Selective Reformulation of Examples in Concept Learning

Jean-Daniel Zucker **Jean-Gabriel Ganascia**

LAFORIA-IBP

Abstract

The fundamental tradeoff that is well known in Knowledge Representation and Reasoning affects Concept Learning from Examples too. Representation of learning examples using attribute-value has proved to support efficient inductive algorithms but limited expressiveness whereas more expressive representation languages, typically subsets of First Order Logic (FOL), are supported by less efficient algorithms. In fact, an underlying problem is that of the number of different ways of matching examples, just one in attribute-value representation and potentially large in FOL representation. This paper describes a novel approach to perform representation shifts on learning examples. The structure of these learning examples, initially represented using a subset of FOL-based languages, is reformulated so as to produce new learning examples that are represented using an attribute-value language. What is considered to be an adequate structure varies according to the learning task. We introduce the notion of *morion* (from the Greek) to qualify this structure and show, through a concrete example, the advantages it offers. We then describe an algorithm which reformulates learning examples automatically and go on to analyze its complexity. This approach to deductive reformulation is implemented in the REMO system that has been experimented on the learning of the construction of Chinese characters.

Résumé

Les *exemples* constituent la matière première des algorithmes d'apprentissage symbolique automatique (ASA). Le problème de leur représentation apparaît souvent se réduire au choix, non encore résolu, d'un vocabulaire pour les décrire. Le choix et la remise en question de ce qui, parmi les exemples, constituera un *exemple d'apprentissage*, restent cependant très peu décrits. Nous pensons que ceci vient du fait que dans de nombreux domaines, il existe des entités naturelles (un "patient" en médecine, un "animal" en zoologie, etc.) qui s'imposent comme les entités à représenter. Nous présentons, sur un exemple concret, l'intérêt pour l'ASA d'une remise en cause du rapport entre les connaissances du domaine et leurs descriptions sous forme d'exemples d'apprentissage. Nous qualifions ce rapport à travers la notion de "morion", néologisme grec, défini comme la *partie* du domaine sur laquelle porte la représentation des exemples d'apprentissage. Nous expliquons en quoi la reformulation des exemples d'apprentissage suivant de nouveaux morions permet de réduire la complexité du processus de généralisation en fixant les appariements. Cette reformulation, baptisée REformulation MOriologique, est expérimentée dans le cadre du système REMO, dont la première application complexe est la détection de similarités sur des caractères chinois.

Published in the Proceedings of the eleventh
International Conference in Machine Learning 1994 (pp 352,360)

Selective Reformulation of Examples in Concept Learning

Jean-Daniel Zucker **Jean-Gabriel Ganascia**
LAFORIA-IBP-CNRS
Université Paris VI - 4, Place Jussieu
75252 Paris Cedex 05 - FRANCE
zucker@laforia.ibp.fr ganascia@laforia.ibp.fr

Abstract

The fundamental tradeoff that is well known in Knowledge Representation and Reasoning affects Concept Learning from Examples too. Representation of learning examples using attribute-value has proved to support efficient inductive algorithms but limited expressiveness whereas more expressive representation languages, typically subsets of First Order Logic (FOL), are supported by less efficient algorithms. In fact, an underlying problem is that of the number of different ways of matching examples, just one in attribute-value representation and potentially large in FOL representation. This paper describes a novel approach to perform representation shifts on learning examples. The structure of these learning examples, initially represented using a subset of FOL-based languages, is reformulated so as to produce new learning examples that are represented using an attribute-value language. What is considered to be an adequate structure varies according to the learning task. We introduce the notion of *morion* (from the Greek) to qualify this structure and show, through a concrete example, the advantages it offers. We then describe an algorithm which reformulates learning examples automatically and go on to analyze its complexity. This approach to deductive reformulation is implemented in the REMO system that has been experimented on the learning of the construction of Chinese characters.

1 INTRODUCTION

There is a kind of consensus in the machine learning community regarding the five basic components needed for concept learning from examples: a language for describing the examples, a language to describe the generalizations, background knowledge, a matching predicate to match instances and generalizations, and finally examples and counter-examples of the concepts to be learned (Mitchell, 82). One central aspect of knowledge representation in concept learning is the choice of languages to describe the examples and generalizations (Michalski, 83). Finding a good representation (Amarel, 68), adapted to the learning task, is a complex problem that has not yet been solved (Carbonell, 89). There are mainly two types of research addressing this question: representation analysis and representation synthesis (either *inductive* or *deductive*).

Representation analysis (Cohen, 90; Schlimmer, 87) is concerned with the study of the adequacy of a representation w.r.t. the learning task. The mathematical frameworks proposed (Holte and Zimmer, 89) have been designed to analyze and compare representation shifts; in addition they can represent the context of the learning task as a problem solving task. Methods have also been proposed to evaluate the effectiveness of a set of representations (Saxena, 91; Dietterich, 89).

Representation synthesis is concerned with *modifying* the existing representation to improve the quality of learning (Subramanian, 89). In the area of *inductive* reformulation, improving the quality of learning involves making the results more compact or more operational, or offering better generalizations (Jianping and Michalski, 89). In this area, the main problem is to modify inductively the language describing the learning examples. An approach called constructive induction (Matheus, 91a) aims at lifting some of the limitations of the initial description language. This approach is used both with representation based on attribute-value (Matheus, 91b) and with subsets of FOL (Lapointe, Ling & Matwin, 93).

In the area of *deductive* reformulation, improving the quality of learning involves finding a representation to solve problems within some computational constraints. Most of the work in this research area is concerned with abstracting the languages used for describing instances (Drastal and Czako, 89) by generating various *abstractions* that progressively introduce details of the initial representation (Riddle, 90). Subramanian, for example, proposes an *irrelevance principle* to minimize a formulation by

"*removing all facts that are either logically or computationally irrelevant to the specified goals"* (Subramanian, 90).

