Options for Query Revision when Interacting with Case Retrieval Systems

Alex Ferguson Derek Bridge
Department of Computer Science,
University College, Cork
a.ferguson@cs.ucc.ie d.bridge@cs.ucc.ie

Abstract

This paper is concerned with using similarity for case retrieval, especially in interactive case retrieval systems (ICRSs). We define and exemplify similarity metrics, which are a generalisation of similarity measures, having any partial order as their result type. Within this new similarity metric setting, we define the notion of the 'best' cases in case retrieval. We then show the variety of ways in which users can proceed through an interaction with an ICRS to widen, narrow or otherwise change the set of retrieved 'best' cases. We show that using similarity metrics rather than similarity measures increases the variety of interaction options open to users.

1 Introduction

This paper is concerned with Interactive Case Retrieval Systems (ICRSs). In an ICRS, as in all CBR systems, a similarity measure is used to compare some or all of the cases in a case base,[1] $c \in \mathrm{CB}$, to some particular probe case, p. In ICRSs, the probe p is supplied by the user, and the cases in CB that are 'most similar' to p are displayed for user perusal. Users who are not satisfied with the results of their single-shot query might then interact further with the ICRS.

In CBR systems other than ICRSs, the 'most similar' case or cases are not simply displayed for the user; they are processed further by the system. For example, they might be automatically evaluated and/or adapted. Much of what we say in this paper might apply in these settings too but the focus is exclusively on ICRSs.

ICRSs are a legitimate and worthwhile field of study in themselves. Many (though not all) of the main CBR research issues can be investigated in the ICRS setting (e.g. storage and indexing of cases, computation of similarity, learning behaviour and case base maintenance). And, the vast majority of fielded CBR systems are actually ICRSs: most CBR-based help-desk support

[1] CBR systems in which the case base is indexed and case base interrogation is a two-stage process are often examples of systems in which the similarity measure is applied to only a subset of the cases in the case base: the first stage exploits the indexes to restrict computational effort to certain cases, and then, in the second stage, a similarity measure is applied only to the results of the first stage. For simplicity of exposition, in this paper we will henceforth assume that the similarity measure is applied to every case in the case base.

systems and e-commerce systems for searching through product catalogues are ICRSs. This paper uses examples from a product catalogue search application.

2 Similarity metrics

In the main, similarity measures have been binary operators that, when applied to two objects of type α, return a number, usually a real from $[0, 1]$, denoting the objects' degree of similarity. That is, their type is most usually $\alpha \to \alpha \to [0, 1]$, for any data type α.

We have elsewhere described *similarity metrics*, our generalisation of similarity measures [7, 8, 9, 2, 4]. A similarity metric is a binary operator that, when applied to two objects of type α, returns some value indicating the degree to which the two arguments are similar. That is, if $\sim$ is a similarity metric, then

$$\sim :: \alpha \to \alpha \to P$$

for any data type α, and some suitable P.

All that we require of P is that it impose a relative degree of similarity, e.g. so that we can say whether objects a and b are more similar to each other than are objects c and d. We require, therefore, that P be a *partial order*,[2] $P = (S, \sqsubseteq)$.

Of course, not every function of this type is a similarity metric. To be a similarity metric, a function must also satisfy a number of similarity metric axioms. These axioms have been discussed in [9, 2].

3 Examples of similarity metrics

We will use product catalogue searching as our example application. The particular mini case base (product catalogue) that will be used in the examples is given at the end of this paper. It comprises computer representations of a number of hotels in the Manchester area. Assume we can 'project out' of these representations the value of certain attributes of the hotels. The lowest price per night of a room, *price*, is an attribute of type $\mathcal{N}$; the amenities of the hotel, *amen*, is a value of the powerset type $\mathcal{P}\{d, f, m, s, w\}$, i.e. a subset of $\{d, f, m, s, w\}$ that indicates whether the hotel has a dining room (d), a fitness room (f), meeting rooms (m), a swimming pool (s) and wheelchair access (w).

On occasion, we might want a boolean-valued similarity measure, i.e. one which simply says whether two values are similar or not. To formulate these as similarity metrics, we construct a suitable partial order Bool as the result type of the function: the partial order is **False** $\sqsubset$ **True**[3]. The similarity metrics will then be of type $\alpha \to \alpha \to \mathsf{Bool}$.

[2] In our earlier work, we required P to be a complete lattice [7, 8, 9, 2], which is a more restrictive requirement. We thought this to be a natural restriction. However, in [4] we lift this restriction and provide example similarity metrics where the result type is a partial order but not a lattice.

[3] And of course **False** $\sqsubseteq$ **False**, **True** $\sqsubseteq$ **True**.

An example Bool-valued similarity metric is:

$$x \sim_{price1} y \mathrel{\hat{=}} abs(x - y) < 10$$

i.e. prices x and y are similar if they are within \$10 of each other. Note that this is not transitive. Hotel P (\$72 per night) and hotel H (\$80) are similar (the absolute difference in price is less than 10); hotels H (\$80) and N (\$85) are also similar; but P and N are not similar (the absolute difference in price is not less than 10).

