Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference

Toward a Knowledge Transfer Model of Case-Based Inference

Santiago Ontañón* **and Enric Plaza**
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain),
{santi,enric}@iiia.csic.es

Abstract

While similarity and retrieval in case-based reasoning (CBR) have received a lot of attention in the literature, other aspects of CBR, such as case reuse are less understood. Specifically, we focus on one of such, less understood, problems: *knowledge transfer*. The issue we intend to elucidate can be expressed as follows: *what knowledge present in a source case is transferred to a target problem in case-based inference?* This paper presents a preliminary formal model of knowledge transfer and relates it to the classical notion of analogy.

Introduction

Case-based reasoning (Aamodt and Plaza 1994) is a problem solving methodology based on the principle that "similar problems have similar solutions". In CBR, a problem is solved by first *retrieving* one or several relevant cases from a case-base, and then *reusing* the knowledge in the retrieved case (or cases) to solve the new problem. The retrieval stage in CBR has received a lot of attention in the literature. However, other aspects of CBR have received less attention and are less well understood; specifically, what knowledge can be reused from a previous case (source) to solve a new (target) case? There is no generally agreed upon model of this process, which we will call the *knowledge transfer* process.

This paper presents a preliminary model of knowledge transfer in case-based inference (CBI). Case-based inference concerns "exploiting experience in the form of previously observed cases in order to predict the outcome of a new situation" (Hüllermeier 2007). In this paper, we intend to model the process of pure knowledge transfer, without intending to model the complete case reuse process, nor trying to encompass the whole variety of approaches to reuse in case-based reasoning, like rule-based adaptation. The issue we intend to elucidate can be expressed as follows: *what knowledge present in the source case is transferred to a target problem in case-based inference?*

In our approach, we do take into account the notion of similarity; nevertheless we downplay the importance of measuring degrees of similarity, and we focus on a more symbolic notion of similarity. In our approach, it is more important to reason about *what is shared* among cases than the *degree* to which two cases share some of their content.

In this paper we model knowledge transfer based on the notions of refinement, subsumption, partial unification and amalgam, defined over a generalization space. This model is applicable to any representation formalism for which a relevant generalization space can be defined.

Finally, after presenting our model of knowledge transfer, we discuss the relation of the classical notion of analogy with case-based inference, and how our model of knowledge transfer neatly provides a formalization of the the relation between analogy and CBI.

The remainder of this paper is organized as follows. First we will introduce the notion of generalization space, necessary to present our model. Then we will present our formal model of knowledge transfer, and finally relate it with the idea of analogy. After discussing relevant related work, we close the paper with conclusions and future work.

Background

In this paper we will make the assumption that cases are terms in some *generalization space*. We define a generalization space as a partially ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$, where $\mathcal{L}$ is a language, and $\sqsubseteq$ is a subsumption between the terms of the language $\mathcal{L}$. We say that a term ψ_1 subsumes another term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when ψ_1 is more general (or equal) than ψ_2[1]. Additionally, we assume that $\mathcal{L}$ contains the infimum element $\perp$ (or "any"), and the supremum element $\top$ (or "none") with respect to the subsumption order.

Given the subsumption relation, for any two terms ψ_1 and ψ_2 we can define their *unification*, ($\psi_1 \sqcup \psi_2$), which is the *most general specialization* of two given terms:

$$\begin{aligned}\psi_1 \sqcup \psi_2 \quad = \quad \psi : \quad & (\psi_1 \sqsubseteq \psi \ \wedge \ \psi_2 \sqsubseteq \psi) \ \wedge \\ & (\nexists \psi' \sqsubset \psi : \ \psi_1 \sqsubseteq \psi' \ \wedge \ \psi_2 \sqsubseteq \psi')\end{aligned}$$

That is to say, the unifier's content is the addition of the content of the two original terms. However, not every pair of terms may be unified: if two terms have contradictory information then they have no unifier $\psi_1 \sqcup \psi_2 = \top$ —which is equivalent to say that their unifier is "none".

*Currently at Drexel University

[1]In machine learning terms, $A \sqsubseteq B$ means that A is more general than B, while in description logics it has the opposite meaning, since it is seen as "set inclusion" of their interpretations.