Our research belongs to the domain of deductive reformulation inasmuch as we aim at reformulations that are computationally more effective. However, our approach is somewhat different from abstraction. What we are concerned with is the very *nature* of the learning examples rather than just the relevant information or level of abstraction to describe them. We exploit the fact that re-representing the domain examples can lead to performance gains (Flann & Dietterich, 86; Lavrac, Dzeroski & Grobelnik, 91) but what mostly interests us is the entity of the domain that will be described to form the learning examples.

To the best of our knowledge, no work to date has formalized the relation between the domain examples and the entity referred to above. In many domains there are "natural entities" (a "patient" in medicine, an "animal" in zoology, a "molecule" in chemistry, etc.) that impose themselves as the entities to be represented. When an expert in a domain provides examples to illustrate a concept, one is tempted to define learning examples in a language which matches that of the natural entities. However, there are cases where it is necessary to shift these entities in order to achieve a more efficient learning. The purpose of our research is to provide a framework for accommodating such cases (see section 4.2). We shall use a neologism, *morion* (from μοριον[1] meaning "part" in Greek) to designate this part or entity of the domain example to be represented.

In the following section (i.e. section 2), we explain the notion of morion through a concrete example. From a theoretical point of view the cost of concept learning algorithms is directly related to the matching problem, just one in attribute-value representation and potentially large in FOL representation. In the third section we introduce a new approach to representation shift that produces representations where matching is controlled by changing the morion. The interest of this approach in terms of complexity is presented, along with a reformulation algorithm that automates one kind of reformulation. Finally, we present an overview of an implemented system, REMO, along with some results obtained on a concrete application aimed at learning concepts about Chinese characters.

[1] Moriology, literally *discourse on the parts,* is a term borrowed from biology. Moriology refers to the study of the anatomical parts of animals. Aristotle (Aristotle, 90) is the founder of this discipline, which is based on the principle of the analytical study of the organism. The principal operations in this discipline are comparison and analogy.

2 DOMAIN EXAMPLES AND LEARNING EXAMPLES

2.1 EXAMPLE IN CONCEPT LEARNING

According to the dictionary (Sikes, 82), an *example* is "a fact or thing illustrating a general rule". In the machine learning community, depending on the context, an *example* has two different acceptations. The first one corresponds more or less to the idea provided by the dictionary. We shall call it the *domain example*. The second one stands for the *representation* of a domain example using a description language. We shall call it the *learning example*.

Domain examples can be considered as a set of references to objects that point to a concept, i.e. a subset of the *extension* (ZhongZhi, 92) of the concept that it refers to. To represent a domain example in a machine, there is the need for a *data structure* and *semantics* to interpret the formula as *intensional* descriptions. With regards to machine learning, describing domain examples to define the learning examples limits the form of learnable concepts. Indeed, the nature of the entities to be matched by the learner is partially determined by the data structure used to describe the domain examples.

In addition, we suggest that the way to relate domain examples to learning examples is often implicit. For instance, in most inductive systems aiming at learning medical diagnosis in terms of symptoms, the *domain example* "patient" is related to the learning example "description of the patient". Changing the morion could lead to using *learning examples* of smaller morions: physical organ (would produce a diagnosis based on the characteristics of the patient's organs), blood (would produce a diagnosis based on the characteristics of the patient's blood), bacteria (would produce a diagnosis based on the characteristics of the patient's bacteria), DNA (would produce a diagnosis based on the characteristics of the patient's DNA). It could also lead to more general morions such as a small group of patients linked by family ties (would produce a diagnosis based on the characteristics of the patient's family), all patients from a country (would produce a diagnosis based on the characteristics of the patient's neighbors), all patients from a given age range (would produce a diagnosis based on the characteristics of the patient's age group), etc.

The morion establishes a link between the domain examples and the learning examples by qualifying the part of the domain that will be the object of description. The morion captures dynamically the nature and evolution of the structure of the learning example.

2.2 FROM DOMAIN EXAMPLES TO LEARNING EXAMPLES

To clarify the notion of morion, let us take a concrete example. Let us consider five cars of the same make whose performance varies according to the make of tire, "M" or "P", they are equipped with. Figure 1 shows five domain examples with their corresponding performance: *average, bad* or *good*. Let us now consider the problem of learning descriptions of the concept of performance.

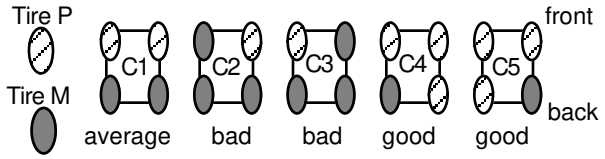

Figure 1: Five domain examples of cars.

It seems *natural* at first to formulate each domain example of a car into *one* learning example, which is a description of that car. The implicit morion is therefore the *car*, inasmuch as each learning example is a *description of a car*. The representation problem consists in finding a language to describe the five examples of cars so as to learn the concept of performance. Let us first consider a powerful representation language based on a subset of FOL, namely Horn clauses:

