Using symbolic descriptions to explain similarity on CBR

Eva Armengol [a,1], and Enric Plaza [a]
[a] *Artificial Intelligence Institute (IIIA-CSIC)*

Abstract. The explanation of the results is a key point of automatic problem solvers. CBR systems solve a new problem by assessing its similarity with already solved cases and they commonly show the user the set of cases that have been assessed as the most similar to the new problem. 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. In particular, we use the notion of anti-unification to build this symbolic description.

Keywords. case-based reasoning, anti-unification, symbolic similarity

1. Introduction

The explanation of the results is a key point of automated problem solvers. These explanations have to support the user in both the understanding of the result and the process to reach it. When this process is not clear and convincing enough the user could reject using the problem solver. There are several ways to explain the results depending on the kind of problem solver and the representation it uses. Thus, rule-based problem solvers can explain the result by showing the rules used to reach a solution. In that way, the user can understand the reasoning process followed by the problem solver and also to detect either lack of knowledge or incorrect rules.

In case-based reasoning (CBR) systems, the solution of a problem is reached based on the similarity among this problem and already solved cases. Here, the key point is the measure used to assess the similarity among the cases. Sometimes the resulting similarity value is difficult to explain, thus CBR systems commonly show the user the set of cases that have been assessed as the most similar to the new problem. Nevertheless when the cases have a complicated structure the inspection could not clarify why these cases have been considered as the most similar.

In this paper we will consider classification problems where the explanation has to justify the membership of a new problem in a solution class, and the generation of such explanations has to be made at the end of each new problem solving process. In our approach, the explanation of the solution given by a CBR method is based on the *symbolic similarity* among the cases. This symbolic similarity is a description of what is shared by the new problem and the most similar cases. Therefore the similarity among

---

[1]Artificial Intelligence Research Institute (IIIA-CSIC) Campus UAB, 08193 Bellaterra, Catalonia (Spain) Tel.: +1 222 333 4444; Fax: +1 222 333 0000; E-mail: eva@iiia.csic.es.

the cases is not a number (as is usual in CBR) but a symbolic description using the same vocabulary of cases; in this way the user can more easily understand the result and the explanation. In particular, we propose to use the *anti-unification* of a set of cases to build the symbolic description to be used as explanation.

The structure of this paper is the following. In the next section we briefly discuss the main issues on explaining the CBR process to the users. Then, in section 3 we explain our proposal for CBR explanation and in section 4 we detail how the anti-unification can be used to explain the result of a CBR method. Finally section 5 summarizes the contributions of this paper.

2. Motivation

Case-based Reasoning is based on the human capability for solving new situations based on the similarity among this new situation and previously solved situations. Aamodt and Plaza [1] defined the CBR cycle as formed by four processes: retrieval, reuse, revise and retain. The *retrieval* consists of, given a new problem to solve, retrieving or selecting from the case base of a subset of the cases most similar to the new problem. An important issue of this task is to select a similarity measure providing an accurate retrieval. The second process of the CBR is the *reuse* that is focused on two aspects: 1) analyzing the differences among the new problem and the retrieved cases and, 2) determining the part of the most similar case that can be useful to reach the solution for the new problem. The third process is the *revision* process consisting on the evaluation of the appropriateness of the modified solution. If the solution is evaluated as correct then the problem and the solution enter to the retain process, otherwise the solution has to be first repaired. Finally, the last process of the CBR is the *retain* that decides whether and how to store the new problem and the solution.

Our approach focuses on the explanations for *retrieve* and *reuse*, i.e. in the explanation concerning the retrieval of the most similar cases and the *reuse* process to justify the similarities and differences among the new problem and the retrieved cases.

In particular, we consider that a main issue is explaining the user a result based on the predictions of a set of most similar cases. In fact, this is the same goal of the recommender systems that have to justify the recommendation of a product according to its similarity among the current user and other users with similar preferences. There are several options for justifying a recommendation. Shimazu [8] in *ExpertClerk* proposes to give the user two different products (both similar to the new problem), and report the advantages and shortcomings on choosing each of them. Herlocker et al. [4] proposes a movie recommender that gives as explanation of the result the histogram of the ratings of acceptation of the same movie by similar users. These approaches explain the result based on similarities of the cases. Other authors, such as Doyle et al. [3] and McSherry [6], point out that sometimes giving as explanation the most similar case and focusing on the similarities is not enough. They claim that sometimes the differences among similar cases may be more explanatory than the similarities specially when the most similar cases have, in fact, more differences than similarities. In particular, McSherry [6] proposes explanations that consist on giving support in favour and against the result.