Bool-valued similarity metrics can be used in ICRSs but they fail to capture the intuitive notion of different degrees of similarity. This motivates numeric-valued similarity functions. These too are special cases of our framework. Any numeric set ordered by $\leq$ or $\geq$ gives the partial order we need as a result type.

A simple numeric metric is to measure the distance between the two values. For example,

$$\sim_{price2} :: \mathcal{N} \to \mathcal{N} \to (\mathcal{Z}, \leq)$$

$$x \sim_{price2} y \mathrel{\hat{=}} -abs(x - y)$$

By this function on prices, P (\$72) and H (\$80) are similar to degree -8 (negated absolute difference) and this is a higher degree of similarity than the similarity of P (\$72) and N (\$85), which are similar to degree -13.

An example numeric-valued similarity metric for hotel amenities is given by the cardinality of the intersection of the sets of amenities:

$$\sim_{amen1} :: \mathcal{P}\{d, f, s, m, w\} \to \mathcal{P}\{d, f, s, m, w\} \to (\mathcal{N}, \leq)$$

$$x \sim_{amen1} y \mathrel{\hat{=}} |\, x \cap y \,|$$

which designates hotel X (which has a dining room and a swimming pool, $\{d, s\}$) to be similar to degree 1 to hotel S (which has a dining room and meeting rooms, $\{d, m\}$).

More conventional [0,1]-valued similarity metrics can also be defined. For example, we could normalise to [0,1] the results of any of the above numeric-valued metrics.

As a final example, we show that set-valued metrics are also possible. We use intersection again, but this time we let the intersections themselves denote the degrees of similarity:

$$\sim_{amen2} :: \mathcal{P}\{d, f, s, m, w\} \to \mathcal{P}\{d, f, s, m, w\} \to (\mathcal{P}\{d, f, s, m, w\}, \subseteq)$$

$$x \sim_{amen2} y \mathrel{\hat{=}} x \cap y$$

Now the similarity of X ($\{d, s\}$) and S ($\{d, m\}$) *is* $\{d\}$ — the intersection itself denotes the degree of similarity. Hotels X ($\{d, s\}$) and P ($\{d, f, m, s\}$) are similar to degree $\{d, s\}$. When we compare the degrees of similarity, we see that our second pair of hotels (X and P) are more similar to each other than our first pair of hotels (X and S) are to each other: $\{d\} \subset \{d, s\}$.

The advantages of our framework are explained in [7, 9, 2]. We state them very briefly here. The first advantage is that the framework subsumes many ways of measuring similarity, e.g. boolean-valued, numeric-valued, set-valued [6], feature structure-valued [10], and linguistic-hedge-valued [2]. In subsuming so many approaches, we know that results we obtain in this framework (theorems, implementation techniques, etc.) will apply quite broadly. The second advantage is the 'naturalness' of the similarity functions that we can define: result types can be chosen to best suit the application. The third advantage is the ease with which we can combine similarity metrics. Uniquely, we believe, similarity metrics of the same, or even quite different, result types can be combined *without inter-conversion.* For example, a numeric-valued and a set-valued similarity metric can be combined into a single metric without first having, e.g., to convert the set-valued metric into a numeric-valued metric [7, 8, 9, 2, 4].

4 Retrieval uses unary metrics

Consider a user of an ICRS who wants to query the system's case base. The system has a similarity metric, $\sim$, and the user supplies a probe, p. Every case in the case base, $c \in \mathrm{CB}$, is compared, using $\sim$, to p. The 'best' cases, those whose similarity to p is maximal, are displayed.

In this scenario, we are applying the function $\sim$ repeatedly. One of the arguments to $\sim$ will be different every time: as we consider each case in the case base in turn. But the other argument to $\sim$ is the same each time: the probe, p. So, instead of presenting the results of this paper using binary functions, we can use unary functions. Suppose we have a similarity metric, $\sim$, of type

$$\sim\, ::\ \alpha \to \alpha \to (S, \sqsubseteq)$$

then, given some probe p, we can construct a unary function, $u^p_\sim$, of type

$$u^p_\sim\ ::\ \alpha \to (S, \sqsubseteq)$$

where

$$u^p_\sim c \mathrel{\hat{=}} p \sim c$$

i.e. $u^p_\sim$ is a unary function which takes in a case, c, and compares it to probe p using similarity metric $\sim$. We have 'frozen' one argument to the similarity function, the result being a unary function. In functional programming, this is referred to as *partial application.*[4]

Here is an example. From $\sim_{price2}$, we can define u^{80}_{price2}, a function that uses $\sim_{price2}$ to compute the similarity of its argument to \$80. Of course, there is a family of such unary functions, one for each probe p. So, from $\sim_{price2}$ we can also define, e.g., u^{90}_{price2}, u^{100}_{price2}, etc., by 'freezing in' different probe values.

[4] If $\sim$ is symmetric, then it does not matter which argument is 'frozen in': defining $u^p_\sim$ as $p \sim c$ is equivalent to defining it as $c \sim p$. However, if asymmetric similarity measures are allowed, the choice of which of these two ways of defining $u^p_\sim$ becomes significant and would need to be decided by the system designer or the user.