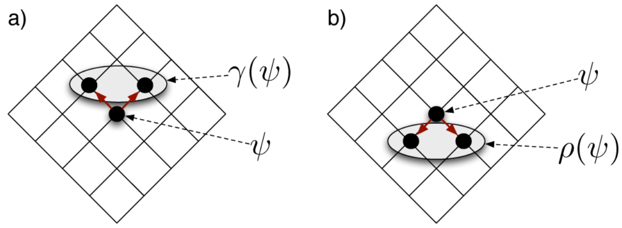

Figure 1: A generalization refinement operator γ, and a specialization operator ρ.

The dual operation to unification is that of *anti-unification*, that is defined as the *least general generalization* of two terms, representing the most specific term that subsumes both. If two terms have nothing in common, then $\psi_1 \sqcap \psi_2 = \bot$. Thus, anti-unification encapsulates in a single description *all* that is shared by two given terms, and is defined as follows:

$$\begin{aligned}\psi_1 \sqcup \psi_2 \;\; = \;\; \psi : \;\; & (\psi \sqsubseteq \psi_1 \;\wedge\; \psi \sqsubseteq \psi_2) \;\wedge \\ & (\nexists \psi' \sqsupset \psi : \;\; \psi' \sqsubseteq \psi_1 \;\wedge\; \psi' \sqsubseteq \psi_2)\end{aligned}$$

Notice that both anti-unification and unification might not be unique. Let us now summarize the basic notions of *refinement operator* over partially ordered sets and introduce the concepts relevant for this paper —see (van der Laag and Nienhuys-Cheng 1998) for a more in-depth analysis. Refinement operators are defined as follows:

Definition 1 *A* downward refinement operator ρ *over a partially-ordered set* $\langle \mathcal{L}, \sqsubseteq \rangle$ *is a function such that* $\rho(\psi) \subseteq \{\psi' \in \mathcal{L} | \psi \sqsubseteq \psi'\}$ *for all* $\psi \in \mathcal{L}$.

Definition 2 *An* upward refinement operator γ *over a partially-ordered set* $\langle \mathcal{L}, \sqsubseteq \rangle$ *is a function such that* $\rho(\psi) \subseteq \{\psi' \in \mathcal{L} | \psi' \sqsubseteq \psi\}$ *for all* $\psi \in \mathcal{L}$.

In other words, upward refinement operators generate elements of $\mathcal{L}$ which are more general, whereas downward refinement operators generate elements of $\mathcal{L}$ which are more specific, as illustrated by Figure 1. Typically, the symbol γ is used to symbolize upward refinement operators, and ρ to symbolize either a downward refinement operator; when specified we will use ρ as a refinement operator in general.

Refinement operators can be used to navigate the space of terms using search strategies, and are widely used in Inductive Logic Programming. For instance, if we have a term representing "a German minivan", a generalization refinement operator would return generalizations like "a European minivan", or "a German vehicle". If we apply the generalization operator again to "a European minivan", we can get terms like "a minivan", or "a European vehicle". A specialization refinement operator would perform the opposite task, and given a term like "a German minivan", would return more specific terms like "a Mercedes minivan", or "a red German minivan". Moreover, in practice it is preferable to have refinement operators that do not perform large generalization or specialization leaps, i.e. that make the smallest possible change in a term when generalizing or specializing.

A Model of Knowledge Transfer

A classical scenario in CBR is where knowledge transfer takes place from one retrieved case to a new problem. It is common to express a case as a problem/solution pair (p, s). In this scenario, solving a problem p' means finding or constructing a solution s' by adapting the solution of the retrieved case (p, s).

In our framework we take a more general approach, where we see a problem and a solution to be two parts of the same description. Thus, we will represent a case as a single term. In this model, an unsolved problem is a partially defined term that needs completion. We will assume a case is an element in a generalization space. Therefore, in our model, a (source) case is a complete description, expressed as a term ψ_s, while a problem (or target case) to be solved is an incomplete description, expressed also as a term ψ_t.

The task of solving ψ_t in our model consists of two steps: (1) (case-based inference) finding a complete description ψ'_t by transferring information from retrieved cases $\psi_1, ...\psi_k$ to ψ_t, and (2) (adaptation) later performing any additional domain specific adaptations required to turn ψ'_t into a valid solution for the domain at hand.

The model presented in this section focuses exclusively on the process of knowledge transfer, rather than on the whole reuse process. Therefore, the outcome of the knowledge transfer process is not a valid solution, but the result of transferring knowledge from the source to the target, which might still need to be adapted by using some domain specific rules, or any other reuse procedure. For that reason, we will refer to the result of knowledge transfer as a *conjecture*. Thus, we say that a conjecture is formed by transferring knowledge from source cases to a target problem —or, in other words, conjectures are the outcome of case-based inference. Some conjectures might constitute solutions, while some others might require adaptation.

