Guiding Retrieval of Textual Cases with Reuse

Adeyanju I.A., Omidiora E.O.
Department of Computer Science and Engineering
Ladoke Akintola University of Technology
Ogbomoso, Nigeria
iaadeyanju@lautech.edu.ng

Wiratunga N., Lothian R.
School of Computing Science and Digital Media
The Robert Gordon University
Aberdeen, United Kingdom

***Abstract*— Textual case-based reasoning (TCBR) solves new problems by reusing previous similar problem-solving experiences documented as text. During reuse, TCBR identifies reusable textual constructs in the retrieved solution content and differentiates from the rest that need revision. However, reuse is heavily influenced by the quality of retrieval since TCBR attempts to adapt retrieved cases to solve a new problem. In scenarios where only the most similar case is adapted during reuse, such best match case might not necessarily be the easiest to adapt. We introduce a technique called Reuse Guided Retrieval to determine a specific similar case whose solution is best adaptable to solve a new query. A reuse metric is also proposed which encodes how easily reusable or adaptable the solution from a particular nearest neighbour is to a query. Experiments on two datasets from the domains of weather forecast revision and health & safety incident reporting indicate that our technique was more effective than a baseline retrieval which always chooses its best match using only the retrieval similarity between the cases and a query.**

***Keywords*—text reuse, text retrieval, case based reasoning**

I. INTRODUCTION

Textual Case Based Reasoning (TCBR) solves new problems by reusing previous similar problem-solving experiences documented as text. TCBR is a subfield of case based reasoning but has evolved as a specialized research area due to challenges associated with reasoning with textual attributes [1] as opposed to structured attributes consisting of numeric and symbolic values. text reuse is highly dependent on the quality of the retrieved similar case since TCBR attempts to adapt retrieved cases to solve a new problem. in scenarios where only the most similar case is adapted during reuse, such best match case might not necessarily be the easiest to adapt. this is because most retrieval mechanisms focus solely on the problem space while reuse is done within the solution space. the similarity of cases in the problem space is typically not the same as their similarity in the solution space. in other words, two cases most similar to each other with respect to their problems might have a third case similar to both but whose solution is more similar to one of the cases than their solutions are to each other.

We introduce a technique called Reuse Guided Retrieval to determine a specific similar case whose solution is best adaptable to solve a new query. This is done by formulating a reuse metric which encodes how easily reusable or adaptable the solution from a particular nearest neighbour is to a query. This should generally increase the overall effectiveness of the TCBR system since the new proposed solution will be similar to the actual solution and also easiest to adapt. Section II discusses our related work while our reuse guided retrieval technique is explained in Section III. Experimental setup, evaluation and discussion of results appear in Section IV followed by conclusion in Section V.

II. RELATED WORK

A textual case has at least one of its attributes (either problem, solution or both components) in free text form. The retrieval stage of the problem-solving cycle deals mainly with the problem component of cases to determine their similarity to a new problem while reuse deals with the solution components. Text reuse is applicable when the solution is in free text form. Verbatim Reuse is the most common form of text reuse and previous TCBR systems that fall under these category includes Experience Book [2], DRAMA [3], and SMILE [1]. With these systems, indexing and retrieval mechanism encode most of the semantics in the textual cases, so that retrieved cases are semantically similar, but give no assistance on how to adapt the retrieved solution.

Substitutional reuse has been applied extensively to textual cases especially when minimal adaptation is required. This involves the identification of specific terms in a retrieved textual solution and proposing suitable modifications due to observed differences between the query and retrieved problem. This approach was used for a substitution based adaptation in some TCBR applications that deal with modification of ingredients to recommend recipes that satisfy a user query [4,5]. It was also used to make suggestions for named entities substitution in a TCBR application for automated email response [6,7]. Structural reuse in the context of TCBR involves proposing suggestions for adaptation to a textual solution that goes beyond substitution. In other words, there might be suggestions to delete and/or insert terms into specific sections of a retrieved textual solution without necessarily replacing other sections thereby changing the overall structure of the solution. More sophisticated strategies might also consider the impact of these structural changes to other solution parts. Structural text reuse were also proposed for report writing applied to air travel incidents investigation [8], a semi-automated email response application [7] and aiding text reuse by suggesting features and values or phrases during authoring of a product description for trading or its review after purchase [9,10].

