IRIT
CNRS - INPT - UPS - UT1 - UTM
Institut de Recherche en Informatique de Toulouse

Working Papers

of the
1st International Workshop on
Similarity and **A**nalogy-based **M**ethods in **AI**

SAMAI 2012

Edited by Gilles Richard

Co-located with the
20th European Conference on Artificial Intelligence
Montpellier, France, August 27, 2012

Rapport Interne IRIT/RR--2012-20—FR

Working Papers

of the

1st International Workshop on

Similarity and Analogy-based Methods in AI

SAMAI 2012

August 27, 2012 - Montpellier (France)

Cover picture: *A chat with Fernando* - Photomontage HP - 2011

Preface

The ability to identify and evaluate similarities (and dissimilarities) between a current situation and already known cases and to take advantage of these evaluations is an important feature of human thinking. We make sense of the new situation and draw inferences for classification, prediction, etc.

For this reason, AI has always been interested in analogical reasoning. Special forms of similarity-based reasoning such as case-based reasoning have been studied and developed. Similarity and analogy-based methods have been applied to a large variety of areas such as problem solving (including IQ tests), theorem proving, case-based question-answering and decision, machine learning, natural language processing, image processing, causality analysis, argumentation, logic programming, diagrammatic reasoning, or creativity.

In the last decade, the interest in AI for analogical reasoning and analogical proportions, case-based reasoning, and other forms of similarity-based reasoning has continued to develop. In particular, different views and modelings have been suggested referring to a large variety of approaches ranging from propositional, first or higher order logics to structure mappings, neural networks, analogical dissimilarity distances, probabilistic models, Kolmogorov complexity theory, and fuzzy similarity-based methods.

The aim of this workshop is

1. to provide an overview of the state of the art in this field,
2. to share results,
3. to confront viewpoints, and
4. to get a better understanding of how to effectively implement these ideas and to bring new solutions to practical problems.

It is a concerted attempt to try to reach a new milestone in this field. These proceedings give an idea of the richness and the diversity of the field.

We take this opportunity to thank

- Our invited speaker, Prof. Laurent Miclet, for having accepted to give an introductory talk at this workshop.
- Our programme committee members for their commitment to the success of this event and for their work (each paper received 3 reviews).
- The participants of SAMAI for the quality of their contribution.
- Finally, we thank our sponsors, namely the IRIT laboratory in Toulouse and the British Institute for Technology and E-commerce in London.

Programme committee

Stergos.Afantenos	University of Toulouse (France)
Danushka Bollegala	University of Tokyo (Japan)
Antoine Cornuejols	AgroParisTech (France)
Bruno Gaume	University of Toulouse (France)
Lluis Godo	Artificial Intelligence Research Institute (Spain)
Ashok Goel	Georgia Institute of Technology (USA)
Eyke Hüllermeier	University of Marburg (Germany)
Bipin Indurkhya	International Institute of Information Technology Gachibowli Hyderabad (India)
Mateja Jamnik	University of Cambridge (UK)
Mark Keane	University College Dublin (Ireland)
Matthew Klenk	Palo Alto Research Center (USA)
Kai-Uwe Kühnberger	University of Osnabrueck (Germany)
Philippe Langlais	University of Montreal (Canada)
Yves Lepage	University of Waseda (Japan)
Jean Lieber	University of Nancy (France)
Ramon Lopez de Mantaras	Artificial Intelligence Research Institute (Spain)
Andrew Lovett	Northwestern University (Evanston-USA)
Arun K. Majumdar	Vivomind (Rockville-USA)
Laurent Miclet	University of Rennes (France)
Diarmuid O'Donoghue	National University of Ireland (Ireland)
Yoshiaki Okubo	University of Hokudai (Japan)
Praveen Paritosh	University of Northwestern (Evanston-USA)
Henri Prade	University of Toulouse (France)
Marco Ragni	University of Freiburg (Germany)
Eyal Sagi	University of Northwestern (Evanston-USA)
Ute Schmid	University of Bamberg (Germany)
Steven Schockaert	University of Cardiff (UK)
John F. Sowa	Vivomind (Rockville-USA)
Georgi Stojanov	American University in Paris (France)
Tony Veale	University College Dublin (Ireland)
Manuela Veloso	Carnegie Mellon University (USA)
Francois Yvon	University Paris-Sud (France)

Workshop programme

9:00 - 9:15: Arrival of participants and welcome

Session 1 - A is to B as C is to ...

9:15 - 9:30: Introductory address (Gilles Richard)
9:30 - 10:30: Invited Talk
Artificial analogy: an introduction (Laurent Miclet)

Coffee Break

Session 2 - Rationality - Similarity - Creativity
(Chair: Henri Prade)

11:00 - 11:30: Rationality Through Analogy: On HDTP And Human-Style Rationality
(Tarek R. Besold, Martin Schmidt, Helmar Gust, Ulf Krumnack, Ahmed Abdel-Fattah, Kai-Uwe Kühnberger)
11:30 - 12:00: Matchmaking: How similar is what I want to what I get (Michael Munz, Klaus Stein, Martin Sticht, Ute Schmid)
12:00 - 12:30: Perceptual similarity and analogy in creativity and cognitive development (Georgi Stojanov, Bipin Indurkhya)

Lunch Break

Session 3 - Analogical proportions - Cased-based and interpolative reasoning
(Chair: Georgi Stojanov)

14:00 - 14:30: Analogical proportions in a lattice of sets of alignments built on the common subwords in a finite language (Laurent Miclet, Nelly Barbot, Baptiste Jeudy)
14:30 - 15:00: Belief revision-based case-based reasoning(Julien Cojan, Jean Lieber)
15:00 - 15:30: Cautious analogical-proportion based reasoning using qualitative conceptual relations (Steven Schockaert, Henri Prade)

Coffee Break

Session 4 - Natural language processing - Argumentation
(Chair: Jean Lieber)

16:00 - 16:30: Issues in Analogical Learning over Sequences of Symbols: a Case Study with Named Entity Transliteration (Philippe Langlais)
16:30 - 17:00: (Re-)discovering the graphical structure of Chinese characters (Yves Lepage)
17:00 - 17:30: Arguing by analogy - Towards a formal view. A preliminary discussion (Leila Amgoud, Youssef Ouannani, Henri Prade)

Final Discussion Panel

Table of Contents

Analogical proportion – A brief survey

Henri Prade and **Gilles Richard**[1]

Abstract. Analogies play an important role in many reasoning tasks. In this paper, we survey a recently proposed modeling for an analogical proportion, i.e. a statement of the form "A is to B as C is to D", and its diverse applications. The logical representation used for encoding such proportions takes both into account what the four situations have in common and how they differ. Thanks to the use of a Boolean modeling extended with suitable fuzzy logic connectives, the approach can deal with situations described by features that may be binary or multiple-valued. It is shown that analogical proportion is a particular case of a more general concept, namely logical proportion. Among the 120 existing logical proportions, we single out four proportions (including the analogical proportion) for their remarkable properties. We emphasize the interest of analogy-related logical proportions for handling a large variety of reasoning tasks, ranging from solving IQ tests, to transductive reasoning for classification, and interpolative and extrapolative reasoning. The approach does not just rely on the exploitation of similarities between two cases (as in case-based reasoning), but rather takes advantage of the parallel made between a situation to be evaluated or to be completed, with three other situations.

1 Introduction

Reasoning is at the core of human intelligence, allowing to infer new knowledge from a pre-existing knowledge. While logical inference allows to generate valid conclusions, analogical inference is less robust and leads only to plausible conclusions. An *analogy* parallels two particular situations on the basis of some *similarities* (the notion of situation and similarity should be understood in a very broad sense). Based on such an analogy, *analogical inference* assumes that these two situations might be similar in other respects and then draws conclusions on this basis. Due to the brittleness of its conclusions, analogy is not easily amenable to a formal logic framework. Nevertheless, diverse modelings coexist, using first order logic as in [9], second order logic as in [50], algebraic oriented frameworks as in [8, 15], or even a complexity-based approach as in [5] that establishes a close link with computational learning theory. From a more practical viewpoint, analogical reasoning has mainly been viewed as a powerful heuristic device [13, 52, 31], and considered as such in AI for a long time, e.g. [10, 57, 16, 26], being useful at the meta level for improving deductive reasoning provers, as well as in cognition, in problem solving and in learning, e.g. [14, 25, 21, 17], involving not only symbolic representations but also numerical calculations as it is the case for quantitative estimation [32].

The above-mentioned works do not usually consider a specific type of analogy, the so-called *analogical proportion*, i.e. a statement of the form "*a is to b as c is to d*", usually denoted $a : b :: c : d$. In that case, a situation is represented as a pair of items (a, b) and the 4 items appearing in the proportion generally have the same type. Inference then amounts to find a value for an unknown item x such that the proportion $a : b :: c : x$ holds.

In the last decade, some authors starting with the pioneering investigation made in [23] with computational linguistic motivations, have started to develop a large panel of algebraic models for analogical proportions, from semi-groups to lattices, through words over finite alphabets and finite trees [53, 54, 28]. Moreover, the use of analogical proportions for machine learning purposes in [3, 27] has been also proposed and studied. Since analogical proportions have various instances in natural language, it is not surprising that the field of natural language processing has also been investigated [1, 24, 22, 56, 4] providing encouraging results for automatic translation or text comprehension [55], or even recently recommendation systems [49].

However, it is only more recently that a propositional logic representation of analogical proportion has been proposed [29]. Still, this view has its roots in the largely ignored work of the anthropologist, linguist, and computer scientist Sheldon Klein [19, 18], who was the first to propose a truth table-like way for finding x such that the analogical proportion $a : b :: c : x$ holds. Besides, in the annex of a 1952 French book by the psychologist Jean Piaget [33], the author informally investigates a similar idea, still without explicitly mentioning analogy (see also [34] pp. 35–37), where a definition of a so-called *proportion logique* is given).

The logical view of analogical proportion has been further developed in a series of works [37, 36, 40, 38] leading to the introduction of other related proportions, the extension to multiple-valued settings [39], and to various applications to reasoning and classification. Our aim in this paper is to provide a brief introduction to the Boolean approach to analogy and a short overview of its potential developments and applications. The paper is organized in two main parts: section 2 and its subsections present the main results on the modeling of analogical proportion and other related logical proportions. The remaining sections discuss the interest of the approach for solving IQ tests, in classification, in interpolation and extrapolation reasoning.

2 Analogical proportion and related proportions

In this section, we present the basic Boolean definitions for analogical proportion, introducing three *sister-proportions*, highlighting their strong link and formalizing the equation solving problem, which is the core of the analogical inference process. Finally, we briefly investigate their multi-valued extensions, before briefly mentioning the existence of other logical proportions.

2.1 Four basic proportions

Generally speaking, the comparison of two items A and B relies on their representation. Let us adopt a logical setting and let φ be a

[1] University of Toulouse, IRIT-CNRS, France, email: prade,richard@irit.fr

property, which can be seen as a predicate: $\varphi(A)$ may be true (in that case $\neg\varphi(A)$ is false), or false. When comparing two items A and B w.r.t. φ, it makes sense to consider A and B *similar* (w.r.t. φ):
- when $\varphi(A) \wedge \varphi(B)$ is true or
- when $\neg\varphi(A) \wedge \neg\varphi(B)$ is true.

In the remaining cases:
- when $\neg\varphi(A) \wedge \varphi(B)$ is true or
- when $\varphi(A) \wedge \neg\varphi(B)$ is true,

we can consider A and B as *dissimilar* w.r.t. property φ. $\varphi(A)$ and $\varphi(B)$ being ground formulas, they can be considered as Boolean variables, denoted a and b by abstracting w.r.t. φ. If the conjunction $a \wedge b$ is true, the property is satisfied by both items A and B, while the property is satisfied by neither A nor B if $\overline{a} \wedge \overline{b}$[2] is true. Moving to the Boolean setting, $a \wedge b$ and $\overline{a} \wedge \overline{b}$ are *indicators of similarity* while $a \wedge \overline{b}$ and $\overline{a} \wedge b$ are *indicators of dissimilarity*. These indicators are the basis [37] for providing a formal definition of analogical proportion and more generally of a logical proportion. An analogical proportion focuses on *differences* and should hold when the differences between a and b and between c and d are the same. If we consider a pair of items (a, b) as completely described via the 4 above indicators, analogical proportion express the identity of their dissimilarity indicators, Then it makes sense to encode an analogical proportion with the following conjunction

$$(a \wedge \overline{b} \equiv c \wedge \overline{d}) \wedge (\overline{a} \wedge b \equiv \overline{c} \wedge d)(1)$$

as it is the logical counterpart of "a differs from b as c differs from d", and conversely. When we generalize this viewpoint by considering a *logical proportion* T as the conjunction of 2 distinct equivalences between indicators, it appears that 120 different proportions can be build, from which 4 emerge as being strongly related to analogical proportion. They are respectively [37]:

- *reverse analogy*: $R(a, b, c, d)$, defined by
$$((a \wedge \overline{b}) \equiv (\overline{c} \wedge d)) \wedge ((\overline{a} \wedge b) \equiv (c \wedge \overline{d}))$$
- *paralogy*: $P(a, b, c, d)$, defined by
$$((a \wedge b) \equiv (c \wedge d)) \wedge ((\overline{a} \wedge \overline{b}) \equiv (\overline{c} \wedge \overline{d}))$$
- *inverse paralogy*: $I(a, b, c, d)$, defined by
$$((a \wedge b) \equiv (\overline{c} \wedge \overline{d})) \wedge ((\overline{a} \wedge \overline{b}) \equiv (c \wedge d))$$

When needed, the analogical proportion $a : b :: c : d$ will be denoted $A(a, b, c, d)$.