Querying an ICRS now becomes a matter of taking a probe from a user, partially applying a similarity metric to it to give a unary function, applying this to every case in the case base, and displaying the best cases.

This is a useful move in this paper as a notational ploy. When we come on to defining the best cases (section 5), the definitions will look simpler if we use these unary functions rather than expressing them in terms of (binary) similarity metrics.

Furthermore, it may inspire wider use of our framework. There are many AI applications where objects need to be scored or ranked according to some criteria; unary [0,1]-valued functions are often used for this purpose. For example, utility measures in action planning and fitness functions in genetic algorithms are typically of type $\alpha \rightarrow [0,1]$. But, in the same way that we generalised similarity measures of type $\alpha \rightarrow \alpha \rightarrow [0,1]$ to similarity metrics of type $\alpha \rightarrow \alpha \rightarrow (S, \sqsubseteq)$, utility measures and fitness functions might also be usefully generalised from functions of type $\alpha \rightarrow [0,1]$ to functions of type $\alpha \rightarrow (S, \sqsubseteq)$. Then the results of this paper (and our other papers) can be applied to action planning and genetic algorithms.

So, the rest of this paper will be couched in terms of unary functions, $u^p_\sim$, rather than binary similarity metrics, $\sim$. We will refer to these unary functions as *unary metrics*, and we will denote them simply by u, except where $u^p_\sim$ would be clearer.

(Note that none of the above is intended to imply that this is the only way to use similarity functions. For example, in data mining and machine learning systems, similarity measures and metrics are used to form 'clusters' of objects. In these systems, pairwise comparisons of objects (e.g. each object with every other object) are often required, and in these situations partial application to form unary functions would serve little purpose).

5 Best cases

Having applied a unary metric to every case in the case base, we then want to determine the best cases, i.e. those whose metric value is maximal: that is, the value of the unary metric, applied to that case, is no worse than the value for any other case in the case base. As the order is partial, in general we may have several different maximal metric values, each incomparable to the other in the ordering (as well as several different cases which yield the same metric value).

We will give the definition first, and then some examples.

Given a case base CB with each case being of type α and given a unary metric, u, of type $\alpha \rightarrow (S, \sqsubseteq)$, the best cases are defined by a best case function $\mathbf{best}_u$:

$$\mathbf{best}_u \,\mathrm{CB} \mathrel{\hat{=}} \{c : c \in \mathrm{CB}, \exists m \in M : (u\, c) = m\}$$

where M is the set of maximal metric values:

$$M \mathrel{\hat{=}} \max_{(S,\sqsubseteq)}\{u\, c : c \in \mathrm{CB}\}$$

and the function $\max_{(S,\sqsubseteq)}$ is the usual maximal set operation on the partial order $(S, \sqsubseteq)$, that is for $S' \subseteq S$:

$$\max_{(S,\sqsubseteq)} S' \mathrel{\hat{=}} \{x \in S', \forall x' \in S', x \not\sqsubset x'\}$$

In words, the best cases with respect to unary metric u, $\mathbf{best}_u$CB, are the cases whose metric values are in M, where M is the best of the metric values of the entire case base; some element of a set S' is 'best', that is to say, maximal iff if it is not lower in the ordering than any other element of S'.

This definition gives those cases which are adjudged the best in the case base with respect to the given unary metric — or at least, those which it is not possible to discriminate further between.

(Note that this definition is not an implementation prescription. If it were, it would appear to involve 'multiple passes' through the case base, first to compute M and then to compute $\mathbf{best}_u$ from M. We have a different but equivalent definition of **best** which, while more complicated and less clear, is suggestive of a 'single pass' implementation, but we will not show this here.)

We give two simple examples using **best**. Consider a user who wants a hotel room costing around \$95 according to $\sim_{price2}$. Unary metric $u^{95}_{\sim_{price2}}$ is applied to every case. The set of maximal values so-computed is $\{-4\}$ (the set is a singleton because the ordering $\leq$ is a total order). Hotels whose prices are -4 similar to \$95 are the best ones; they are shown in Figure 1a.

(a) $\mathbf{best}_{u^{95}_{\sim_{price2}}}$

M	99	$\{d, w\}$
T	99	$\{d, f, m, w\}$
U	91	$\{\}$

(b) $\mathbf{best}_{u^{\{d,f,m,s,w\}}_{\sim_{amen2}}}$

A	158	$\{d, f, m, s\}$
B	80	$\{d, f, m, s\}$
C	102	$\{d, m, s, w\}$
D	67	$\{d, f, m, s\}$
F	112	$\{d, m, s, w\}$
G	139	$\{d, m, s, w\}$
P	72	$\{d, f, m, s\}$
R	158	$\{d, f, m, w\}$
T	99	$\{d, f, m, w\}$
V	115	$\{d, f, s, w\}$

Figure 1: Query results

Now consider a user who wants a hotel whose facilities are as similar (using $\sim_{amen2}$) as possible to $\{d, f, m, s, w\}$. When $u^{\{d,f,m,s,w\}}_{\sim_{amen2}}$ is applied to each case, the maximal values are $\{\{d, f, m, s\}, \{d, m, s, w\}, \{d, f, m, w\}, \{d, f, s, w\}\}$ — there is no longer a unique maximal value. For example, hotel A is similar to the probe to degree $\{d, f, m, s\}$ and, in the ordering used in $\sim_{amen2}$ (i.e. $\subseteq$), this is neither higher, lower nor equal to the similarity of hotel C to the probe, i.e. $\{d, m, s, w\}$. This is one of the consequences of generalising from similarity measures (whose result types, [0,1], are totally ordered) to similarity metrics (whose result types are partially, and thus not necessarily totally, ordered).