Table 1: Representation of the cars using Horn Clauses

C1= Perf(C1,average)∨
¬Make(C1,$x_{1,1}$,P)∨¬Make(C1,$x_{1,2}$,P)∨
¬Make(C1,$x_{1,3}$,M)∨¬Make(C1,$x_{1,4}$,M)∨
¬Pos(C1,$x_{1,1}$,front_l)∨¬Pos(C1,$x_{1,2}$,front_r)∨
¬Pos(C1,$x_{1,3}$,back_l) ∨¬Pos(C1,$x_{1,4}$,back_r)
C2= Perf(C2,bad)∨
¬Make(C2,$x_{2,1}$,M)∨¬Make(C2,$x_{2,2}$,P)∨
¬Make(C2,$x_{2,3}$,M)∨¬Make(C2,$x_{2,4}$,M)∨
¬Pos(C2,$x_{2,1}$,front_l)∨¬Pos(C2,$x_{2,2}$,front_r)∨
¬Pos(C2,$x_{2,3}$,back_l)∨¬Pos(C2,$x_{2,4}$,back_r)
C3= Perf(C3,bad)∨
¬Make(C3,$x_{3,1}$,P)∨¬Make(C3,$x_{3,2}$,M)∨
¬Make(C3,$x_{3,3}$,M)∨¬Make(C3,$x_{3,4}$,M)∨
¬Pos(C3,$x_{3,1}$,front_l)∨¬Pos(C3,$x_{3,2}$,front_r)∨
¬Pos(C3,$x_{3,3}$,back_l)∨¬Pos(C3,$x_{3,4}$,back_r)
C4= Perf(C4,good)∨
¬Make(C4,$x_{4,1}$,P)∨¬Make(C4,$x_{4,2}$,P)∨
¬Make(C4,$x_{4,3}$,M)∨¬Make(C4,$x_{4,4}$,P)∨
¬Pos(C4,$x_{4,1}$,front_l)∨¬Pos(C4,$x_{4,2}$,front_r)∨
¬Pos(C4,$x_{4,3}$,back_l)∨¬Pos(C4,$x_{4,4}$,back_r)
C5= Perf(C5,good)∨
¬Make(C5,$x_{5,1}$,P)∨¬Make(C5,$x_{5,2}$,P)∨
¬Make(C5,$x_{5,3}$,P)∨¬Make(C5,$x_{5,4}$,M)∨
¬Pos(C5,$x_{5,1}$,front_l)∨¬Pos(C5,$x_{5,2}$,front_r)∨
¬Pos(C5,$x_{5,3}$,back_l)∨¬Pos(C5,$x_{5,4}$,back_r)

To take into account the semantics of the Position predicate (called Pos in Table 1), let us add some background knowledge *BK* that will be used to generalize the argument values of the Position predicate:

¬Position(c,x,front_l) ∨ Position(c,x,front)
¬Position(c,x,front_l) ∨ Position(c,x,left)
¬Position(c,x,front_r) ∨ Position(c,x,front)
¬Position(c,x,front_r) ∨ Position(c,x,right)
¬Position(c,x,back_l) ∨ Position(c,x,back)
¬Position(c,x,back_l) ∨ Position(c,x,left),
¬Position(c,x,back_r) ∨ Position(c,x,back)
¬Position(c,x,back_r) ∨ Position(c,x,right)

2.3 THE MATCHING PROBLEM

Let us now consider an algorithm to learn descriptions of the concept of performance. We shall restrict ourselves to classical algorithms which learn descriptions of concepts as either conjunctions or a disjunction of conjunctions. The algorithms used in concept learning can be distinguished according to whether or not they perform matching on examples in order to generalize (Kodratoff, 89). When a FOL-based representation is used, it is necessary to perform learning example matching as soon as one predicate is repeated in any of the learning examples. The total number of different matches of a set of clauses is equal to the *product* of all the number of *selections*[2] of each predicate (Zucker & Ganascia, 94a). Let us call N_Sel(Make) the number of selections of the Make predicate in the car problem. The number of possible matches is N_Match = N_Sel(Perf) x N_Sel(Make) x N_Sel(Pos) = 1×4^5 x 4^5 = 1048576. However, Plotkin (Plotkin,70) shows that to obtain a least general generalization, it is only necessary to explore the generalization of all *selections*. The number of all different selections is the *sum* of the number of selections of each predicate. In the car problem, there are potentially $1 + 4^5 + 4^5 = 2049$ selections to explore.

Nevertheless, to generalize all possible selections without constraints as in LGG (Plotkin, 71) leads to high cost algorithms. Taking into account background knowledge does increase even more the number of selections and therefore the general complexity. To be precise, if no learning bias is introduced, the complexity of algorithms supporting representation allowing for more than one matching is exponential with the number of examples [CIGOL (Muggleton and Buntine, 88), AGAPE (Ganascia, 87a), GOLEM (Muggleton and Feng, 90), FOIL (Quinlan, 90), ITOU (Rouveirol, 91)]. In Inductive Learning Programming (ILP), which "*has greater representational power than learners based on attribute-value representations*" (Lapointe & al., 93), the representational power is sacrificed whenever trying to avoid this matching problem. In fact, the difficulty of reasoning with a representation language "*increases as the expressive power of the*

[2] A selection (Plotkin, 70; page 160) captures the notion of predicate matching accross clauses. Let $S=\{C_i \mid i=1,n\}$ be a set of clauses. A set of literals, $K=\{L_i \mid i=1,n\}$ is a *selection* from S iff $L_i \in C_i(i=1,n)$.

language increases" (Levesque & Brachman, 85).

One way of reducing the number of possible selections is to restrict the number of examples to be matched. In KBG (Bisson, 92) the choice of the examples to be matched results from a similarity measure. Although it is efficient, one drawback of such an approach is that firstly, it relies on a numerical distance that needs to be computed between all learning examples, and secondly, that possible fruitful matches may be overlooked. An alternative to this approach involves choosing a representation, such as attribute-value representations, where just one matching is possible. Such representations are used in ID3 (Quinlan, 86), CHARADE (Ganascia, 87), etc. Our claim is that, for a given task, it is always possible to define attribute-value representations which give the same results as those that could be obtained using a FOL based representation, but more efficiently.

There exists methods to re-represent FOL based representations into attribute-value representations. However, these methods do not restrict drastically the number of selections but rather use attribute-value pairs to represent relations (Lavrac et al., 91). Our method relies on representing a restriction on the possible selections, in an attribute-value language. In the car problem, let us restrict the selections to the ones where $x_{i,k}$ is matched with $x_{j,k}$ (see Table 1), which corresponds intuitively to match the tires' position (front_l, front_r, back_l and back_r). Such constraint on the matching allows to re-represent the five cars in an attribute-value language (see Table 2) using the four attributes front_l_Make, front_r_Make, back_l_Make and back_r_Make.

Table 2: The five cars in an attribute-value representation.