2.2 Basic properties of A, R, P, I

Considered as Boolean formula, logical proportions can be seen via their truth tables. Table 1 exhibits the truth tables of A, R, P, I, where only the 6 lines leading to the truth value 1 are shown. Starting from the definition, it can be easily shown:

- $A(a, b, a, b)$ and $A(a, a, b, b)$, but *not* $A(a, b, b, a)$;
- $A(a, b, c, d) \Rightarrow A(c, d, a, b)$ (symmetry);
- $A(a, b, c, d) \Rightarrow A(a, c, b, d)$ (central permutation).

which correspond to the usual postulates of analogical proportion and which mimics the properties of the numerical proportion $\frac{a}{b} = \frac{c}{d}$. Similar postulates hold for R, P and I. The following properties are easy to check on the truth tables and establish a strong link between the four proportions A, R, P, I:
- $R(a, b, c, d)$ iff $A(a, b, d, c)$
- $P(a, b, c, d)$ iff $A(a, d, c, b)$
- $I(a, b, c, d)$ iff $A(a, \overline{d}, \overline{c}, b)$

[2] The overline denotes Boolean negation.

Table 1. Analogy, Reverse analogy, Paralogy, Inverse Paralogy truth tables

A				**R**			
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
0	1	0	1	0	1	1	0
1	0	1	0	1	0	0	1
P				**I**			
0	0	0	0	1	1	0	0
1	1	1	1	0	0	1	1
1	0	0	1	1	0	0	1
0	1	1	0	0	1	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0

2.3 Equation solving

Given a proportion T and 3 items a, b, c, the problem of finding a fourth item x such that $T(a, b, c, x)$ holds is known as the equation-solving problem. The equations $A(a, b, c, d)$, $R(a, b, c, x)$, $P(a, b, c, x)$ and $I(a, b, c, x)$ have not always a solution $x \in \{0, 1\}$. For analogical proportions, the existence condition for a solution is $(a \equiv b) \vee (a \equiv c) = 1$, which just states that $A(1, 0, 0, x)$ and $A(0, 1, 1, x)$ have no solution. When a solution exists, it is unique and given by $x = a \equiv (b \equiv c)$ for the three proportions A, R, P [29, 37], as first guessed from anthropological observations by Klein [19] (without distinguishing the three proportions). Obviously, this equation-solving problem is at the core of the inference process associated to analogical proportion. When we know that a proportion $A(a, b, c, x)$ holds (under the above conditions), we can infer the value of x from the known values of a, b, c.

As a direct illustration of the equation solving process, let us consider an analogical puzzle, as the ones considered early by Evans [10] where a series of 3 first items a, b, c is given and the 4th item d has to be chosen among several plausible options. In [11] or [47], an optimization mechanism looks for the candidate solution that maximizes the similarity of the set of rules describing the change from a to b, with the set of rules describing the change from c to each candidate solution. Here, the application of the equation solving process for each feature contrasts with such an approach, since the solution is computed directly. This is illustrated on Figure 1, and the details of the equations for the different features are given below. When the items are pictures, our method may also apply with an encoding of the image at the pixel level; see [41] for a discussion.

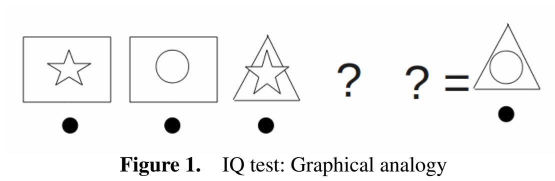


Figure 1. IQ test: Graphical analogy

square: $(1, 1, 0, x_1)$ holds $\Rightarrow x_1 = 0$, i.e. no square;
triangle: $(0, 0, 1, x_2)$ holds $\Rightarrow x_2 = 1$, i.e. triangle;
star: $(1, 0, 1, x_3)$ holds $\Rightarrow x_3 = 0$, i.e. no star;
circle: $(0, 1, 0, x_4)$ holds $\Rightarrow x_4 = 1$, i.e. circle;
black point: $(1, 1, 1, x_5)$ holds $\Rightarrow x_5 = 1$, i.e. black point;
hexagon: $(0, 0, 0, x_6)$ holds $\Rightarrow x_6 = 0$, i.e. no hexagon.

There are many examples of analogical proportions that may look slightly different at first glance. Take for instance `Paris: France :: Roma : Italy`. In such a case, there is the (obvious) relation "is the capital of" between a and b, which also holds between c and d. This is a binary relation, not a unary feature, and the proportion cannot just be captured with features such that "is a capital", and "is a country" (since then `Roma : France :: Paris : Italy` would hold as well). However, with more specific features such as "capital of what is in the next slot", "country to which belongs what is in the previous slot", one may still have a faithful encoding of the proportion. Checking that the new proportion obtained by central permutation `Paris : Roma :: France : Italy` is still valid, would require the introduction of slightly different features – a problem that does not appear with unary features. Lastly, solving the equation $a : b :: c : x$ amounts in this case to look for the relation(s) that hold(s) for (a, b), say R_i and then to look for x such as $R_i(c, x)$ holds (see [49] for instance).

2.4 Multiple-valued logic extension

If we consider the Boolean expression of the analogical proportion, one may think of many possible multiple-valued extensions, depending of the operations chosen for modeling $\wedge$, $\equiv$ (involving $\rightarrow$), and $\neg$. Moreover, a formula such as (1) can be written in many equivalent forms in Boolean logic. These forms are no longer necessarily equivalent in a non-Boolean setting where $[0, 1]$ is now the truth space. The choice for $\neg a$ interpretation is quite standard as $1-a$, but it is important to make proper choices for the remaining connectors that are in agreement with the intended meaning of the considered proportion. Some properties seem very natural to preserve, such as

i) the independence with respect to the positive or negative encoding of properties (one may describe a price as the extent to which it is cheap, as well as it is not cheap), which leads to require that $A(\neg a, \neg b, \neg c, \neg d)$ holds if $A(a, b, c, d)$ holds;

ii) the knowledge of a and of the differences between a and b and between b and a, should enable us to recover b. Indeed in the Boolean case, we have $b = (a \wedge (a \rightarrow b)) \vee \neg(b \rightarrow a)$. A careful analysis [39] of the requirements leads to choose

i) the minimum operator for $\wedge$; ii) $s \equiv t = 1 - |s - t|$;

iii) Łukasiewicz implication $s \rightarrow t = min(1, 1 - s + t)$.

Note also that with these choices $s \equiv t = (s \rightarrow t) \wedge (t \rightarrow s)$.

This leads to the following expressions which both generalize the Boolean case to multiple-valued entries and introduce a graded view of the analogy-related proportions.

For analogy, we have $A(a, b, c, d) =$

$1- \mid (a - b) - (c - d) \mid$ if $a \geq b$ and $c \geq d$, or $a \leq b$ and $c \leq d$

$1 - max(|a - b|, |c - d|)$ if $a \leq b$ and $c \geq d$ or $a \geq b$ and $c \leq d$

Thus, $A(a, b, c, d)$ is all the closer to 1 as the differences $(a - b)$ and $(c - d)$ have the same sign and have similar absolute values. Note that $A(1, 0, c, d) = 0$ as soon as $c \leq d$.

For reverse analogy, we have $R(a, b, c, d) =$

$1- \mid (a - b) - (d - c) \mid$ if $a \leq b$ and $c \geq d$ or $a \geq b$ and $c \leq d$

$1 - max(|a - b|, |c - d|)$ if $a \geq b$ and $c \geq d$, or $a \leq b$ and $c \leq d$

The definition of paralogy is a little bit simpler:

$P(a, b, c, d) = min(1 - |(a \wedge b) - (c \wedge d)|, 1 - |(a \vee b) - (c \vee d)|)$, with $a \vee b = 1 - (1 - a) \wedge (1 - b)$. Again we take $a \wedge b = min(a, b)$; see [39] for justifications. The definition for I is deducible from the definition of P and the link $I(a, b, c, d) \equiv P(a, b, \overline{c}, \overline{d})$.

With respect to equation solving, it can be shown that it exists x such that $A(a, b, c, x) = 1$ if and only if $x = c + b - a \in [0, 1]$, and when it exists, the solution is unique. Similar equations may be solved as well for the three other proportions.

2.5 The other logical proportions

As said above, there are 120 distinct ways to combine indicators to build a logical proportion. It appears that all of them are *true for 6 lines* in their truth table and *false for the 10 other lines*. Since we have $\left[{16 \atop 6}\right] = 8008$ truth tables with exactly 6 valuations leading to true, logical proportions can be considered as quite rare. When focusing on their syntactic forms, 5 classes of proportion can be distinguished (see [38] for the details), but is also interesting to identify the proportions that possess some noticeable property, such as

- full identity, i.e. being true for the pattern (x, x, x, x), which means true for $(1, 1, 1, 1)$ and $(0, 0, 0, 0)$. There are 15 such proportions, including P, A, R [40].

- code independency. The idea underlying this semantic property is that a proportion should be independent from the coding convention, i.e., representing true by 1 and false by 0. So if we switch the values $(0, 1)$ in the coding of a given valuation, the truth value of a proportion should remain the same. There are 8 proportions satisfying code independency including P, A, I, R.

- transitivity. There are 6 transitive proportions among which A and P. But R and I are not transitive.

Thus, if we require full identity and code independency, we only have P, A, R. Both formal and empirical investigations of these proportions can be found in [43, 44].

3 Reasoning with proportions

Due to their dissimilarity / similarity semantics, logical proportions, and specially analogy-related ones, seem to have a great potential in reasoning about particular situations. Inferring the value of d starting from the values of a, b, c and the fact that some proportion holds in the 4-tuple (a, b, c, d) is an equation solving problem: find d such that the considered proportion holds knowing the truth values of a, b, c. Such an equation may have no solution, or may have one or two solutions. Regarding the unicity of the solution when it exists, the solution will be always unique for proportions such that each of the 6 lines of their truth table starts with a different triple of values for a, b, c. There are 64 proportions that have this property, and there are 56 proportions for which the 4-tuple (a, b, c, x) may have 2 solutions for some entries a, b, c. Besides, since any logical proportion relating (a, b, c, d) is true for only 6 patterns of values, and (a, b, c) may take $2^3 = 8$ different triples of values, there are at least 2 entries a, b, c leading to no solution. Thus, as already said for analogy, the two equations $A(1, 0, 0, x)$ and $A(0, 1, 1, x)$ have no solution.

Since logical proportions are Boolean formulas, it is natural to describe what can be inferred from a given situation with a set of valid inferences, involving these proportions in the premises of the inference schemes. Let us start from a simple example to understand our point. Suppose we observe $\neg a$, b and $\neg c$ and we get a new d knowing only that d is in *analogical* proportion with the 3 previous values. We are faced to the problem of inferring the value of d. One may use the clausal form of this proportion [36], namely

$\{\neg a \vee b \vee c, \neg a \vee b \vee \neg d, a \vee \neg c \vee d, \neg b \vee \neg c \vee d,$
$a \vee \neg b \vee \neg c, a \vee \neg b \vee d, \neg a \vee c \vee \neg d, b \vee c \vee \neg d\}$[3]

Each clause is falsified by a pattern of 3 literals for which there does not exist a 4th literal with which they form a proportion. Thus, the first clause $\neg a \vee b \vee c$ expresses syntactically that a $\neg b$ $\neg c$ (i.e.,

[3] Similarly, the clausal form for paralogy is: $\{\neg a \vee c \vee d, \neg a \vee \neg b \vee d, a \vee b \vee \neg c, b \vee \neg c \vee \neg d, a \vee \neg c \vee \neg d, a \vee b \vee \neg d, \neg a \vee \neg b \vee c, \neg b \vee c \vee d\}$. More generally, analogy, paralogy, reverse analogy, inverse paralogy are each described by a set of 8 clauses which cannot be further reduced by resolution, and these 4 sets do not share any clause.