III. REUSE GUIDED RETRIEVAL

The basic idea in reuse guided retrieval (RGR) is to determine the best match to a query using not just its retrieval similarity in the problem space but also how much of its solution can be reused without adaptation. This should generally increase the overall effectiveness of the TCBR system since the new proposed solution will be similar to the actual solution but also easiest to adapt. In other words, retrieval performance is improved by determining a case in the query's neighborhood whose solution is easiest to adapt. Our best match case in this scenario can be a case other than the nearest neighbour (1NN). We propose a new metric which determines the utility of a retrieved case for solving a query by combining the reusable proportion of its solution and its similarity to the query. Intuitively, this metric assigns a high score to a retrieved case whose problem is very similar to the query and whose solution can be reused with very little adaptation. We leverage a reuse architecture called Case Retrieval Reuse Net (CR2N) [11,12] to determine what proportion of the solution of a query's neighbour is reusable. Figure 1 lists the algorithm for reuse guided retrieval.

The algorithm uses function getReuseUtilityScore which returns the reuse utility score for any case in the query's specified neighbourhood. We show the pseudo codes for this function in Figure 2 separately to allow an explanation of how it incorporates the CR2N technique. The parameters used in the algorithm are casebase (CB), problem/solution vocabulary (Vp/Vs), query (Q), similarity threshold to assess a case solution as containing a similar term (attribute) to a retrieved solution term (σ), size of the query's neighbourhood (ret_k) and the reuse neighbourhood size (rs_k) which can be tuned to obtain the best performance. We expect the best value for rs_ k to be identical or very similar to the optimal k-value obtained from CR2N empirical evaluations in any domain. Other parameters are weight of the retrieval similarity (α) and reuse proportion (β) in reuse utility score. We expect ret_k to be a very small value, typically less than 10. This is because cases ranked lower during retrieval are likely to be less similar and therefore less reusable except the query belongs to a densely-populated cluster with very close similarity values between the query and several cases in its neighbourhood. The RET function retrieves cases given a partial case description and an indexing vocabulary while SelectK returns top k cases.

```
Require: CB= {C_1, ..., C_n}, set of cases in the case base
Require: V_p= {pie_1, ..., pie_m}, set of problem IEs in CB
Require: V_s= {sie_1,...,sie_l}, set of solution IEs in CB
Require: C= {P, S}, where (C ∈ CB) ∧ (P ⊂ V_p) ∧ (S ⊂ V_s)
Require: Q= a query, where Q ⊂ V_p
Require: ret_k= query's retrieval neighbourhood
Require: rs_k= optimal reuse neighbourhood size
    <!- based on empirical evaluation of CR2N on best match case ->
Require: σ= similarity threshold between a retrieved solution attribute and other solu
    tions
Require: α= weight on retrieval similarity in reuse utility score
Require: β= weight on reuse proportion in reuse utility score
    <!- where α + β = 1 ->
CB_local ← SelectK(RET(V_p, Q), ret_k)
Initialise RM ← {rm_1, ..., rm_ret_k}, reuse utility score for each retrieved case
for each (C_i ∈ CB_local) do
    rm_i ← getReuseUtilityScore(C_i,Q,V_p,V_s,rs_k,σ,α,β)
end for
max= getMaxValue(RM)
index= getIndex(max, RM)
return C_index as C_best, where C_index ∈ CB_local
```