Examples / Attributes	**C 1**	**C 2**	**C 3**	**C 4**	**C 5**
Performance	average	bad	bad	good	good
front_l_Make	P	M	P	P	P
front_r_Make	P	P	M	P	P
back_l_Make	M	M	M	M	P
back_r_Make	M	M	M	P	M

With the exception of the over-specialized rules covering each example, no description of the concept of performance (as a disjunction of conjunctions) can be learned inductively from these five learning examples (Table 2). However, it is clear from the domain examples of Figure 1 that one would wish to induce the following concept description R1: "If a tire of make M is in a *front* position of a given car C_X then C_X's Performance is bad". The reason why such a description is not learned is that the background knowledge *BK* is not used in this representation. Indeed, in the chosen selections (i.e. $x_{i,k}$ with $x_{j,k}$) only tires at the same position are matched to one another whereas BK is precisely useful to generalize values of the Position predicate for tires that are at different positions (Zucker & Ganascia, 94a). In order to make use of the background knowledge BK and learn R1, one should have selected a different restriction on the selections than $x_{i,k}$ with $x_{j,k}$. A good restriction on the selections, w.r.t. learning R1, is one that leads to the building of attributes front_Make, back_Make, right_Make and left_Make.

What is to be considered here, is how the adequate matching varies according to the learning task and the background knowledge. Our goal is to associate a semantics to selective matching and find an attribute-value representation that is adequate with regards to the selected matches.

2.4 INTUITIVE APPROACH TO REFORMULATION BY SELECTIVE MATCHING

We propose to change the nature of the entity used to describe the learning example from *car* to *tire*. Let us now represent the five domain examples of cars with 20 new learning examples of tires, arbitrarily indexed T1 to T20. In this representation, each of the four tires of each of the five cars becomes in itself a learning example:

Table 3: Representation of the twenty tires.

Examples / Attributes	T1	T2	T3	T4	...	T20
Ctxt_Ref_car	C1	C1	C1	C1	...	C5
Ctxt_Perf	average	average	average	average	...	good
Make_att	P	P	M	M	...	M
Position_att	front_l	front_r	back_l	back_r	...	back_r

In this description, each tire is represented with its own properties (its position and make), the properties of the car it comes from and a reference to the car (the context of the tire). It is important to note that the tires do not have any Performance as such (tire T1 does not have an "average" Performance) but they are in the *context* of an entity (car C1) that has in itself an average performance. Moreover, the background knowledge *BK* associated to the Position predicate can be used as is by the Position_att attribute. Using this representation, two rules, each of which covers two learning examples of tires (T5 and T10 respectively T16 and T19), can be learned inductively:

If Make_att = M ∧ Position_att = front
Then Ctxt_Performance = bad (R1)
If Make_att = P ∧ Position_att = back
Then Ctxt_Performance = good (R2)

The coverage of these two rules is in terms of *tires*. It would therefore be useful to reformulate the results obtained at the level of the domain examples. Here, by analyzing to which cars the tires covered by R1 and R2 belong, it is possible to map R1 and R2 back into the initial Horn clauses representation as:

¬Position(c,x,front) ∨ ¬Make(c,x,M) ∨ Perf(c,bad) (R'1)
¬Position(c,x,back) ∨ ¬Make(c,x,M) ∨ Perf(c,good) (R'2)

Reformulating the learning examples modifies the possible matching and may therefore be viewed as a learning bias (Ganascia, 91) to define specific matching. The morion is the entity (in the sense of links, structures, etc.) that characterizes the matching. In order to obtain more general results on the real interest of such reformulation, we shall now provide some definitions related to morions and moriological reformulation and study in terms of complexity one representation obtained after a simple reformulation.

3 MORIOLOGICAL REFORMULATION

3.1 DEFINITIONS

Definition 1: *Any entity, be it composed of, or part of, domain examples, shall be called a* ***morion****.*

For each representation of the domain examples in learning examples, the entity that the learning examples describe is called their *attached morion*. A morion composed of domain examples is called a *super-morion*. A morion which is a *part* of the domain examples is called a *sub-morion.*

Definition 2: *A homogenous sub-morion will refer to a sub-morion that occurs the same number of times* ν *in each domain example.* ν *is called the homogeneity factor of the sub-morion.*

In the car problem the *tire* can be considered as a homogenous sub-morion of the car morion with a homogeneity factor of four (see Figure 2). In this problem, if we had considered domain examples of cars including sidecars (sort of "car" with three tires), the sub-morion *tire* would have lost its homogeneity property.

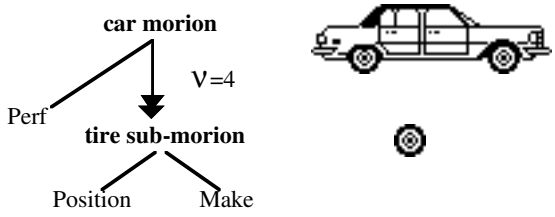

Figure 2: Graphical representation of the morion attached to the learning examples of Table 1.

Definition 3: *A* ***moriological formulation*** *of domain examples is their representation as learning examples using a given morion M. A* ***moriological reformulation*** *is a transformation of learning examples that produces new learning examples using a given morion M. A* ***terminal*** *moriological reformulation produces learning examples where just one matching is possible.*

Formulation and reformulation are therefore different, the first one being a representation of the domain examples in learning examples, the second being a representation shift of learning examples using a new morion. The *initial learning examples* are the learning examples obtained after the first formulation. The initial morion refers to the morion attached to these initial examples.