1 0 0 in semantical terms) cannot be analogically completed, while $a \vee \neg b \vee \neg c$ expresses the same w. r. t. $\neg a \;\; b \;\; c$ and 0 1 1. Since the unknown x may be any of the 4 literals of the proportion, this makes 2×4 clauses. Going back to our inference example, we have:

$$\frac{\neg a \quad b \quad \neg c \quad a : b :: c : d}{d}$$

by resolution from the clausal form of $a : b :: c : d$ (6th clause). As expected, there are 6 valid inferences given in Table 2.

Table 2. Valid inferences with an analogical proportion

$\frac{a \;\; b \;\; c \;\; a:b::c:d}{d}$	$\frac{\neg a \;\; \neg b \;\; \neg c \;\; a:b::c:d}{\neg d}$	$\frac{\neg a \;\; \neg b \;\; c \;\; a:b::c:d}{d}$
$\frac{a \;\; \neg b \;\; c \;\; a:b::c:d}{\neg d}$	$\frac{\neg a \;\; b \;\; \neg c \;\; a:b::c:d}{d}$	$\frac{a \;\; b \;\; \neg c \;\; a:b::c:d}{\neg d}$

4 Analogical-proportion based reasoning

In the above section, we have assumed the knowledge that some particular logical proportion holds in the inference patterns we considered. Where may such knowledge come from? A natural answer is to consider that the proportion has been observed between four items for some features, and we assume on this basis that the same proportion holds for other features as well. This makes sense at least for proportions such as the 3 analogy-related proportions that express regularities in change as well as they leave room for the expression of similarity for other features.

More formally, let us consider items, cases, or situations described by vectors $(a_1, \ldots, a_n)$ of Boolean values that encode the different binary features that describe a situation. More generally, one may have multiple-valued features. Starting from a triple of items completely informed with respect to all features, we consider a new item $d = (d_1, \ldots, d_n)$, which is only partially informed, i.e. where only some features $k(d) = (d_1, \ldots, d_p), p < n$, are known, the values of the missing features $u(d) = (d_{p+1}, \ldots, d_n)$ having to be predicted. For that purpose, we adopt the following transfer pattern (where $T \in \{A, R, P\}$ denotes any analogy-related proportion):

$$\frac{\forall i \in [1, p], \;\; T(a_i, b_i, c_i, d_i)}{\forall j \in [p+1, n], \;\; T(a_j, b_j, c_j, d_j)}$$

It simply means that *if the known part $k(d)$ of d is componentwise in formal proportion T with $k(a), k(b)$ and $k(c)$ then it should be also true for the unknown part $u(d)$ of d for the same proportion T*. This form of reasoning is clearly not sound, but may be useful for trying to guess unknown values. Then, for all $j \in [p+1, n]$, one may infer the truth value of d_j assuming that $T(a_j, b_j, c_j, d_j)$ holds, given a_j, b_j, and c_j.

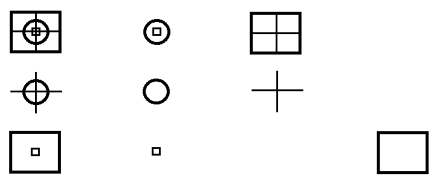
Let us consider an *example* where $n = 5$ and $a = (1, 1, 0, 0, 1)$, $b = (1, 0, 1, 1, 0)$, $c = (0, 1, 0, 0, 1)$, with an incompletely known item $d = (0, 0, 1, d_4, d_5)$ (here $p = 3$, $k(d) = (0, 0, 1), u(d) = (d_4, d_5)$). We can check that analogical proportion $k(a) : k(b) :: k(c) : k(d)$ holds for the first three features. Then using the above inference scheme, we should have $0 : 1 :: 0 : d_4$ and $1 : 0 :: 1 : d_5$, which leads to $d_4 = 1$ and $d_5 = 0$. Although this type of reasoning basically amounts to copy existing similarity / dissimilarity relationships, it is very powerful since it may produce new compound patterns where the vector representing d is not similar in all respects to any of the vectors representing a, b, or c, (as in the example above).

The transfer pattern also encompasses the following basic analogical reasoning schema. We have 2 situations or cases at hand, x and y, that both share a property P, i.e. $P(x)$ and $P(y)$ hold, and x also satisfies another property Q, the schema amounts to conclude that $Q(y)$ also holds, by an "analogical jump". Indeed the proportion $P(x) : Q(x) :: P(y) : Q(y)$ can be justified in our approach, viewing the presence of instances, properties or functions as features (see [42] for details). Similarly, one can also justify the extended analogical pattern $x : f(x) :: y : f(y)$. This pattern, combined with the general ones, can be successfully used for solving IQ tests, starting from the completion of a sequence of 3 geometric figures as in [41] to more complex tests like the Raven's tests [7] that we summarize in the following section.

5 Solving IQ tests

IQ tests play a special role in the AI litterature as they can be, in some sense, considered as a kind of scale on which to measure the effectiveness of an AI theory or system (see [20] for instance where the authors target Bennett Mechanical Comprehension Test Problems). Among the most well-known IQ tests, we have the Raven Progressive Matrices (RPM) [48]. They are visual tests where a sequence of 8 pictures has to be completed in a logical way and the solution has to be chosen among a set of 8 candidate pictures. The resulting performance is considered as a measure of the reasoning ability of the participant. An example[4] is given with its solution (a simple big square) in Figure 2. Solving an RPM heavily relies on the representation of the space and objects at hand. A Raven matrix *pic* is a 3×3 matrice where $pic[i, j]$ $(i, j \in \{1, 2, 3\} \times \{1, 2, 3\})$ denotes the picture at row i and column j and where $pic[3, 3]$ is unknown. Assuming that the Raven matrices can be understood in the following way, with respect to rows and columns:

Figure 2. Modified Raven test 12 and its solution



$$\forall i \in [1, 2], \exists f \text{ such that } pic[i, 3] = f(pic[i, 1], pic[i, 2])$$

$$\forall j \in [1, 2], \exists g \text{ such that } pic[3, j] = g(pic[1, j], pic[2, j])$$

the two complete rows (resp. columns) are examples supposed to help to discover f (resp. g), and then to predict the missing picture $pic([3, 3])$ as $f(pic[3, 1], pic[3, 2])$ (resp. $g(pic[1, 3], pic[2, 3])$). This representation is summarized in Figure 3. In fact, the problem can be restated as an analogical equation-solving problem. using an extended scheme where proportion $(a, b) : f(a, b) :: (c, d) : f(c, d)$

Figure 3. Raven matrix representation

pic[1,1]	pic[1,2]	f(pic[1,1],pic[1,2])
pic[2,1]	pic[2,2]	f(pic[2,1],pic[2,2])
g(pic[1,1],pic[2,1])	g(pic[1,2],pic[2,2])	

[4] For copyright reasons and to protect the security of the tests, the original Raven test is replaced by specifically designed examples (still isomorphic in terms of logical encoding to the original ones).

holds for lines and proportion $(a, b) : g(a, b) :: (c, d) : g(c, d)$ for columns, which translates into:

$$(pic[1, 1], pic[1, 2]) : pic[1, 3] :: (pic[2, 1], pic[2, 2]) : pic[2, 3]$$

$$(pic[1, 1], pic[2, 1]) : pic[3, 1] :: (pic[1, 2], pic[2, 2]) : pic[3, 2]$$

Similar analogical proportions are supposed to relate line 1 to line 3 and line 2 to line 3, and similarly for the columns. We have two options to solve the problem:

- Each picture $pic[i, j]$ is represented as a Boolean vector of dimension n. The Boolean coding is done manually so far. In the example of Figure 3, we consider 4 binary features in the following order: `big square`, `small square`, `circle`, `cross`. For instance, the contents of $pic[1, 2]$ is encoded by the vector (0110) in Figure 4.

Figure 4. Raven test 12 encoding

	$column1$	$column2$	$column3$
$row1$	1111	0110	1001
$row2$	0011	0010	0001
$row3$	1100	0100	????

  Considering the first missing feature, the horizontal pattern (1 0) to be completed in line 3 appears in line 1 and leads to the solution 1. For the second missing feature, there is no corresponding horizontal pattern for (1 1), so we have to move to a vertical analysis and to look for a vertical pattern starting with (0 0) for this second feature. We fail again. We have then to consider that a Raven matrix expresses a set of analogical proportions without any consideration of a particular feature. This is why, even if we fail to find a suitable pattern in row or column for this feature, we look for a similar valid pattern but related to another feature. In other terms, in order to find the solution for feature i, we allow us to take lesson from other features, then looking for a solution in a row or a column coming from another feature j. In the example, we thus find 0 for the second feature, using the context of feature 3 in column 1 or in column 2 (in fact when proceeding this way, we have to make sure that there is no contradictory patterns (here it would be (1 1 1)) observable in row or in column. Using the same approach for feature 3 (`circle`) we get 0 from observing feature 2 in row 2, or feature 4 in column 2. Feature 4 is easier, we get 0 from column 1 for the same feature. Altogether, we get (1 0 0 0) as the answer, which is indeed the encoding of the solution. This method is sufficient for solving 32 Raven tests over 36 [7].
- If we consider an image as a matrix of pixels of dimension $n \times m$ (considering only non compressed format as BMP for instance), this is simply a Boolean-like coding automatically performed by the picture processing device (e.g. the camera or the scanner). In that case, instead of dealing with 8 hand-coded Boolean vectors, we deal with BMP files for instance. Apart from the fact that we have to take care of the headers of the files (which do not obey any proportion pattern), there is no reason to change our method and our basic algorithm still applies allowing to solve 16 tests (over 36).

6 Classification by transduction

Transduction [12] is the name given to a form of reasoning that amounts to predict the class of a new piece of data on the basis of a set S of previously observed pieces of data whose class is known, without any attempt at inducing a generic model for the observed data (which would be then applied to the new piece of data in order to determine its class). A simple example of transduction mechanism (also known as lazy learning) is the k-Nearest Neighbors method, where the class that is the most frequent among the k closest neighbors of x is inferred for x.

The application of the transfer pattern to classification provides the basis for another transduction mechanism, proposed in [40] for binary classification and binary-valued features, and then extended to multiple-valued features and to multiple class problems [45]. Each piece of data x is described by a vector $(x_1, \ldots, x_n)$ of feature values that are normalized in the interval $[0, 1]$ together with its class $cl(x)$. Then a classification method, quite different from k-NN methods, is applied since the new item d to be classified is not just compared with the classified items on a one-by-one basis. Once chosen some fixed analogy-related proportion T, we look for 3-tuples $(a, b, c) \in S^3$ such that the proportion $T(cl(a), cl(b), cl(c), cl)$ has a solution cl. This requires that $cl(a)$, $cl(b)$, and $cl(c)$ correspond to a *maximum of two* distinct classes: either $cl(d) = cl(a) = cl(b) = cl(c)$, or there are two distinct classes ($cl(a) = cl(b) \neq cl(c) = cl(d)$ or $(cl(a) = cl(c) \neq cl(b) = cl(d))$. Only such triples (a, b, c) are retained as potentially useful. Indeed the other triples (a, b, c) are useless since whatever the coming d, they cannot constitute a logical proportion with d. This processing of the suitable set of triples can be done offline.

Then, we have to look, among the set of suitable triples, for the one(s) that seem(s) the most appropriate to predict the class $cl(d)$. For doing this, each suitable triple we consider is evaluated by means of the following vector $(T(a_1, b_1, c_1, d_1), \ldots, T(a_i, b_i, c_i, d_i), \ldots, T(a_n, b_n, c_n, d_n))$. Then the vectors (and thus the triples) are ordered in a lexicographic decreasing manner[5]. Then we may choose for $cl(d)$ the class associated to the triple having the best evaluation, or the most frequent class among the k best triples.

This approach has been tested on different benchmark problems and has given rather good results [45]. This agrees with the results previously obtained in [3, 27], where the authors have developed a binary classifier on the basis of an "analogical dissimilarity" measure AD: for a given tuple (a, b, c, d), $AD(a, b, c, d)$ is a positive number which is zero if and only if the proportion $a : b :: c : d$ holds. The reasons behind this success are investigated in [6].

