Computing and Informatics, Vol. 25, 2006, 153–171

SYMBOLIC EXPLANATION OF SIMILARITIES IN CASE-BASED REASONING

Eva Armengol, Enric Plaza

IIIA – Artificial Intelligence Research Institute
CSIC – Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
e-mail: {eva, enric}@iiia.csic.es

Revised manuscript received 3 February 2006

Abstract. CBR systems solve problems by assessing their similarity with already solved problems (cases). Explanation of a CBR system prediction usually consists of showing the user the set of cases that are most similar to the current problem. Examining those retrieved cases the user can then assess whether the prediction is sensible. Using the notion of symbolic similarity, our proposal is to show the user a symbolic description that makes explicit what the new problem has in common with the retrieved cases. Specifically, we use the notion of anti-unification (least general generalization) to build symbolic similarity descriptions. We present an explanation scheme using anti-unification for CBR systems applied to classification tasks. This scheme focuses on symbolically describing what is shared between the current problem and the retrieved cases that belong to different classes. Examining these descriptions of symbolic similarities the user can assess which aspects are determining that a problem is classified one way or another. The paper exemplifies this proposal with an implemented application of the symbolic similarity scheme to the domain of predicting the carcinogenic activity of chemical compounds.

Keywords: Case-based reasoning, explanation, symbolic similarity

1 INTRODUCTION

Explaining the results of automated problem solving systems is a key issue concerning their acceptability and understandability. These explanations have to support the user in both the understanding of the outcome and the process to reach it.

When this process is not clearly explained in a convincing way the user could reject using that problem solving system. Case-based reasoning (CBR) systems predict the solution of a problem based on the similarity between this problem (the *current case*) and already solved problems (cases). Clearly, the key point is the measure used to assess the similarity among the cases. Since the resulting similarity value is difficult to explain, CBR systems often show the retrieved cases (the set of cases that have been assessed as the most similar to the new problem) to the user as an explanation of the prediction: the solution is predicted *because* the problem was similar to the cases shown. Nevertheless, when the cases have a complex structure, simply showing the most similar cases to the user may not be enough.

In this paper we will propose a way to summarize the similarity of a new case with the retrieved cases by means of a symbolic description. These descriptions capture in a symbolic way those aspects of the retrieved cases that are similar to current case; these symbolic similarity descriptions will be shown to the user as an explanation of the CBR prediction for the current case.

In our experience, we observed that for classification problems using the *k-NN* algorithm sometimes the k retrieved cases are classified in different classes. Commonly, these situations are solved using criteria such as the *majority rule* (i.e. the new problem is classified in the same class as the majority of the retrieved cases) to give the classification of the new problem. Nevertheless, this situation needs to be well explained to the user, specially when the majority is not overwhelming (for instance, when $k = 5$ and three of the retrieved cases are in a class $C1$ and the other two cases are in another class $C2$). The approach in this paper is that, in addition to make explicit the similarities of the current case with the retrieved cases, it is useful to make explicit the similarities among the new problem and the retrieved cases in each class separately.

While CBR systems customarily use numeric assessment techniques (usually metrics or distance measures), the key notion in our approach is that of *symbolic similarity.* The use of numeric assessment techniques makes sense to rank the cases according to the importance they have with respect to the current case. Then, the cases at the top of the ranking can be selected as the set of *retrieved cases.* In order to provide explanations, however, numeric values provide less leverage than symbolic explanations. In a nutshell, the notion of *symbolic similarity* between two cases is that of a description that expresses those aspects that are common to (or shared by) these two cases.

In fact, taking the notion of generalization from Machine Learning (ML), we can see that any *generalization* of two cases is a description of *some* aspects they both have in common; in other words, a *symbolic similarity* description can be built by any generalization process. The difference is that generalizations in inductive ML are built to symbolically describe necessary (and often also sufficient) conditions for a case to belong to a class; since many generalizations can be built, inductive ML techniques can then be seen as a search process in the space of generalizations [14]. Moreover, inductive ML techniques often focus on finding discriminant generalizations, i.e. general descriptions that predict a case to be of a specific class and not of any other. In addition, inductive ML techniques have explicit or implicit *biases* with respect to the generalizations they produce or prefer to produce; e.g. one such bias is to prefer shorter generalizations.

Our situation, however, is different: CBR already provides a way to predict a solution. We consider the generalization of two or more cases (e.g. the current case and one or more retrieved cases) as a description of what is similar, what is shared, among them. Moreover, we will not be building discriminant descriptions, instead we will use generalizations as explanations of the current CBR prediction being endorsed by the currently retrieved cases [15]. Clearly, our goal and biases are different form those in inductive ML techniques. Therefore, although a *symbolic similarity* description is technically a *generalization*, the way we use generalizations is different from that of inductive ML. For this reason, and to avoid confusion with ML usage, we will call the explanations we will build *symbolic similarity descriptions.*

2 AN APPROACH TO SYMBOLIC SIMILARITY

The goal of our approach is to explain a CBR system prediction in a way understandable by a user who is an expert in some domain but not in the CBR process itself. In the present paper we propose an explanation scheme for classification problems that is independent of the CBR method used to solve the problem. Our hypothesis is that the result of the retrieval process is a *retrieval set* C with the k cases most similar to the problem; our approach is independent of how the cases in the retrieval set are determined, although in the rest of the paper we will assume they are the most similar cases to the new problem following some k-NN technique.

The explanation scheme we propose is based on applying to the set C the concept of *least general generalization (lgg)*, commonly used in machine learning. The relation $\geq$ is called *more (or equally) general than*, and $g \geq g'$ means that g is more (or equally) general that g' – or equivalently, that g' is more specific than g. The more specific than relation sometimes is noted as $g \sqsubseteq g'$, meaning that g *subsumes* g'. Since relation $\geq$ is an order relation it induces a lattice over a generalization space $\mathcal{G}$. In this lattice, the *lgg* of two generalizations is their corresponding least upper bound. Therefore, we can define the *least general generalization* or *anti-unification* of a collection of descriptions (either generalizations or cases) as follows:

Anti-unification: $AU(d_1, \ldots, d_n) = g$ such that $g \geq d_1 \wedge \ldots \wedge g \geq d_n$ and does not exist a g' such that $g > g'$ and $g' \geq d_1 \wedge \ldots \wedge g' \geq d_n$.