3.2 REPRESENTING SIMPLE MORIONS

Definition 1 (section 3.1) is a very general definition of a morion. To illustrate some concrete uses of morions, let us consider a restricted language to describe morions: a type of subset of Horn Clauses where only morions containing one sub-morion SM with a homogeneity factor ν may be described. Let us consider the description language in which the $i^{\underline{th}}$ domain example formulated using the morion M is represented by the following Horn Clause:

$$\begin{aligned}
Ex[M,i] = {} & pred_1(i,v_{1i}) & & & \vee \\
& \neg pred_2(i,v_{2i}) & \vee \ldots \vee & \neg pred_\alpha(i,v_{\alpha i}) & \vee \\
& \neg s_pred_1(i,1,v_{1i1}) & \vee \ldots \vee & \neg s_pred_\beta(i,1,v_{\beta i1}) & \vee \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & & \\
& \neg s_pred_1(i,k,v_{1ej}) & \vee \ldots \vee & \neg s_pred_\beta(i,k,v_{\beta ej}) & \vee \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & & \\
& \neg s_pred_1(i,\nu,v_{1e\nu}) & \vee \ldots \vee & \neg s_pred_\beta(i,\nu,v_{\beta e\nu}) & \vee
\end{aligned}$$

In these predicates, "i" labels the domain example and "j" labels the ν occurrences of the sub-morion SM.

- α predicates: $pred_1(i,v_{1i})$,..., $pred_r(i,v_{ri})$,..., $pred_\alpha(i,v_{\alpha i})$ (each predicate occurring once), where the predicates "$pred_r$" and "v_{ri}" qualify the morion's properties and values respectively.

- β predicates: $s_pred_1(i,j,v_{1ij})$,..., $s_pred_s(i,j,v_{sij})$,..., $s_pred_\beta(i,j,v_{\beta ij})$ (each predicate occurring ν times) where the predicates "s_pred_s" and "v_{sij}" qualify the $s^{\underline{th}}$ sub-morion's properties and values respectively.

In the car problem, this corresponds to the representation of Table 1 where α=1, β=2, ν=4, $pred_1$=Perf, s_pred_1=Make and s_pred_2=Position.

3.3 AMORE1: AN ALGORITHM FOR MORIOLOGICAL REFORMULATIONS

This section describes an algorithm, AMORE1, which performs a terminal moriological reformulation of examples formulated using the restricted representation of a morion M described in section 3.2. This reformulation generalizes that described in section 2.4. It takes as input N learning examples E[M,i] represented using the morion M and *reformulates* them using a sub-morion of M composed of μ SM entities (such sub-morion is termed μ–SM). The reformulation presented in section 2.4 can be viewed as a 1-Tire moriological reformulation, where M is a *car* with a homogeneity factor ν=4, SM a *tire* and μ =1. Each learning example of *car* produces ν=4 learning examples of entities composed of one tire.

A 2-Tire reformulation would consist in reformulating each learning example of *car* using an entity composed of two tires. The new learning examples of 2-Tire can be described by 4 different attributes, one attribute Make and one attribute Position for each of the two tires: Make1, Position1, Make2 and Position2. The number of different 2-Tire entities equals the number of arrangement of two tires amongst four, namely A_2^4 =12 entities (two tires T1 and T3 of a given car produce two 2-Tire entities (T1 T3) and (T3 T1) so as to be independent of the tires labeling).

More generally, a reformulation of N examples E[M,i] with the morion μ–SM produces $N \times A_\mu^\nu = N \times \nu! / (\nu-\mu)!$ learning examples E[μ–SM,i,j]. Each initial learning example is reformulated into as many learning examples as the number of possible entities composed of μ parts amongst ν. To describe such entities, 1+α+μ×β attributes are needed: one attribute (Ctxt_Ref_M) to mark the reformulated examples with the label of the learning examples they come from, α attributes ($Ctxt_att_1$, ..., $Ctxt_att_\alpha$) that contain the exact same value as in the learning examples they come from, and μ×β attributes ($S_att_{1,1}$, ..., $S_att_{\mu,\beta}$) to fully describe the morion μ–SM.

The reformulation algorithm AMORE1 (Zucker & Ganascia, 94b) simply consists in reformulating each learning example Ex[M,i] into (ν! / (ν—μ)!) new learning examples of the form Ex[μ–SM,i,j] as a conjunction of attributes:

$$\begin{aligned}
Ex[\mu\text{–}SM,i,j] = {} & Ctxt_Ref_M = i \wedge \\
& Ctxt_att_1 = v_{1i} \wedge \ldots \wedge Ctxt_att_\alpha = v_{\alpha i} \wedge \\
& S_att_{1,1} = v_{1ij_1} \wedge \ldots \wedge S_att_{\beta,1} = v_{\beta ij_1} \wedge \\
& \ldots\ldots\ldots\ldots\ldots \wedge \ldots \wedge \ldots\ldots\ldots\ldots\ldots \wedge \\
& S_att_{\mu,1} = v_{1ij_\mu} \wedge \ldots \wedge S_att_{\mu,\beta} = v_{\beta ij_\mu}
\end{aligned}$$

In the car problem, this corresponds to the representation of Table 3 where α=1, β=2, ν=4, Ctxt_Ref_M = Ctxt_Ref_car, $Ctxt_att_1$ = Ctxt_Perf, $S_att_{1,1}$ = Make_att, $S_att_{2,1}$ = Position_att. After such terminal moriological reformulation the number of learning examples increases; it is therefore necessary to show that the complexity gained in reformulating is not offset by the complexity due to the larger number of learning examples. The following section presents a short analysis of a complexity criterion associated to the AMORE1 algorithm.

3.4 COMPLEXITY CRITERION

Let us now analyze the AMORE1 algorithm in terms of complexity. Let N_Ex be the number of initial learning examples and N_Pred the initial number of predicates.

$$N_Ex = N \quad \text{and} \quad N_Pred = \alpha + \beta$$

The total number of possible selections over the N_Ex learning examples is:

$$N_Sel = \alpha + (\beta \times \nu) \times \nu^{N-1} = \alpha + \beta \times \nu^{N}$$

because each of the β predicates s_pred_i of the ν sub-parts

can match any of the predicates of the ν parts of any of the (N-1) other examples. And there is one selection for each of the α predicates $pred_j$. In the car problem, where α=1, N=5, ν=4 and β=2, the number N_Sel equals 2049.