Figure 1. Reuse guided retrieval algorithm

```
Input: V_p= {pie_1,...,pie_m}, set of problem IEs in the casebase
Input: V_s= {sie_1,...,sie_l}, set of solution IEs in the casebase
Input: Q= a query, where Q ⊂ V_p
Input: C_nn= {P, S}, where(P ⊂ V_p) ∧ (S ⊂ V_s)
    <!- A case in the nearest neighbourhood of Q ->
Input: rs_k= reuse neighbourhood size (optimal)
Input: σ= similarity threshold between a retrieved solution attribute and other solutions
Input: α= weight on similarity between Q and C_nn
Input: β= weight on reuse proportion in C_nn, where α + β = 1
Output: Proportion of reusable terms in C_nn
rc ← 0, counter for number of reusable terms in solution of C_nn
RS_1 ← SelectK(RET(V_s, C_nn), rs_k)
for each ({sie_i} ∈ C_nn) do
    RS_2 ← SelectT(RET(V_s, {sie_i}), σ)
    AS ← RS_1 ∩ RS_2
    BS ← RS_1 \ RS_2
    S̄_A ← (1/|AS|) Σ_{a∈AS} SIM(a, Q)
    S̄_B ← (1/|BS|) Σ_{b∈BS} SIM(b, Q)
    if S̄_A ≥ S̄_B then
        rc ← rc+1, [REUSE attribute {sie_i}]
    end if
end for
size ← getSizeSolutionTerms(C_nn), total number of terms in solution of C_nn
ret_sim ← SIM(C_nn, Q)
rs_prop ← rc ÷ size, reuse proportion
score ← α * ret_sim + β * rs_prop
return score
```

Figure 2. Function getReuseUtilityScore

The reuse guided retrieval algorithm begins in line 1 with the retrieval of nearest neighbors of the query in the problem space using the RET function which incorporates the similarity metric. The top ret_k neighbors (CB_{local}) are then selected by the SelectK function from which one of them will be assessed to be the best match. The remainder of the algorithm on Lines 2-8 calculates a reuse utility score for each case from these top neighbors and selects the case with the highest reuse utility score as the best match. It should be noted that the case returned as the best match might be the same as the retrieval best match (1NN) if the other neighbors are not found to have a better utility score.

The reuse utility score is calculated as a weighted average of retrieval similarity and reuse proportion values as shown in Figure 2. Lines 2-12 of function *getReuseUtilityScore* computes the number of reusable terms in the solution of a given nearest neighbour (C_{nn})of the query. Although we used the absolute values of the reuse proportion in our utility score computation, other functions such as binary logarithm as used in information entropy [13,14] might also be used to minimize the effects of the solution length (size in the algorithm on Figure 2) on the computed score. One advantage of absolute reuse proportion value is that it ensures that the cost of inserting new terms during reuse/ adaptation is higher than the cost of deleting terms in the proposed solution. This is intuitive because deleting terms from a piece of text is generally easier (less costly) than adding other new terms when modifying the text. For instance, if there were two solutions with 3 and 4 terms out of which 2 and 3 terms were determined to be reusable respectively. These will give reuse proportion values of 0.67 (2/3) and 0.75 (3/4) while logarithmic equivalent will be 0.578 ($\log_2 2/3$) and 0.415 ($\log_2 3/4$). But the solution with 3 reusable out of 4 terms will typically be preferred since the cost of deleting a term from this solution will be less than adding a term to the other solution text with 2 out of 3 terms.