7 Interpolation, extrapolation and non-monotonic reasoning

A problem formally similar to transduction is the problem of reasoning from an incomplete collection of parallel if-then rules of the form "if X_1 is A_1 and ... and X_n is A_n then Y is B" where the A_i's and the B's are labels belonging to ordered domains; for instance, the labels associated with the domain of X_i could be 'small', 'medium', and 'large'. Such a rule base may be incomplete in the sense that for some combination of the labels of the condition variables X_i, there may not be a corresponding rule. Such a problem may be handled by considering that the A_i's and the B's are represented by fuzzy sets. Another route may take advantage of the idea of analogical proportion, as suggested in [46]. The analogical proportions will no longer apply between feature values pertaining to different pieces of data as in classification, but between labels appearing in the expression

[5] $(u_1, \ldots, u_i, \ldots, u_n) >_{lexicographic} (v_1, \ldots, v_i, \ldots, v_n)$, *once the components of each vector have been decreasingly ordered*, iff $\exists j < n \; \forall i = 1, j \; u_i = v_i$ and $u_{j+1} > v_{j+1}$.

of generic rules. The approach is based on the assumption that the mapping from the X_i's to Y which is partially described by the set of available rules, is sufficiently "regular" for completing missing rules, as in the following example. Assume we know the two rules

[rule 1] "if X_1 is $small$ and X_2 is $small$ then Y is $large$"

[rule 2] "if X_1 is $small$ and X_2 is $large$ then Y is $small$"

where the possible labels associated with variables X_1, X_2, and Y are $small$, $medium$, or $large$. Then, if we wonder what may be a plausible conclusion for the rule

[rule 3] "if X_1 is $small$ and X_2 is $medium$ then Y is ..."

we may observe that a kind of analogical proportion of the form $rule\ 1 : rule\ 3 :: rule\ 3 : rule\ 2$ holds. Indeed, one may consider that we have for variable X_1: $small : small :: small : small$, which certainly holds on the basis of pure identity, and for variable X_2 we get $small : medium :: medium : large$, which holds as much as the increase from $small$ to $medium$ is the same as the increase from $medium$ to $large$. What we have here are analogical proportions on a discrete scale. This leads to an equation of the form $large : x :: x : small$ for the label of Y in rule 3, with $medium$ as a solution. In case we would rather know rule 1 and rule 3, and we have to guess the conclusion of rule 2, we would obtain the equation $large : medium :: medium : x$ (with $x = small$ as a solution). We can thus perform extrapolation as well as interpolation. Such an approach fully agrees with a more cautious treatment of the ideas of "betweenness" and "parallelism" between symbolic situations in conceptual spaces; see [51] for further discussions.

Non-monotonic reasoning and analogical reasoning are very different in nature since one handles generic default rules while the other one parallels particular situations, even if both leads to brittle conclusions. Still there are noticeable connections between them; see [42] for detailed discussions. Let us only mention that in the setting of nonmontonic consequence relations, one may propose a representation of "a is analogous to b", which agrees with the pattern of inference originally hinted by Polya [35]: "a is analogous to b, a is true, b plausibly holds". This may be related to the idea that two situations are analogous if they only differ on *non important* features.

8 Concluding remarks

A Boolean view of the ideas of similarity and dissimilarity have led to the concept of logical proportions, which is a logical generalization of the numerical concept of proportion. Analogy-related proportions are especially remarkable, seem to play an important role in classification and other types of reasoning tasks where particular patterns can be put in parallel. In spite of its wide scope, this overview has left aside some worth-mentioning issues such as the study of the relation of analogical proportion with formal concept analysis [30], or the evaluation of natural language analogical proportions in terms of Kolmogorov complexity [2].

REFERENCES

[1] S. Ando and Y. Lepage. Linguistic structure analysis by analogy: Its efficiency, 1996.

[2] M. Bayoudh, H. Prade, and G. Richard, 'Evaluation of analogical proportions through Kolmogorov complexity', *Knowledge-Based Systems*, **29**, 20–30, (2012).

[3] S. Bayoudh, L. Miclet, and A. Delhay, 'Learning by analogy: A classification rule for binary and nominal data', *Proc. Inter. Conf. on Artificial Intelligence IJCAI07*, 678–683, (2007).

[4] D. Bollegala, T. Goto, N. Tuan Duc, and M. Ishizuka, 'Improving relational similarity measurement using symmetries in proportional word analogies', *Information Processing and Management*, (to appear in 2012).

[5] A. Cornuéjols, 'Analogie, principe d'économie et complexité algorithmique', in *In Actes des 11èmes Journées Francaises de l'Apprentissage. Sète, France*, (1996).

[6] W. Correa, H. Prade, and G. Richard, 'Trying to understand how analogical classifiers work', in *Proc. 6th Inter. Conf. on Scalable Uncertainty Management (SUM'12), Marburg, Germany*, ed., S. Link E. Hüllermeier, Springer Verlag, LNAI, to appear, Sept. 17-19, (2012).

[7] W. Correa, H. Prade, and G. Richard, 'When intelligence is just a matter of copying', in *Proc. 20th Europ. Conf. on Artificial Intelligence, Montpellier, Aug. 27-31*, volume to appear, (2012).

[8] M. Dastani, B. Indurkhya, and R. Scha, 'An algebraic approach to modeling analogical projection in pattern perception', *Journal of Experimental and Theoretical Artificial Intelligence*, **15**(4), 489 – 511, (2003).

[9] T. R. Davies and S. J. Russell, 'A logical approach to reasoning by analogy', in *IJCAI-87*, pp. 264–270. Morgan Kaufmann, (1987).

[10] T. G. Evans, 'A heuristic program to solve geometry-analogy problems', in *Proc. A.F.I.P. Spring Joint Computer Conf.*, volume 25, pp. 5–16, (1964).

[11] K. Forbus, A. Lovett, K. Lockwood, J. Wetzel, C. Matuk, B. Jee, and J. Usher, 'Cogsketch', in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pp. 1878–1879. AAAI Press, (2008).

[12] A. Gammerman, V. Vovk, and V. Vapnik, 'Learning by transduction', in *Proc. 14th Conf. Uncertainty in AI*. Morgan Kaufmann, (1998, 148-155).

[13] D. Gentner, 'The Mechanisms of Analogical Learning', in *Similarity and Analogical Reasoning*, 197–241, Cambridge Univ. Pr., (1989).

[14] D. Gentner, K. J. Holyoak, and B. N. Kokinov, *(Eds.) The Analogical Mind: Perspectives from Cognitive Sciences*, MIT Press, 2001.

[15] H. Gust, K. Kühnberger, and U. Schmid, 'Metaphors and heuristic-driven theory projection (hdtp)', *Theoretical Computer Science*, **354**(1), 98 – 117, (2006).

[16] D. H. Helman, ed., *Analogical Reasoning: Perspectives of Artificial Intelligence*, Cognitive Science, and Philosophy, Kluwer, 1988.

[17] M. Helms and A. Goel, 'Analogical problem evolution from inception: Complementing problem-solution co-evolution', *Proc. of 5 International Conference on Design Computing and Cognition (DCC'12)*, (2012 (to appear)).

[18] S. Klein, 'Analogy and mysticism and the structure of culture (and Com-ments & Reply)', *Current Anthropology*, **24 (2)**, 151–180, (1983).

[19] S. Klein, 'Culture, mysticism & social structure and the calculation of behavior', in *Proc. Europ. Conf. Artif. Intel.*, pp. 141–146, ((ECAI) 1982).

[20] M. Klenk, K. Forbus, E. Tomai, and H. Kim, 'Using analogical model formulation with sketches to solve Bennett mechanical comprehension test problems.', *Journal Experimental and Theoretical Artificial Intelligence - Special Issue on "Test-Based AI"*, **23**(3), (2011).

[21] A. Kulakov, G. Stojanov, and D. Davcev, 'A model of an expectancy-driven and analogy-making actor', *Proceedings of the 2nd International Conference on Development and Learning*, 61 – 66, (2002).

[22] P. Langlais, F. Yvon, and P. Zweigenbaum, 'Analogical translation of medical words in different languages', in *GoTAL*, pp. 284–295, (2008).

[23] Y. Lepage, 'Analogy and formal languages', *Electr. Notes Theor. Comput. Sci.*, **53**, (2001).

[24] Y. Lepage, J. Migeot, and E. Guillerm, 'A measure of the number of true analogies between chunks in japanese', in *Human Language Technology. Challenges of the Information Society, Third Language and Technology Conference, LTC 2007, Poznan, Poland, October 5-7, 2007, Revised Selected Papers*, eds., Z. Vetulani and H. Uszkoreit, volume 5603 of *LNCS*, pp. 154–164. Springer, (2009).

[25] A. Lovett, K. Forbus, D. Gentner, and E. Sagi, 'Using analogical mapping to simulate time-course phenomena in perceptual similarity', *Cognitive Systems Research*, **10**, (2009).

[26] E. Melis and M. Veloso, 'Analogy in problem solving', in *Handbook of Practical Reasoning: Computational and Theoretical Aspects*. Oxford Univ. Press, (1998).

[27] L. Miclet, S. Bayoudh, and A. Delhay, 'Analogical dissimilarity: definition, algorithms and two experiments in machine learning', *JAIR, 32*, 793–824, (2008).

[28] L. Miclet and A. Delhay, 'Relation d'analogie et distance sur un alphabet defini par des traits', Technical Report 1632, IRISA, (July 2004).

[29] L. Miclet and H. Prade, 'Handling analogical proportions in classical logic and fuzzy logics settings', in *Proc. 10th Europ. Conf. on Symb. and Quantit. Appr. to Reasoning with Uncertainty (ECSQARU'09)*, pp.

638–650. (C. Sossai, G. Che-mello, eds.), Verona, Jul. 1-3, Springer, LNCS 5590, (2009).

[30] L. Miclet, H. Prade, and D. Guennec, 'Looking for analogical proportions in a formal concept analysis setting', in *Proc. 8th Inter. Conf. on Concept Laticces and Their Applications (CLA'11)*, eds., A. Napoli and V. Vychodil, pp. 295 – 307. INRIA, Nancy, (2011).

[31] D. P. O'Donoghue, A. J. Bohan, and M. T. Keane, 'Seeing things: Inventive reasoning with geometric analogies and topographic maps', *New Generation Comput.*, **24**(3), 267–288, (2006).

[32] P. K. Paritosh and M. E. Klenk, 'Cognitive processes in quantitative estimation: Analogical anchors and causal adjustment', (2006).

[33] J. Piaget, *Essai sur les transformations des opérations logiques: les 256 opérations ternaires de la logique bivalente des propositions*, Presses Univ. de France, Paris, 1952.

[34] J. Piaget, *Logic and Psychology*, Manchester Univ. Press, 1953.

[35] G. Polya, *Mathematics and Plausible Reasoning. Vol.2: Patterns of Plausible Inference*, Princeton Univ. Press, 2nd ed. 1968, 1954.

[36] H. Prade and G. Richard, 'Analogical proportions: another logical view', in *Proc. 29th Ann. Int. Conf. on Artif. Intel.* (AI'09), Cambridge, Springer, 121-134, (2009).

[37] H. Prade and G. Richard, 'Analogy, paralogy and reverse analogy: Postulates and inferences', in *Proc. 32nd Ann. Conf. on Artif. Intellig. (KI 2009), Paderborn*, volume LNAI 5803, pp. 306–314. Springer, (2009).

[38] H. Prade and G. Richard, 'Logical proportions - typology and roadmap', in *Computational Intelligence for Knowledge-Based Systems Design: Proc. 13th Inter. Conf. on Inform. Proces. and Management of Uncertainty (IPMU'10), Dortmund, June 28 - July 2*, eds., E. Hüllermeier, R. Kruse, and F. Hoffmann, volume 6178 of *LNCS*, pp. 757–767. Springer, (2010).

[39] H. Prade and G. Richard, 'Multiple-valued logic interpretations of analogical, reverse analogical, and paralogical proportions', in *Proc. 40th IEEE Inter. Symp. on Multiple-Valued Logic (ISMVL'10)*, pp. 258–263, Barcelona, May, (2010).

[40] H. Prade and G. Richard, 'Reasoning with logical proportions', in *Proc. Int. Conf. on Principles of Knowledge Representation and Reasoning.(KR'10),Toronto, May 9-13*, pp. 545–555. AAAI Press, (2010).

[41] H. Prade and G. Richard, 'Analogy-making for solving IQ tests: A logical view', in *Proc. 19th Inter. Conf. on Case-Based Reasoning, Greenwich, London, 12-15 Sept.*, LNCS, pp. 561–566. Springer, (2011).

[42] H. Prade and G. Richard, 'Cataloguing/analogizing: A non monotonic view', *Int. J. Intell. Syst.*, **26**(12), 1176–1195, (2011).

[43] H. Prade and G. Richard, 'Homogeneous logical proportions: Their uniqueness and their role in similarity-based prediction', *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning KR2012*, 402 – 412, (2012).