The complexity of a learning algorithm depends, amongst other factors, on the number of learning examples N_Ex, the number of predicates used in an example description N_Pred and the number of possible selections N_Sel. Let us define a complexity criterion as a polynomial function of these three parameters: $Q \times N_Ex^a \times N_Pred^b \times N_Sel^c$. The term 'Q' represents the influence of other factors than the one mentioned above. Before applying AMORE1 this complexity criterion can be expressed as:

$$Complexity(M) = Q \times N^a \times (\alpha + \beta)^b \times (\alpha + \beta \times \nu^N)^c$$

After applying AMORE1 (cf. 3.3):

$$N_Ex = N \times (\nu! / (\nu-\mu)!) \quad \text{and} \quad N_Pred = 1+\alpha+\mu \times \beta$$

Because each of the N_Pred predicate of Ex[M,i,j] is uniquely matched, the total number of possible selections over the N_Ex learning examples is equal to:

$$N_Sel=N_Pred= 1 + \alpha + \mu \times \beta$$

The new complexity after moriological reformulation is:

$$Complexity(\mu\text{–}SM) = Q \times (N \times \nu!/(\nu-\mu)!)^a \times (1+\alpha+\mu \times \beta)^b \times (1+\alpha+\mu \times \beta)^c$$

The ratio of Complexity(M) over Complexity(μ–SM) is exponential in terms of the number of initial examples: $Q' \times \nu^{N \times c}$. In the car problem, where N=5, μ=1, α=1, ν=4, β=2, and for an algorithm with the following parameters a=b=c=1, this ratio equals 96. Even for very small numbers for a, b, c and N, this ratio confirms the gain in terms of complexity after this moriological reformulation.

3.5 LEARNABILITY AND COMPOSITION OF RESULTS

Thanks to reformulation, the complexity can be reduced from an exponential function to a polynomial function in terms of the number of initial learning examples. After each of these terminal moriological reformulations there is just one possible matching. However, these reformulations restrict the form of the learnable concept, since only those concepts that can be described with properties of less than (or equal to) μ parts can be learned after the reformulation using μ–SM. In the reformulation presented in section 2.4 that can be viewed as a 1-Tire moriological reformulation a rule such as "if three tires of a given car C_X are of make M, then C_X's Performance is good" cannot be learned. In fact such a rule would need a 3-Tire reformulation to be learnable. More generally, the morion characterizes the entities that will be matched. In a μ–SM-reformulation only the concepts that can be expressed with μ sub-parts are learned. In the car problem, if car C6 (see Figure 3) is added to the domain example, no concepts are learned after a reformulation with one tire (both rules R1 and R2 are canceled out by C6). However, by reformulating using a 2-Tire morion, rule R3 can be learned:

If Make1 = M $\wedge$ Make2 = M $\wedge$
Position1 = front $\wedge$ Position2 = back
Then Perf = bad (R3)

One of the drawbacks of moriological reformulation is the *distance* between concepts learned from the learning examples after reformulation and from the initial learning examples. There is a need for recomposition of the results after reformulation. This phase consists in mapping the results back into the initial representation by analyzing the context of the examples covered by the rules obtained after reformulation. For instance rule R3 may be mapped back into R'3 (using the notations of 3.2):

$\neg$Make(c,x,M) $\vee$ $\neg$Position(c,x,front) $\vee$
$\neg$Make(c,y,M) $\vee$ $\neg$Position(c,y,back) $\vee$
Perf(c,bad) (R'3)

4 THE REMO SYSTEM

The REMO system has been developed to assist the user to perform moriological reformulations and so to find incrementally the reformulations that are best fitted for the learning task.

4.1 AN OVERVIEW OF THE SYSTEM

Two operators enabling moriological *reformulation* have been defined within the REMO system, BREAK-UP and AGGREGATE. A third operator has been defined to support moriological *formulation*: DIVIDE-UP. The first two can, in certain cases, be fully automated, but not the last one.

What the first operator does is to break a morion up into smaller morions, and then to reformulate the current learning examples with this new morion. So far the heuristic to choose a new smaller morion is guided by an "Occam's razor" type criteria and uses the AMORE1 algorithm (see Section 3.3). What the second operator does is to aggregate morions into bigger morions, and then to reformulate the learning examples with these new morions. The third and last operator assists the formulation of the domain examples using a new morion and suggests new morions. This latter operator is by essence the most complex one and the least automatable. It is also the one that appears to be the most creative, the most capable of producing original morions.

average
Fig 3 : a new domain example of car

The moriological approach that we propose is directed at

reformulating initial learning examples using a new morion. Since many domains would appear to have a "natural entity" to be described, our approach to example reformulation starts with acquiring an initial morion. In the REMO system, a morion that can be described using the representation defined in Section 3.1. can be acquired. REMO allows the domain examples to be formulated using that initial morion. REMO is coupled to CHARADE (Ganascia, 87), which is used to induce rules over the terminal reformulated representation. CHARADE represents learned concepts as a disjunction of conjunctive rules from a set of learning examples represented by attribute-value pairs. The REMO system's general algorithm can be outlined as follows:

1. Acquisition of the initial morion for example formulation using the DIVIDE-UP operator.
2. Acquisition of the domain examples and background knowledge (expressed as axioms).
3. Automatic definition of a new morion using one of the two operators BREAK-UP or AGGREGATE.
4. Reformulation of the current learning examples using the newly defined morion.
5. Concept learning using CHARADE if the reformulation is terminal else return to phase 3.
6. Mapping the results back into the representation of the initial learning examples .
7. Assessment of the results and return to phase 3, if necessary.

