ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010
© 2010 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-60750-606-5-777

Learning to Author Text with textual CBR

Ibrahim Adeyanju[1] and **Nirmalie Wiratunga** and **Juan A. Recio-García**[2] and **Robert Lothian**

Abstract. Textual reuse is an integral part of textual case-based reasoning (TCBR) which deals with solving new problems by reusing previous similar problem-solving experiences documented as text. We investigate the role of text reuse for text authoring applications that involve feedback or review generation. Generally providing feedback in the form of assigning a rating from a likert scale is far easier compared to articulating explanatory feedback as text. When previous feedback generated about the same or similar objects are maintained as cases, there is opportunity for knowledge reuse. In this paper, we show how compositional and transformational adaptation techniques can be applied once sentences in a given case are aligned to relevant structured attribute values. Three text reuse algorithms are introduced and evaluated on a dataset gathered from online Hotel reviews from TripAdvisor. Here cases consists of both structured sub-rating attributes together with textual feedback. Generally, aligned sentences linked to similar sub-rating values are clustered together and prototypical sentences are then extracted to enable reuse across similar authors. Experiments show a close similarity between our proposed texts and actual human edited review text. We also found that problems with variability in vocabulary are best addressed when prototypes are formulated from larger sets of similar sentences in contrast to smaller sets from local neighbourhoods.

1 Introduction

The task of authoring documents that include pre-defined attributes along with some textual content is common to several domains. Such documents include reviews, student feedback, medical notes and incident reports. Review of products and services is one of such web applications where authoring is increasingly being encouraged by e-commerce websites. This is very useful for both the manufacturers/service providers to improve their products/services and the customer to make informed choices. Review typically consist of pre-defined attributes which authors can rate on a likert scale. For example, a customer reviewing a hotel visited recently might be asked to rate the cleanliness and service enjoyed on a scale of 1 to 5 where 1 is *terrible* and 5 is *excellent*. Another component of such reviews is a free text section where authors can explain their ratings and elaborate further. However, they are sometimes reluctant to write free text especially a comprehensive one since it takes more time to put their thoughts into writing.

Textual case base reasoning (TCBR) [15] is a research area that deals with solving new problems by reusing previous similar experiences documented as text. Text reuse is an integral part of TCBR and is not only helpful in solving a new similar problem but can assist in authoring new experiences. TCBR is particularly suited to support authoring of textual contents because it can propose useful initial text from previous reviews that are similarly rated. Reusing previous textual contents is challenging as it is difficult to know sections in the text that are associated with structured attributes corresponding to the set of ratings. It is also important to avoid irrelevant verbose details that are not easily reusable across several authors.

Our focus in this paper is to assist authors to write better and more comprehensive reviews by proposing useful text which they can easily edit to taste. We propose two novel mechanisms to align rated attributes to review sentences and abstract a group of similar sentences into a prototypical sentence. These mechanisms led to the development of three text reuse techniques which differ mainly in terms of how similar case(s) are retrieved, what sentence(s) are used from these cases and whether such sentences are global or local prototypes. Our hypothesis is that each of these techniques will generate useful initial text but one of them might significantly outperform the others. We evaluate these algorithms on hotel reviews dataset and our results show a close similarity between the proposed and actual review texts. Algorithms presented in this paper have the advantage of being domain-independent and so are applicable in domains containing cases with both pre-defined structured attributes and complementary textual content.

An overview of our domain of application is given in Section 2 followed by details of our alignment approach and text reuse techniques in Sections 3 and 4. Experimental setup and discussion of results appear in Section 5 with related work in text reuse in Section 6. We conclude with the contributions of this work in Section 7.

2 Hotel Reviews Domain

User generated experiential content is readily available on the world wide web in the form of blogs, forum posts, reviews and other social applications. This provides an opportunity to reuse these experiences [12] for similar web related tasks such as search and browse, review generation and other forms of problem solving. However, reuse will only make sense if there are several experiences authored about similar/same objects (or problems). Hotel reviews are particularly useful in this regard because several reviews are available for the same or indeed similar hotels. Each review typically has some attributes rated on a likert scale and a complementary text. Hotel reviews are generally suitable for text reuse as the assumption is that authors with similar ratings will use similar explanatory feedback text. However, such review texts are prone to grammatical errors since authors rarely use spell checkers. They also contain a lot of verbose details that might not be related to hotels since unedited reviews are uploaded.