IV. Experimental Evaluation

Our reuse guided retrieval (RGR) technique described in Section III is evaluated for its effectiveness. This is done by examining the average retrieval effectiveness of the RGR technique relative to a retrieve-only system which simply chooses the best match based solely on retrieval similarity; this baseline is denoted as CRN. In other words, we compare the best match solutions proposed by RGR and CRN to an actual solution. There are five parameters to be tuned to obtain an optimal performance according to the RGR algorithm listed in Figure 1. These are the query's retrieval neighbourhood (ret_k), optimal reuse neighbourhood size (rs_k), similarity threshold between a retrieved solution term and other solutions (σ), and weights on retrieval similarity and reuse proportion in the reuse utility score (α and β respectively). We chose ret_k=3 based on analysis of empirical experiments on other values such as ret_k=5,7 which showed that less than 2% of cases computed as best match by our technique were originally ranked below third position using only retrieval similarity. For instance, on the weather forecast dataset, only 41 out of 2414 cases (1.7%) in a cross validation experiment was adjudged as best match by RGR (with equal weightings for α and β, σ=0 and rs_k=3) were originally ranked below third position while 372 cases (15%) were ranked second or third. The value for rs_k is chosen to vary increasingly up to the size of the casebase. This allows us to investigate if the optimal neighbourhood size for the CR2N gives the best retrieval and reuse effectiveness for RGR. σ is given a value of 0 since we used the CR2N at the keyword level for our experimental domains and any neighbour's solution containing a term annotated from the retrieved solution will have a similarity greater than zero. For the weights in the reuse utility score (α and β), we used values α= 0.25, 0.5 & 0.75 where β=1−α to measure the effect of different weighting schemes. Note that α=1 is the same as CRN since only retrieval similarity is used for determining the best match. We therefore compared the following algorithms.

1. CRN which uses only retrieval similarity in determining the best match, as baseline.

2. RGR, explained in Section III. Three versions of this algorithm were tested by varying the retrieval weights (α= 0.25, 0.5 & 0.75) in the reuse utility score.

We used a ten-fold cross validation experimental design with cosine coefficient for similarity computation at both retrieval and reuse stages. Each information entity (IE) in the CR2N represents a keyword from our domain vocabulary because the size of each retrieved solution text in our application domain is small. The problem and solution texts are parsed into keywords and suitable stop words are also removed and keywords stemmed to cater for morphological variations except stated otherwise. We analyse evaluation results from our empirical experiments on datasets from weather forecast revision and medical health and safety incident reports . Test of significance on all our evaluation results is done at 95% confidence with a non-parametric method (Kruskal-Wallis test) since their deviation from the normal distribution is significant, that is p-value<0.05. Evaluation result values are shown in tables with bold font (apart from the column titles) indicating that a value is statistically better at 95% confidence than others in the same column. An italicized result value in a table shows that it is the highest but not significantly while an underlined value is significantly worse than others.

A. Weather Forecast Revision

The wind dataset was extracted from a post-edit corpus [15,16] of an NLG weather forecast system called Sumtime Mousam (SM). The dataset consists of weather forecast text generated from numerical data by SM and its edited form after revision by domain experts. A case in our experiments therefore consists of the NLG system generated text (Unedited Text) as problem and its revised form by domain experts (Edited text) as solution. The expectation is that similar edit operations are applicable on solution texts. The indexing vocabulary is small i.e. 71/140 keywords in problem/solution vocabulary respectively. A total of 2414 cases (from 14690) were extracted for experiments and we ensured that the average size of problem/solution text is about 1 sentence since the reuse techniques were tested at keyword granularity.