That is to say, g is the most specific generalization of all those generalizations that cover all the descriptions $d_1, \ldots, d_k$. The interpretation of anti-unification from the point of view of symbolic similarity is the following: consider the anti-unification of two cases $g = AU(d_1, d_2)$, then g is the description of all that is common to (or shared by) d_1 and d_2. Therefore, anti-unification builds a symbolic similarity that describes *all* aspects in which two ore more cases are similar.

In the rest of the article we will use the formalism of feature terms to describe both generalizations and cases (see [2] for a more detailed account on feature terms and their anti-unification). A *feature term* was defined as follows:

Feature Terms Given (1) a signature $\Sigma = \langle S, F, \preceq \rangle$ (where S is a set of sort symbols, F a set of feature symbols, and $\preceq$ is a decidable partial order on S such that $\bot$ is the least element called *any*) and (2) a set υ of variables, a *feature term* is an expression of the form

$$\psi ::= X : s[f_1 = \phi_1, \ldots, f_n = \phi_n] \tag{1}$$

where X (the *root* of the feature term) is a variable in υ, s is a sort in S, $f_1, \ldots, f_n$ are features in F, $n \geq 0$, and each ϕ_i is, in turn, a set of feature terms and variables. Notice that when $n = 0$ we are defining a term with no features.

The partial order $\preceq$ gives an informational order among sorts since $s_1 \preceq s_2$ (s_2 is a subsort of s_1) means that s_1 provides *less* information than s_2. Feature terms have an informational order relation among them based of the sort informational order; this relation is called *subsumption*($\sqsubseteq$). Subsumption between two terms $\psi \sqsubseteq \psi'$ (we say that ψ *subsumes* ψ') means that all the information contained in ψ is also contained in ψ' . This relation is the converse of the $\geq$ relation, that is to say $\psi \sqsubseteq \psi'$ means that $\psi \geq \psi'$ (ψ is more general that ψ').

Using the partial order $\preceq$ we can define the *least upper bound (lub)* of two sorts $lub(s_1, s_2)$ as the most specific super-sort common to both sorts. In order to illustrate *feature terms* and the notion of *lub* we will use examples of the Toxicology domain (`ntp.niehs.nih.gov/ntpweb/`). The Toxicology domain has a collection of chemical compounds classified as positive or negative for carcinogenicity on both sexes of two rodent species: rats and mice. We used feature terms to describe the molecular structure of chemical compounds [4] and also we defined an ontology based on the IUPAC nomenclature for chemical compounds. Figure 1 shows the sort hierarchy representing this chemical ontology. The most general sort is *organic-compound* and most specific sorts are the leafs of this hierarchy (e.g. *pentane*, *hexane*, *benzene*, *furane*, etc). Thus, when comparing two sorts, for instance *benzene* and *furane*, *organic-compound* is a super-sort of both. For instance, if we consider *lub(benzene, furane)* we see from Figure 1 that their least upper bound (the most specific super-sort of both) is the sort *monocycle*. Similarly, *lub(benzene, xantene)=ring-system*, and *lub(methane, O-compound)=organic-compound.*

Figure 2 shows an example of chemical compound represented as a feature term called *C-127*. *C-127* is a feature term that has *organic-compound* as root sort, and this root has three features named main-group, radical-set and p-radicals. The values of these features are, in turn, feature terms. Thus, the value of the main-group feature is noted $B1$: *benzene*, meaning it is a feature term $B1$ of sort *benzene* with no features. The feature radical-set has as value a set of three feature terms: $A1$, $N1$, and $C1$. Figure 2 shows that each of the three values is a term: $A1$ is of sort *amine* (corresponding to NH_2), $N1$ is of sort *nitro-derivate* (corresponding

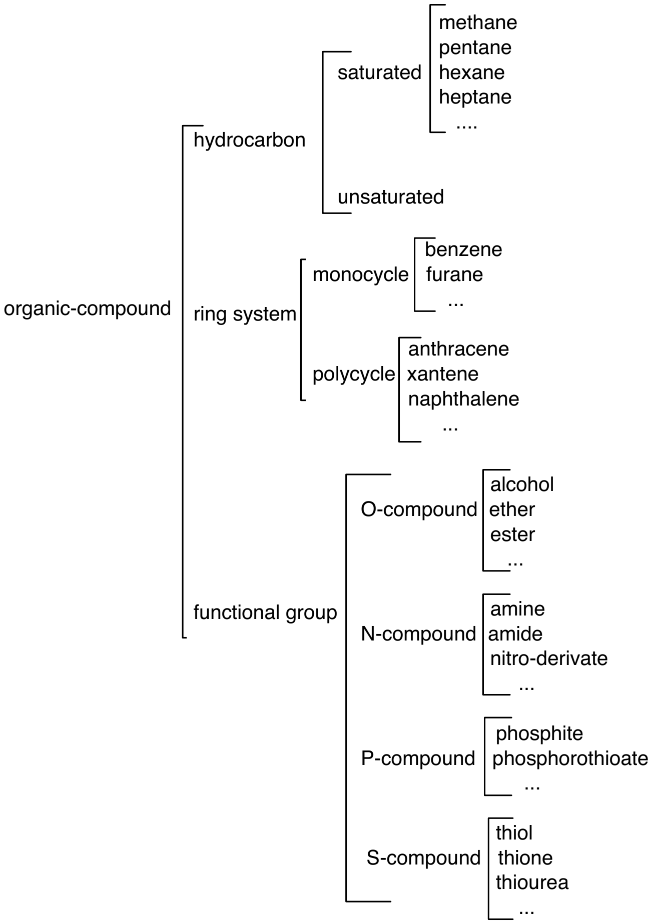
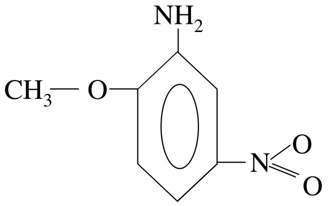