Hotels with maximal similarity to the probe are shown in Figure 1b; they have equal or incomparable degrees of similarity to the probe.

6 What can a user do next?

Interaction with an ICRS, especially for product catalogue search, is unlikely to comprise only a single-shot query. While subsequent interaction might comprise unrelated further single-shot queries, more likely is that a user, faced with a set of best cases (e.g. as per Figure 1) will want to use these cases as a basis for exploration in subsequent interaction.

An ICRS might offer users the following options at this point:

- changing the probe,
- changing the way degrees of similarity are computed,
- changing the case base contents,
- requesting the next best cases,
- requesting a subset of the next best cases, and
- indicating indifference between degrees of similarity.

The last two of these are new possibilities, opened up by our similarity metric framework, reported for the first time in this paper. For completeness, we describe all six options. The six options do not form an exhaustive list; they are simply the main possibilities in simple ICRSs.

6.1 Changing the probe

An obvious way to revise a query in the hope of seeing a different set of retrieved cases is to change the probe. For example, supplying a desired price of \$90, in place of the \$95 used earlier, so that the best cases are computed using u^{90}_{price2} instead of u^{95}_{price2}, will result in the user viewing the cases in Figure 2a.

(a) $\mathbf{best}_{\underset{\sim}{u}^{90}_{price2}}$			(b) $\mathbf{best}_{\underset{\sim}{u}^{85}_{price2}}$		
U	91	$\{\}$	N	85	$\{d, m\}$

Figure 2: Query results

This option is universally available in ICRSs. But, in some ICRSs the only way to change the probe is to enter a new probe into a form on the screen. The problem with this is that it is not easy for a user to deploy this sensibly. Seemingly small changes in probe values can radically change the set of retrieved cases: queries that, in the user's mind, were related may not produce related results. For example, changing the probe value to $85 will result in the user viewing the cases in Figure 2b, which are disjoint from those in Figures 1a and 2a.

This problem can be lessened by displaying not just the best cases but other highly ranked cases too (see subsections 6.4, 6.5 and 6.6). Then, cases retrieved in one query may still appear in the result set of a subsequent query that uses a different, but not very different, probe.

An interesting possibility, offered by only some ICRSs, is to allow the user to select one or more new probes from the set of cases that have been so far retrieved and are currently being displayed on the screen, rather than having the user type in new probes from scratch. If this is done, then assuming that the similarity metric is not transitive, a different, but related, set of cases will be retrieved by the new query.

6.2 Changing the way degrees of similarity are computed

In some ICRSs, users can change the way that degrees of similarity are computed. For example, the user might have been using $\sim_{price1}$ (repeated below), and might replace it in subsequent queries by $\sim_{price3}$ or $\sim_{price4}$ to narrow or broaden the set of cases that will be retrieved by the **best** function.

$$x \sim_{price1} y \mathrel{\hat{=}} abs(x - y) < 10$$

$$x \sim_{price3} y \mathrel{\hat{=}} abs(x - y) < 5$$

$$x \sim_{price4} y \mathrel{\hat{=}} abs(x - y) < 20$$

In fact, in existing ICRSs, it would be very rare for users to change individual similarity metrics in this kind of way. More usually, individual similarity metrics are installed and maintained by the application designer.

There is, however, one aspect of similarity metrics that some ICRSs do allow users to change: the way that individual similarity metrics are combined to form overall similarity metrics. For example, an ICRS might have a similarity metric for comparing hotel room prices and another for comparing hotel amenities. These two functions need to be combined into a single function so that hotels can be compared for similarity on both room prices and amenities together.

In the traditional similarity measure setting ($\alpha \to \alpha \to [0, 1]$), the functions are combined using numerically-weighted sums, products or averages. Some ICRSs will allow users to alter the weights so that individual functions contribute less or more to the overall degree of similarity.

Our similarity metric framework also offers this option to users. But, our ways of combining functions have to be sensitive to the fact that the way we denote degrees of similarity is not restricted to real numbers from [0,1] but can be values from any partially ordered set. We have covered this issue in a recent paper [4] and, due to its length, we will not repeat that material here.

6.3 Changing the case base contents

It is obvious that a change to the contents of the case base can change the set of best cases retrieved. It is less obvious that this could be an option open to users of ICRSs. No existing CBR shell provides for this, for example.

However, in a number of reasonable future developments, this could become an option. For example, imagine a web-based ICRS in which parts of a case base are down-loaded for client-side similarity-based retrieval, instead of the more common server-side retrieval. In other words, the comparisons take place on the user's machine; the server simply delivers an initial set of cases. It might then be an option, over the course of an interaction with the system, to down-load further partitions of the case base: users might elect to down-load a second partition if their queries are not satisfactorily answered on the first partition that they down-load.

