction of noisy compression of the regression data using the footprint sets as a set of representative cases. In order to test the quality of the reconstruction of footprint sets, we performed a regression analysis where we used each footprint set as the the training data. The test data are the cases that belong to neither

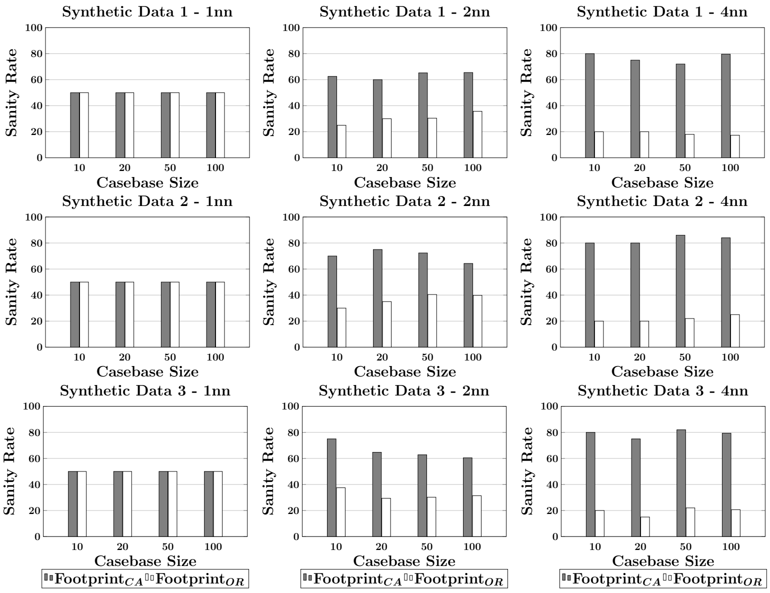

Fig. 7: Sanity Rate of footprint cases in Synthetic Datasets

the footprint$_{CA}$ set nor the footprint$_{OR}$ set. The reconstruction error (RE) is evaluated by the mean square error and we compared the reconstruction error obtained by both the training data. The comparison of results for all the three synthetic datasets are shown in Figure 8. The comparison is done based on the percentage of the difference between the reconstruction error received by the two training sets, with respect to the footprint$_{OR}$ set error. The comparison measure is given in Equation 6. As we compute the reduction with respect to the footprint set, a high error percentage indicates a significant improvement by the footprint$_{CA}$ set.

$$\text{Reduction w.r.t footprint}_{OR}\text{ RE} = \frac{\text{footprint}_{OR}\text{ RE} - \text{footprint}_{CA}\text{ RE}}{\text{footprint}_{OR}\text{ RE}} \times 100 \quad (6)$$

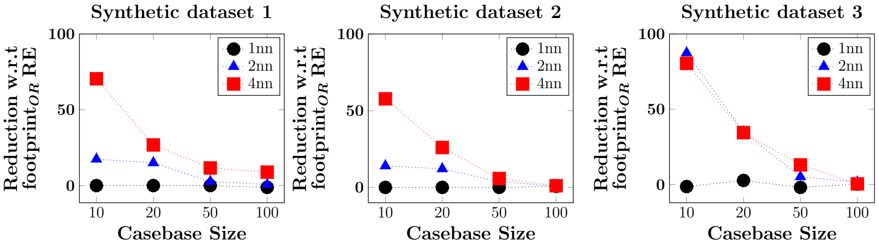

Fig. 8: Reconstruction Error (RE) Analysis

The k value for finding the neighbors is varied from 1 to 4. At $k = 1$, the reduction is close to zero which indicates footprint$_{CA}$ and footprint$_{OR}$ perform similarly in single case adaptation. For $k > 1$ and casebase size =10, we can observe a high reduction which signifies a notable improvement by the footprint$_{CA}$ set. This reveals the sanity of the retention score.

6 Footprint$_{CA}$ in Tutoring Application

Encyclopedic resources like Wikipedia and dictionary do not have rich pedagogical content, tailored to suit the users learning goals [7]. The concepts in Wikipedia (articles) as well as in dictionary (words) are not arranged in a learning order where as an ideal textbook explains a concept before referring it which results in a sequential order for learning [11]. So, sequencing the concepts in Wikipedia like resources may help the online learners to fulfill their learning goal. Each article in Wikipedia is explained in terms of other articles which, in turn explained using other articles. These articles are interconnected using hyperlinks. In the CBR perspective, Wikipedia articles are the cases and the concepts in Wikipedia that help in understanding a target concept are composed together to explain the target [7]. Hence, those set of cases acts as a composite solution of the target. The definition of a Wikipedia article can be approximated as the first sentence in the article [17]. So, the articles pointed to, by hyperlinks in the first sentence can be assumed as the concepts or cases that help in understanding the corresponding concept. We can construct a graph of Wikipedia casebase by marking these set of concepts as the cases that provide one composite solution for a Wikipedia article. In such graph, it is possible to adapt many composite solutions to explain a concept by using the transitivity property of the graph. This is because every Wikipedia concept is explained in terms of other concepts. Fig 9 illustrates an example of casebase graph constructed from English Wikipedia. Each node corresponds to Wikipedia articles. The Edges are drawn from the concepts in the first sentence of each article. For example, the concept *atom* is explained in terms of *chemical element* and *matter*. Hence, the arc between the edges from *chemical element* and *matter* to *atom* which forms an *AND* relation indicates that the cases *chemical element* and *matter* are composed together to explain *atom*.