We downloaded several reviews from a hotel recommender website [3] where each review is written by an author who visited a hotel

[1] School of Computing,Robert Gordon University, Aberdeen, Scotland, UK, email: [iaa|nw|rml]@comp.rgu.ac.uk

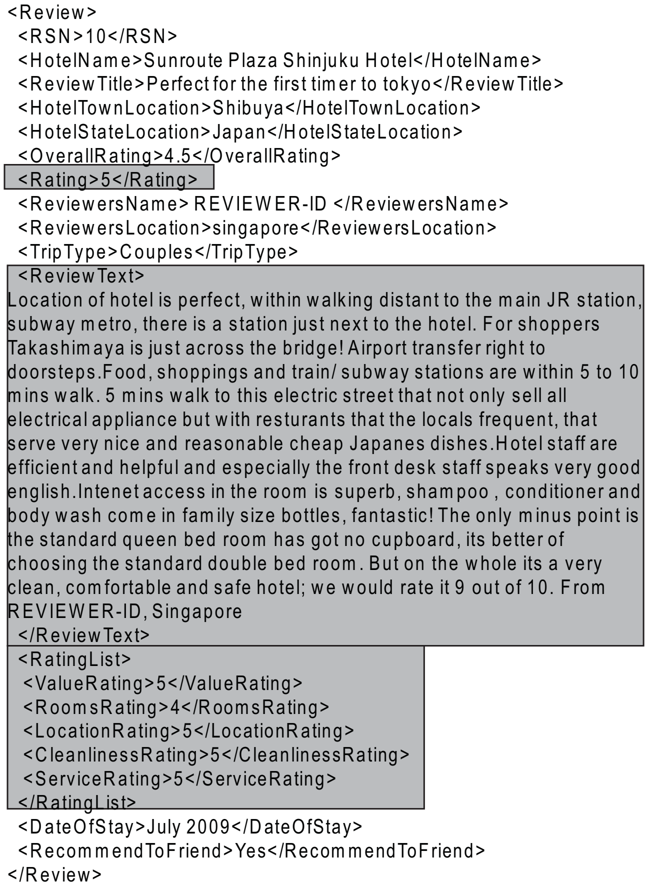
[2] Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Madrid, Spain, email: jareciog@fdi.ucm.es

[3] www.tripadvisor.co.uk

1. Hotel name	2. Hotel town	3. Hotel country (or US state)
4. Overall rating	5. Review Title	6. Author **rating**(up to 5 stars)
7. Author ID	8. Author location	9. Trip type (solo, couple etc)
10. **Review text**	11. Date of stay	12. Recommend to friend(y/n)
13. **Sub-ratings** for value, room, location, cleanliness & service		

Table 1. Complete list of possible attributes extracted for each hotel review

and presents her opinion of the place. The 13 attributes shown in Table 1 were extracted for each review; however, some of these attributes were absent in some reviews. 39, 870 reviews were extracted from our downloads cutting across 6, 564 hotels in 104 different countries (or states in USA). Based on an analysis of the corpus, we discovered that the downloaded corpus contained a small number of reviews (< 50) per hotel or author. The overall rating of a hotel is also an average of authors' ratings and not those given by regulators such as ISO (International Organization for Standardization). It is therefore more intuitive to reuse similar reviews across all hotels.



```
<Review>
 <RSN>10</RSN>
 <HotelName>Sunroute Plaza Shinjuku Hotel</HotelName>
 <ReviewTitle>Perfect for the first timer to tokyo</ReviewTitle>
 <HotelTownLocation>Shibuya</HotelTownLocation>
 <HotelStateLocation>Japan</HotelStateLocation>
 <OverallRating>4.5</OverallRating>
 <Rating>5</Rating>
 <ReviewersName> REVIEWER-ID </ReviewersName>
 <ReviewersLocation>singapore</ReviewersLocation>
 <TripType>Couples</TripType>
 <ReviewText>
Location of hotel is perfect, within walking distant to the main JR station,
subway metro, there is a station just next to the hotel. For shoppers
Takashimaya is just across the bridge! Airport transfer right to
doorsteps.Food, shoppings and train/ subway stations are within 5 to 10
mins walk. 5 mins walk to this electric street that not only sell all
electrical appliance but with resturants that the locals frequent, that
serve very nice and reasonable cheap Japanes dishes.Hotel staff are
efficient and helpful and especially the front desk staff speaks very good
english.Intenet access in the room is superb, shampoo , conditioner and
body wash come in family size bottles, fantastic! The only minus point is
the standard queen bed room has got no cupboard, its better of
choosing the standard double bed room. But on the whole its a very
clean, comfortable and safe hotel; we would rate it 9 out of 10. From
REVIEWER-ID, Singapore
 </ReviewText>
 <RatingList>
  <ValueRating>5</ValueRating>
  <RoomsRating>4</RoomsRating>
  <LocationRating>5</LocationRating>
  <CleanlinessRating>5</CleanlinessRating>
  <ServiceRating>5</ServiceRating>
 </RatingList>
 <DateOfStay>July 2009</DateOfStay>
 <RecommendToFriend>Yes</RecommendToFriend>
</Review>
```

Figure 1. Example of a hotel review from tripadvisor.co.uk

Another finding from the corpus analysis is that the rating and sub-rating attributes have the greatest effect on the contents of a review text because most authors enter values for these attributes. We therefore limit our structured attributes to rating for the hotel and sub-ratings for *cleanliness*, *location*, *rooms*, *service* and *value* so that review texts can be reused across a wider range of authors. These attributes are completed on a likert scale of terrible (1), poor (2), average (3), very good (4) and excellent (5). An example review is shown in Figure 1. The author's ID is anonymised due to privacy issues and highlighted portions relate to attributes used in our experiments. The review text shown is typical where authors write not just content in relation to their ratings but also elaborate on associated concepts that contributed towards the overall experience such as local restaurants.