We can envisage many other similar scenarios. For example, initial sets of cases to which similarity-based retrieval is applied might be obtained by using a traditional database query on a legacy database or by instructing a softbot to collect cases from the Internet. By issuing further database queries or by waiting for the softbot to discover further cases on the Internet, the user will be interacting with an evolving set of cases.

6.4 Requesting the next best cases

It is common to offer the user the option of requesting more than just the best cases. We can retrieve larger sets of cases by retrieving those of successive ranks.

In general, we can define O_u^n, the nth rank of cases with respect to u, as follows. The rank 1 cases w.r.t. u are the best cases w.r.t. u:

$$O_u^1 \mathrm{CB} \mathrel{\hat{=}} \mathbf{best}_u\, \mathrm{CB}$$

The next lower rank of cases are those that are best when the 'real' best ones are taken out of the picture. Inductively:

$$O_u^{n+1} \mathrm{CB} \mathrel{\hat{=}} \mathbf{best}_u\, (\mathrm{CB} - \bigcup_{i=1}^{n} O_u^i)$$

(Again, the definition is not intended as an implementation prescription.)

The rank 2 cases for u_{price2}^{95} and $u_{amen2}^{\{d,f,m,s,w\}}$ are shown in Figures 3a and 3b. (The rank 1 cases are, of course, those already shown in Figure 1.)

Relatedly, we may wish to find, rather than a given number of ranks, simply a given number of cases, i.e. the 'best-n' For example, we may wish for the best 50 cases, regardless of whether this means calculating 50 ranks or only 1. We can obviously define this as the union of successive ranks, until the cardinality of the answer set is 50. The problem with this is that we have to allow for the possibility of 'ties' for 'last place': that is, the final rank we add may give us more, and possibly even many more, cases than the 'n' we requested. One

(a) $O^2_{u^{95}_{\sim price2}}$		
E	100	$\{d\}$

(b) $O^2_{u^{\{d,f,m,s,w\}}_{\sim amen2}}$		
K	104	$\{d,m,w\}$
L	88	$\{d,m,w\}$
Z	104	$\{f,m,s\}$

Figure 3: Query results

might compare this to the 'cut' rule in golf, where one allows, for example, '40th place and ties' into the final rounds.

Accordingly, we define $\mathbf{best}^n_u$ to be the optimal n entries in the case base, with respect to the metric u, plus any entries which are incomparable with those n cases (and, therefore, are effectively just as good for the purposes of that query). It is given by:

$$\mathbf{best}^n_u \mathrel{\hat=} B_m$$

such that:

$$|B_m| \geq n, |B_{m-1}| < n$$

where $B_0 \ldots B_m$ are a family of sets of best cases:

$$B_i \mathrel{\hat=} \bigcup_{j=1}^{i} (O^j_u \text{ CB})$$

6.5 Requesting a subset of the next best cases

Consider a user who is viewing a set of retrieved best cases. These cases are ones that are highly similar to the user's probe. The user may favour some of these cases over others. Let's suppose that the ICRS's interface offers the user a way of selecting these favoured cases (e.g. by clicking on check-boxes). Thus the user selects a 'distinguished' subset of the retrieved cases, which we designate below by D. The user might like to see only those cases in the next rank that are most like those in D.

We have already, in subsection 6.1, dealt with the situation where the user uses the cases in D as new probes. To avoid confusion, let us state explicitly that we are now describing a different possibility. Here, the probe remains as before and we are dealing with the retrieval of cases from the next rank.

So, the set of cases on the screen, of which D is a subset, are the most similar to the probe. Assuming all else remains unchanged (e.g. that the probe and similarity metric remain unchanged), any further cases we retrieve will come from the next rank of cases. But, we are suggesting here that, rather than retrieve all cases in the next rank, the user may want only those in the next rank that are most similar to the favoured ones in D.

The function for computing the next rank in this selective manner, the best in CB w.r.t. u and D, O^D_u, is:

$$O^D_u \mathrm{CB} \hat{=} \mathbf{best}_u \{c \in \mathrm{CB}, \exists d \in D : (u\, c) \sqsubset (u\, d)\}$$

In fact, the idea we are presenting in this subsection is only useful in situations where the similarity metric's result type is not totally ordered. We will demonstrate this with two examples.

Suppose the user is looking at the cases in Figure 1a, computed using u^{95}_{price2}. The subset of hotels that the user prefers in this result set is, say, $\{U\}$, i.e. $D = \{U\}$. If the user then applies $O^{\{U\}}_{u^{95}_{price2}}$ to the case base, the cases retrieved are the best of the cases that have lower similarity to the probe than the ones in $\{U\}$. Since hotel U was -4 similar to the probe, the cases now retrieved are those that are -5 similar to the probe. But, this is the whole of the next rank, as depicted in Figure 3a. You can see that, in this situation, using $O^{\{U\}}_{u^{95}_{price2}}$ is no different from using $O^2_{u^{95}_{price2}}$ from earlier. Picking a particular element from the current rank is unnecessary, since they are all equivalent as far as a totally-ordered metric is concerned.