Fig. 1. A partial view of the sort hierarchy in the Toxicology ontology

to the functional group NO_3), and $C1$ is a term with root sort *organic-compound* and has two features named main-group and radical-set. The value of main-group is the feature term $O1$ of sort *ether* (the oxygen O); and the value of radical-set is a feature term $M1$ of sort *methane* (CH_3). Returning to the root of *C-127*, we consider now its feature p-radicals and we see it has as value a set of three feature terms: $p1$, $p2$ and $p3$. The feature term $p1$ (as well as $p2$ and $p3$) is of sort *relative-position* and has two features: radicals and distance. The values of radicals are the same feature terms $A1$ and $N1$ in the feature radical-set. The value of distance is the number 2 (meaning that there is a distance of two carbon atoms between the radicals $A1$ and $N1$). The description of the feature terms $p2$ and $p3$ is similar to the $p1$ description.



```
        ┌ organic-compound
        │  main-group = B1 : benzene
        │  radical-set = A1 : amine
        │                N1 : nitro-derivate
        │
        │                    ┌ organic-compound
        │                 C1 │  main-group = O1 : ether
        │                    └  radical-set = M1 : methane
C-127=  │
        │                   ┌ relative-position
        │ p-radicals = p1   │  radicals = A1  N1
        │                   └  distance = 2
        │                   ┌ relative-position
        │              p2   │  radicals = A1  C1
        │                   └  distance = 1
        │                   ┌ relative-position
        │              p3   │  radicals = C1  N1
        │                   └  distance = 3
        └
```

Fig. 2. A feature term describing the chemical compound *C-127* (*5-nitro-O-anisole*)

The anti-unification of the chemical compounds *C-127* (Figure 2) and *C-084* (Figure 3) is the feature term *AU(C-127, C-084)*, shown in Figure 4. The set of features common to *C-127* and *C-084* is {main-group, radical-set, p-radicals}. For each one of these common features, their values will be anti-unified recursively. Since the value of the feature main-group is *benzene* in both *C-127* and *C-084*, the value of main-group in *AU(C-127, C-084)* is also *benzene*.

The value of the feature radical-set in *C-127* is the set $V_1 = \{A1, N1, C1\}$ and in *C-084* it is the set $V_2 = \{A2, A3, C2\}$. Thus, the anti-unification of V_1 and V_2 will be a set containing three (recursively) anti-unified feature terms, i.e. $AU(V_1, V_2) = \{g_1, g_2, g_3\}$. Anti-unification considers all compatible pairings of elements from V_1 and V_2, anti-unifies them, and returns the three most specific anti-unified values [2]. In this example, the first value is $g_1 = AU(A1, A2)$, i.e. a feature term of sort *amine* with no features. The next most specific anti-unification is $g_2 = AU(N1, A3)$, that yields a feature term of sort *N-compound* without features. Finally, the third most specific anti-unification is $g_3 = AU(C1, C2)$, yielding a feature term of sort *organic-compound* with the features main-group and radical-set whose values are *ether* and *methane* respectively (since both $C1$ and $C2$ have values *ether* and *methane* for those features).

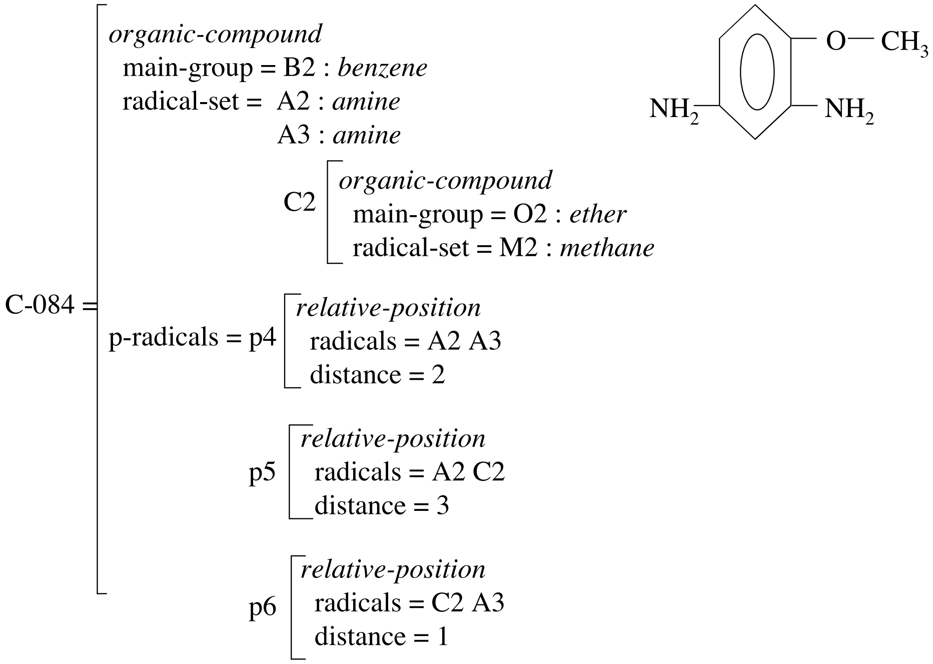


Fig. 3. A feature term describing the chemical compound *C-084* (*2,4-diamino anisole*)

The feature p-radicals in *C-127* has as value the set $V_3 = \{p1, p2, p3\}$ and in *C-084* has as value the set $V_2 = \{p4, p5, p6\}$. Their anti-unification proceeds similarly, and the complete feature term resulting from the process is shown in Figure 4. Notice that the anti-unification of V_3 and V_4 results is three terms (g_4, g_5 and g_6) of sort *relative-position* and that their values for feature radicals are precisely the terms g_1, g_2 and g_3 already produced on the previous step we explained for feature radical-set.