4.2 AN EXPERIMENT WITH CHINESE CHARACTERS

REMO has been experimented within the framework of an interdisciplinary research program called EURECAR whose goal is to provide a system based on machine learning techniques to support the analysis of the construction of Chinese characters (Zucker & Mathieu, 93). Chinese is radically different from other languages as it is neither an alphabetic language nor a phonetic description of the language. In Chinese, a sentence is a set of characters that can be a word or a component of a word. The main facets of a character are its meanings, pronunciations and written form. For example, the Chinese character meaning *feelings or situation* is pronounced qi2ng (this notation, called Pinyin, is one of the three main representations of character pronunciation) and is written «È. It can be described as containing the two radicals ,‡ and «‡ which are respectively identified as "radical 61" and "radical 174" in the Kang Xi dictionary (Zhang, 79) which contains 214 radicals.

To learn concepts related to the characters, it appeared natural to take the entity *character* as the initial morion. The written form can be represented by a set of *radicals* (,‡ and «‡) assembled in various ways (left /right) to form the character («È). We have therefore considered *radical* as a sub-morion. Each radical can be decomposed into a set of elementary strokes ("ª,ÿ≠,ÿØ,"", etc.). We have therefore considered *stroke* as a sub-morion of *radical*. Figure 4 summarizes the structure of the initial non-homogeneous morion acquired using the REMO system. The two ranges ρ_1 (res. ρ_2) designates the number of possible radicals (res. strokes) in each character (res. radical).

Having acquired the initial morion, we then acquired the 400 characters contained in the Bellassen method for learning Chinese characters (Bellassen & Pengpeng, 91). They are amongst the most frequent characters and represent 67% of the characters printed in standard publications in China. In collaboration with sinologues, we have identified about twenty predicates to describe the characters in the representation language introduced in section 3.2. Using the BREAK-UP operator on the sub-morion *radicals*, REMO has produced a corpus of 817 learning examples. In a second reformulation on the sub-morion *stroke*, REMO has produced a corpus of 2761 learning examples.

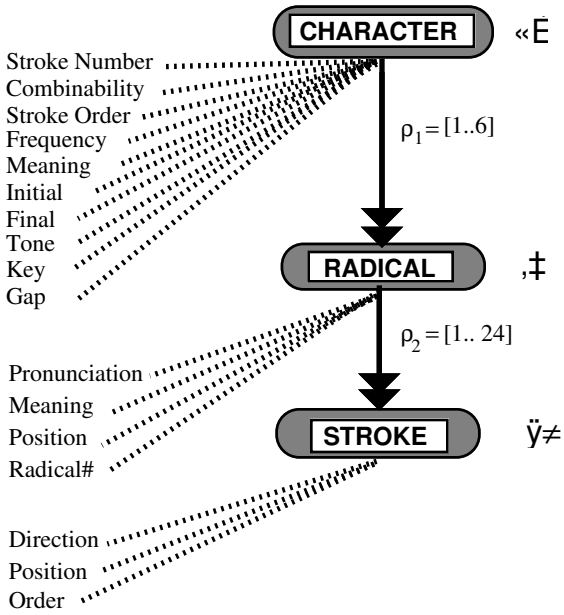

Figure 4: Initial morion *character*.

In the first reformulation, *radicals* were matched to one another and many relations between the presence of a radical and some properties of the characters have been learned. For instance:

¬Radical(x,y) ∨ ¬Radical#(y,«‡)∨ ¬Position(y,"right")
∨ Final(x,"ing")
¬Radical(x,y)∨¬Radical#(y,«‡)∨ ¬Position(y,"right")
∨ Initial(x,"palatal")

5 CONCLUSION

The problem of example matching is crucial in learning from examples. A classical approach to this problem considers *a priori* all the possible selections and then uses a learning bias to restrict those that will be generalized. Our approach is different in that we propose a reformulation of the examples that itself restricts the possible selections. Although this reformulation can be viewed as a deductive reformulation, it is different from the classical approach based on abstraction because we are concerned not so much with the selection of relevant information to characterize examples as with the nature of the new learning examples captured by the notion of morion. Such reformulations are efficient in terms of complexity. We have shown in the case of a simple morion that a complexity criterion would drop from an exponential number of selections in function of the number of examples to a polynomial of this number. Moreover, when a terminal moriological reformulation is obtained it becomes possible to use algorithms based on attribute-value representation.

Nevertheless, in many cases the right morions for a given representation are unknown. It is therefore important to discover appropriate morions by successive reformulations. The REMO system allows morion acquisition, domain example formulation and moriological reformulations to be carried out, and concepts to be learned using the CHARADE algorithm on learning examples resulting from terminal reformulations. In the framework of the REMO system, we are working in three other domains (medical diagnosis, musical composition, protein folding) in order to obtain a strong empirical evaluation of each of the tasks with and without moriological reformulation and also to enrich the moriological operator DIVIDE-UP. This operator is especially useful to guide the user in cases where there are no implicit domain examples or natural entities. Our current work is focused on the acquisition of more complex morions that fit best a given learning task. In the long term, we shall study how our work can be applied as an alternative to selection choice in algorithms used in the context of Inductive Logic Programming.

Acknowledgments

We would like to thank Joël Bellassen, Vincent Corruble, Geber Ramalho who have given us continuous encouragement and technical support, Rosalind Greenstein for proofreading the paper and the anonymous reviewers for their remarks that have enabled us to greatly enhance the paper.

References

Amarel S., "On Representation of Problems of Reasoning about Actions", in *Machine Intelligence* 3, edited by D. Mitchie, pp. 131-171, Edinburgh University Press, 1968.

Aristotle, "*Les parties des animaux*", pp. V-XVIII, Ed. Les Belles Lettres, Paris, 1990.

Bellassen J. & Pengpeng Z., "*Perfectionnement à la langue and à l'écriture chinoises*", pp. 296-347, La Compagnie, Paris, France, 1991.

Bisson G., "Learning in FOL with a Similarity Measure", in *Proceedings of AAAI*, pp. 82-87, San Jose, 1992.

Carbonell J.G., "Paradigms for machine learning", in *Artificial Intelligence 40,* pp. 1-9, North Holland, Amsterdam, 1989.

Cohen W., "An analysis of Representation Shift in concept learning", in *Proceeding of the seventh international conference on Machine Learning*, pp. 104-112, Austin, Texas, 1990.