But, now we give an example in which the similarity metric is $\sim_{amen2}$, whose result type is not a total order.

This time, the user is viewing the cases in Figure 1b. Perhaps wheelchair access is especially important, so the user selects the hotels on the screen that offer wheelchair access, i.e. $D = \{C, F, G, R, T, V\}$. Now, we compute $O^{\{C,F,G,R,T,V\}}_{u^{\{d,f,m,s,w\}}_{amen2}}$. Hotels C, F, G, R, T and V were similar to the probe to degrees $\{d, m, s, w\}$, $\{d, f, m, w\}$ and $\{d, f, s, w\}$. The new cases displayed are the best of those whose degrees of similarity are lower than these degrees of similarity. They are shown in Figure 4, and should be compared with those in Figure 3b. (Case Z no longer appears in the result set because, although

$O^{\{C,F,G,R,T,V\}}_{u^{\{d,f,m,s,w\}}_{\sim amen2}}$		
K	104	$\{d, m, w\}$
L	88	$\{d, m, w\}$

Figure 4: Query results

it is in the next rank, it is not of comparable similarity to the user's favoured cases.)

6.6 Indicating indifference between degrees of similarity

We introduce the notion of degrees of indifference between degrees of similarity and use this to change the set of retrieved cases. While we will use $\sim_{price2}$ and $\sim_{amen2}$ in our examples, this idea is one that is more useful when the result type of the similarity metric is not totally ordered.

Consider the similarity metric $\sim_{price2}$, from above. Someone trying to find a suitable hotel room is unlikely to regard small differences in the degrees of similarity of two rooms according to this metric as being decisive in the choice of which match is best. For example, if the probe value is \$90, it would probably be unwise to regard an \$89 room (similar to the probe to degree -1) to be decisively more similar than an \$88 room (similar to the probe to a strictly lower degree, -2). Yet, by our definition of **best**, such small differences would indeed be decisive in determining what appears in the best set. Better in the case of small differences (where 'small differences' for a hotel room metric might be differences as large as, say, 10) would be to broaden the best set to include cases that, while not maximally similar to the probe, are 'as good as' maximally similar to the probe (i.e. similar to the probe to a degree that is so close to the maximal degree that we are indifferent about the shortfall). This is not necessarily the same as taking the next rank of cases, as you will see in one of our examples, after we have given the definitions.

We need a new definition of **best** that is insensitive to small differences in degrees of similarity. But it also means that we have to say what we mean by 'small differences in degrees of similarity'. For this we introduce the idea of degrees of indifference between degrees of similarity.

In order to do this, we must change the signatures of similarity metrics. We augment the result type with a boolean-valued binary function that will be used to indicate whether we are indifferent between two degrees of similarity:

$$\sim :: \alpha \rightarrow \alpha \rightarrow (P, \simeq_P)$$

where P, as before, is a partial order, $P = (S, \sqsubseteq)$.

Then, we need to formalise this notion of indifference between degrees of similarity ($\simeq_P$). The relations that formalise this notion will generally *not* be equivalence relations. To use an equivalence relation would have the same effect as simply using a less discriminating result type in the similarity metric; there would still be cases of small differences in the similarity metric that were having decisive effects.

For example, suppose we wished to declare, for the price metric, similarities within 10 to be equivalent:

$$\sim_{price2} :: \mathcal{N} \rightarrow \mathcal{N} \rightarrow ((\mathcal{Z}, \leq), R_{price2})$$

$$x \sim_{price2} y \mathrel{\hat{=}} -\,abs(x - y)$$

$$s_1 R_{price2} s_2 \mathrel{\hat{=}} s_1 \ \mathit{div\ 10} = s_2 \ \mathit{div\ 10}$$

Here R_{price2} is indeed an equivalence relation. But this is, in essence, the same as simply using $=$ as the indifference relation with prior integer division by 10:[5]

$$\sim_{price2} :: \mathcal{N} \rightarrow \mathcal{N} \rightarrow ((\mathcal{Z}, \leq), =)$$

[5]It is also the same, in this case, as using integer division on the argument types:

$$x \sim_{price2} y \mathrel{\hat{=}} -\,abs(x \ \mathit{div\ 10} - y \ \mathit{div\ 10})$$

This suffers the same weakness that we explain above.

$$x \sim_{price2} y \hat{=} -abs(x - y)\ div\ 10$$

The problem with these (equivalent) formulations is that they still have points where small differences in the degrees of similarity are decisive, while at other points a larger difference proves not to be. For example, similarity degrees of -9 and -10 would be mapped to distinct values (0 and -1), whereas similarity degrees of -10 and -19 are both mapped to -1.

More reasonable is an *ϵ-equality*, similar to that used for comparing floating point numbers, i.e. we are indifferent between two degrees of similarity, s_1 and s_2, iff $abs(s_1 - s_2) < \epsilon$. For our price metric, ϵ would be 10. Now, we would be indifferent between similarity degrees of -9 and -10, and of -10 and -19, but not of -9 and -19.