Although we have shown the anti-unification of two feature terms for simplicity sake, anti-unification can be applied to a collection of feature terms (as shown in [2]) obtaining a new feature term with what is shared by all the cases of that collection.

After introducing the process of building the most specific generalization we move to the next section to show how this is used to explain the similarity among cases.

3 THE EXPLANATION SCHEME

This section presents the way in which descriptions resulting from the anti-unification of a collection of cases can be used to provide explanation of the classification of a new problem in CBR systems. Let CB be a case base containing cases classified in one of the solution classes $S = \{S_1, \ldots, S_m\}$. Let us suppose that p is a new

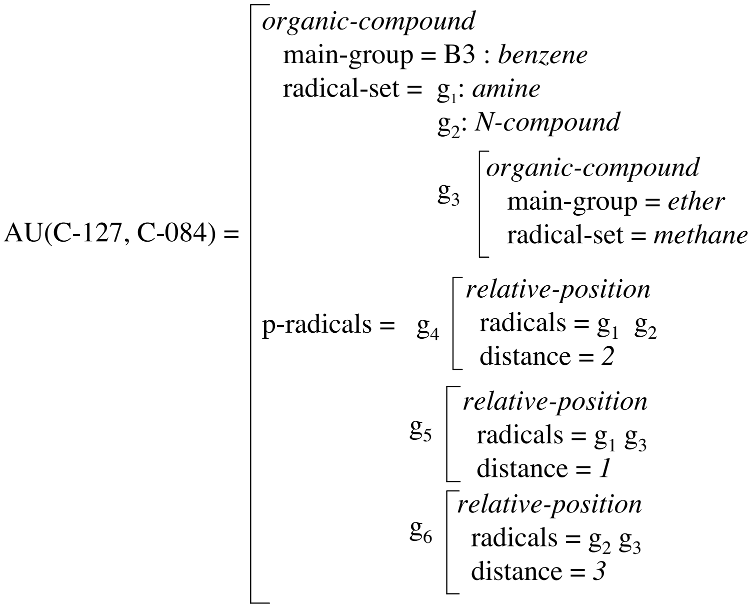


Fig. 4. Anti-unification of the chemical compounds *C-127* and *C-084*

problem to be solved and the *retrieval set* is $C = \{c_1, \ldots, c_k\}$ – i.e. the set of the k cases more similar to c. There are two possible situations:

- cases in C belong to one class S_i
- cases in C belong to several classes.

Concerning the first situation, where a problem p is classified as belonging to S_i, usually the explanation would be to show the user the cases in C. Our approach is that the explanation of why p is in a class S_i is given by what c shares with the retrieved cases in that class. In other words, the anti-unification $AU(c_1 \ldots c_k, p)$ is an explanation of why the cases in C are similar to p, since it is a description of all that is shared among the retrieved cases and the new problem. For instance, consider Figure 5 where the problem is the chemical compound *C-068* and the k-NN algorithm obtains a *retrieval set* with three compounds: *C=(C-074, C-027, C-000)*. Since the three retrieved cases are carcinogenic for mice, *C-068* will also be classified as carcinogenic for mice. The explanation of this classification (as shown in the right hand side of Figure 5) is that all the compounds are saturated hydrocarbons with (at least) three chlorine (Cl) radicals.

However, very often the second situation above with multiple possible solution classes occurs. For simplicity we will consider in our approach that some cases in C belong to one solution class (say S^+) and some others belong to another class (say S^-), but our explanation scheme is also applicable to situations with more than two classes.

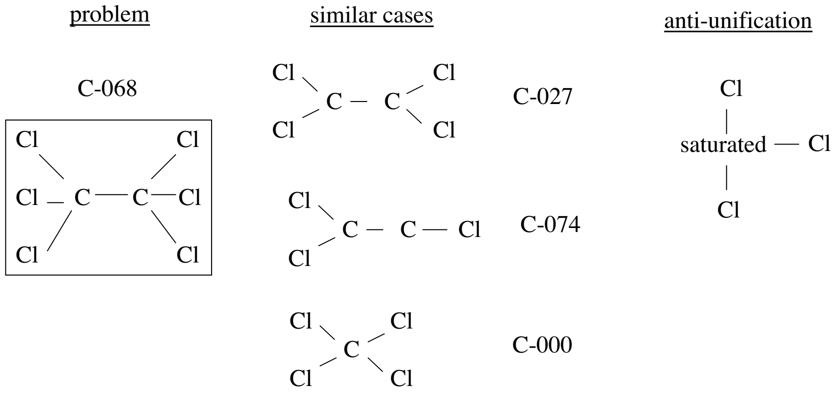


Fig. 5. The problem is the compound *C-068* on the left, while the three compounds in the retrieval set are in the center, and their anti-unification is on the right

Let $C^+ \subseteq C$ the subset of retrieved cases in class S^+, and $C^- \subseteq C$ the subset of retrieved cases in class S^- ($C = C^+ \cup C^-$). In addition to the particular classification of p by using the majority rule or some other aggregation criterion, the user should understand why the cases in C have been considered similar to p. Our approach, as in the first situation above, is to use the anti-unification as explanation, but now we will also do so for each class present in the retrieval set. Assuming two classes, the explanation scheme we propose is composed of three descriptions:

- AU^*: the anti-unification of p with all the cases in C. This description shows what aspects of the problem are shared by all the retrieved cases, i.e. the k retrieved cases are similar to p because they have in common what is described in AU^*.
- AU^+: the anti-unification of p with the cases in C^+. This description shows what has p in common with the cases in C^+.
- AU^-: the anti-unification of p with the cases in C^-. This description shows what has p in common with the cases in C^-.