[44] H. Prade and G. Richard, 'Logical proportions - further investigations', in *Computational Intelligence for Knowledge-Based Systems Design: Proc. 14th Inter. Conf. on Inform. Process. and Management of Uncertainty in Knowledge-Based Systems (IPMU'12), Catania, Italy, July 9-13*, ed., S. et al. Greco, volume 297 of *CCIS*, pp. 208–218. Springer, (2012).

[45] H. Prade, G. Richard, and B. Yao, 'Enforcing regularity by means of analogy-related proportions - a new approach to classification', *Inter. J. of Computer Information Systems and Industrial Management Appl.*, **4**, 648–658, (2012).

[46] H. Prade and S. Schockaert, 'Completing rule bases in symbolic domains by analogy making', in *Proc. 7th Conf. Europ. Soc. for Fuzzy Logic and Technology (EUSFLAT), Aix-les-Bains, July 18-22*, (2011).

[47] M. Ragni, S. Schleipen, and F. Steffenhagen, 'Solving proportional analogies: A computational model.', pp. 51–55. Institute of Cognitive Science, Osnabrück University, (2007).

[48] J. Raven, 'The Raven's progressive matrices: Change and stability over culture and time', *Cognitive Psychology*, **41**(1), 1 – 48, (2000).

[49] T. Sakaguchi, Y. Akaho, K. Okada, T. Date, T. Takagi, N. Kamimaeda, M. Miyahara, and T. Tsunoda, 'Recommendation system with multi-dimensional and parallel-case four-term analogy', *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3137 – 3143, (2011).

[50] U. Schmid, H. Gust, K. Kühnberger, and J. Burghardt, 'An algebraic framework for solving proportional and predictive analogies', *Eur. Conf. Cogn. Sci. 295-300*, (2003).

[51] S. Schockaert and H. Prade, 'Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces', in *Working Papers of the 1st Inter. Workshop on Similarity and Analogy-based Methods in AI (SAMAI'12), Montpellier, Aug. 27*, ed., G. Richard, pp. 38–44. Res. Rep. IRIT/RR–2012-20, (2012).

[52] J. F. Sowa and A. K. Majumdar, 'Analogical reasoning', in *Proc. Inter. Conf. on Conceptual Structures*, pp. 16–36. Springer, LNAI 2746, (2003).

[53] N. Stroppa and F. Yvon, 'An analogical learner for morphological analysis', in *Online Proc. 9th Conf. Comput. Natural Language Learning (CoNLL-2005)*, pp. 120–127, (2005).

[54] N. Stroppa and F. Yvon. Analogical learning and formal proportions: Definitions and methodological issues. ENST Paris report, 2005.

[55] P. D. Turney, 'A uniform approach to analogies, synonyms, antonyms, and associations', in *COLING*, pp. 905–912, (2008).

[56] T. Veale and M. Keane, 'The competence of sub-optimal theories of structure mapping on hard analogies', *International Joint Conference on Artificial Intelligence (IJCAI97)*, **15**(1), 232–237, (1997).

[57] P. H. Winston, 'Learning and reasoning by analogy', *Commun. ACM*, **23**(12), 689–703, (1980).

Rationality Through Analogy: On HDTP And Human-Style Rationality

Tarek R. Besold and **Martin Schmidt** and **Helmar Gust** and **Ulf Krumnack** and **Ahmed Abdel-Fattah** and **Kai-Uwe Kühnberger**[1]

Abstract. At times, human behavior seems erratic and irrational. Therefore, when modeling human decision-making, it seems reasonable to take the remarkable abilities of humans into account with respect to rational behavior, but also their apparent deviations from the normative standards of rationality shining up in certain rationality tasks. Based on previous work on computational analogy-making, on the computational side we give a sketch of the Heuristic-Driven Theory Projection (HDTP) analogy-making system, focusing on the heuristics applied in the current implementation of the system, and subsequently conceptually outline a high-level algorithmic approach for an HDTP-based system for simulating and, thus, predicting human-style rational behavior.

1 Introduction

At times, human behavior seems erratic and irrational. Still, from a top-down perspective, it is widely undoubted that humans have the ability to act in a rational way and, in fact, even appear to act rational most of the time. Even more, the degree of rationality shining up in a human's behavior in many cases is even taken as an indicator for the agents level of intelligence.[2] The study of rationality itself has a long history in science and philosophy that resulted in the formation of mainly four different families of abstract models: logic-based systems, probability-based frameworks, game theory-based models, and accounts based on the use of heuristics. Unfortunately, when comparing the different conceptions, it shows that the resulting definitions are in many cases almost orthogonal to each other (as are the frameworks). Additionally, the predictive, positive power of the frameworks is very limited (if to be found at all), which we see as one of the reasons why rationality after all these years still is a discussed and actively pursued research topic, with new proposals and theories being developed, and new fields joining the debate (cf., e.g., [1]).

The aim of the present paper is twofold: Starting out from previous work on computational analogical reasoning, which resulted in the development of the Heuristic-Driven Theory Projection framework for computational analogy-making, and the corresponding computational implementation HDTP (cf. e.g. [19]), on the technical side we first provide a description of how the heuristics underlying HDTP work, and subsequently on the conceptual side sketch how HDTP could be used as basis for a computational architecture that links rationality to analogy as a basic cognitive capacity of humans.

The general architecture proposed in the conceptual part of the paper, on a very abstract level, can functionally be subdivided into four steps: Given a problem description and domain, select and retrieve analogical situations from memory (**retrieval**). Use the problem as target domain for an analogy, the retrieved situation as source domain, and establish an analogy between both (**mapping**). Transfer solution-relevant knowledge from the source domain to the target domain via the analogical mapping (**transfer**). Apply the newly obtained knowledge in the target domain (i.e. the problem domain) for solving the problem (**application**). One of the main advantages of such a cognitively-inspired architecture is the positive and predictive, rather than normative, character of the underlying theoretical paradigm, actively allowing for deviations from classical paradigms of rationality and seeming "rationality errors", thus also accounting for peculiarities of human-style rationality (as, e.g., to be found in Tversky and Kahneman's Linda Problem experiments [21]).[3]

2 Heuristic-Driven Theory Projection

The Heuristic-Driven Theory Projection (HDTP) framework has been conceived as a mathematically sound framework for analogy-making (cf., e.g., [19]). In the following we will give an outline of the overall basic principles and functionality of HDTP, before having a closer look at its heuristics in the subsequent section.

As explained in [8], HDTP has been created for computing analogical relations and inferences for domains which are given in form of many-sorted first-order logic representations. Source and target of the analogy-making process are defined in terms of axiomatisations, i.e., given by a finite set of formulae. From there, HDTP tries to align pairs of formulae from the two domains by means of anti-unification. Anti-unification is the dual to the more prominent unification problem, and has to the best of our knowledge firstly been studied by Plotkin in [16]. Basically anti-unification tries to solve the problem of generalizing terms in a meaningful way, yielding for each term an anti-instance, in which distinct sub-terms have been replaced by variables (which in turn would allow for a retrieval of the original terms by a substitution of the variables by appropriate sub-terms). The goal of anti-unification is to find a most specific anti-unifier, i.e., the least general generalization of the involved terms.[4] HDTP now extends Plotkin's classical first-order anti-unification to a restricted form of higher-order anti-unification, as mere first-order structures

[1] Institute of Cognitive Science, University of Osnabrück, Germany, email: {tbesold | martisch | gust | krumnack | ahabdelfatta | kkuehnbe}@uos.de

[2] Recently, a feasible positive model of rationality has also been considered as a decisive part in constructing and evaluating systems of artificial general intelligence. For an example see a proposal for a cognitively-inspired decomposition of Turing's classical test for machine intelligence in [2].

[3] A more detailed treatment of these aspects, together with a deeper elaboration of some of the underlying theoretical assumptions and claims, can, e.g., be found in [3] or [4].

[4] Plotkin [16] has shown that for a proper definition of generalization, for a given pair of terms there always is a generalization, and that there is exactly one least general generalization (up to renaming of variables).

have shown to be too weak for the purpose of analogy-making: Think of structural commonalities which are embedded in different contexts, and therefore not accessible by first-order anti-unification only.

Restricted higher-order anti-unification as used in HDTP was first presented in [13]. In order to restrain generalizations from becoming arbitrarily complex, a new notion of substitution is introduced. First of all, classical first-order terms are extended by the introduction of variables which may take arguments (where classical first-order variables correspond to variables with arity 0), making a term either a first-order or a higher-order term. Now, anti-unification can be applied analogously to the original first-order case, yielding a generalization subsuming the specific terms. As already indicated by the naming, the class of substitutions which are applicable in HDTP is restricted to (compositions of) the following four cases: renamings, fixations, argument insertions, and permutations. In [13], it is shown that this new form of (higher-order) substitution is a real extension of the first-order case, which has proven to be capable of detecting structural commonalities not accessible to first-order anti-unification. Unfortunately, the least general generalization loses its uniqueness (which in turn may be interpreted as corresponding to the multiple possibilities humans may find in drawing analogies between a source and a target domain). Therefore, HDTP ranks generalizations according to a complexity order on the complexity of generalization (which in turn is based on a complexity measure for substitutions), and finally chooses the least complex generalizations as preferred ones. From a practical point of view, it is also necessary to anti-unify not only terms, but formulae. Therefore, HDTP extends the notion of generalization also to formulae by basically treating formulae in clause form and terms alike (as positive literals are structurally equal to function expressions, and complex clauses in normal form may be treated component wise). Furthermore, analogies do in general not only rely on an isolated pair of formulae from source and target, but on two sets of formulae. Here, a heuristics is applied when iteratively selecting pairs of formulae to be generalized: Coherent mappings outmatch incoherent ones, i.e., mappings in which substitutions can be reused are preferred over isolated substitutions, as they are assumed to be better suited to induce the analogical relation. Once obtained, the generalized theory and the substitutions specify the analogical relation, and formulae of the source for which no correspondence in the target domain can be found may by means of the already established substitutions be transferred to the target, constituting a process of analogical transfer between the domains.

3 HDTP: Heuristics at Work

Mapping in HDTP is the process of selecting pairs of formulae to be handed to anti-unification in a manner that minimizes the complexity of alignment between two domains. The complexity of alignment here means the sum of complexities of the preferred generalizations for the pairs of formulae that constitute the alignment. Therein lies a meta heuristic that is a key principle in HDTP namely "*an alignment with minimal complexity produces a good analogy*". A brute force approach to finding an alignment for domains with minimal complexity would simply mean to compute the complexity of all possible generalizations of two domains (which minimally is equal to the number of all possible alignments, assuming the ideal case that for each pair of formulae the preferred generalization is unique). Substitutions that were already made by anti-unification earlier are considered complexity-free when reused for generalizations later in the mapping process (even when used for anti-unification of a different pair of formulae). From this arises that the complexity for a generalization of a pair of formulae not only depends on the formulae but also on the substitutions that are regarded as free during anti-unification. Therefore the order in which formulae are anti-unified matters, expanding the search space of mappings even further. Thus, in order to handle this problem space (growing faster than proportional to the formulae in each domain), we must explore techniques to reduce the number of evaluated mappings between domains.

We may distinguish two aspects of heuristics used for mapping. The first viewpoint we can take on heuristics is that they could incorporate knowledge that is not solely based on the formal structure of domains. For instance, the heuristics could use information that is based on previous knowledge of anti-unifying domains from similar fields (bridging this system-theoretical consideration back to a more cognitive scenario, this would correspond to a memory-based reuse of generalizations). Or they could guide the search in such a way that complexity is not solely minimized, rather other goals are incorporated by pruning the search in an appropriate form. Whereas HDTP in its most primitive form is solely a symbol manipulating system, this represents a point where "intelligence" can be incorporated: Mapping for analogical reasoning as here described has an exponential explosion in the search tree and cannot be solved by mere brute force search. The task of intelligence in form of heuristics here is then to advert the threat of exponential search explosion by guiding search in certain, by some standard plausible directions only.