We note that this relation is reflexive and symmetric, but *not* transitive. Instead, it satisfies the generally weaker property of *convexity*. Define a relation R to be P-convex, over any partial order P, iff:

$$\forall x, y, z \in P : x \sqsubseteq_P y \sqsubseteq_P z, x\, R\, z \Rightarrow x\, R\, y, y\, R\, z$$

That is, given three ordered 'points', if the top and the bottom points are related by R, then the middle point must also be related to both.

To generalise then, what we require of an indifference relation on degrees of similarity, $\simeq_P$, is that it satisfy reflexivity, symmetry, and convexity. It need not satisfy transitivity.

Now we can redefine the **best** function with respect to a unary metric of type $\alpha \to (S, \sqsubseteq)$, incorporating the notion of degrees of indifference between degrees of similarity:

$$\mathbf{best}_u\ \mathrm{CB} \hat{=} \{c : c \in \mathrm{CB}, \exists m \in M : (u\, c) \simeq_{(S,\sqsubseteq)} m\}$$

where (as before)

$$M \hat{=} \max_{(S,\sqsubseteq)}\{u\, c : c \in \mathrm{CB}\}$$

In words, if M is the set of maximal metric values, the best cases are the cases whose metric values are sufficiently close to a value in M (as determined by the indifference function, $\simeq_P$).

Users have quite some control over the way they use this idea by deciding just how much indifference between degrees of similarity they will exercise. The more they are indifferent between quite distant degrees of similarity, the more cases they will see. At one extreme, if they are indifferent between all different degrees of similarity, they will see all cases in the case base. At the other extreme, they can choose to exercise no indifference at all. In this case $\simeq_P$ is $=$ (i.e. users are indifferent between two degrees of similarity only if they are equal). When $\simeq_P$ is $=$, the revised definition of **best** computes the same answer set as our original definition of **best**.

For example, suppose the user is looking at the cases in Figure 1a, computed using u^{95}_{price2}. If $abs(s_1 - s_2) < 2$ is the indifference function, then the new definition of **best** will retrieve cases with maximal similarity to the probe (-4) and cases whose similarity to the probe is within one of the maximal value. The effect will be to retrieve the cases shown in Figure 5a. Using $abs(s_1 - s_2) < 10$ would compute the larger result set in Figure 5b.

(a) $\mathbf{best}_{u^{95}_{price2}}$ with indifference function $abs(s_1 - s_2) < 2$		
E	100	$\{d\}$
M	99	$\{d, w\}$
T	99	$\{d, f, m, w\}$
U	91	$\{\}$

(b) $\mathbf{best}_{u^{95}_{price2}}$ with indifference function $abs(s_1 - s_2) < 10$		
C	102	$\{d, m, s, w\}$
E	100	$\{d\}$
J	104	$\{\}$
K	104	$\{d, m, w\}$
L	88	$\{d, m, w\}$
M	99	$\{d, w\}$
N	85	$\{d, m\}$
Q	88	$\{d, m\}$
S	88	$\{d, m\}$
T	99	$\{d, f, m, w\}$
U	91	$\{\}$
Z	104	$\{f, m, s\}$

Figure 5: Query results

More interesting is an example that uses $\sim_{amen2}$, whose result type is not a total order. Figure 1b shows the result computed by the original definition of **best**. The set of maximal degrees of similarity to the probe was $\{\{d, f, m, s\}, \{d, m, s, w\}, \{d, f, m, w\}, \{d, f, s, w\}\}$. Suppose users are indifferent between two degrees of similarity if they include wheelchair access: $w \in s_1 \wedge w \in s_2$. This is a quite 'loose' definition of indifference; it means users are indifferent between degrees of similarity such as $\{d, m, s, w\}, \{d, m, w\}, \{f, w\}, \{w\}$ and others. The result we compute using the new definition of **best** and this indifference function, shown in Figure 6, is quite different from anything we could compute using O^n_u.

6.7 Combining the options

We have described six ways a user can proceed in an interaction with an ICRS. But they are not mutually exclusive and there may be great value in combining them.

For example, the user might reasonably desire the following combinations: taking the different ranks *and* using an indifference relation to change the values that are considered to be in each rank; choosing a new probe *and* changing the method of computing similarity; and also picking favoured elements from the current rank, *together with* fetching more elements into the case base. However, we have not yet investigated these in any detail.

$\mathbf{best}_{u_{amen2}^{\{d,f,m,s,w\}}}$ with indifference function $w \in s_1 \wedge w \in s_2$		
A	158	$\{d,f,m,s\}$
B	80	$\{d,f,m,s\}$
C	102	$\{d,m,s,w\}$
D	67	$\{d,f,m,s\}$
F	112	$\{d,m,s,w\}$
G	139	$\{d,m,s,w\}$
K	104	$\{d,m,w\}$
L	88	$\{d,m,w\}$
M	99	$\{d,w\}$
O	67	$\{m,w\}$
P	72	$\{d,f,m,s\}$
R	158	$\{d,f,m,w\}$
T	99	$\{d,f,m,w\}$
V	115	$\{d,f,s,w\}$

Figure 6: Query results

7 Conclusions

We have considered various ways in which an interaction with an ICRS may proceed. Some of these ways are new; they exploit the main aspect of our similarity metric framework, *viz* that the result types of similarity metrics can be any partial order, rather than a total order.