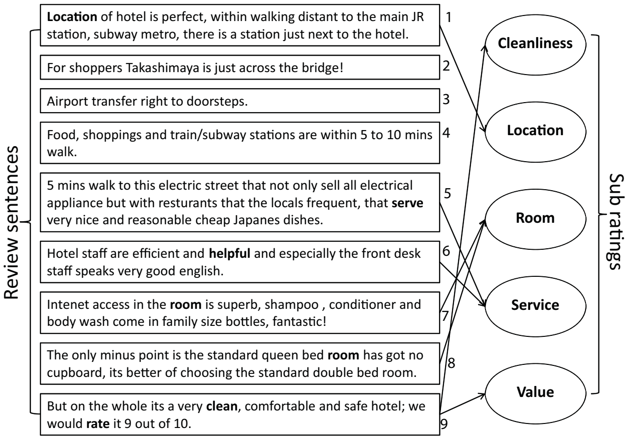
3 Text Alignment

A CBR case typically consists of two components: a problem and its solution. When both have multiple attributes, each solution attribute might depend on a specific combination of problem attributes rather than all. Knowledge of such problem-solution attribute alignment allows for better retrieval accuracy. This is because the best values for solution attributes can be retrieved from different cases with aligned problem attributes most similar to the query. However, learning such relationships or alignment between problem and solution attributes remains a challenge when they are not explicitly expressed in the domain. This applies to TCBR where it is difficult to predict which section of a text (e.g. sentence) aligns to specific problem attributes.

We propose a method that aligns sub-rating attributes to specific sentence(s) in the text of a review. This enables the reuse of sentences from different authors with similar sub-ratings to a query. The basic idea in our text alignment method is to bridge the vocabulary in the problem and solution spaces. This is done by compiling a list of seedwords related to each sub-rating; these seedwords were obtained from WordNet [8] by checking for synonyms of sub-rating descriptors and manually refining the list. A sample list of example seedwords extracted for the five sub-ratings is given in Table 2. Although our list of seedwords is non-exhaustive, it is a good foundation to test our text alignment hypothesis.

Sub-rating name	**Seedwords (sample)**
cleanliness	clean, neat, kempt, tidy, cleanse
location	location, position, place
room	room, bedroom
service	help, serve, service, reception, star
value	esteem, rate, valuate, value, worth

Table 2. Seedwords used for text alignment between sub-ratings and review sentences



1 **Location** of hotel is perfect, within walking distant to the main JR station, subway metro, there is a station just next to the hotel.

2 For shoppers Takashimaya is just across the bridge!

3 Airport transfer right to doorsteps.

4 Food, shoppings and train/subway stations are within 5 to 10 mins walk.

5 5 mins walk to this electric street that not only sell all electrical appliance but with resturants that the locals frequent, that **serve** very nice and reasonable cheap Japanes dishes.

6 Hotel staff are efficient and **helpful** and especially the front desk staff speaks very good english.

7 Intenet access in the **room** is superb, shampoo , conditioner and body wash come in family size bottles, fantastic!

8 The only minus point is the standard queen bed **room** has got no cupboard, its better of choosing the standard double bed room.

9 But on the whole its a very **clean**, comfortable and safe hotel; we would **rate** it 9 out of 10.

Figure 2. Example of review sentences' alignment to sub-ratings

Each review text is parsed into sentences using the GATE [6] libraries available as part of the jColibri [7] framework. Every sentence in the text is then categorised as belonging to a sub-rating if it contains any of its seedwords. Figure 2 illustrates alignment between review sentences and sub-ratings using the review text in Figure 1; here, seedwords are in bold. The text has 9 sentences of which only 6 are aligned to sub-ratings. It can be observed that most of the aligned sentences are intuitively reasonable; for example, sentence 1 is about the proximity of the hotel to rail station and is correctly aligned to *location* sub-rating. However, sentence 5 is better aligned to *location* than *service* sub-rating since it highlights the hotel's proximity to restaurants and local shops. The unaligned sentences (i.e. 2, 3 & 4) are related to *location* but were not linked because they contain none of the seedwords. This highlights the need for a representative set of relevant seedwords.

Overall, the text alignment process approximates the relationship between sub-ratings and review sentences. The alignment link is a many-to-many relationship as a sentence can belong to more than one sub-rating and vice versa. This is illustrated in Figure 2 where sentences 7 & 8 are linked to *room* sub-rating while sentence 9 is linked to *cleanliness* and *value* sub-ratings. Unaligned review sentences are regarded as verbose details (sometimes unrelated to the hotel) that cannot be easily reused across authors. For example, not every hotel will be in a town with an airport; therefore in Figure 2, sentence 3 cannot be reused by such authors. In our experiments, only aligned sentences are used since this allows the system to propose generic texts while insertion of contextual details is left for authors.