Instead of a full breadth first search through all possible pairings of formulae HDTP focuses the search by employing a locally greedy search for alignment: First a selection heuristic selects pairs of formulae that do not constrain mapping possibilities too fast, but collect support for symbol mappings incrementally in small steps. If no prior mapping is available the pair of formulae is selected that is best according to an heuristic that approximates the anti-unification complexity itself (if multiple such pairs exist, the ones that have most matching predicates on the top-level will be selected). For example aligning $f(a)$ and $f(b)$ would be preferred over the pair $f(a)$ and $g(a)$, as aligning arguments is less complex than aligning functions themselves. We then take the source formula from that pair as a starting point. If the mapping already contains substitutions we simply select the formula as source formula that contains the least amount of symbols that are unmapped or do not appear in the other domain. If multiple such formulae exist we narrow down the choice again by the anti-unification heuristic and choose the source formula that is contained in the pair of formulae that has the lowest approximated complexity. Because the anti-unification heuristic used is not admissible we will only heuristically select the source formula and not at the same time the target formula. All unaligned formulae in the target domain will then have to be considered to form a pair for alignment with the source formula. However, this is optimized by working through them in the order given by the complexity computed by the anti-unification heuristic. To make this computationally more efficient, we exploit the fact that the restricted higher-order anti-unification algorithm is able to search for generalizations up to a specific complexity. The anti-unification heuristic should be good enough to put a formula at the beginning of the list of possible target formulae in such a way that the first ones have relatively low computed anti-unification complexity in comparison to the formulae further down the list. We then pass the computed minimal complexity for generalizations with a specific selected source term on to the following anti-unifications with different target formulae. If they do not have generalizations that are less or equally complex the process can terminate earlier than without a given cutoff value. If the anti-unification heuristic has not accurately predicted the pair of formulae which contains the source formula selected that has minimal anti-unification complexity, this will not generate a suboptimal solution. The algorithm is still locally optimal in that, given a source formula, it will select the best target formula that still needs to be aligned. However, because we employ a greedy search we will not compute anti-unifications for an arbitrary order of source formulae in the alignment, which makes the algorithm globally not optimal. An alignment does not need to include all formulae of the two corresponding domains, but could be partial or even empty. When domains consist of an unequal number of formulae a full one-to-one alignment trivially will not be possible at any rate. Usually formulae from the source domain are left over with no possible partner to form a pair, even if as much pairs of formulae as possible are incorporated in an alignment. Therefore, besides pairing up a source formula with a target formula the case of not pairing it up (and therefore not aligning it) is also considered. For the heuristic that gives the complexity of not matching a formula the balance between not trying to force mappings to hard and not leaving a lot of formulae unaligned is aimed for. We therefore use the metric $(\textit{number of symbols} * 2) + 1$ to make it more complex to not align a formula, than to anti-unify it with a formula that is equal in argument structure. At the same time this does not dismiss alignments when a term with many matching or already mapped symbols is available. The mapping process accordingly is finished when all source formulae have been aligned or determined to have no match. The strategy will always terminate, because in each iteration of the main mapping loop a formula will be taken from the source formula set which is finite (whether that formula will be aligned or not is a different matter). In the end no source formulae are left for inclusion in the alignment, thereby, trying to make the mapping cover as many formulae as possible in both domains with minimal complexity.

In analogy-making restrictions on mapping symbols between domains are often imposed so as to disallow arbitrary mappings from symbols to symbols. The up to now described form of mapping domains allows for multiple mappings of one symbol to symbols within the other domain. However, this may not be desired. Gentner's Structure-Mapping Theory is based on the finding in [6] that people prefer alignments that are structurally consistent. This means that there should be a one-to-one correspondence between elements in source and target domain. Enforcing this constraints further narrows down the search for alignments, as under this restriction an anti-unifier for a pair of formulae does not always exist.

Another addition to HDTP presented in [18] is the encoding of additional knowledge by incorporating sorts. Sorts describe the type of an entity at a general level and can be interpreted as high-level concepts as, e.g., object, massterm, time or number. They reflect our wish not to consider the universe as a homogeneous collection of objects. Their purpose in HDTP is to help restrict the possible mappings during the analogical mapping process by using background knowledge about relations between classes of symbols in the involved domains. Domains used in HDTP are described by a many-sorted logic and sortal ontology. Adding a complexity measure for adjusting sorts in mappings makes HDTP prefer mappings where entities with the same sort are mapped to each other. It thereby serves as another heuristic to speed up computation and produce good mappings.

4 Cornerstones of an Architecture for Human-Style Rationality

In this section, we outline how solving a rationality puzzle can be modeled in terms of HDTP, by this also pointing towards principles for a HDTP-based architecture for a cognitive rationality system. For some first example applications of HDTP-style modeling to classical rationality puzzles we refer the reader to [3] or [14].

As already stated before, in HDTP, source and target domains for analogy-making are represented as theories in a first-order logical language. In the following, we additionally assume that the system has access to a library of previously formalized situations and scenes (i.e., domains that had already initially been pre-compiled, or that have been learned and acquired during runtime up to the present moment in time), corresponding to a human's (episodic) memory of previously seen and experienced happenings and events (here, constraints on human memory could for example be modeled by limiting the number of domains available to the system).

Given the (rationality) problem at hand as target domain for the analogy, the *retrieval* problem within HDTP comes down to selecting a fitting domain from memory as source domain. This can be done in different ways, for example by means of a separate module (similar to the MAC stage in the MAC/FAC model [5]), or by forcing HDTP to construct analogies between all possible pairings of the target domain with a candidate source domain, subsequently taking the heuristics value HDTP computed when constructing the analogy as a measure for analogical distance between domains and proceeding for example with the analogically closest domain as source domain for the analogy. Of course, the outcome of the retrieval process does not have to be unique, and always strongly depends on the heuristics or distance measures used, thereby introducing a degree of uncertainty into the system (matching the uncertainty and irregularities humans exhibit in their decision and rationality behavior). Once a source and target domain have been identified, HDTP constructs an analogical relation between both, *mapping* between elements from source and target domain. The construction of this mapping is based on the previously outlined generalization mechanism, guided by heuristics which try to keep the analogy as simple (i.e. less general) as possible, whilst still maximizing the sub-theories of the sources which can be re-instantiated from the generalization (a trade off close in spirit to the precision/recall problem in pattern recognition and information retrieval). Also here, in most cases the mappings between elements of the respective domains do not have to be unique (e.g. different elements of the source could be mapped to one certain element of the target domain), again introducing a source of uncertainty. In the transfer phase, knowledge from the (with respect to problem solutions richer) source domain is transferred to the target domain (i.e. the problem at hand). Making use of the mappings established in the previous step, the concepts from the source domain are re-instantiated from the generalized theory into the target domain, enriching the latter and giving additional information needed for computing a solution to the problem. In the last step, the newly added knowledge is *applied* in the target domain (e.g. used for reasoning and inference), in most cases yielding a solution to the problem (sometimes, although additional knowledge has been provided via the analogical process, the problem solving process still will fail, a phenomenon reminiscent of human failure in seemingly familiar, in the past already mastered problem situations). This step also includes a consolidation process, integrating the transferred knowledge into the target domain, giving an expanded or richer domain.

Of course, this type of architecture leaves ample space for uncertainty and deviating behavior: Apart of the already mentioned systemic influences, a certain chance of deviation from HDTP's predicted outcome for a certain problem situation is automatically introduced by the use of logical theories as descriptive framework for situations and problems. As with every symbolic formalization, decisive information might accidentally be left out of considerations when formulating the domain descriptions. Nonetheless, we don't see this as a major drawback, but rather as a natural constraint every system trying to predict a phenomenon as complex as human rational behavior has to face, and which even holds in the case where humans try to predict each other. We want to go even one step further, in the future trying to make this uncertain or vague formalization process to a certain extent part of the system via introducing a bayesian-style learning and inference process at the stage of obtaining (complete) domain theories from (most likely only partial) observations.

5 Related Work & Conclusion

We are not alone in our overall plea for a "more cognitive" view on rationality models and systems. Of course, Simon's epochal work on *bounded rationality* [20] has to be seen as an early predecessor. Also, there are tight programmatic connections to the *ecological rationality* [17] movement. On the cognitive science side, amongst others, Kokinov challenged traditional views on rationality in [12]: Observing that rationality fails as both, descriptive theory of human-decision making and normative theory for good decision-making, Kokinov reaches the radical conclusion that the concept of rationality as a theory in its own right ought to be replaced by a multilevel theory based on cognitive processes, proposing analogy as means of unifying the different mechanisms. Utility making then would have to be rendered as an emergent property, emerging in most (but not all) cases, thereby converting rationality itself into an emergent phenomenon and assigning rational rules the status of approximate explanations of human behavior. And also on the empirical side, evidence for the applicability and suitability for the resulting position can be found. In [15], Petkov and Kokinov present JUDGEMAP, a computational model of judgement and choice based on the general-purpose cognitive architecture DUAL [11], and the corresponding AMBR analogy-making system. JUDGEMAP is capable of performing both tasks, giving a judgement on a scale and deciding a choice situation, by means of a process of making forced analogies, exclusively using mapping principles inherited from the underlying AMBR system. JUDGEMAP has been demonstrated to replicate phenomena known from observations of human judgement as, e.g., range and frequency effects, or sequential assimilation effects. Additionally, several simulations run on JUDGEMAP have shown that mechanisms designed for modeling analogy can have influence on judgement and choice, possibly reproducing contextual effects in tasks which at first sight don't seem to be related to analogy-making.

Earlier work has been conducted addressing the application of analogy-making systems to basic problem-solving scenarios (cf., e.g., [9, 10]). Still, concerning the general application of analogy engines to rationality tasks, to the best of our knowledge the number of existing coordinated research projects is still quite limited,[5] and most proposals and frameworks only are in an early stage, at best having proof-of-concept status. Nonetheless, already at this level, available theoretical and practical results are promising enough to justify serious and dedicated research efforts. Here, both sides, the "classical" cognitive and analogy computation movement and the "classical" rationality schools, could profit from further intensifying their interaction and cooperation: The mostly symbolic paradigms used in many applications and implementations dealing with analogy-making could be expanded and enriched by some of the alternate techniques applied when modeling rationality, whilst the numerous frameworks for rationality could significantly profit from including more cognitive aspects and properties of humans into their models and theories (for an example, see some of the considerations concerning subject-related aspects of rationality in [7]).

[5] We explicitly do not consider frameworks and theories for case-based reasoning in decision-making or for case-based decision theory, as the underlying mechanism differs significantly from analogy-making.

REFERENCES

[1] T. R. Besold, 'Rationality in/through/for AI', in *Proc. of Extended Abstracts Presented at the International Conference Beyond AI 2011*, eds., J. Romportl, P. Ircing, E. Zackova, R. Schuster, and M. Polak, (2011).

[2] T. R. Besold, 'Turing Revisited: A Cognitively-Inspired Decomposition', in *Theory and Philosophy of Artificial Intelligence*, ed., V. C. Müller, SAPERE, Berlin, (2012). Springer. (to appear).

[3] T. R. Besold, H. Gust, U. Krumnack, A. Abdel-Fattah, M. Schmidt, and K. Kühnberger, 'An argument for an analogical perspective on rationality & decision-making', in *Proc. of the Workshop on Reasoning About Other Minds (RAOM-2011), Groningen, The Netherlands*, eds., J. van Eijck and R. Verbrugge. CEUR-WS.org, Vol. 751, (2011).

[4] T. R. Besold and K.-U. Kühnberger, 'E Pluribus Multa in Unum: The Rationality Multiverse', in *Proc. of the 34th Annual Conference of the Cognitive Science Society*. Cognitive Science Society, (2012).

[5] K. Forbus, D. Gentner, and K. Law, 'MAC/FAC: A model of Similarity-based Retrieval', *Cognitive Science*, **19**(2), 141–205, (1995).

[6] D. Gentner, 'Structure mapping: A theoretical framework for analogy', *Cognitive Science*, **7**, 155–170, (1983).

[7] I. Gilboa, 'Questions in decision theory', *Annual Reviews in Economics*, **2**, 1–19, (2010).

[8] M. Guhe, A. Pease, A. Smaill, M. Martinez, M. Schmidt, H. Gust, K.-U. Khnberger, and U. Krumnack, 'A computational account of conceptual blending in basic mathematics', *Cognitive Systems Research*, (2011).

[9] R. Jones and P. Langley, 'Retrieval and learning in analogical problem solving', in *Proc. of the 17th Annual Conference of the Cognitive Science Society*, (1995).

[10] M. Klenk and K. Forbus, 'Cognitive modeling of analogy events in physics problem solving from examples', in *Proc. of the 29th Annual Conference of the Cognitive Science Society*, (2007).

[11] B. Kokinov, 'The DUAL cognitive architecture: A hybrid multi-agent approach', in *Proc. of the 11th European Conference of Artificial Intelligence*, ed., A. Cohn, pp. 203–207. John Wiley & Sons, Ltd., (1994).

[12] B. Kokinov, 'Analogy in decision-making, social interaction, and emergent rationality', *Behavioral and Braind Sciences*, **26**(2), 167–169, (2003).

[13] U. Krumnack, A. Schwering, H. Gust, and K.-U. Kühnberger, 'Restricted higher-order anti-unification for analogy making', in *20th Australian Joint Conference on Artificial Intelligence (AI'07)*, volume 4830 of *Lecture Notes of Artificial Intelligence*, Gold Coast, Australia, (2007). Springer.