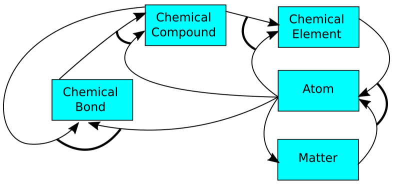

Fig. 9: An example of Casebase network from Wikipedia

Concepts	Retention Score
Atom	2.0
Matter	1.19
Chemical Element	1.18
Chemical Compound	1.12
Chemical Bond	1

Table 2: Retention score values

We can construct a casebase graph for a given topic, and our case competence model can identify a competent subset of concepts which covers the rest of the concepts in that topic. The retention score ordering implies the importance of each concept based on the extent to which the concept to be retained. A concept with high retention value is likely to be a basic concept as its coverage will be high due to its repetitive usage in defining other concepts. Thus, the ordering based on retention score provides an order in which one can learn the entire set of concepts under a specific topic.

The retention scores obtained for the Wikipedia concepts in the network shown in Fig 9 are given in Table 2. The footprint$_{CA}$ set for this example is obtained as $\{$*atom, chemical element, chemical compound*$\}$. This set can cover the entire casebase. The ordering of elements in the footprint$_{CA}$ set indicates the learning ordering where the position in the order implies the level of completion of learning. For example, let the learning goal be *Chemical Compound.* To satisfy the learning goal, one can learn the concepts in footprint$_{CA}$ in the retention score ordering. While learning each concept in the footprint$_{CA}$, the concepts that are solved by the elements in footprint$_{CA}$ can be learnt. Note that these concepts may not be present in the footprint$_{CA}$ set. A learner who is familiar with any of the concept in footprint$_{CA}$ can skip all the concepts that are positioned before this concept in the footprint$_{CA}$. This is because a concept subsumes all the previously present concepts in footprint$_{CA}$. Thus, footprint$_{CA}$ and the retention score ordering helps a learner to satisfy his/her goal.

6.1 Empirical Results

The effectiveness of retention score and footprint$_{CA}$ set is analyzed on the casebase extracted from the Wikipedia and dictionary. We extracted the articles in Wikipedia Artificial Intelligence (AI) category[2] and sub-categories up to three levels. The composed solution cases of each article are marked from the hyperlink articles that are present in the first sentence. This casebase (wikiAI) contains 6,536 cases. In the dictionary, concepts (cases) are the words that are defined in it and the content words in the definition are marked as the cases that are used for compositional adaptation to define a word. We make simplifying assumptions that the words in the dictionary are sense disambiguated. So, the content words present in the first definition of the first sense is considered as the composed solution of each word. Thus, we have taken definitions from the Longman dictionary of contemporary English (ldoce) and WordNet (wn). The graph constructed from this casebase results in an *AND-OR* graph due to the presence of multiple compositional solutions. Thus, we have 81,653 cases in the casebase. Similarly, other casebases are constructed using only WordNet (wn) and only Longman dictionary (ldoce). The wn casebase includes 79,582 cases and ldoce casebase contains 26,984 cases. All these four casebases are used for the analysis of retention score and footprint$_{CA}$ in tutoring application.

Casebase Coverage We analyzed the casebase coverage by the footprint$_{CA}$ set and footprint$_{OR}$ set in all the casebases. The footprint$_{CA}$ is observed as covering the full casebase whereas the footprint$_{OR}$ set does not cover the entire casebase due to the presence of *AND* composition. Thus, the entire dictionary words can be defined using the words in the footprint$_{CA}$. We analyzed the casebase

[2] `https://en.wikipedia.org/wiki/Category:Artificial_intelligence`

coverage by the footprint$_{OR}$ and this is shown in Fig 10. In all the casebases except wikiAI, the footprint$_{OR}$ set solves only less than 30% of the cases in the casebase. The higher coverage of footprint$_{OR}$ in wikiAI can be because of the less number of hyperlinks in the first sentence of each article which is considered as the cases in the composed solution.