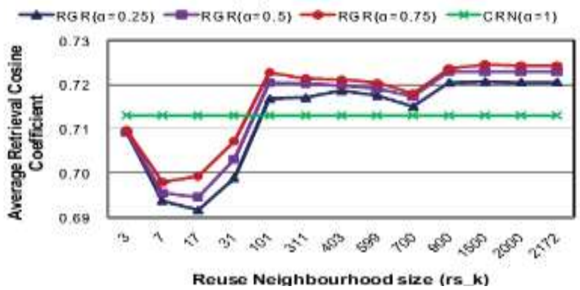

Figure 3. Evaluation results for RGR in weather forecast revision

Figure 3 shows the evaluation results of RGR on our weather forecast dataset using cosine coefficient similarity between the retrieved and actual textual solutions. The results for varying weights of the retrieval similarity in the utility score, α, are also shown with the average retrieval cosine coefficient plotted against reuse neighbourhood sizes. Cosine coefficient values are also listed in Table 1; values for k= 7, 17 and 2000 are not shown due to space limitations but can be approximated from the graph. The average cosine coefficient values of RGR with different weightings are better than the baseline CRN at our reuse strategy's (CR2N) optimal reuse neighbourhood size of 101 and above. In Figure 3, the cosine curves of RGR at α= 0.25, 0.5 and 0.75 are all above the CRN curve from k= 101. CRN's cosine values from this point are also significantly worse than RGR's across the different neighbourhood sizes as shown in Table I. With increasing reuse neighbourhood size, the average cosine values for RGR generally increases and tends to a constant value once most of the casebase is being used for reuse evidence computation. The marginal decrease on RGR's curves at rs_k = 7, 17, 31 and 700 can be due to reuse annotation errors from CR2N which can lead to falsely computing a high reuse proportion. Nevertheless, this should not affect RGR's performance since these rs_k values are not optimal for CR2N's effectiveness.

Table I. Cosine evalution results in weather forecast revision

	3	31	101	311	403	599	700	900	1500	2172
α = 0.25	0.709	0.699	0.717	0.717	0.719	0.718	0.715	0.721	0.721	0.721
α = 0.5	0.709	0.703	0.721	0.720	0.720	0.719	0.717	0.723	0.723	0.723
α = 0.75	0.710	0.707	*0.723*	*0.722*	*0.721*	*0.720*	*0.718*	*0.724*	*0.725*	*0.724*
CRN (α = 1)	*0.713*	*0.713*	0.713	0.713	0.713	0.713	0.713	0.713	0.713	0.713

Comparison of results across RGR's different weight parameters suggests that retrieval similarity should be weighted more than reuse in the utility score computation. This is because the curve for α=0.75 is above that of α=0.5 which in turn is above α=0.25. The difference in performance between these weighting schemes are generally marginal and not statistically significant. Therefore, assigning equal weights to the retrieval similarity and reuse proportion in RGR's utility score should provide a comparable performance without the need to experiment with different values for α and β.

B. Health and Safety Incident Reporting

We also evaluated RGR on health and safety (H&S) incident reports from hospitals provided by NHS Grampian. A report consists of a textual description of an incident and action taken by the health personnel on duty. Each record is also labelled with 1 of 17 care stage codes which identifies a group of records such as accidents that result in personal injuries, incidents during treatment etc. The intention is to build a TCBR system that assists less experienced health personnels when resolving/recording incidents by using previous similar experiences. Therefore, incident description serves as our problem while the solution is the action taken to resolve the incident for each case in our experiments.

We extracted a total of 362 cases that were grouped under a similar care stage code and having just 1 sentence in both the problem and solution texts. This allows us not only to evaluate our reuse technique at keyword granularity but makes it comparable to results from the weather domain. During evaluation, synonym keywords were matched using WordNet [17] as well as keywords with the same lemma but different stems (e.g gave and given). The same evaluation methodology as Section IV(A) is used and CR2N was used for reuse annotation at keyword granularity level since average size of a problem/solution text is small and to make the results comparable to those obtained from our RGR experiments on the weather forecast revision dataset.

Evaluation results of the RGR technique on our H&S incidents dataset with different cosine similarity is shown in Figure 4 for varying weights of the retrieval similarity (α). Average cosine coefficient values are also given in Table II to indicate significant differences where applicable. Cosine coefficient average values of RGR are better than baseline CRN for most reuse neighbourhood sizes in Figure 4 and Table II. RGR's cosine curves show an unstable start with peaks and troughs but stabilises into a near-constant value once the neighbourhood size includes most of the casebase. These might be due to the fact that our evaluation measure is unable to capture sentence-level variation across texts. The initial increase and decrease can be attributed to the reuse strategy's (CR2N) effectiveness not being optimal at these neighborhood sizes.