4 Text Reuse Algorithms

We propose three different techniques to assist with reusing textual contents. These techniques make use of text alignment between sub-ratings and review sentences as explained in Section 3. The differences between them are in terms of what neighbourhood of a query is used in generating a proposed solution and how sentences are combined from similar cases. Here, techniques similar to CBR substitutional, transformational and compositional adaptation are applied to textual cases in relation to sentence aggregation from different neighbours of a query.

4.1 Baseline retrieval

Given a query, Q, consisting of a set of rating and sub-ratings, baseline (BASE) retrieves the nearest neighbour and reuses its review text. In Figure 3, *Retrieve* returns the most similar case (C_{best}). Here, sub-ratings are termed ratings since they are graded on the same likert scale. Sentences in the review text aligned to the five sub-ratings are then identified and concatenated to form the proposed solution, $SOLN$. Identification of aligned sentences is achieved with the *selectAlignedSentences* method for each rating in C_{best}. The use of $SOLN$ as a set ensures that duplicate sentences in the proposed solution are removed because each sentence can be aligned to more than one rating. Our baseline technique is essentially a retrieve-only system except for the removal of unaligned sentences. BASE generates five or more sentences in a proposed solution text since there can be multiple sentences aligned to each sub-rating.

```
CB= $\{C_1, \ldots, C_n\}$, set of cases in the case base
$R = \{r_1, \ldots, r_p\}$, set of rating types i.e. pre-defined attributes
$V = \{v_1, \ldots, v_q\}$, set of possible values for each rating type
IE= information entity consisting of a rating with distinct value,
    where $(ratingType(IE) \in R) \wedge (ratingValue(IE) \in V)$
$C_i = \{$IE$_{i1}, \ldots,$IE$_{ip}$, ReviewText$_i\}$, where $(i \in \{1 \ldots n\})$
    i.e. a case consists of $p$ rating values and a review text
$Q = \{$IE$_1, \ldots,$IE$_p\}$ , a query with $p$ rating values

    $C_{best} = Retrieve($CB$, Q, 1)$, retrieve most similar case
    Initialise $SOLN$= {},
        to contain sentences in the proposed solution text
    ReviewText= $getReviewText(C_{best})$
    for each IE$_j \in C_{best}$
        $r_j = ratingType($IE$_j)$
        $S_j = selectAlignedSentences(r_j,$ ReviewText$)$
        $addSentences(S_j, SOLN)$
    Concatenate sentences in $SOLN$ for reuse
```

Figure 3. Baseline text reuse algorithm (BASE)

4.2 Transformational approach to text reuse

This approach denoted as XFRM uses multiple nearest neighbours of a query to propose a review solution text rather than the nearest neighbour used in baseline retrieval discussed in Section 4.1. Given a query of rating and sub-ratings attributes, specified $k-$nearest neighbours are retrieved. To reuse review texts from these neighbours, we propose and progressively transform aligned sentences from the best match solution. This takes place only if there are mismatches between the query and best match's sub-ratings. Sentences aligned to mismatched sub-ratings are removed if they are not aligned to any other sub-ratings and replaced with aligned sentences from nearest neighbours matching the query's sub-rating. This approach is similar to CBR transformational adaptation [5] where solution elements are re-organised through add and delete operations. However it is also similar to substitutional adaptation [16, 9, 1] if seen as successive replacement of aligned sentences in baseline text (see Section 4.1).

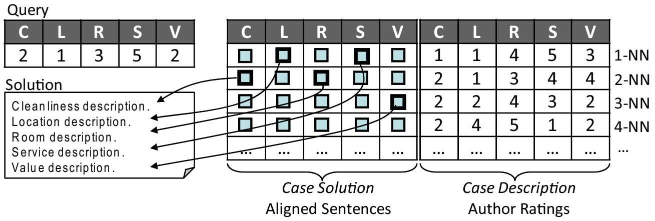


Figure 4. Transformational approach to text reuse

Figure 4 illustrates this approach with a query and four nearest neighbours with their sub-rating values and aligned sentences. Here, sentences for *location* and *service* are chosen from the first neighbour. Any mis-matched values are resolved by extracting aligned sentences from the neighbourood (2NN & 3NN). Note that if a query sub-rating value is not matched in any of the nearest neigbours, no sentence is generated for such sub-rating and number of sentences can be less than 5. However in reality, there are multiple sentences per rating resulting in a reuse solution with five or more sentences.

The transformational approach is also formalised as an algorithm (see Figure 5). Here, we compare each sub-rating (IE_j) in the query with similar sub-ratings of neighbouring cases (CB$_{local}$). Functions $ratingType$ and $ratingValue$ returns the sub-rating type (e.g. location) and values (e.g. 4) respectively. The conditional statement SOL_j=$null$ ensures that aligned sentences are chosen from the first similar case whose sub-rating values matches the query.