In our experience, we observed that for classification problems using the *k-NN* algorithm with $k > 1$ sometimes the k cases are classified in different classes. Commonly,

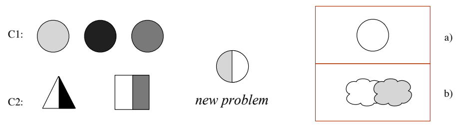

Figure 1. The left part of the figure represents a case base composed of 5 objects: three of them are in class C1 and the other two in the class C2. The goal is to classify the *new problem* in the center. The right part represents the features that the new problem has in common with the objects in class C1 (a) and and those in class C2 (b)

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, these situations need to be well explained to the user, specially when the majority is not significant (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 among all the retrieved cases, it would be useful to make explicit the similarities among the new problem and the retrieved cases in each class separately. Notice that the similarity of the new problem with each one of the classes also makes explicit some differences among the cases in each one of the classes.

For instance, let us suppose the situation in Fig. 1 where the goal is to classify the new problem using *k-NN* with $k = 5$. The k retrieved objects are in two classes, $C1$ and $C2$, and by applying the majority rule, the new problem will be classified in the class $C1$ although there is no clear majority of cases in that class. How this result could be explained to the user? In our approach we propose to give an explanation scheme composed of two kinds of explanations. The first one is a symbolic description justifying the retrieval of the k cases that in the example is that all them have medium size. The second kind of explanations are descriptions of the similarities among the new problem and the cases belonging to each one of the classes. Thus, in the example in Fig. 1, there are two explanations: 1) the new problem could be classified in $C1$ because it is a cercle (Fig. 1a), 2) the new problem could be classified in $C2$ because it is filled with two colors (Fig. 1b). Notice that with these explanations the user could decide which feature, the form or the color, is more relevant for the classification

3. Approach

The goal of our approach is to explain the CBR result in a way understandable by the user. Plaza, Armengol and Ontañón [7] proposed a lazy method that uses a symbolic similarity for the retrieval task. In fact, this symbolic similarity is a description using the same features that describe the cases. Moreover, the symbolic similarity can be seen as an explanation of the result since it is build using the features assessed as the most relevant to classify a new problem.

In the present paper we propose an explanation scheme for classification problems that follows the same idea of the symbolic similarity but is independent of the CBR method used to solve the problem. Our hypothesis is that the result of the retrieval process is a set C with the k cases most similar to the new problem. The explanation scheme we propose is based on the concept of *least general generalization*, commonly used in Machine Learning, on the set C. The relation *more general than* ($\geq_g$) forms a lattice

robot	smiling	holding	has tie	body shape	head shape	class
R1	yes	balloon	yes	square	square	friendly
R2	yes	flag	yes	octagon	octagon	friendly
R3	yes	sword	yes	round	octagon	unfriendly
R4	yes	sword	no	square	octagon	unfriendly
R5	no	flag	no	octagon	round	unfriendly

Table 1. Table describing the five robots introduced in [5]

over a generalization language $\mathcal{G}$. Using the relation $\geq_g$ we can define the *least general generalization* or *anti-unification* of a collection of descriptions (either generalizations or instances) as follows:

- $AU(d_1, ..., d_k) = g$ such that $g \geq_g d_1 \wedge ... \wedge g \geq_g d_k$ and not exists $g' \geq_g d_1 \wedge ... \wedge g' \geq_g d_k$ such that $g >_g g'$

In other words the anti-unification g of a set of descriptions is the most specific generalization of these descriptions in the sense that there is no other generalization g' of all these descriptions that is more specific than g. The anti-unification is a description composed of all the properties shared by the descriptions. Therefore, the anti-unification can be seen as a symbolic description of the similarity among these descriptions.

For instance, let us suppose the domain of Robots [5] consisting of the descriptions of six robots that belong to two solution classes: friendly and unfriendly (see table 1). Each robot is described using five features: *smiling*, *holding*, *has-tie*, *body-shape* and *head-shape*. The anti-unification of the robots R4 and R5 (Table 1) is the following description:

- AU(R4, R5) = (smiling = boolean) and (holding = object) and (has-tie = no) and (head-shape = X) and (body-shape = Y)

The feature *has-tie* in AU(R4,R5) takes as value *no* because both R4 and R5 have as value *no* in this feature. Other features, take as value the *lub* of the values. For instance, *smiling*, take as value *boolean* because the type common to *yes* and *no* that is boolean. The features *head-shape* and *body-shape* take as value X and Y respectively, meaning that could take any of the possible values for shape (i.e. {square, octagon, round}).

The anti-unification of the robots R1 and R2 (Table 1) is the following object:

- AU(R1, R2) = (smiling = yes) and (holding = object) and (has-tie = yes) and (head-shape = X) and (body-shape = X)

The features *smiling* and *has-tie* in AU(R1,R2) takes as value *yes* because both R1 and R2 have as value *yes* in these features. Feature *holding* takes as value *object* because both robots hold a different object. Notice that the value of *head-shape* and *body-shape* is X. This means that both robots have the *same shape* in the head and in the body. See [2] for more information about the anti-unification.