Although we have not described it here, we have implemented a system that can search a case base using similarity metrics. It offers all the different interaction options that we have described in this paper. However, it has only a command-line user-interface.

Future work will include proper consideration of the user-interface, with reference to work such as [3], [5] and [1]. In a direct manipulation user-interface, we will consider which of the interaction options that we have discussed are useful and natural.

References

[1] Aha,D.W., Maney,T. & Breslow,L.A.: Supporting Dialogue Inferencing in Conversational Case-Based Reasoning, in B.Smyth & P.Cunningham (eds.), *Advances in Case-Based Reasoning (Procs. of 4th European Workshop on Case-Based Reasoning)*, LNAI-1488, pp.262-273, Springer, 1998

[2] Bridge, D.G.: Defining and Combining Symmetric and Asymmetric Similarity Measures, in B.Smyth & P.Cunningham (eds.), *Advances in Case-Based Reason-*

ing (Procs. of 4th European Workshop on Case-Based Reasoning), LNAI-1488, pp.52-63, Springer, 1998

[3] Dearden, A.M.: Improving the Interface to Interactive Case Memories, in I.D.Watson (ed.), *Progress in Case-Based Reasoning (Procs. of 1st U.K. Workshop on Case-Based Reasoning)*, LNAI-1020, pp.73-84, Springer, 1995

[4] Ferguson, A. & Bridge, D.G.: Generalised Prioritisation: A New Way of Combining Similarity Metrics, in Derek Bridge *et al.* (eds.) *Procs. of 10th Irish Conference on Artificial Intelligence & Cognitive Science (AICS'99)*, pp.137-142, 1999

[5] Hammond,K.J., Burke,R. & Schmitt,K.: A Case-Based Approach to Knowledge Navigation, in D.B.Leake, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, pp.125-136, MIT Press, 1996

[6] Matuschek, D. & Jantke, K.P.: Axiomatic Characterization of Structural Similarity for Case-Based Reasoning, in D.D.Dankel (ed.), *Procs. of Florida AI Research Symposium*, Florida AI Research Society, pp.432-436, 1997

[7] Osborne,H.R. & Bridge,D.G.: Similarity Metrics: A Formal Unification of Cardinal and Non-Cardinal Similarity Measures, in D.B.Leake & E.Plaza (eds.), *Case-Based Reasoning Research and Development (Procs. of 2nd International Conference on Case-Based Reasoning)*, LNAI-1266, pp.235-244, Springer, 1997

[8] Osborne, H. & Bridge, D.G.: We're All Going on a Summer Holiday: An Exercise in Non-Cardinal Case Base Retrieval, in G.Grahne (ed.), *Frontiers in Artificial Intelligence and Applications (Procs. of 6th Scandinavian Conference on Artificial Intelligence)*, pp.209-219, IOS Press, 1997

[9] Osborne,H.R. & Bridge,D.G.: Models of Similarity for Case-Based Reasoning, in *Procs. of the Interdisciplinary Workshop on Similarity and Categorisation*, pp.173-179, Edinburgh, 1997

[10] Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases, in M.Veloso & A.Aamodt (eds.), *Case-Based Reasoning Research and Development (Procs. of 1st International Conference on Case-Based Reasoning)*, LNAI-1010, pp.265-276, Springer, 1995

Hotel case base

We use a very small case base to illustrate this paper. It has few cases, and each case has few attributes. Our implementation can comfortably handle much larger case bases; but the purposes of this paper are better served by using a manageable example case base.

We extracted information pertaining to 26 hotels in the Manchester region from Yahoo's hotel shopping service (`http://travel.yahoo.com/destinations/travelocity_hotel/yfinal_hotel.html`). We identify the hotels by unique letters, rather than by names and addresses, and, in this paper, we consider only their lowest nightly room prices and their amenities. The room prices are in U.S. dollars. The amenities are shown on the Yahoo web site as up to 10 highlighted icons representing different amenities (dining room, swimming pool, etc.). To keep the examples small, we considered only 4 of the 10 amenities, and we have represented them as sets of letters: d designates a dining room, f a fitness room, m meeting rooms, s a swimming pool and w wheelchair access.

Identifier	**Lowest price**	**Amenities**
A	158	$\{d, f, m, s\}$
B	80	$\{d, f, m, s\}$
C	102	$\{d, m, s, w\}$
D	67	$\{d, f, m, s\}$
E	100	$\{d\}$
F	112	$\{d, m, s, w\}$
G	139	$\{d, m, s, w\}$
H	80	$\{\}$
I	150	$\{d, m\}$
J	104	$\{\}$
K	104	$\{d, m, w\}$
L	88	$\{d, m, w\}$
M	99	$\{d, w\}$
N	85	$\{d, m\}$
O	67	$\{m, w\}$
P	72	$\{d, f, m, s\}$
Q	88	$\{d, m\}$
R	158	$\{d, f, m, w\}$
S	88	$\{d, m\}$
T	99	$\{d, f, m, w\}$
U	91	$\{\}$
V	115	$\{d, f, s, w\}$
W	120	$\{\}$
X	131	$\{d, s\}$
Y	77	$\{\}$
Z	104	$\{f, m, s\}$