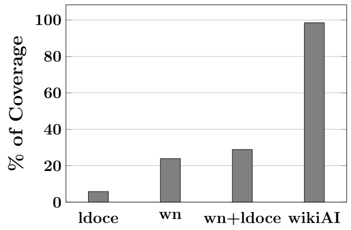

Fig. 10: Casebase Coverage of Footprint$_{OR}$

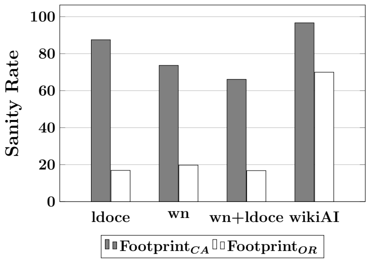

Fig. 11: Sanity Rate Analysis

Sanity Check The sanity of the footprint$_{CA}$ and footprint$_{OR}$ are analyzed using the sanity rate formulated in Section 5.2. The results are given in Fig 11. We can observe that the sanity rate of footprint$_{CA}$ cases in all the casebases are more than 65% and that of footprint$_{OR}$ cases are less than 20% except the wikiAI dataset which might be due to the lack of compositional information in the dataset. This indicates that the footprint$_{CA}$ set is useful for compositional adaptation applications.

7 Conclusion and Future Work

We start with the observation that the Smyth's footprint-based approach [15] is not designed for compositional adaptation applications. We proposed a measure called retention score to estimate the retention quality of a case that involves compositional adaptation. Using the retention score, we proposed a revised approach to identify the footprint$_{CA}$ set where compositional adaptation is required. We tested the effectiveness of the footprint$_{CA}$ using regression datasets and compared it with the Smyth's footprint set. The empirical results demonstrated the improved performance of our model when compositional adaptation is required; the proposed model performs equally well as Smyth's model during single case adaptation process. We also illustrated and tested the effectiveness of our method in a tutoring application which uses compositional adaptation.

The proposed retention score measure assumes that the compositional adaptation operator is a disjunction over conjunctions which makes a *hard-AND* relation between the cases that solves a problem using compositional adaptation. In some applications, the *soft-AND* relation might solve the problem. For example, the mean value of the solutions of the similar cases is taken as the composed solution for the target problem in applications such as pollution prediction in Aiquap CBR system [3]. The dropping of any of the similar cases might not affect the resulting solution. It would be interesting to introduce the softness in the retention score.

References

1. Arshadi, N., Badie, K.: A Compositional Approach to Solution Adaptation in Case-Based Reasoning and its Application to Tutoring Library. Proceedings of 8th German Workshop on Case-Based Reasoning, (2000)
2. Atzmueller, M., Baumeister, J., Puppe, F., Shi, W., Barnden, J. A.: Case-Based Approaches for Diagnosing Multiple Disorders. FLAIRS, 154–159, (2004)
3. Lekkas, G. P., Avouris, N. M., Viras, L. G.: Case-Based Reasoning in Environmental Monitoring Applications. Applied Artificial Intelligence An International Journal, 8(3), 359–376, (1994)
4. Lieber, J.: A Criterion of Comparison between Two Case Bases. Advances in Case-Based Reasoning: Second European Workshop, EWCBR, 87–100 (1994)
5. Markovitch, S., Scott, P.D.: Information Filtering: Selection Mechanisms in Learning Systems. Machine Learning, 10(2), 113–151 (1993)
6. Massé, A. B., Chicoisne, G., Gargouri, Y., Harnad, S., Picard, O., Marcotte, O.: How is Meaning Grounded in Dictionary Definitions?. Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing, 17–24, (2008)
7. Mathew, D., Eswaran, D., Chakraborti, S.: Towards Creating Pedagogic Views from Encyclopedic Resources. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, 190–195 (2015)
8. Müller, G., Bergmann, R.: Compositional Adaptation of Cooking Recipes using Workflow Streams. Computer Cooking Contest, Workshop Proceedings ICCBR, (2014)
9. Müller, G., Bergmann, R.: Workflow streams: A means for compositional adaptation in process-oriented CBR. Case-Based Reasoning Research and Development, 315-329 (2014)
10. Patterson, D., Rooney, N., and Galushka, M.,: A regression based adaptation strategy for case-based reasoning. Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, 87–92 (2012)
11. Agrawal, R., Chakraborty, S., Gollapudi, S., Kannan A., Kenthapadi K.: Quality of Textbooks: An Empirical Study. ACM Symposium on Computing for Development, (2012)
12. Reinartz, T., Ioannis I.,Thomas R.: Review and Restore for CaseBase Maintenance. Computational Intelligence, 17.2, 214–234 (2001)
13. Smyth, B., Keane, M.T.: Remembering to Forget. In Proceedings of the 14th international joint conference on Artificial intelligence (IJCAI), 377–382 (1995)
14. Smyth, B., McKenna, E.: Modelling the Competence of Casebases. In Advances in Case-Based Reasoning, 208–220 (1998)
15. Smyth, B., McKenna, E.: Footprint-based retrieval. In Case-Based Reasoning Research and Development, 343–357 (1999)
16. Wilke, W., Bergmann, R.: Techniques and Knowledge used for Adaptation during Case-Based Problem Solving. In Tasks and Methods in Applied Artificial Intelligence, 497–506 (1998)
17. Ye, S., Chua, T., Lu, J.: Summarizing Definition from Wikipedia. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on NLP of the AFNLP, 199–207, (2009)