4.3 Text generation with sentence clustering

Here, a proposed text is generated in response to a query by combining sentences from several similar cases. Hence it is called compo-

```
Initialise $SOLN = \{SOL_1, \ldots, SOL_p\}$,
    set of proposed sentences for each rating
CB$_{local} = Retrieve($CB$, Q, k)$, retrieve $k$ similar cases
for each IE$_j \in Q$
    $qr = ratingType($IE$_j)$; $qv = ratingValue($IE$_j)$
    for each C$_i \in$ CB$_{local}$, in order of decreasing similarity
        ReviewText= $getReviewText(C_i)$
        $r_j = ratingType($IE$_j, C_i)$; $v_j = ratingValue($IE$_j, C_i)$
        if ($qr = r_j$ AND $qv = v_j$ AND $SOL_j = null$)
            $S_j = selectAlignedSentences(r_j,$ ReviewText$)$
            $addSentences(S_j, SOL_j)$
Concatenate all sentences in $SOLN$ for reuse
```

Figure 5. Transformational text reuse algorithm (XFRM)

sitional (COMP) text reuse because of its similarity to CBR's compositional [5, 3] or constructive [13] adaptation where a solution is obtained by combining solution elements of several partially similar cases. Sentences are considered to be contextually similar when they are aligned to an identical sub-rating value. For example, all sentences aligned to a cleanliness sub-rating of 3 can be regarded as similar. Aggregating several pieces of similar sentences into a single meaningful prototype is not trivial. Concatenation is inappropriate since it leads to tautology and summarisation methods might not work because sentences are semantically similar yet lexically different. Thus, we introduce a mechanism that combines several similar sentences into a single meaningful text called the *prototypical* sentence. For prototypes, a term frequency vector is first created for each sentence. Each vector length is the size of unique keywords in all similar sentences for which a prototype is being determined. A centroid is calculated for these vectors as the average term frequency across each unique keyword. Accordingly, a prototypical sentence is a sentence whose vector is most similar to the centroid vector. Intuitively, our prototype will contain common keywords used across sentences. This is because values of such keywords in the prototype vector will be closer to the average.

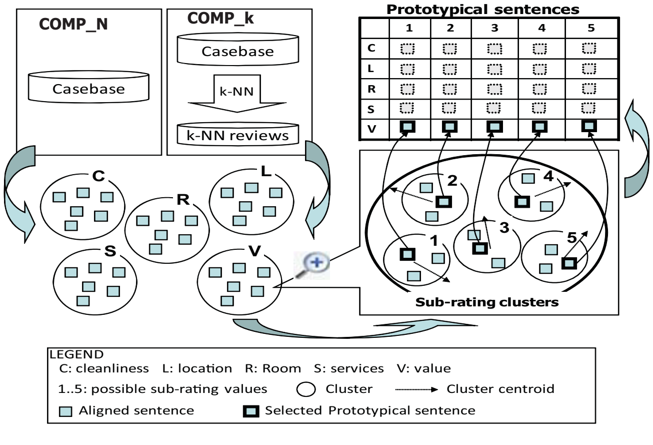


Figure 6. Clustering similar aligned sentences in review texts

The generation of prototypical sentences is illustrated in Figure 6. These prototypes can be generated from either k nearest neighbours to the query (COMP_k) or all reviews in the casebase (COMP_N). Aligned sentences across the specified reviews (local or global) are clustered according to the class they belong to given the five sub-ratings. Each cluster is then further re-clustered into five groups using their rating value (i.e. 1 to 5). The smaller group of clusters shown for the *value* sub-rating also applies to the other four sub-ratings. The outcome of this clustering process is 25 small clusters and a prototypical sentence per cluster.

```
Initialise
$G = \left\{ \begin{array}{c} g_{11}, \ldots, g_{1q} \\ g_{21}, \ldots, g_{2q} \\ \ldots \\ g_{p1}, \ldots, g_{pq} \end{array} \right\}$, set of clustered similar sentences; each cluster belongs to a pair from $p$ ratings and $q$ values
    CB$_{local} = Retrieve($CB$, Q, k)$; retrieve $k$ similar cases
    Initialise $SOLN$= {},
        to contain sentences in the proposed solution text
    for each C$_i \in$ CB$_{local}$, in order of decreasing similarity
        ReviewText= $getReviewText(C_i)$
        for each IE$_j \in C_i$
            $r_j = ratingType($IE$_j)$; $v_j = ratingValue($IE$_j)$
            $g_j = getClusteredSimilarSentences(G, r_j, v_j)$
            $S_j = selectAlignedSentences(r_j,$ ReviewText$)$
            $addSentences(S_j, g_j)$
    for each IE$_k \in Q$
        $r_k = ratingType($IE$_k)$; $v_k = ratingValue($IE$_k)$
        $g_k = getClusteredSimilarSentences(G, r_k, v_k)$
        $ps_k = getPrototypicalSentence(g_k)$
        $addSentences(ps_k, SOLN)$
    Concatenate sentences in $SOLN$ for reuse
```

Figure 7. Compostional text reuse algorithm (COMP_k)

The COMP_k algorithm shown in Figure 7 is illustrated in Figure 8 where the query consists of 5 (p in algorithm) ratings of $2, 1, 3, 5, 2$ for *cleanliness*, *location*, *room*, *service* and *value* respectively. Five sentences are then obtained from the prototypical sentences with identical sub-rating values to the query and concatenated as proposed text ($SOLN$). In this algorithm, each prototypical sentence is generated from an element in the matrix of sentence clusters (G) having $p \times q$ elements. A major difference between COMP_k that use reviews from neighbours and COMP_N that uses all reviews is that it might generate less than five sentences since a small neighbourhood may not contain all sub-rating values required by a query.