Dietterich T., "Limitations on inductive learning", in *Proceeding of the sixth International Workshop on Machine Learning*, pp. 124-128, Morgan Kaufmann Publishers, California, 1989.

Drastal G. & Czako G., "Induction in an abstraction space: A form of constructive induction", in *Proceedings* of *the twelfth International Jointed Conference in Artificial Intelligence (IJCAI)*, pp. 708-712, 1989.

Flann N. & Dietterich T., "Selecting Appropriate Representations for Learning from Examples", in *Proceedings of AAAI*, pp. 460-466, 1986.

Ganascia J-G., "CHARADE: A rule System Learning System", in *Proceedings of the tenth International Jointed Conference in Artificial Intelligence (IJCAI)*, Milan, 23-28 August 1987.

Ganascia J-G., "Deriving the Learning Bias from Rule Properties.", Machine Intelligence (12), Oxford University Press, Oxford, 1991.

Holte R. & Zimmer R., "A Mathematical Framework for Studying representation", in *Proceeding of the sixth international Workshop on Machine Learning*, pp. 454-456, Morgan Kaufmann Publishers, California, 1989.

Kodratoff Y., "Characterizing Machine Learning Programs", *Artificial Intelligence, Research Directions in Cognitive Sciences. European Perspectives,* Vol. 5, D. Sleeman and N.O. Bernse (Eds.), Lawrence ErlFraun, 1989.

Lapointe S., Ling C. & Matwin S., "Constructive Inductive Logic Programming", *Proceedings of the 14th IJCAI*, pp. 1030-1036, 1993.

Lavrac N. , Dzeroski, S., & Grobelnik M., "Learning non-recursive definitions of Relations with LINUS.", in *Proceedings of the European Working Session on Learning,* pp. 265-281, Springer-Verlag (482), 1991.

Levesque H. & Brachman R., "A Fundamental Tradeoff in

Knowledge Representation and Reasoning (Final Version)", in *Readings in Artificial Intelligence,* pp. 41-71, Brachman & Levesque Eds, Morgan Kaufmann, 1985.

Matheus C., "The Need for Constructive Induction", in *Proceeding of the eighth international Workshop on Machine Learning*, pp. 173-177, Morgan Kaufmann Publishers, California, 1991a.

Matheus C., "Adding Knowledge to SBL through Feature Construction", in *Proceedings of the European Working Session on Learning,* pp. 803-808, Springer-Verlag (482), 1991b.

Michalski R.S., "A theory and methodology of inductive learning", in *Machine Learning: An artificial intelligence approach,* Tioga, 1983.

Mitchell T. ,"Generalization as Search", *Artificial Intelligence Journal 18*. pp. 203-226, 1982.

Muggleton S. & Buntine W., "Machine invention of first order predicates by inverting resolution", in *Proceeding of the fifth international Workshop on Machine Learning*, pp. 339-352, Morgan Kaufmann Publishers, California, 1988.

Muggleton S. & Feng C., "Efficient Induction of logic program", in P*roceeding of the first conference on algorithmic learning theory,* Tokyo, 1990.

Plotkin G., "A further note on inductive generalization", in *Machine Intelligence* 6, (eds. by B. Meltzer and D. Mitchie), pp. 101-124, Edinburgh University Press, 1971.

Plotkin G., "A note on inductive generalization", in *Machine Intelligence* 5, pp. 153-163, Edinburgh University Press, 1970.

Quinlan J-R., "Induction of Decision Trees", in *Machine Learning* (1), pp. 81-106, 1986.

Quinlan J-R., "Learning Logical Definitions from Relations", in *Machine Learning* (5) pp. 239-266, 1990.

Riddle P., "Automating Problem Reformulation", *Change of Representation and Inductive Bias*, pp. 105-124, edited by D. Benjamin, Kluwer Academic Publisher, Boston, 1990.

Rouveirol C., "Semantic Model of Induction of First Order Theories", in *Proceedings of the 12th IJCAI*, pp. 685-690, 1991.

Saxena S., "Evaluating Alternative Instance Representations", in *Proceedings of the European Working Session on Learning (EWSL-91),* pp. 465-467, Springer-Verlag, 1991.

Schlimmer J., "Incremental Adjustment of Representations for learning", in *Proceeding of the fourth international Workshop on Machine Learning*, pp. 79-90, Morgan Kaufmann Publishers, California, 1987.

Subramanian D., "Representational Issues in Machine Learning", in *Proceeding of the sixth international Workshop on Machine Learning*, pp. 426-429, Morgan Kaufmann Publishers, California, 1989.

Subramanian D., "A theory of justified reformulation", *Change of Representation and Inductive Bias*, pp. 147-167 edited by D. Benjamin, Kluwer Academic Publisher, Boston, 1990.

Sikes J-B. (ed.),"*The Concise Oxford Dictionary of Current English*", pp. 335, Oxford University Press, 1982.

Jianping J. and Michalski R., "A Description of preference Criterion in a constructive Learning", in *Proceeding of the sixth international Workshop on Machine Learning*, pp. 17-22, Morgan Kaufmann Publishers, California, 1989.

Zhang Y., "*Kang Xi Dictionary*", The Commercial Press, Beijing, 1979.

ZhongZhi S., "*Principles of Machine Learning*", International Academic Publishers, pp. 70, Beijing, 1992.

Zucker J-D. & Mathieu J., "Machine Learning Contributions to a Guided Discovery Tutoring Environment for Chinese Characters", in *Proceedings of the International Conference on Computers in Education*, pp. 25-30, Taiwan, 1993.

Zucker J-D. & Ganascia J-G., "Selections and Matches", Rapport Interne du LAFORIA, to appear, 1994a.

Zucker J-D. & Ganascia J-G., "Une approche moriologique de la reformulation d'exemples en apprentissage", in *Proceedings of the Neuvième Journées de l'Apprentissage*, pp. I1-I14, 1994b.