There are multiple scenarios that define different knowledge transfer tasks:

- Transfer may be from a single or multiple retrieved cases.
- The problem description ψ_t can be understood as a hard requirement ($\psi_t \sqsubset \psi'_t$) or not (i.e. ψ_t might just express some preferences over the final solution ψ'_t).

For the sake of clarity, in this paper we will only provide a formalization of the single case with hard requirements scenario. However, we will provide insights into how the other situations can be easily modeled in our framework.

Before providing a formalization of knowledge transfer, we will introduce the notions of amalgam and partial unification, that are at the core of our model.

Amalgams

The notion of *amalgam* can be conceived of as a generalization of the notion of unification over terms. The unification of two terms (or descriptions) is a new term, the unifier, which contains all the information in these two terms. Thus, if a term ϕ is a unifier of two other terms ($\phi = \psi_a \sqcup \psi_b$), then all that is true for one of these terms is also true for ϕ. For instance, if ψ_a describes "a red vehicle" and ψ_b describes

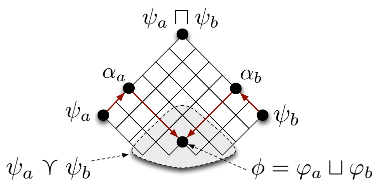

Figure 2: Illustration of the idea of amalgam between two terms ψ_a and ψ_b.

"a German minivan" then their unification ϕ is the description "a red German minivan." Two terms are not unifiable when they possess contradictory information; for instance "a red French vehicle" is not unifiable with "a red German minivan" since being French and German at the same time is not possible for vehicles. The strict definition of unification means that any two descriptions with only one item with contradictory information cannot be unified.

An *amalgam* of two terms (or descriptions) is a new term that contains *parts from these two terms*. For instance, an amalgam of "a red French vehicle" and "a German minivan" is "a red German minivan"; clearly there are always multiple possibilities for amalgams, since "a red French minivan" is another example of amalgam. Thus, all unifications of two terms are amalgams, but not all their amalgams are unifications. The notion of amalgam, as a form of partial unification, was formally defined in (Ontañón and Plaza 2010), where its relationship with the idea of merging operator is also discussed. For the purposes of this paper, we will introduce only the necessary concepts.

Definition 3 (Amalgam) *The set of* amalgams *of two terms ψ_a and ψ_b is the set of terms such that:*

$$\psi_a \curlyvee \psi_b = \{\phi \in \mathcal{L}^+ | \exists \alpha_a, \alpha_b \in \mathcal{L} : \alpha_a \sqsubseteq \psi_a \ \wedge \ \alpha_b \sqsubseteq \psi_b \ \wedge \ \phi = \alpha_a \sqcup \alpha_b\}$$

where $\mathcal{L}^+ = \mathcal{L} - \{\top\}$

Thus, an amalgam of two terms ψ_a and ψ_b is a term that has been formed by unifying two terms α_a and α_b such that $\alpha_a \sqsubseteq \psi_a$ and $\alpha_b \sqsubseteq \psi_b$ —i.e. an amalgam is a term resulting from combining some of the information in ψ_a with some of the information from ψ_b, as illustrated in Figure 2. Formally, $\psi_a \curlyvee \psi_b$ denotes the set of all possible amalgams; however, whenever it does not lead to confusion, we will use $\psi_a \curlyvee \psi_b$ to denote one specific amalgam of ψ_a and ψ_b.

The terms α_a and α_b are called the *transfers* of an amalgam $\psi_a \curlyvee \psi_b$. α_a represents all the information from ψ_a which is *transferred* to the amalgam, and α_b is all the information from ψ_b which is transferred into the amalgam. As we will see later, this idea of transfer is akin to the idea of *transferring* knowledge from the source to target in CBR, and also in computational analogy (Falkenhainer, Forbus, and Gentner 1989).

Intuitively, an amalgam is *complete* when all which can be transferred from both terms into the amalgam has been transferred, i.e. if we wanted to transfer more information, α_a and α_b would not have a unifier.