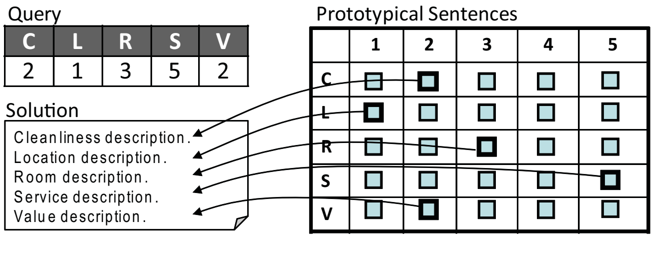


Figure 8. Compositional approach to text reuse

5 Experimental Setup

We compare three text reuse techniques.

1. Baseline retrieval (BASE) in Section 4.1
2. Transformational approach to text reuse (XFRM) in Section 4.2
3. Text generation with sentence clustering (COMP_k & COMP_N) in Section 4.3

A ten-fold cross validation is employed in our experiments. We want to ensure that retrieved reviews have very similar ratings to the query. For example, it will be very difficult to reuse text from a review with rating 4 (very good) for a query with rating 2 (poor). Therefore, similar cases are retrieved with an interval of 2 between rating (or sub-rating) attributes. This means that a difference of 1 between two ratings gives a 0.5 similarity while a difference greater than 1 gives zero similarity. Similarities across attributes are aggregated using a weighted average; 0.25 for rating and 0.15 for each sub-rating.

The effectiveness of the text reuse techniques is measured using cosine coefficient similarity between aligned sentences in our actual solutions and the proposed text. Cosine similarity is employed as opposed to precision/recall because it allows us to compare the performance of the reuse techniques with a single metric which also takes the texts' length into account. We are also interested in the effect of different neighbourhood sizes (k) on reuse performance for COMP_k and XFRM. Experiments for the two techniques were therefore repeated using increasing values of k ($k = 3, 5, 10$ & 25).

5.1 Dataset

A sample dataset from the hotel reviews (Section 2) was created by selecting reviews with sentences aligned to each sub-rating; 641 of such reviews were found. Review texts were normalised by substituting named entities such as person names, currencies, locations and dates with generic labels. Table 3 lists some of these entities extracted with GATE [6] together with the general category label.

Category	Named entity examples
person name	yang, vincent, susanne, patrick, katherine
currency	yen, pounds, francs, euros, dollars, cents
date	september 2009, mid august 08, last year,april 26th
time	9.30pm, 8:00 a.m., 5pm, 3:45pm, 17:45

Table 3. Examples of named entities found in Hotel Reviews

5.2 Discussion of results

Figure 9 shows the average cosine similarities between proposed text and actual solution across the three text reuse techniques. The different k-neighbourhoods are shown in brackets for compositional (COMP_k) and transformational (XFRM) approaches. The baseline (BASE) which recommends a subset of sentences from the best match case by ignoring sentences unaligned to any sub-rating does well as compared to COMP_k and XFRM. This is because they use similar neighbouring case(s) unlike COMP_N which uses all cases.

COMP_k where local prototypical sentences are proposed improves with increasing neighbourhood size. This trend suggests that the performance will match up with COMP_N as the neighbourhood size tends toward the entire casebase. This shows that local prototypical sentences tend to capture less keywords that are reusable across authors as compared to the global prototypes. COMP_N which uses all cases to generate prototypical sentences for each sub-rating clearly outperforms the rest. An advantage of this approach is that these generic sentences are likely to be more similar to the actual solution compared to a local sentence which might express the same opinion using different terms. Although at first surprising, this result compliments findings in other related studies [10, 11] in text reuse. On the other hand, there is very little improvement in performance as we increase the neighbourhood size for XFRM. This means that most of the query sub-ratings are easily matched in the smaller neighbourhoods (i.e $k = 3, 5$). However, aligned sentences generated from such neighbourhoods are not as good as prototypical sentences from larger neighbourhoods.