[14] M. Martinez, T. R. Besold, A. Abdel-Fattah, K.-U. Kühnberger, H. Gust, M. Schmidt, and U. Krumnack, 'Towards a Domain-Independent Computational Framework for Theory Blending', in *AAAI Technical Report of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*, pp. 210–217, (2011).

[15] G. Petkov and B. Kokinov, 'JUDGEMAP - integration of analogy-making, judgement, and choice', in *Proc. of the 28th Annual Conference of the Cognitive Science Society*, pp. 1950–1955, (2006).

[16] Gordon D. Plotkin, 'A note on inductive generalization', *Machine Intelligence*, **5**, 153–163, (1970).

[17] J. Rieskamp and T. Reimer, *Encyclopedia of Social Psychology*, chapter Ecological rationality, 273–275, Sage, Thousand Oaks, CA, 2007.

[18] M. Schmidt, U. Krumnack, K.-U. Kühnberger, and H. Gust, 'Refinements of restricted higher-order anti-unification for heuristic-driven theory projection', in *KI 2011: Advances in Artificial Intelligence*, eds., J. Bach and S. Edelkamp, volume 7006 of *Lecture Notes in Computer Science*. Springer, (2011).

[19] A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust, 'Syntactic principles of heuristic-driven theory projection', *Journal of Cognitive Systems Research*, **10**(3), 251–269, (2009).

[20] H. Simon, 'Theories of decision making in economics and behavioural science', *American Economic Review*, **49**(3), 253–283, (1959).

[21] A. Tversky and D. Kahneman, 'Extensional versus intuitive reasoning: The conjunction fallacy in probability judgement', *Psychological Review*, **90**(4), 293–315, (1983).

Matchmaking: How similar is what I want to what I get

Michael Munz and **Klaus Stein** and **Martin Sticht** and **Ute Schmid** [1]

Abstract. We introduce matchmaking as a specific setting for similarity assessment. While in many domains similarity assessment is between pairs of entities with equal status, this is not true for matchmaking in general. Usually, in matchmaking there exists a source request which triggers search for the most similar set of available entities. Whether an entity is acceptable depends highly on the application domain. We describe a specific scenario where elderly people request support or companionship for activities away from home. The focus is primarily based on neighbourly help, like helping someone to carrying his or her shoppings, finding someone else to enjoy a performance or simply for taking a walk around the block. Then, the scenario is used to formulate requirements for a matchmaking framework and for the matchmaking service.

1 INTRODUCTION

Cognitive scientists consider similarity to play a crucial role in most cognitive processes such as concept acquistion, categorization, reasoning, decision making, and problem solving [5, 4]. Major approaches to similarity in cognitive science as well as in artificial intelligence can be characterized on two dimensions: First, whether basic information about objects is metrical or categorial and second, whether objects are characterized by feature vectors or structural information [10, 5]. In psychology, the typical task under investigation is that subjects are asked to rate similarity of two objects. In this setting, the entities for which similarity is assessed play equivalent roles and often occur as first or second position during evaluation. Furthermore, entities are dissociated from the person who does the rating. However, there are many scenarios, where similarity between a "driver" entity and a series of candidates needs to be assessed. This type of similarity assessment is to the core of information retrieval research and can be characterized by the questions *how similar is what I want to what I get*?

In this paper we introduce matchmaking as a special domain of information retrieval. In general, matchmaking is the process of identifying similar or compatible entities. Requirements stated as a query by a user are matched with descriptions (e.g. of services or social events) provided by other users. Typically, a good match is obtained by identifying features or constraints which are *similar* and – in addition – by features or constraints which are *complementary* for request and candidate entities. Complementary or fitting features are defined by a request/provides relation.

There exists a wide range of application to matchmaking, such as (online) dating, sports, eSports and business [12, 11]. In those domains, the matching process is based on different assumptions about what "similarity" means. In (online) dating a matchmaker tries to bring together people with similar interests or similar personality. Whereas in the area of sports a matchmaker has to consider the skills and competence of sportsmen and of teams when it comes to a matching. In business, a matchmaker could have the job of finding appropriate services for a request. Here, similarity depends on what kind of service one is interested in. The examples are all from different domains, that means finding something that is similar to a request depends on the domain of application.

The paper is organised as follows: first, we review three different approaches of matchmaking applied to different domains. Then we present two different kinds of scenarios, where older people search support or compagnions for activities. The scenarios are used to derive requirements for a matchmaking framework. In chapter 5 we present the components of framework and conclude with a short discussion and future plans. The focus of this paper is on the presentation of the system architecture (backend) by which matchmaking can be realized. We are not concerned with the user interface (frontend).

2 APPROACHES TO MATCHMAKING

In this section we discuss three existing approaches for matchmaking with respect to four major questions:

1. How is the data (advertisements, queries) represented?
2. Does the approach make use of background knowledge?
3. Which matching algorithm is applied?
4. Which fitting measurement is used?

The described approaches are applied in different domains. The approaches [9] and [3] are related to the business domain whereas [2] is related to dating and meeting people. Furthermore, they have a different understanding of what actually similarity means as previously discussed. It is this difference that drives the matchmaking process in different directions.

2.1 Matching Resources With Semistructured Data

The classad matchmaking framework [9] is a centralized resource management system for managing distributed resources. It allocates information, like availability, capacity, constraints, and other attributes describing a resource. Those information are used in the matchmaking process to find a proper match. The idea here is to use classads (classified advertisement), a *semi-structured data* model [1] comparable to records or frames, to describe a resource request or to announce a resource to the system. Classads are modelled via lists of pairs, each containing an attribute name and one or more values, to store semi-structured data. Data pairs are used to describe offered and requested services. For example, when considering to use a workstation, a requester would probably store information about the CPU's capabilities or the disk space, while a provider offering a printing service would describe the printer's throughput. It's possible to define

[1] University of Bamberg, Germany, WIAI, email: {name.surname}@uni-bamberg.de

constraints, restricted user groups and rules to rank each other. Both, service provider and service requester use classad descriptions. This makes it easy to compare the query with the suppliers' offers, looking at similarities of attributes and constraints and to rank the offers found in this process.

For a given request, the matchmaker tries to match the classad of the request to a resource with respect to any constraints given in the classads. The rule-based process evaluates expressions like *other.memory >= self.memory*. The authors focus on the data structure and do not specify a specific matching algorithm. They state that the profiles can be matched in "a general manner" using the specified constraints. Additionally, as goodness criteria, the ranking rules can be applied to find out which classads fit better than others. Unfortunately, further details are not given by the authors.

Finally, the matched entities will be informed by sending them the classads of each other by the matchmaker and the resource provider decides to accept or decline the given request.

2.2 Matching Activities Using Ontologies

R-U-In? [2] is a social network primarily based on activities and interests of users. A user looking for company for an activity (e. g. going to the cinema or to a jazz club etc.) queries system with a short description, including time and place. The matchmaker returns contacts found by the user's social network profile, who have similar interests and are located in close proximity. The found contacts need not be known by the querying user yet. For example, the new person might be a social-network "friend" of a "friend" identified by some social network service.

Users can post their interests and planned activities on the platform in real-time, i .e. planned activities are dynamic and can often change at the last minute. As a result of this, participants in an activity get updates about changes immediately.

An ontology is used to realise the matching process. There are reasoning mechanisms for ontologies based on Description Logic [6] and therefore for ontologies based on OWL [7]. Banerjee et al. used an OWL-based *context model* for their activity-oriented social network. Interests are provided by the user itself and are based on tags. Each interest can be tagged via the dimensions *location*, *category* and *time*. In this way, one can find similar interests by matching on all dimensions: the time (e.g. *evening*, *8 pm*, . . .), the category (*horror movie*, *skating*, *jazz*, . . .) and the location (*Bamberg, jazz-club*).

Tags entered by the user (for describing or querying an activity) are considered as concepts of the ontology. The matchmaker queries the context model which in return gives a set of similar tags. Those tags are then matched with the tags specified in the user profile. Based on the search criteria of a user, activities might match exactly or just partially. The search result of any match is then ranked by its geographical distance to the current location of the requesting user. Suppose, a user stores the activity (Park, skating, 3 pm) and a second user searches for (skating, afternoon). While the activity *skating* is an exact match, *afternoon* matches only partially with *3 pm*. As afternoon subsumes 3 pm it is still possible to match the activity.

In general, the ontology is used to store background knowledge by modelling concepts and relations. For the presented prototype, this is done manually. After a query, the matching process is performed in two steps. First, the context-model is used to get semantically similar tags which are then compared to the tags of the other user's activity descriptions. However, details on how the tags are compared and matched and how the results are ranked (beside of the geographical distance) are not discussed by the authors.

2.3 Matching Web Services Using Clustering

Fenza et al. [3] propose an agent-based system to match semantic web services. There are two different kinds of agents in the system: a broker agent (kind of *mediator*) and one or more advertiser agents. A request for a service is handled only by the broker itself. When it encounters a request it converts it into a *fuzzy multiset* [8] representation. With these multisets a *relevance degree* is assigned to each possible ontology term that describes a web service according to the place, where the term occurs. For example, if the term occurs in the input specification of the service then it will get a relevance degree of 1. If it occurs in the textual description, then it will get a degree of 0.3 and so on. In this way, it is possible to weight the term for different occurrences via categories.

Advertiser agents interact with web services and with a single broker agent. Each web service description[2] is converted into a fuzzy multiset representation. Note that the broker does the same with the user's request. So in the end, a broker has a fuzzy multiset of a request and advertiser agents have a fuzzy multiset for each registered service. The broker sends the fuzzy multiset of the request to the advertiser agents to find an appropriate web service. If a web service matches with a request then the matched web service is returned by the broker, the corresponding fuzzy multiset is stored to a central cluster and its job is done. Otherwise, the broker tries to find an approximate service by using a knowledge base which is divided into two distinct sets of knowledge: *static knowledge* and *dynamic knowledge*.

There are several ontologies modelled to specific domains in the static part of the background knowledge. To calculate an approximation, the broker modifies the original request by utilizing the domain ontologies. The dynamic part of the knowledge consists of the cluster of fuzzy multisets where the web service descriptions of the known providers are stored (encoded as fuzzy multisets). It compares the fuzzy multiset of the modified request with the fuzzy multiset of each cluster center and selects the services most similar to the request. That is, services with the minimal distance to the request are candidates for an approximation. The similarity is therefore measured using the distances in the fuzzy cluster.

3 SCENARIO

Many older people at a specific age often don't leave their home on their own, because of several factors: they might be more anxious in late life or may have physical health problems. They also might be more socially isolated, have significant changes in living arrangements, the loss of mobility, fewer flexibility, and loss of their independance. All this factors contribute to withdrawal from social life and thereby reduce quality of life.

To have an independent life at an old age mobility is crucial to being active and to stay in contact with other people. Therefore, the goal is to improve mobility and social connections of (older) people. That is, to bring together people who do need help, but also people who want to meet others and people who offer help.

The idea is to build a platform mainly based on collaborative help, but also includes service providers. In this paper we focus on the matchmaking framework of the platform. The context of collaborative help means to match people asking for help to people offering help and vice versa. People looking for help are going to be mostly elderly people and people offering help are going to be mainly volunteers.

[2] A specific ontology for describing web-services named OWL-S is used.

In the following, we are looking at two kinds of scenarios the system might be confronted with. They represent two different approaches the matching service has to deal with. The first scenario describes matchmaking based on best fit, while the second scenario describes matchmaking based on similarity. Fitting and Similarity are discussed in more detail in chapter 5.

Scenario 1 (a) *Mrs. Weber is looking for a babysitter for her 3 years old daughter on weekends on a regular basis from 1 pm until at least 4 pm.* (b) *Mrs. Peters is an 82 years old lady who needs attendance in taking the public bus lines. She thinks it's too complicated for her, because she has to know how to buy a ticket, where to change bus lines, and the bus station to get off.* (c) *Zoey plays guitar and goes to the music lessons every Wednesday after school at 3 pm and takes the bus line 901. She would agree in attending someone else.* (d) *Aylia Özdan is 31 and will help someone else on Sundays if it's between 11 am – 8 pm."*

The examples provide some information: Zoey works as a volunteer every now and then. She would accompany someone else under some conditions. She is using the bus line 901 at a very specific time (3 pm). So the person to accompany should use the same bus line and should be there before the bus arrives at the bus stop. If these conditions are met, Zoey will accompany a person until the bus arrives at the bus stop she has to get off. The request of Mrs. Peters will match this offer, if she also uses bus line 901 and waits at the same bus stop.