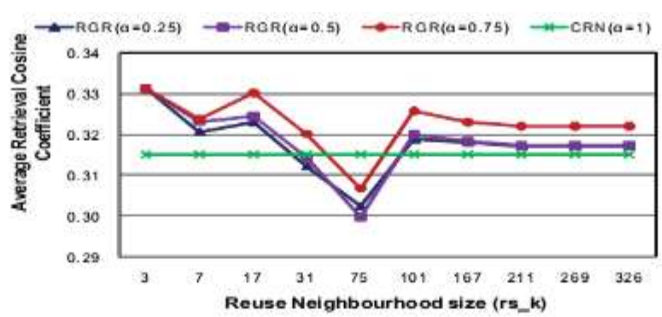

Figure 4. Evaluation results for RGR in H&S incident reporting

Table II. Cosine coefficient results in H&S Incident Reporting

	3	7	17	31	75	101	167	211	269	326
$\alpha = 0.25$	*0.331*	0.321	0.323	0.312	0.302	0.319	0.318	0.317	0.317	0.317
$\alpha = 0.5$	*0.331*	0.323	*0.325*	0.314	0.300	0.320	0.318	0.317	0.317	0.317
$\alpha = 0.75$	*0.331*	*0.324*	0.330	*0.320*	0.307	*0.326*	*0.323*	*0.322*	*0.322*	*0.322*
CRN ($\alpha = 1$)	0.315	0.315	0.315	0.315	*0.315*	0.315	0.315	0.315	0.315	0.315

Comparison of values across the RGR cosine curves for different weightings indicates that α=0.75 is more effective than α=0.5 which is comparable to α=0.25. Cosine results do indicate that on average, reuse text from RGR will be more similar to the actual text than those from the CRN.

C. Further Discussion on Reuse Guided Retrieval

Results from experiments on two domains indicates that RGR improves the quality of retrieval and reuse. RGR can also be viewed as a retrieval framework which can utilize different reuse strategies. This allows for choosing a reuse strategy that performs best on a particular domain for use within the RGR framework to achieve good retrieval performance. An important observation across both domains is that retrieval similarity should be weighted higher than the counter reuse as this was most effective. However, we also note that equal weights should be sufficient, when repeated experiments to determine the optimal weights is costly. This is because the differences in performance for the various RGR weight configurations used in our experiments were marginal and statistically insignificant.

The medical health and safety incident reporting domain proved to be more difficult when compared to weather forecast. Low result values in the medical domain were mainly due to solution texts having significant variation in terminology usage and writing styles (concise vs. verbose). This led to having several cases with identical/nearly identical problems but whose solutions though semantically similar were syntactically dissimilar. Unfortunately, the automated evaluation metric used in our experiments were unable to capture these semantic variation.

I. Conclusion And Future Work

This paper proposes a reuse guided retrieval (RGR) mechanism which ranks cases based on a utility score which is a weighted combination of the retrieval similarity and reuse proportion. This is because though similar problems should have similar solutions, it is sometimes possible that the solution of the most similar case to a query is not the easiest to adapt during reuse. The proposed technique determines the ease of adaptability of a retrieved textual solution. Our intuition is that the easiest solution to adapt will also have the highest number of terms annotated as reuse and the least number of terms as adapt. The reuse proportion (which is a quotient of the number of terms annotated as reuse and the total number of terms) aptly captures this intuition.

Result from empirical experiments supports RGR as its performance surpassed a baseline standard retrieval system which always chooses its best match using only the retrieval similarity between the cases and a query when tested on two datasets. We also discovered that RGR's performance is best when retrieval similarity is weighted higher than reuse proportion by the reuse utility function.