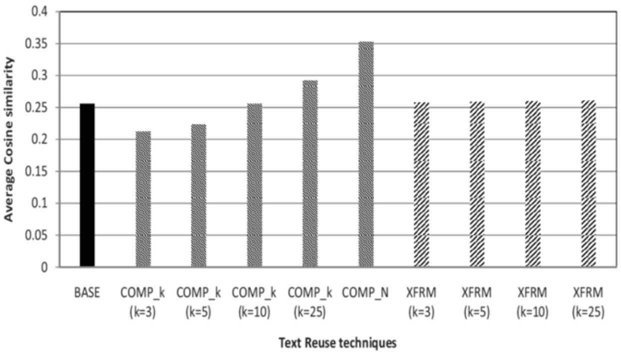
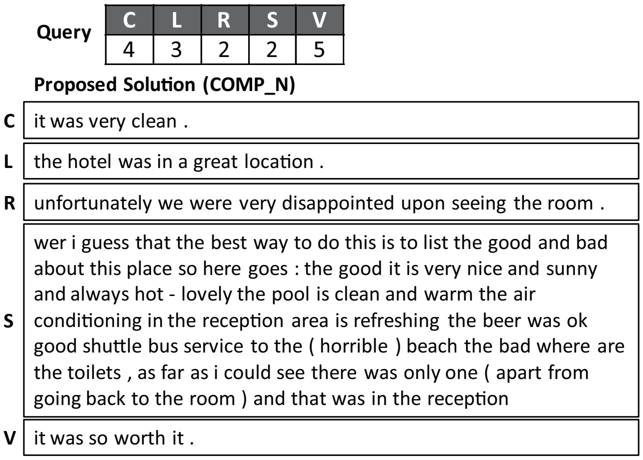


Figure 9. Graph of cosine similarity across text reuse techniques



Query

C	L	R	S	V
4	3	2	2	5

Proposed Solution (COMP_N)

C it was very clean .

L the hotel was in a great location .

R unfortunately we were very disappointed upon seeing the room .

S wer i guess that the best way to do this is to list the good and bad about this place so here goes : the good it is very nice and sunny and always hot - lovely the pool is clean and warm the air conditioning in the reception area is refreshing the beer was ok good shuttle bus service to the (horrible) beach the bad where are the toilets , as far as i could see there was only one (apart from going back to the room) and that was in the reception

V it was so worth it .

Figure 10. An example of the proposed text from COMP_N

Generally, a low cosine similarity (less than 0.5) is seen between the proposed texts and their actual solution. Closer examination of proposed text suggests that low similarity values does not necessarily mean poor solution quality. Figure 10 shows a sample of the proposed text generated by COMP_N technique. Most of the sentences seem reasonable to the given query ratings except for *service* which is verbose. Such long sentences contain specific details that will adversely affect the cosine similarity. Nevertheless, the results indicate that proposed text were similar to the actual and it might be easier to edit them than writing from scratch. Also, our proposed texts will encourage new authors to write reasons for each sub-rating value rather than a lot of verbose but unnecessary details thereby making future reviews more useful to others.

6 Related Work

Automated reuse of text remains a challenge especially when they are available in the unstructured form. There are few studies [14, 11, 2, 4] available in this area due to difficulties with mapping such text to a structured representation, measuring semantic similarity and automated evaluation . A restricted form of textual reuse is presented for report writing applied to the air travel incident domain [14]. Here, textual cases consist of incident reports with one or more paragraphs grouped under a specific heading as a section. The most similar document to a query is retrieved and textual reuse is facilitated for each section of the retrieved report. This is done by presenting a cluster of other documents containing similar text under the same heading. This technique ignores the context of each section within the entire report which might lead to unuseful clusters and is restrictive as it cannot be used where common section headings are absent. Therefore, this approach is not applicable directly to domains such as hotel reviews authoring with no sectional headings. The approach is similar to one of our reuse techniques because similar sections (or sentences in our work) are grouped together. However ours differ in that we propose prototypical sentences generated from sentence clusters.

The drawbacks observed in the work reviewed above are addressed by a text reuse technique called *Case Grouping* (*CG*) [11]. The technique demonstrated on a semi-automated email response application involves reuse of previous email messages to synthesize new responses to incoming requests. A response is a sequence of statements satisfying the content of a given request and requires some personalization and adjustment of specific information to be reused in a new context. The reuse technique annotates sentences as reuse if there is sufficient evidence that similar past problems contain this sentence. The case base is divided into two clusters that contain similar sentence and those that don't to quantify this evidence. Query similarity to a centroid case formed for each cluster determines whether or not to reuse. The centroid case has the average value for each feature across all cases in a cluster. Our mechanism of prototypical sentence (see Section 4.3) is also based on a centroid vector. However, we form a single feature vector for each similar sentence rather than entire text (usually several sentences) in CG. This reduces the effect of aggregating the same features across unrelated sentences.