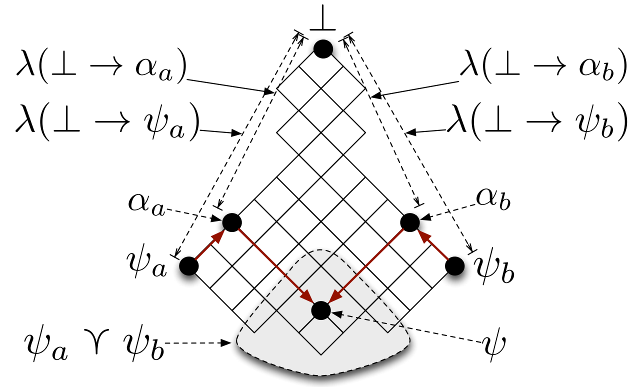

Figure 3: A schema illustrating the preservation degree $\mathbf{p}(\psi)$ of an amalgam ψ.

Definition 4 (Complete Amalgam) *An amalgam $\phi = \psi_a \curlyvee \psi_b$ with transfers α_a and α_b is* complete *when*

$$\forall \alpha'_a, \alpha'_b | \alpha_a \sqsubset \alpha'_a \sqsubset \psi_a \wedge \alpha_b \sqsubset \alpha'_b \sqsubset \psi_b \Rightarrow \alpha'_a \sqcup \alpha'_b = \top$$

that is to say, there are no transfers α'_a and α'_b (that are more specific than transfers α_a and α_b) that have a unifier.

For the purposes of case reuse, we introduce the notion of asymmetric amalgam, where one term is fixed while only the other term is generalized in order to compute an amalgam.

Definition 5 (Asymmetric Amalgam) *The* asymmetric amalgams $\psi_s \overrightarrow{\curlyvee} \psi_t$ *of two terms ψ_s (called* source*) and ψ_t (called* target*) is the set of terms such that:*

$$\psi_s \overrightarrow{\curlyvee} \psi_t = \{\phi \in \mathcal{L}^+ | \exists \alpha_s \in \mathcal{L} : \alpha_s \sqsubseteq \psi_s \ \wedge \ \phi = \alpha_s \sqcup \psi_t\}$$

In an asymmetric amalgam, the target term is transferred completely into the amalgam, while the source term is generalized. The result is a form of partial unification that conserves all the information in ψ_t while relaxing ψ_s by generalization and then unifying one of those more general terms with ψ_t itself. Finally, when an asymmetric amalgam is complete then all knowledge in ψ_s that is consistent with ψ_t is transferred to the solution ψ'_t.

Definition 6 (Preservation Degree) Given an amalgam $\psi \in \psi_a \curlyvee \psi_b$ which is a unification $\psi = \alpha_a \sqcup \alpha_b$ of two terms such that $\alpha_a \sqsubseteq \psi_a$ and $\alpha_b \sqsubseteq \psi_b$, *the* preservation degree $\mathbf{p}$ *for ψ is:*

$$\mathbf{p}(\psi_a, \alpha_a, \psi_b, \alpha_b) = \frac{\lambda(\perp \xrightarrow{\rho} \alpha_a) + \lambda(\perp \xrightarrow{\rho} \alpha_b)}{\lambda(\perp \xrightarrow{\rho} \psi_a) + \lambda(\perp \xrightarrow{\rho} \psi_b)}$$

where $\lambda(\psi \xrightarrow{\rho} \psi')$ is the minimal number of times a refinement operator has to be used to reach ψ' from ψ —i.e. the distance between ψ and ψ' in the generalization space.

As described in (Ontañón and Plaza 2012), $\lambda(\psi \xrightarrow{\rho} \psi')$ is a good measure of the amount of information that ψ' has and ψ does not (assuming all refinements add the same amount of information). Thus, $\mathbf{p}$ (see Figure 3) is the ratio of information preserved in the amalgam ψ with respect to the information present in the original terms ψ_a and ψ_b. The information preserved is measured by the addition of the information contained in the two amalgamable terms α_a and α_b yielding the amalgam ψ. When nothing is preserved $\mathbf{p}$ is 0 since $\psi = \bot$, while $\mathbf{p}$ is 1 when $\psi = \psi_a \sqcup \psi_b$.

As shown in Figure 3, the preservation degree is high when ψ_a and ψ_b had to be generalized very little in order to obtain the amalgam ψ. In other words, if the λ-distances between ψ_a and α_a and between ψ_b and α_b are low, preservation is high. If ψ_a and ψ_b had to be greatly generalized before finding amalgamable generalizations, then the preservation degree of the resulting amalgam will be low.