This explanation scheme supports the user in the understanding of the classification of a problem p. Figure 6 shows the intuitive idea of our approach. The problem p is on the border of the two solution classes. This means that it is similar both to some cases belonging to S^+ and to some other cases belonging to S^-. In fact, in the situation shown in Figure 6 (p similar to 4 cases of the class S^+ and to 3 cases of the class S^-) the only reason to classify p in S^+ is that there is only one more case in C^+ than in C^-. With the explanation scheme we propose, the similarities among p and the cases of each class are explicitly given to the user, who can decide the final classification of p.

AU^* is the anti-unification of all the cases considered as the most similar to p, i.e. is a description containing all the commonalities of the similar cases. When this

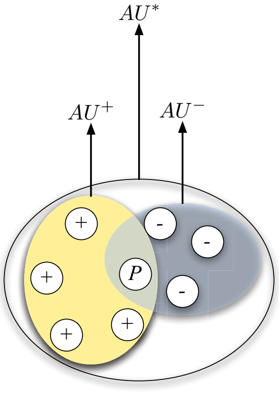
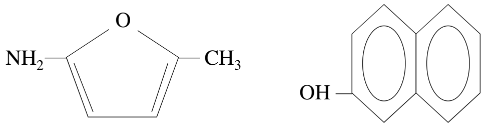


Fig. 6. Construction of the explanation schema for a retrieval set with seven cases



AU* = *organic-compound*
main-group = *ring-system*
radical-set = *organic-compound*

Fig. 7. Molecular structure of the *2-amino, 3-methylfuran* (left) and *6-hydroxynaphtalene* (right), and their anti-unification AU^* expressed as a feature term

description is too general (e.g. most of the features hold the most general sort as value), the meaning is that the cases have low similarity. Conversely, when AU^* is a description with some features holding some specific value, this means that the cases share something more than only the general structure. For instance, the AU^* of the chemical compounds *6-hydroxynaphtalene* and the *2-amino, 3-methylfuran* (shown in Figure 7) is a feature term that describes a molecule that is a ring system (since the *2-amino, 3-methylfuran* is a monocycle and the *6-hydroxynaphtalene* is a polycycle) holding one radical with no specific sort – since the *lub* of the alcohol (OH) and both the amine (NH_2) and the methane (CH_3) is *organic-compound*. Therefore, for this example the AU^* is not very informative. Instead, the explanation of the classification of the chemical compound *C-068* (Fig 5) gives more information since it explains that all the compounds are saturated hydrocarbons with three chlorine radicals.

AU^+ shows the commonalities among the problem p and the retrieved cases belonging to C^+. This allows the user to focus on those aspects that could be relevant to classify p as belonging to C^+. As before, the more specific AU^+ is the more information it gives for classifying p. Notice that AU^+ could be as general as AU^*; in fact, it is possible that both feature terms are equal. This situation means that p has not too many similar aspects with the cases of C^+ that differ from those p shares with C^-. A similar situation may occur with AU^-.

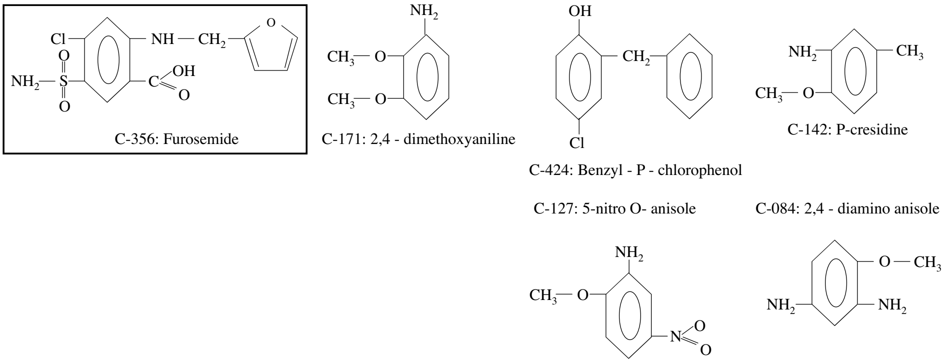

C-356: Furosemide
C-171: 2,4 - dimethoxyaniline
C-424: Benzyl - P - chlorophenol
C-142: P-cresidine
C-127: 5-nitro O- anisole
C-084: 2,4 - diamino anisole
Cl
NH — CH2
O
OH
NH2
CH3
CH2
CH3— O
N

Fig. 8. Molecular structure of the chemical compound *C-356* (left) and of five compounds in the retrieval set

Let us illustrate the complete explanation scheme with an example on the Toxicology domain. Figure 8 shows a chemical compound, namely *C-356*, for which we want to assess its carcinogenicity for male rats. The retrieval set C formed by five chemical compounds considered most similar to *C-356* is also shown in Figure 8. The retrieval set C can be partitioned in two subsets, namely C^+ containing those compounds that are *positive* for carcinogenesis, and C^- containing those compounds that are *negative* for carcinogenesis; specifically, $C^- = \{$*C-242, C-171*$\}$ and $C^+ = \{$*C-084, C-127, C-142*$\}$.

Following our approach, the explanation scheme for chemical compound *C-356* is as follows:

- The description AU^* is the chemical structure shown on the left hand side of Figure 9. The description AU^* shows that *C-356* and the compounds in C have in common that they are all benzenes with at least three radicals: one of these radicals is a functional group derived from the oxygen (i.e. an alcohol, an ether or an acid) called *O-compound* in the figure; another radical (called *rad1* in the figure) is in the position next to the functional group (chemically this means that both radicals are in disposition *ortho*). Finally, there is a third radical (called *rad2* in the figure) with no specific position.
- The description AU^- is the chemical structure shown in Figure 9, and shows that *C-356* and the chemical compounds in C^- have in common that they are benzenes with three radicals: one radical derived from an oxygen (*O-compound*), a radical *rad1* with another radical (*rad3* in the figure) in position *ortho* with the *O-compound*, and finally a third radical (*rad2*) with no specific position.