An approach to text reuse is proposed in [4] where users are given suggestions to support the authoring process applied to a waste exchange service that links people over the web to enable transfer of unwanted items to those who can use such items. Suggestions are generated from previous successful item descriptions; these are descriptions where users have been able to complete transfer of items to others using the service. The approach extracts feature-value pairs from all previous successful descriptions using regular expressions that are manually defined. The most similar successful description is retrieved during authoring of a new item description. This is done iteratively as the author adds a specified amount of text (e.g. a sentence). Features from the similar case are then compared to those extracted from the new partial description. Top k common values of features from the retrieved case whose features are absent in the new description are ranked from top similar cases and shown to the user as suggestions. Such suggestions support the authoring process by assisting a user to write an item description that can lead to the item being transfered successfully. A major drawback is that repeated suggestions are distractive to users and can lead to more time being spent on authoring. The aim in this work is similar to ours and their use of extracted features is similar to our structured attributes. However, we suggest whole texts rather than in bits which removes unnecessary distraction to the author. Also, their technique cannot be integrated into an existing authoring system without modification to the user interface but our techniques can be integrated directly.

7 Conclusion

This work introduced two novel concepts in relation to text reuse: text alignment and sentence aggregation. Text alignment links rated attributes to specific sentences in a review text while sentence aggregation abstracts similar sentences into a single meaningful prototype. These concepts are generally applicable in domains where cases consists of pre-defined attributes along with written text. These mechanisms led to the development of three text reuse techniques that generate proposed texts related to the pre-defined attributes' ratings. Our results show that proposed texts were similar to the actual and will assist authors to write better and more useful reviews. We also obtained better results with global than local prototypical sentences meaning that higher level abstractions are more reusable across authors.

We intend to improve the choice of seedwords by learning introspectively from our corpus as opposed to using a external ontology like WordNet. This might be done by searching for sentences containing defined patterns and limiting our seedwords to specific parts of speech. We plan to introduce alternative evaluation measures such as edit distance and experiment with other related domains.

Acknowledgements This work is funded by the Northern Research Partnership and UK-India Education & Research Initiative.

REFERENCES

[1] Ibrahim Adeyanju, Susan Craw, Abhishek Ghose, Allyson Gray, and Nirmalie Wiratunga, 'RaGoÛt: An arpeggio of tastes', in *ECCBR 2008 workshop Proceedings (Cooking contest)*, pp. 229–238. Tharax, (2008).

[2] Ibrahim Adeyanju, Nirmalie Wiratunga, Rob Lothian, Somayajulu Sripada, and Luc Lamontagne, 'Case Retrieval Reuse Net: Architecture for reuse of textual solutions', in *Proc. of ICCBR'09*, pp. 14–28, (2009).

[3] Rim Bentebibel and Sylvie Despres, 'Using compositional and hierarchical adaptation in the SAARA system', in *ECCBR'06 workshop proceedings (Reasoning with Text)*, pp. 98–108, (2006).

[4] Derek Bridge and Aidan Waugh, 'Using experience on the read/write web: The GhostWriter system', in *ICCBR09 workshop proceedings (WebCBR)*, pp. 15–24, (2009).

[5] Chun-Guang Chang, Jian-Jiang Cui, Ding-Wei Wang, and Kun-Yuan Hu, 'Research on case adaptation techniques in case-based reasoning', in *Proc. of ICMLC'04*, pp. 2128–2133, (2004).

[6] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan, 'Gate: A framework and graphical development environment for robust NLP tools and applications', in *Proc. of ACL '02*, (2002).

[7] Belén Díaz-Agudo, Pedro A. González-Calero, Juan A. Recio-García, and Antonio Sánchez, 'Building CBR systems with jCOLIBRI', *Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming*, **69**, 68–75, (2007).

[8] *WordNet: An Electronic Lexical Database*, ed., Christiane Fellbaum, MIT press, 1998.

[9] Pedro A. González-Calero, Mercedes Gómez-Albarran, and Belén Díaz-Agudo, 'A substitution-based adaptation model', in *ICCBR'99 workshop proceedings (Challenges for CBR)*, (1999).

[10] Luc Lamontagne and Guy Lapalme, 'Applying case-based reasoning to email response', in *Proc. of ICEIS-03*, pp. 115–123, (2003).

[11] Luc Lamontagne and Guy Lapalme, 'Textual reuse for email response', in *Proc. of ECCBR'04*, pp. 234–246, (2004).

[12] Enric Plaza, 'Semantics and experience in the future web', in *Proc. of ECCBR 08*, pp. 44–58, (2008).

[13] Enric Plaza and Josep-Lluís Arcos, 'Constructive adaptation', in *Proc. of ECCBR'02*, pp. 306–320, (2002).

[14] Juan A. Recio-García, Belén Díaz-Agudo, and Pedro A. González-Calero, 'Textual CBR in jCOLIBRI: From retrieval to reuse', in *ICCBR'07 workshop proceedings (Textual CBR)*, pp. 217–226, (2007).

[15] Rosina Weber, Kevin Ashley, and Stefanie Bruninghaus, 'Textual case-based reasoning', *Knowledge Engineering Review*, **20**(3), 255–260, (2006).

[16] Wolfgang Wilke and Ralph Bergmann, 'Techniques and knowledge used for adaptation during case-based problem solving', in *Proc. of IEA-AIE'98*, (1998).