Knowledge Transfer with Hard Requirements

Let us define the task of knowledge transfer for single case reuse with hard requirements as follows.

Given A case base $\Delta = (\psi_1, \dots \psi_m)$ and a target description ψ_t

Find A complete case ψ_t' such that $\psi_t \sqsubset \psi_t'$ (a *conjecture*)

Clearly, if there is some $\psi_i \in \Delta$ such that $\psi_t \sqsubset \psi_i$ then ψ_i is a solution, and the conjecture can be built simply by unifying query and solution: $\psi_t \sqcup \psi_i = \psi_i$. This specific situation is called in CBR literature "solution copy with variable substitution" (Kolodner 1993). Also, notice that determining whether a case is *complete* or not corresponds to the intuitive notion of whether the case represents a properly specified problem and solution, and is domain dependent.

In general, when there is no case such that $\psi_t \sqsubset \psi_i$, unification is not enough, and knowledge transfer requires the use of amalgams, and in particular of the asymmetric amalgam. Knowledge transfer from a source ψ_s with hard requirements produces *hard conjectures*, defined as follows:

Definition 7 ***(Hard Transfer)*** *A* hard transfer α *for target* ψ_t *from a source* ψ_s *is a term* $\alpha \sqsubseteq \psi_s$ *such that* $\alpha \sqcup \psi_t \neq \top$, *i.e. a generalization of* ψ_s *that unifies with* ψ_t. *Thus, the set of* hard transfers *for target* ψ_t *from a source* ψ_s *is:* $G(\psi_s, \psi_t) = \{\alpha \in \mathcal{L} | \alpha \sqsubseteq \psi_s \wedge \alpha \sqcup \psi_t \neq \top\}$

Definition 8 ***(Hard Conjecture)*** *Given a* hard transfer $\alpha \in G(\psi_s, \psi_t)$, *a* conjecture *for target* ψ_t *is a term in* $\psi_s \overrightarrow{\curlyvee} \psi_t$ *where* α *is the transfer. The set of hard conjectures* K_H *for target* ψ_t *from a source* ψ_s *is* $K_H(\psi_s, \psi_t) = \psi_s \overrightarrow{\curlyvee} \psi_t$.

In order to illustrate our model with an example let us consider the example of room design —introduced in (Ontañón and Plaza 2010). In this example, the goal is to design a room (decide which furniture to have and where to place it) in order to satisfy some goals (e.g. to have an adequate work space). The case base consists of a collection of already designed work rooms with different spatial configurations, and the problem is a new room with a collection of restrictions (e.g. where the door or windows are located). We can see, first of all, that there is no clear distinction between problem and solution. A case is just a complete room design, whereas a problem is just a partially designed room. Figure 4 illustrates our model showing the following elements: a target problem consisting of an incomplete room design (in this case we specified the location of door and windows, and

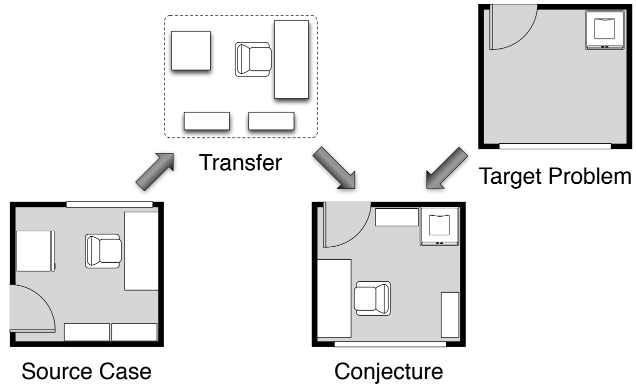

Figure 4: Exemplification of the concepts of source, target, transfer and conjecture in a room design domain.

also that we want a specific metallic cabinet in the north-east corner of the room); a source case consisting of a complete room design of a similar room; a transfer, consisting of the pieces of furniture, and some spatial relations among them (like that the chair goes in front of the table, the table near the window, etc); and finally the conjecture, corresponding to the unification of the transfer and the target problem. In this example, the 'wood cabinet' from the source has been generalized to 'cabinet' in the transfer, and then unified with the 'metallic cabinet' in the target. We show one possible conjecture, but notice that many different conjectures could be formed here, depending on the specific way in which we formalize the generalization space of room designs.