- The description AU^+ is the chemical structure in Figure 9, and shows that *C-356* and the chemical compounds in C^+ have in common that they are benzenes with three radicals: one of the radicals is derived from an oxygen (*O-compound*), another radical is an *amine* (NH_2) in position *ortho* with the *O-compound*, and a third radical (*rad1*) is at distance 3 of the *O-compound* (chemically this means that both radicals are in disposition *para*).

Using the majority rule, the compound *C-356* will be classified in the class C^+ (positive carcinogenesis) because $card(C^+) = 3$ and $card(C^-) = 2$. The explanation scheme shows to the user the description AU^* that states that all the retrieved compounds are benzenes with three radicals, one of them an *O-compound* in *ortho* position with respect to another radical. Moreover, the description AU^- states that all the compounds with negative carcinogenesis (those in C^-) are also benzenes with three radicals. One of the radicals is an *O-compound* in position *ortho* with another radical that has, in turn, a radical. From both descriptions AU^- and AU^* the user may infer that the position *ortho* among the radical *O-compound* and the radical *rad1* is only important when *rad1* has radicals since in such situation all the compounds in C^- are negative for carcinogenesis.

On the other hand, description AU^+ in Figure 9 states that the compounds in C^+ are also benzenes with three radicals. One of these radicals is an *O-compound* that is in position *ortho* with a radical *amine* (NH_2) and in position *para* with another radical. By comparing both terms AU^+ and AU^- the user may conclude that both the kind of radical in position *ortho* with the *O-compound* and the position of the third radical are important to classify a compound as positive. In other words, from the descriptions AU^- and AU^+ the user is able to observe that the presence of the *amine* may hypothetically be a key factor in the classification of a compound as positive for carcinogenesis. Once the symbolic similarity description gives a key factor (such as the amine in our example), the user can proceed to search the available literature for any empirical confirmation of this hypothesis. In this particular example, a cursory search in the Internet has shown that there is empirical evidence supporting the hypothesis of *amine* presence in aromatic groups (such as benzene) being correlated with carcinogenicity [16, 1].

Finally, in situations where more than two classes are present in the retrieval set, our explanation scheme is simply to build one anti-unification description for each one of them. For instance, if cases in the retrieval set belong to 4 classes the explanation scheme consists on the following symbolic descriptions: AU^*, AU^1, AU^2, AU^3, and AU^4.

4 DISCUSSION

The anti-unification is the least general generalization of a set of cases. Since any kind of generalization is expressed in the same language as the cases, they can be easily understood by a user familiar with the application domain. Very often CBR systems give an explanation consisting on the set of retrieved cases C for

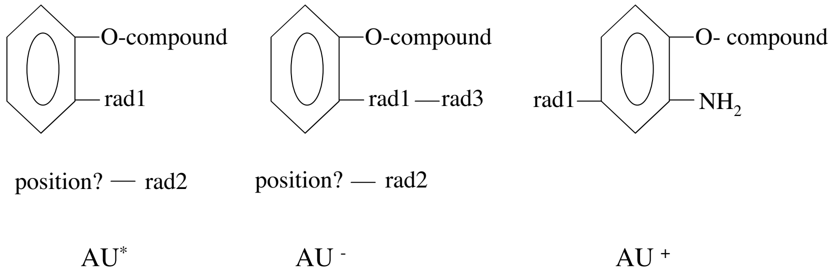


Fig. 9. *AU** is the chemical structure common to all the compounds in Figure 8. *AU*$^-$ is the chemical structure common to *C-356* and the negative compounds (i.e. *C-242* and *C-171*). *AU*$^+$ is the chemical structure common to *C-356* and the positive compounds (i.e. *C-084, C-127* and *C-142*)

a problem p. The main shortcoming of this kind of explanation is that when the cases have a complex structure or when the solution required some adaptation, the user can have some difficulties in understanding how these cases endorse the solution proposed by the CBR system [9, 13]. However, using the explanation scheme we propose, the anti-unification *AU** gives a symbolic explanation of why the cases in C have been considered as the most similar – and also gives a different symbolic description (i.e. *AU*$^+$ and *AU*$^-$) to explain the similarity of the problem p to each class. Examining these symbolic descriptions the user can understand why the problem could be classified as belonging to a class. Notice that this explanation scheme supports the user in taking the final decision to classify a problem when the cases more similar to p belong to different classes, but this explanation is independent of the classification produced by the CBR system.

The anti-unification of two numeric values v_1 and v_2 such that $v_1 \neq v_2$ is, following the definition of least general generalization, the sort *number*. Nevertheless, we propose that features with numeric values have in the explanation the mean of the numbers (i.e. $(v_1 + \ldots + v_n)/n$).

The anti-unification is indeed a generalization but is not a discriminant generalization for a class. That is to say, the anti-unification can cover not only the examples used in the generalization process but also some unseen examples of a different class. As Figure 10 shows, the *AU*$^-$ is the generalization of the problem p and the cases in C^-, nevertheless it can also cover some cases in C^+. The reason why the anti-unification is not discriminant is that *AU*$^-$ is built without using counter-examples, e.g. no case in C^+ is used in the generalization process that builds the description *AU*$^-$ from C^-.

Therefore, the anti-unification gives only an explanation for the problem at hand focusing on what is shared and describing all that is shared. However, it is not a discriminant description that distinguishes cases in C^+ from cases in C^-. For this purpose counterexamples should be used to obtain a generalization, say G^+ for C^+, such that G^+ covers every case in C^+ and none in C^-. Notice that $G^+ \geq$ *AU*$^+$ (i.e.

it is not the least general generalization), and therefore does not contain all that is common to p and C^+. In fact, using a standard top-down induction technique to build G^+ we would usually obtain the *most general* discriminant generalization. Although this approach is useful for inductive learning, from our point of view a lot of useful information about what is shared is lost. This is the reason to use anti-unification for explanations instead of discriminant generalizations as those that can be built, for example, using a decision tree induction technique.