4. The anti-unification as explanation

In this section we will propose how the description resulting from the anti-unification of a set of retrieved cases can be interpreted as the explanation of the classification of a new problem. Let CB be a case base containing cases classified in one of the solution classes $S = \{S_1, ..., S_m\}$. Let us suppose that c is a new problem to be solved and $C = \{c_1, ..., c_k\}$ the set of the k cases more similar to c. There are two possible situations:

- All the cases in C are in one class S_i
- the cases in C are in several classes

Concerning the first situation, most of CBR methods classify c as belonging to S_i and give as explanation of this classification the k cases in C. Our approach is that the explanation of why c is in S_i is given by the features that c shares with all the retrieved cases. In other words, the anti-unification $AU(c_1...c_k, c)$ is a good explanation of why the cases in C have been considered as the more similar to c since the anti-unification is a description of all that is shared by these cases with the new problem. For instance, let R6 be a robot with the following description: (smiling = yes, holding = sword, has-tie = no, body-shape = square, head-shape= round) and let us suppose that the more similar robots are R3 and R4. Because both R3 and R4 are *unfriendly* robots, R6 will also be classified as *unfriendly*. The explanation of the similarity of robots R3, R4 and R6, given by the anti-unification, is AU(R3, R4, R6) = (smile = true, holding = sword); that is, the three robots have in common that they smile and hold a sword while the other features are not relevant for classifying R6.

However, very often the second situation above with multiple possible solution classes occurs. For simplicity we will make our explanations considering that some cases in C belong to one solution class (say S_1) and some others belong to another class (say S_2), but the explanation scheme is also applicable to situations in which C contains cases in several classes. Let $C_1 \subseteq C$ the subset of cases in class S_1, and $C_2 \subseteq C$ the subset of cases in class S_2 ($C = C_1 \cup C_2$). In addition to the particular classification of c by using the majority rule or some other aggregation criterion, the user should understand why the cases in C have been considered similar to c. As we justified in the first situation above, the anti-unification is a good explanation when all the cases in P belong to the same solution class but this is not the situation now. The explanation scheme we propose for this situation is composed of three descriptions:

- AU^0: the anti-unification of c with all the cases in C. This description shows what is shared by the retrieved cases, therefore it explains why the k retrieved cases are similar to c.
- AU^1: the anti-unification of c with the cases in C_1. This description shows what is shared among c and the cases in C_1.
- AU^2: the anti-unification of c with the cases in C_2. This description shows what is shared among c and the cases in C_2.

Let us to illustrate this second situation with an example on the Toxicology data set from the NTP (ntp.niehs.nih.gov/ntpweb/). The goal is to classify a given chemical compound as positive or negative for carcinogenicity on both sexes of two rodent species: rats and mice. In our example the goal is to assess the carcinogenicity of the chemical compound C-356 (shown in Fig. 2) for male rats. Figure 2 also shows the structure of the five chemical compounds (forming the set C) that have been assessed as the most similar cases to C-356. The set C can be partitioned in two subsets, namely C_+ and C_- according to the solution class they belong. Thus the set C_+ contains those compounds that are *positive* for carcinogenesis, and the set C_- contains those compounds that are

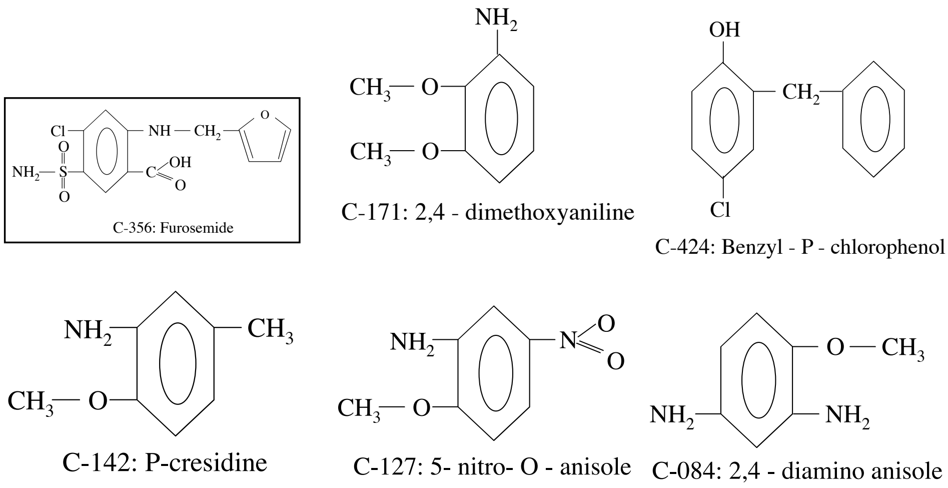

Figure 2. Molecular structure of the chemical compound C-356 and the five compounds that have been retrieved as the most similar to C-356.

negative for carcinogenesis. In particular, in the example in Fig. 2, $C_- = \{$C-424, C-171$\}$ and $C_+ = \{$C-084, C-127, C-142$\}$.