Although reuse from multiple cases is outside of the scope of this paper, it is easy to generalize the notion of transfer to multiple cases, since the amalgam operation can be easily defined amongst a set of terms rather than just two.

The result of CBI is a *conjecture* in the sense that it is a plausible solution for ψ_t. Notice that, (1) a conjecture may be an incomplete solution, and (2) a conjecture is not assured to be correct. Moreover, since there may be more than one conjecture, (3) the issue of which conjecture should be selected has also to be specified. Let us review them in turn.

Conjecture Incompleteness

The purpose of knowledge transfer in case reuse is to transfer to the target as much knowledge as possible (consistent with the target). This "as much as possible" is satisfied if we take as *transfer* a term α that is one of the most specific generalizations of the source that are unifiable with the target. Let $\Gamma(\psi_s, \psi_t)$ be the set most specific terms in $G(\psi_s, \psi_t)$ that satisfy this condition, then we need $\alpha \in \Gamma(\psi_s, \psi_t)$. Nevertheless, some information is lost in the generalization path $\psi_s \xrightarrow{\gamma} \alpha$, which corresponded to the *remainder* (Ontañón and Plaza 2012). Specifically, the remainder $r(\psi, \alpha)$ of a term ψ and a generalization $\alpha \sqsubset \psi$ is a term ϕ such that $\alpha \sqcup \phi = \psi$ (and there is no $\phi' \sqsubset \phi$ such that $\alpha \sqcup \phi' = \psi$). That which is lost from the source case will be called *source differential* in our model.

Definition 9 ***(Source Differential)*** *The source differential ψ_D of a source term ψ_s with respect to a transfer $\alpha \in G(\psi_s, \psi_t)$ is the remainder $r(\psi_s, \alpha)$.*

Therefore, we assumed the source ψ_s to be a consistent and complete case in a case base, but since now the source can be seen as having two parts with respect to the target, namely $\psi_s = \alpha \sqcup r(\psi_s, \alpha)$, and only one of this parts (α) is transferred to the target, we can not assume, in general that the solution for the new case ($\alpha \sqcup \psi_t$) is complete.

Depending on the task a CBR system is performing, this partial solution may be enough. Classical analogy systems take this approach: the goal is to transfer knowledge from source to target —there is no notion of an externally enforced task that demands some kind of completeness to solutions. When a partial solution is not enough, there is usually some form of completeness test that checks possible solutions proposed in the Reuse process. Such a completeness test rejects partial solutions, to which typically the CBR system backtracks to consider other conjectures not yet tested.

Conjecture Correctness

A conjecture $\psi_t \sqcup \alpha$ may be complete, but even so this might be a correct solution or not with respect to ψ_t. If we see ψ_t as a set of requirements that the complete solved target case must satisfy, then if a conjecture $\psi_t \sqcup \alpha$ is complete, then the conjecture $\psi_t \sqcup \alpha$ is correct. Although this supplementary assumptions makes sense in theory (if ψ_t expresses the "requirements" to be satisfied), often CBR systems operate in domains where it is not feasible to assure that ψ_t is complete; it is more reasonable to assume that ψ_t is a partial requirement and the final acceptability or correctness is left to be assessed by the Revise process.

Therefore, knowledge transfer in case reuse produces a solution that is consistent, possibly partial, and not assured to be correct; i.e. produces a conjecture. Since there are multiple knowledge transfers that can produce multiple conjectures, we turn now into the issue of assessing, comparing, and ranking conjectures.

Conjecture Ordering

Multiplicity of complete conjectures for a given source-target pair (ψ_s, ψ_t) may have two causes. The first cause is that $\Gamma(\psi_s, \psi_t)$ is not unique. The second cause is that, even when $\Gamma(\psi_s, \psi_t)$ is unique, more than one source is taken into account: a set of k precedent cases $\mathbb{P}_k = (\psi_1, \ldots, \psi_k)$ produce a set of transfers $\Psi(\mathbb{P}_k) = \Psi_1 \cup \ldots \cup \Psi_k$, which in turn generates a set of conjectures.