There is also another issue that has to be analyzed in more detail in the future: the overgeneralization of the explanations. A possible situation is that the cases in C^+ (or in C^-) are very different. This means that the anti-unification (a generalization) has to be general enough to be satisfied by all the cases, therefore it is not actually a good explanation for that situation.

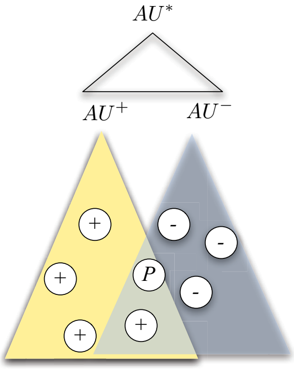


Fig. 10. The anti-unification of a set of cases may cover cases outside that set

5 RELATED WORK

A common form of explanation in CBR is to show the user the case that has been considered as the most similar to the problem at hand. Nevertheless, there is a lot of work focusing on the appropriateness of this explanation [8, 13]. Cunningham et al. [8] performed some experiments on classification tasks in order to evaluate the importance of giving an explanation on the user acceptance of the result. They compared the acceptance of the results of two systems: a CBR and a rule-based system. The experiment showed that the results of the CBR with explanations were more convincing than those of the rule-based system.

McSherry [13] argues that the most similar case (in addition to the features that have been taken as relevant for selecting that case) also has features that could act as arguments against that case. For this reason, McSherry proposes that the explanation of a CBR system has to explicitly distinguish between the case features in favor of an outcome and the case features against it. In this way, the user could decide about the final solution of the problem. A related idea, proposed in [12], is to use the differences among cases to support the user in understanding why some cases do not satisfy some requirements. Finally, several studies by Bridge and Cummins [6] conclude that the cases near the frontiers between classes produce more convincing explanations.

Our approach is based on generating an explanation scheme from the similarities among a problem and a set of cases. As the approaches of McSherry and McCarthy et al., the explanation scheme of our approach is also directed to the user. We make two assumptions: 1) a set C of the most similar cases has been generated from a CBR method, and 2) the cases in C can belong to different classes. From this set of cases, the explanation scheme shows the symbolic similarity of the problem with all the cases retrieved (AU^*) and also with the retrieved cases of each class ($AU^1, AU^2, \ldots, AU^k$). This means that the user can analyze the similarities and, by comparing the descriptions AU^*, $AU^1, AU^2, \ldots, AU^k$, can determine by him/herself the importance of the similarities and the differences among the descriptions. The difference of our approach with that of McSherry is that we explain the result using a set of similar cases whereas McSherry explains it using the similarities and differences within the most similar case compared to the problem at hand.

Other approaches, such as that of Leake [11] and Cassens [7], consider that the form of the explanation should be different depending on the user goals. This statement has been shown in the application presented by Bélanger and Martel [5] where the explanations for expert and novice users are completely different. In particular, these authors propose both 1) to give to expert users more technical explanations (i.e. concordance and discordance matrices) oriented to explain the process that leads to the solution and, 2) to give explanations more intuitive than matrices for novice users. Leake [11] sees the process of explanation construction as a form of goal-driven learning where the goals are those facts that need to be explained and the process to achieve them gives the explanation as result. Cassens [7] uses the Activity Theory to systematically analyze the evolution of a user in using a system, i.e. how the user model is changing. The idea is that in using a system the user can change his/her expectations about it and, in consequence, the explanation of the results would also have to change. In our approach we are considering classification tasks, therefore the user goals are always the same: to classify a new problem. This means that the explanation has to be convincing enough to justify the classification and we assume that the kind of explanation has always the same form, i.e. it does not change along the time.

In this paper we used the notion of symbolic similarity to produce explanations on the performance of CBR systems. In addition, to show the retrieved cases to the user, our proposal also shows the most specific generalizations covering the retrieved cases and the new problem.

Since CBR systems perform lazy learning, and lazy learning builds local approximations of the target concepts, we can view the explanations in this framework. For instance, the retrieved cases in C^+ are an *extensional description* of the local approximation to the carcinogenicity concept, while the most specific generalization AU^+ is an *intensional description* of the local approximation to the carcinogenicity concept. Thus, our approach complements the classical explanation in CBR based on extensional descriptions of the local approximation with several intensional descriptions (AU^*, AU^+, and AU^-) that allow the user to focus on what is shared (and not shared) among the new problem and the retrieved cases.

The idea of *symbolic similarity* was introduced in [3] but was used there to build a discriminant generalization. In this approach, a *symbolic similarity* description is considered as a local approximation of a class description. This local approximation is obtained using the most relevant features of the new problem; then cases that do not satisfy this approximation were discarded. The result is a symbolic description that is satisfied only by cases that belong to one of the classes; thus, that description can be considered as a partial description of the class. This symbolic description can then be used to explain why a problem has been classified into a class and the cases covered by that description form the retrieved set that can be shown also to the user as endorsing the system prediction.

6 CONCLUSIONS AND FUTURE WORK

In this paper we focused on the issue of how to explain to the user the classification given by a CBR system. In particular, we assumed that CBR produces a retrieval set, i.e. a set of k cases considered to be the most similar (under some specific criteria) to the problem at hand. These k cases can belong to different classes, therefore the system has to explain to the user both a) why these cases have been considered as the most similar to the problem (even though they belong to different classes), and b) why the problem could be classified in each one of these classes. Our approach allows to give an explanation scheme composed by several general descriptions, each one explaining different aspects of the CBR process. Thus, one of the descriptions (AU^*) gives symbolic description of what shared by the problem and all the retrieved cases; therefore, the user understands why these k cases have been considered as the most similar – namely the content of the AU^* description.