According to the explanation scheme explained above, the explanation for C-356 is as follows:

- The description AU^0 is the chemical structure shown in figure 3a. That is to say, the compounds in C and C-356 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, the there is a third radical (called *rad2* in the figure) that is in no specific position.
- The description AU^- is the chemical structure shown in Figure 3b. This description shows that C-356 and the chemical compounds in C_- have in common that they are benzenes with three radicals: one radical (*O-compound*) derived from the oxygen, 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 3c. This description 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 an oxygen derived (*O-compound*), another radical is an *amine* (NH_2) in position *ortho* with the *O-compound*, the third radical (*rad1*) is at distance 3 of the *O-compound* (chemically this means that both radicals are in disposition $para$).

From the descriptions AU^- and AU^+ (Fig. 3) the user can easily observe the similarities and differences among the compounds in C_- and those in C_+. Thus, similari-

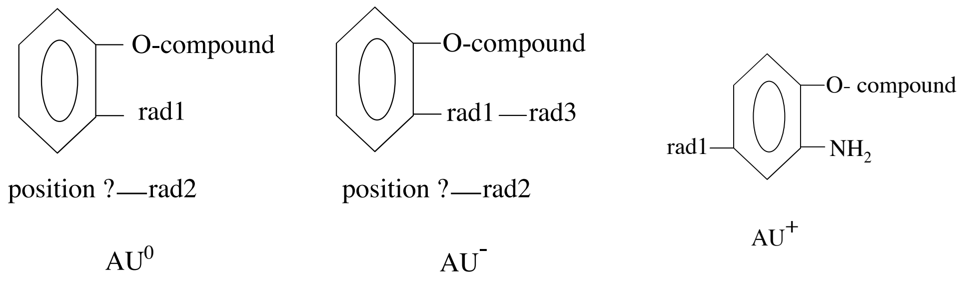

Figure 3. AU^0 is the chemical structure common to all the chemical compounds in Fig. 2. AU^- is the chemical structure common to C-356 and the negative compounds (i.e. C-242 and C-171). AU^+ is the chemical structure common to C-356 and the positive compounds (i.e. C-084, C-127 and C-142).

ties among the compounds in both C_- and C_+ are those in AU^0 (Fig. 3), i.e. they are benzenes with three radicals, one of them an *O-compound* and another radical (*rad1*) in position *ortho* with respect to the *O-compound*. The difference is that for the compounds in C_- the radical in position *ortho* has, in turn, another radical (*rad3*) whereas for the compounds in C_+ the radical in position *ortho* is an *amine* (NH_2). 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.

5. Conclusions

In this paper we discussed how the notion of symbolic similarity can be used 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 the *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^0, AU^+, and AU^-) that allow the user to focus on what is shared (and not shared) among the new problem and the retrieved cases.

As future research we plan to use symbolic similarities in a CBR system for purposes of self-assessment. We are interested in developing confidence measures that could allow a CBR to reliably assess the confidence the system has in each specific solution it predicts. The symbolic similarities we use will cover in general positive and negative cases with respect to a solution class, and this fact can be used to estimate a degree of confidence in a predicted solution.

Acknowledgments

This work has been supported by the project SAMAP (TIC2002-04146-C05-01).

References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7(1):39–59, 1994.

[2] E. Armengol and E. Plaza. Bottom-up induction of feature terms. *Machine Learning*, 41(1):259–294, 2000.

[3] D. Doyle, A. Tsymbal, and P. Cunningham. 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.

[4] J.L. Herlocker, J.A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.

[5] Nada Lavrač and Sašo Džeroski. *Inductive Logic Programming. Techniques and Applications*. Ellis Horwood, 1994.

[6] D. McSherry. Explanation in recommendation systems. In *Proceedings of the ECCBR 2004 Workshops, 125-134. Technical Report 142-04*, pages 125–134. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain, 2004.

[7] E. Plaza, E. Armengol, and S. Ontañón. The explanatory power of symbolic similarity in case-based reasoning. *Artificial Intelligence Review*, to appear, 2005.

[8] H. Shimazu. Expertclerk: A conversational case-based reasoning tool for devevolping salesclerk agents in e-commerce webshops. *Artificial Intelligence Review*, 18:223–244, 2002.