Conjectures in $K_H(\mathbb{P}_k, \psi_t)$ may be complete, but from a practical point of view it is useful to rank them according to their estimated plausibility, their degree of completeness, or any other heuristic that can be used in a particular application domain. Typically, the Retrieve phase estimates relevance of precedent cases with some similarity measure, so we can use the similarity degrees $(s_1 \geq \ldots \geq s_k)$ of the k retrieved cases $\mathbb{P}_k = (\psi_1, \ldots, \psi_k)$ to induce a partial order on the set of transfers: $\langle \Psi(\mathbb{P}_k), \geq \rangle = (\Psi_1 \geq \ldots \geq \Psi_k)$. Thus, the conjectures coming from transfers originating in more similar precedent cases (or those transferring more knowledge from more similar cases, in the case of multi-case reuse) are preferred to those from less similar cases.

Since conjectures are in general partial solutions, using some measure that estimates the degree of completeness of conjectures may also be used for ranking conjectures. Domain knowledge can be used to estimates conjecture completeness, but if not available we can use the measure of Preservation Degree (Definition 6) for this purpose.

Definition 10 *The preservation degree ordering of a set of hard conjectures $\langle K_H(\mathbb{P}_k, \psi_t), \prec \rangle$ is as follows:*

$$\forall \psi, \psi' \in K_H(\mathbb{P}_k, \psi_t) : \\ \psi \prec \psi' \Leftrightarrow \mathbf{p}(\psi_s, \alpha, \psi_t, \psi_t) < \mathbf{p}(\psi_s, \alpha', \psi_t, \psi_t)$$

This ordering can be combined with the similarity based ordering to establish a combined partial order on conjectures. We turn now to examine how this model is related to the classical view that case-based inference is analogy.

Knowledge Transfer in Analogy

We turn now to discuss how the classic concept of analogical reasoning (Falkenhainer, Forbus, and Gentner 1989) is related to our model of knowledge transfer. Analogy is typically defined as the process of transferring knowledge or inferences from a particular domain (source) to another particular domain (target).

John Stuart Mill (Shelley 2003) argued that analogy is simply a special case of induction. That is to say, analogy between two situations A and B can be interpreted as having two steps. In a first step we perform an inductive leap. Assume that A and B are similar (we write $A \sim B$), and assume that there is some knowledge α in A not in B (expressed as $A \rightarrow \alpha$). We proceed by induction finding $A \sqcap B$ (that information shared between A and B), and we assume $A \sqcap B$ is the cause of α —i.e. $A \sqcap B \rightarrow \alpha$. Then, in a second step, the inductive assumption $A \sqcap B \rightarrow \alpha$ is used to derive that $B \rightarrow \alpha$ (i.e. that α can also be derived from B).

This view of inductive analogy can be defined as follows in a generalization space.

Definition 11 *Given two terms $\psi_s, \psi_t \in \mathcal{L}$ (called source and target respectively) a formula $\beta \neq \top$ is derived by analogy whenever:*

1. *$\exists \alpha : \alpha \sqsubseteq \psi_s \wedge \alpha \not\sqsubseteq (\psi_s \sqcap \psi_t)$ (α is true in source only)*
2. *$\beta = \alpha \sqcup \psi_t$ (knowledge α is transferred to target)*

where α is the knowledge transferred from source to target.

Since $\alpha \not\sqsubseteq \psi_s \sqcap \psi_t$ we cannot (deductively) derive that α is true in ψ_t. Therefore, this analogical reasoning requires an inductive step, which can be seen as a defeasible or conjectural inference. This "inductive analogy" model is illustrated in Figure 5. The solid lines depict sound inference, i.e. the subsumption relationships among terms (ψ_s, ψ_t and α). Analogy makes some conjectural inferences shown as dotted lines. Specifically, if α is not inconsistent with ψ_t (i.e. $\alpha \sqcup \psi_t \neq \top$) then possibly both situations may also have α in common; this is represented as the term $\phi_\alpha = \alpha \sqcup \psi_t$ that is conjectured to be true. Now, assuming ϕ_α is true, and ψ_t is true, we can then conjecture that $\beta = \psi_t \sqcup \alpha$ is true (i.e. that α can be "transferred to" ψ_t).

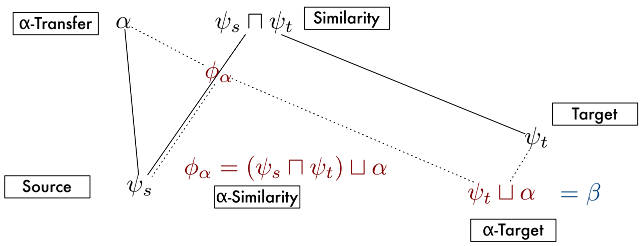

Figure 5: Subsumption relations among the terms involved in analogy $\psi_s \xrightarrow[\psi_t]{a} \beta$.