We intend to experiment with RGR at other levels of text granularity such as phrases and sentences. A qualitative evaluation (human validation) of our technique is needed to address problems encountered with quantitative evaluation on the health and safety incident report.

Acknowledgment

We are grateful to Dr. Somayajulu Sripada and Prof. Ehud Reiter of The University of Aberdeen, UK for providing us the Weather forecast revision datasets. We would also like to thank the Grampian National Health Scheme (NHS) board in Aberdeen, UK for entrusting us with the anonymous Health and Safety incident reports.

References

[1] S. Bruninghaus and K.D. Ashley, "Reasoning with textual cases". Proceedings of ICCBR'05, Springer , Berlin, 2005, pp. 137–151

[2] M. Kunze and A. Hubner, "CBR on Semi-structured Documents: The ExperienceBook and the FAllQ Project", Proceedings of the Sixth German Workshop on Case-Based Reasoning, 1998, pp. 77–85

[3] D. Wilson, "CBR Textuality", Expert Update, 2000, vol. 3, pp. 28–37

[4] I. Adeyanju, S. Craw, A. Ghose, A. Gray and N. Wiratunga, "RaGoUt: : An Arpeggio of Tastes", ECCBR'08 workshop Proceedings, Tharax-Verlag, Hildesheim, 2008, pp. 229–238

[5] J. DeMiguel, E. Plaza, and B. Dıaz-Agudo, "ColibriCook: A CBR system for ontology-based recipe retrieval and adaptation", ECCBR 2008 Workshop Proceedings, Tharax Verlag, 2008, pp. 199–208

[6] L. Lamontagne and G. Lapalme, "Applying case-based reasoning to email response", ICEIS '03 Proceedings, 2003, pp. 115–123

[7] L. Lamontagne and G. Lapalme, "Textual reuse for email response", ECCBR 2004, LNAI 3155, Springer, Heidelberg, 2004, pp. 234–246.

[8] J.A. Recio-Garcıa, B. Dıaz-Agudo and P.A. Gonzalez-Calero, "Textual CBR in jCOLIBRI: From retrieval to reuse" ICCBR'07 workshop proceedings (Textual CBR), 2007, pp. 217–226

[9] D. Bridge and A. Waugh, "Using experience on the read/write web: The GhostWriter system", ICCBR Workshop Proc., 2009, pp. 15–24

[10] P. Healy and D. Bridge, "The GhostWriter-2.0 System: Creating a Virtuous Circle in Web 2.0 Product Reviewing", ICCBR'10 Workshop Proceedings, 2010, pp. 121–130

[11] I. Adeyanju, N. Wiratunga, R. Lothian, S. Sripada and L. Lamontagne, "Case Retrieval Reuse Net (CR2N): An Architecture for Reuse of Textual Solutions", Proceedings of ICCBR'09, LNAI 5650, Springer-Verlag, Berlin Heidelberg, 2009, pp. 14-28

[12] I. Adeyanju, N. Wiratunga, R. Lothian, S. Sripada and S. Craw, "Solution reuse for textual cases", Proceedings of UKCBR Workshop, CMS Press, Greenwich, 2008, pp. 54-62

[13] R.B. Ash, Information Theory, Dover Publications, 1990

[14] C. E. Shannon, "A mathematical theory of communication", Bell System Technical Journal, 1948, vol. 27, pp.379–423

[15] S.G. Sripada, E. Reiter, J. Hunter and J. Yu, "SUMTIME-METEO: Parallel Corpus of Naturally Occurring Forecast Texts and Weather Data",Technical Report AUCS/TR0201, Department of Computer Science, University of Aberdeen, 2002

[16] S. Sripada, E. Reiter and L. Hawizy, "Evaluation of an NLG system using post-edit data: Lessons learnt", Proceedings of European Natural Language Generation Workshop, 2005, pp. 133–139

[17] C. Fellbaum (editor), ".WordNet: An Electronic Lexical Database", MIT press, 1998