Each one of the class specific explanations ($AU^1, AU^2, \ldots, AU^m$) describe in a symbolic way the similarities among the new problem and the subset of cases belonging to each class. These explanations give a symbolic description of what is shared by the problem with the retrieved cases of a specific class. Examining the descriptions in the explanation scheme the user can easily understand why these cases have been retrieved and also (as in the example we described) can detect which aspects of these descriptions are determining the prediction of a solution class for the current problem. In addition, the analysis of the explanation scheme can support the user in doing an oriented search in the literature.

There are several lines of research spawning from the approach presented here that we plan to pursue. Concerning the toxicology domain, current ML and statistical techniques have shown a limited proficiency in prediction [10]; the explanation scheme using symbolic similarity that we provide seem to be helpful in improving our understanding of this challenging application domain.

Another line of future research is the use of the symbolic similarity descriptions in a CBR system for purposes of self-assessment. We are interested in developing confidence measures that could allow a CBR system to reliably assess its confidence in each specific prediction. We know that, in general, the symbolic similarity descriptions we use may cover examples and counterexamples with respect to a solution class; our hypothesis is that this fact can be used to estimate a confidence degree for each single prediction. There are several ways in which this assessment can be made and experiments in several data sets are needed to determine their usefulness.

Finally, symbolic similarity descriptions could be used to determine in an adaptive way the granularity of the local approximations; for instance, in a CBR system using k-nearest neighbor symbolic similarity descriptions could be used to determine for each specific problem which value of k offers a better confidence in the predicted solution.

Acknowledgements

This work has been supported by the project SAMAP (TIC2002-04146-C05-01) and CBR-PROMUSIC (TIC2003-07776-C02-02). The authors also thank to Dr. Lluis Bonamusa for his assistance in developing the representation of the chemical compounds.

REFERENCES

[1] Ambs, S.—Neumann, H. G.: Acute and Chronic Toxicity of Aromatic Amines Studied in the Isolated Perfused Rat Liver. Toxicol. Applied Pharmacol. Vol. 139, pp. 186–194, 1996.

[2] Armengol, E.—Plaza, E.: Bottom-Up Induction of Feature Terms. Machine Learning Journal, Vol. 41, 2000, No. 1, pp. 259–294.

[3] Armengol, E.—Plaza, E.: Remembering Similitude Terms in Case-Based Reasoning. In 3rd Int. Conf. on Machine Learning and Data Mining MLDM-03, No. 2734 in Lecture Notes in Artificial Intelligence, pp. 121–130. Springer-Verlag, 2003.

[4] Armengol, E.—Plaza, E.: An Ontological Approach to Represent Molecular Structure Information. In J. L. Oliveira et al., editors, Biological and Medical Data Analysis, ISBMDA 05, Lecture Notes in Computer Science, pp. 294–304, Springer, 2005.

[5] Bélanger, M.—Martel, J. M.: An Automated Explanation Approach for a Decision Support System Based on MCDA. AAAI Fall Symposium on Explanation-Aware Computing, pp. 21–34. AAAI Press, 2005.

[6] Bridge, D.—Cummins, L.: Knowledge Lite Explanation Oriented Retrieval. In AAAI Fall Symposium on Explanation-Aware Computing, pp. 35–42. AAAI Press, 2005.

[7] Cassens, J.: Knowing What to Explain and When. In Proceedings of the ECCBR 2004 Workshops. Technical Report 142-04, pp. 97–104. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain, 2004.

[8] Cunningham, P.—Doyle, D.—Loughrey, J.: An Evaluation of the Usefulness of Case-Based Explanation. In Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003), pp. 122–130, Springer, 2003.

[9] Doyle, D.—Tsymbal, A.—Cunningham, P.: A Review of Explanation and Explanation in Case-Based Reasoning. In Technical report TCD-CS-2003-41, Department of computer Science. Trinity college, Dublin, 2003.

[10] Helma, C.—Kramer, S.: A Survey of the Predictive Toxicology Challenge 2000–2001. Bioinformatics, pp. 1179–1200, 2003.

[11] Leake, D. B.: Issues in Goal-Driven Explanation. Proceedings of the AAAI Spring Symposium on Goal-Driven Learning, pp. 72–79, 1994.

[12] McCarthy, K.—Reilly, J.—McGinty, L.—Smyth, B.: Thinking Positively – Explanatory Feedback for Conversational Recommender Systems. In Proceedings of the ECCBR 2004 Workshops, Technical Report 142-04, pp. 115–124. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain, 2004.

[13] McSherry, D.: Explanation in Recommender Systems. Artificial Intelligence Review, Vol. 24, 2005, pp. 179–197.

[14] Mitchell, T. M.: Generalization as Search. Artificial Intelligence, Vol. 2, 1982, pp. 203–226.

[15] Plaza, E.—Armengol, E.—Ontañón, S.: The Explanatory Power of Symbolic Similarity in Case-Based Reasoning. Artificial Intelligence Review, Special Issue on Explanation in Case-based Reasoning, Vol. 24, 2005, pp. 145–161.

[16] Sorensen, R. U.: Allergenicity and Toxicity of Amines in Foods. In Proceedings of the IFT 2001 Annual Meeting, New Orleans, Louisiana, 2001.

Eva Armengol received her Ph. D. in Computer Science from the Technical University of Catalonia in 1997. Since 1990 she has been working at the Artificial Intelligence Research Institute (IIIA-CSIC). Her research has been focused in the areas of knowledge representation, machine learning, case-based reasoning and learning in multiagent systems.

Enric Plaza is a scientist of the IIIA (Artificial Intelligence Research Institute) in Bellaterra, Catalonia. His research work is currently focused on machine learning and case-based reasoning, and also on how they can be applied to achieve multiagent systems with learning capabilities. Previous research work focused on knowledge acquisition, knowledge modeling, and expert systems validation. The IIIA is a member institute of the Spanish Council for Scientific Research (CSIC) `http://www.iiia.csic.es/People/enric.html`.