This conjectural inference can be seen in two ways: induction or knowledge transfer, that nonetheless are equivalent. In the knowledge transfer approach, we derive β by conjecturing α is also true in the target (i.e. we derive $\alpha \sqcup \psi_t$); this is to say, we use the idea of asymmetric amalgam to derive β by transferring α to ψ_t. In the inductive model of analogy we conjecture that the implicit generalization should also include α as being true (that is we move from $\psi_s \sqcap \psi_t$ to ϕ_α). Later we infer from this conjecture that α is true in the target (since it is assumed that also share this) and therefore (deductively) α is true in the target. Figure 5 shows how both views arrive at the same conjecture.

Discussion

This paper has presented a preliminary model of knowledge transfer in case-based inference based on the idea of partial unification. Specifically, we have focused on the situation where cases are represented as terms in a generalization space. In our model, case reuse is seen as having two steps: a first step (case-based inference) where knowledge is transferred from one or several source cases to the target case (called a conjecture), and a second step (adaptation) where the conjecture might need to be adapted. This paper has focused on a model of the first step.

Our model of knowledge transfer offers insights on the relation between case reuse and analogical reasoning. Previous work on relating CBR with analogy has focused on superficial aspects such as CBR being typically intra-domain, where as analogy is inter-domain (Seifert 1989). In our model, we can see that analogical reasoning is related to the knowledge transfer step of case reuse rather than with the second (adaptation) step. An interesting line of future work is the relation of knowledge transfer with conceptual blending (Fauconnier 2001). We have seen that analogy can be likened to an asymmetric amalgam, where as conceptual blending could be seen as a form of symmetric amalgam.

Our work is related to existing general models of case reuse, like (Bergmann and Wilke 1998). However, such models focus on the adaptation step, and typically oversimplify the process of knowledge transfer (transfer is seen as a mere "solution copy" from source to target). We believe that this oversimplification of the knowledge transfer problem is at the root of the difficulty of finding general models of multi-case reuse. The work presented in this paper is a step towards that direction, since we envision it can easily cope with transferring knowledge from multiple sources.

Also related is the work on case-based inference (Hüllermeier 2007), but they focus on classification and regression tasks where the outcome is a form of similarity-based inference. The difference with our work is that we have focused on how complex solutions and conjectures can be formed by aggregating (transferring) knowledge from one or multiple source cases to a partial target case.

Part of our long term goal is understanding case reuse and its relation to other forms of reasoning. We envision case-based inference as a form of conjectural or defeasible inference, like other forms of non-monotonic reasoning (induction, abduction and hypothetical reasoning). The model presented in this paper is one step towards this goal. As future work, we want to formalize the process of knowledge transfer from multiple source cases, and develop case reuse methods based on the idea of knowledge transfer.

Acknowledgements. This research was partially supported by project Next-CBR (TIN2009-13692-C03-01).

References

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

Bergmann, R., and Wilke, W. 1998. Towards a new formal model of transformational adaptation in case-based reasoning. In *European Conference on Artificial Intelligence (ECAI'98)*, 53–57. John Wiley and Sons.

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41:1–63.

Fauconnier, G. 2001. Conceptual blending and analogy. In Gentner, D.; Holyoak, K. J.; and Kokinov, B. K., eds., *The analogical mind Perspectives from cognitive science*, number 1998. MIT Press. 255–285.

Hüllermeier, E. 2007. *Case-Based Approximate Reasoning*, volume 44 of *Theory and Decision Library*. Springer.

Kolodner, J. 1993. *Case-based reasoning*. Morgan Kaufmann.

Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *Case-Based Reasoning. Research and Development, 18th International Conference on Case-Based Reasoning, ICCBR 2010*, 257–271.

Ontañón, S., and Plaza, E. 2012. Similarity measuress over refinement graphs. *Machine Learning Journal* to appear.

Seifert, C. M. 1989. Analogy and case-based reasoning. In *Proc. of a Workshop on Case-Based Reasoning*, 125–129.

Shelley, C. 2003. *Multiple Analogies in Science and Philosophy*. John Benjamins Pub Co.

van der Laag, P. R. J., and Nienhuys-Cheng, S.-H. 1998. Completeness and properness of refinement operators in inductive logic programming. *Journal of Logic Programming* 34(3):201–225.