The request of Mrs. Weber looking for a babysitter on weekends matches only partial with the offer of Aylia Özdan to help someone on Sundays. Note, there is a difference between the help of Zoey and Aylia Özdan. While Zoey is helping the old lady by courtesy as long as it doesn't interfere with her plans, is Aylia Özdan helping someone else on purpose by spending some of her spare time.

Scenario 2 (a) *Mr. Beck is 70 years old and interested in playing Backgammon regularly. None of his acquaintances is playing it, and he doesn't know any other person who might play it.* (b) *Mr. Miller plays regularly Poker with his buddies on Friday evening and is always looking for new participants who are interested in it.* (c) *Mr. Novak wants to play Skat with a friend and they are looking for a third player.*

In this scenario Mr. Beck is looking for someone who is playing Backgammon. Because there is nobody else in the system who is interested in it, no exact match is possible. But there are other requests stored in the system, like Skat and Poker, the system could offer instead as similar matches. The implication is, if one is interested in playing Backgammon, one could also be interested in playing other games. Here, similarity means finding someone with similar interests. At the level of a "parlour game" all these requests are similar. So they should appear in the result list as possible matches.

4 REQUIREMENTS

In chapter 3 we described two different kinds of scenarios where matchmaking is either a best fit or a similarity match. From these scenarios various requirements arise which have to be considered in a matchmaking framework.

Activities A matchmaking framework has to deal with different kinds of requests when it comes to a matchmaking. The essence of scenario 1 is that someone is searching for help and someone else is offering a helping hand. Here, a request for help and an offer to help should be matched. In scenario 2 the situation is different. Users search for other users with similar interests. Here, a matching service should match users with same or similar interests. All requests have in common that they concern "activities". As a result, requests are essentially a search for activities. Therefore, one requirement to a matching framework is to handle those activities properly.

While R-U-In? (Sec. 2.2) matches users with similar interests it does not handle fitting of offers with requests. Classads (Sec. 2.1), on the other hand, support these different roles but only provides the data structure and no matching algorithm. The fuzzy multiset approach (Sec. 2.3) also supports the different roles of requester and provider as long as all parameters can be expressed as fuzzy multisets.

Constraints Activities are essentially sets of constraints. We distinguish between hard ("must") and soft ("should") constraints. If only one hard constraint can't be satisfied the whole activity won't be satisfied at all. If one soft constraint can't be satisfied, the activity will still be available as a possible match. In the context of matching similar activities, a soft constraint evaluated to false means matched activities do not fit so well. Note, what is seen as hard and soft constraints depends highly on user expectations. Scenario 2 (Backgammon) is an example where a lot of soft constraints exist: time, place, day of week, and even the activity itself. Whereas the examples of scenario 1 have a lot of hard constraints, like time, place, bus line, and day of week.

The classad approach supports modelling of constraints, but the authors do not distinguish between hard and soft constraints. To overcome this, one could think of utilising annotations to distinguish categorically between hard and soft constraints. Both of the other approaches do not have explicit constraining mechanisms. The activity-oriented network R-U-In? models interests of users via three dimensions: "time", "category", and "location". The model could be extended by an additional dimension specifying constraints. The downside of this approach is all dimensions of the model are represented by an ontology. That means, only the concepts of hard and soft constraints could be modelled, but no instances. Otherwise, constraints would be predefined and too inflexible. The fuzzy multiset approach directly supports (weighted) matching of soft constraints, but the datastructure has to be adapted to directly support hard constraints.

Roles The scenarios (Sec. 3) present two basic situations. On one hand there are people needing help or searching for other people with same interests. On the other hand, there are people offering their help. But there are differences in the degree of helping someone. Some users do volunteering work and other users just do someone a favour. In the context of neighbourly help it's important to distinguish between these, because there is a difference in the social commitment. Via user roles these differences can be modelled. Roles can represent the different expectations users have, when searching for activities. For example, users searching for help expect to find someone offering help. The same is true vice versa. That is, a volunteer who is looking for users needing help expects to find posted activites.

As already stated, the goal of the proposed framework is to improve social life of older people by focusing on mutual assistance and neighbourly help. Therefore, the default role represents those users who are searching for help or looking for company. The difference in helping is taken into account by another two roles. That is, volunteers and favours are represented by separate roles, because they have different characteristics. People doing volunteering jobs do it on a regular basis and offer assistance explicitly. Usually, they have weaker conditions under which they are willing to help and try to be more flexible in scheduling an appointment with someone who needs help. Furthermore, they are willing to spend some of their spare time in helping others. In a third role are those users represented doing favours. The difference to volunteers is, they don't help out regularly and they have stronger conditions under which they are willing to help someone. Doing someone a favour is usually a very time-limited act, so time is a hard constraint. Another characteristic of a favour is most people will do it only, if it doesn't interfere with their own plans and they don't have to change their schedules.

Classads and fuzzy multisets are designed to match available resources to requests of resources. In those situations there exist only the two roles of service providers and service requesters, and no further differentiation is needed. An activity-oriented social platform like R-U-In? does only have one group of users. All users are interested in doing activities in their spare time. This leads to clear expectations when using the platform. They either search for or post activities. Because of an activity-oriented user group only one role is needed to represent them.

Knowledge The matchmaking process can be improved by providing knowledge. For matchmaking based on similarity different sources of knowledge are suitable. That is, background knowledge and user profiles.

Background knowledge is the general knowledge available in the system. Activities are represented by it and the knowledge is used to tell how similar different activities are. Then, the matching service can offer similar activities by evaluating it (e. g. by taxonomic relationships). Suppose, someone searches for Backgammon, but there is no direct match (as the situation is in scenario 2). The matchmaking service can offer Skat and Poker instead and ignore other available activities. Background knowledge has a disadvantage, though. It is often static and explicit. It doesn't change often and represents knowledge to a specific time. Moreover, updating static knowledge is often time consuming. To overcome this we consider to utilise user input. The initial background knowledge would be more dynamic and converge to requested user activities.

A user profile is also helpful in the matching process. It has two advantages: first, in the profile are those information stored a user normally doesn't want to re-enter everytime a search is submitted. Second, information stored in the profile can be used to filter off matched activities which do not fit. In this way, the result list can be improved. Information in the profile could be among other things: interests of a users, trust to other users, constraints, and a user rating. Activities of other users should be withhold in the result list, if a user marked others as disliked or even untrusted. Trust and user ratings are really important in the context of neighbourly help and are valuable information in the matching process. A matching will get a much more higher rating, if there already exists a relationship of trust between users. The implication is, they did some activities in the past, know each other and would like to do future activities together.

Classads [9] have in some extend a user profile, but they do not have any background knowledge. In classads only a resource can specify a list of trusted and untrusted requesters, so the relationship here is unidirectional. The activity network R-U-In? [2] uses both background knowledge and user profiles for a matching. While user profiles are updated in real-time, the background knowledge has no dynamic update mechanism so far. Moreovere, there exists a policy repository where a user can define policies for participants when attending an activity. The downside of the platform is one can't rate users, can't mark them as liked or unliked, and it's not possible assigning any status of trust. In the fuzzy multiset approach [3] there is a distinction between background knowledge and fuzzy multisets. The background knowledge is realised in the form of domain ontologies and is static, according to the paper [3]. Whereas, fuzzy multisets are dynamic and are updated according to changes of services.

Requests The matching framework should be able to differentiate between two different classes of requests, *immediate request* and *stalled request*. They represent different searches of activities. Suppose, a user wants to play Backgammon and issues a search. In the profile aren't defined any preferences, like hard constraints. Further assume no exact match is possible, but there are two other activites stored (Skat and Poker), as the situation in scenario 2. As a result, the best matches are Poker and Skat. The user has now the choice of either choosing any of the matches he or she is interested in by contacting the other person or to store the request in the system. A user should have the opportunity to store it, if he or she doesn't like any of the activities found or the results are not as expected

Everytime a user initiates a new search for activities to the system he or she immediately receives all matching results best fitting the search. It is an immediate request. The result list is ordered according to a weighting so the best fitting activities are on top. In case, the user isn't happy about the found matching results, he or she has the opportunity to initiate a stalled request. The request of the user is stored in the system's activity database and is from now on in monitoring modus. Depending on the preferences stored in the corresponding user profile the user will be notified about new activities of other users similar to his or her activity request. Utilising a stalled request one can find a match that best fits over a period of time while an immediate request matches the best fit of the current available activities.

Classads [9] and the fuzzy multiset approach [3] match a request to the current available set of services, only. They do not have to distinguish between different kinds of requests in their systems. Whereas, in R-U-In? [2] you can search for and post activities. Activities are stored in a so-called activity groups reposititory. The difference here is, stored activities are not in any monitoring mode, so users are not being informed about searches of other users. Rather, in R-U-In? a user will only be informed, if the requester is interested explicitly in an activity by sending him or her a message.

For the proposed system based on neighbourly help the described requirements are mandatory to the process of matchmaking. Because none of the approaches is appropriate for our needs we propose a matching framework with the required components.

5 COMPONENTS OF A MATCHMAKING FRAMEWORK

We introduce a framework with respect to the requirements identified in chapter 4. Figure 1 depicts all components of the proposed matchmaking framework. It shows the interaction between the components, in which the matchmaker is the key component. A user searching for activities initiates a request to the system. All interaction between a user and the system is via a mediator. The mediator decides whether it is an immediate request or a stalled request. If it's an immediate request the matchmaker will be called. For finding similar activities or activities which fit to a given request the matching algorithm uses the underlying databases. That is, the background knowledge, the user profiles and the stored activities. A result list is then returned in response to the mediator. If the request is a stalled

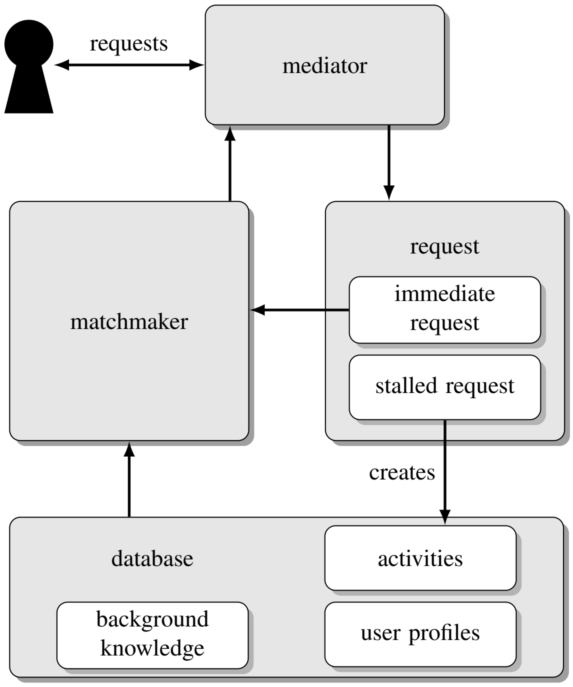


Figure 1. Components of the matching framework and their interaction. The matchmaker is the key component of the system. It uses the underlying database for a matching and propagates the results to the mediator.

request an activity will be created in the activities database. There are two things to be aware of: first, the activity is in a monitoring mode. Second, a stalled request can only follow up on an immediate request. Whenever there is a new match for a stalled request the user will be informed.

5.1 Representing Constraints

Descriptions of activities as those mentioned in chapter 3 consist of features, such as gender, time, location, and the name of the activity itself. These features describing an activity are viewed as constraints for a matching and are classified by two dimensions:

similarity	$\leftrightarrow$	complement
hard constraints	$\leftrightarrow$	soft constraints

Some features need to be similar like the activity. Here reflexivity of mapping holds. On the other hand, some features need to be complementary. For example, the relationship between *needs car/offers car*. Here we speak of fitting and not of similarity. The mapping of fittings can be modelled in such a way that the resulting scale corresponds to a similarity mapping. So that both similarity and fitting can be processed together.

Hard constraints can be encoded using arbitrary complex boolean formulas on object properties while sets of *weighted* propositions are used for soft constraints. For example, let's assume that Mrs. Peters from scenario 1 only wants help from women who are at least 30 years old (hard constraint). This can be formalised as:

$$other.gender = female \quad \wedge \quad other.age \geq 30 \qquad (1)$$

where *other* is a reference to a potential activity partner (similar to [9]). Consider the request of Mrs. Peters finding someone assisting her in riding the public bus as activity a_1:

$$requires(a_1, assistance) \qquad (2)$$

Requires relations are matched to corresponding *provides* relations of other activities. Say a_2 given by another user, namely Aylia Özdan. Both relation will be used to check, if the activities fit, as:

$$provides(a_2, assistance) \qquad (3)$$

The matchmaker must know that the relations *requires* and *provides* are matchable. However, matching two *requires* relations would not solve any problems. Whereas, relations of the same type (e.g. *likes*) would match in a similarity check:

$$likes(other, backgammon) \simeq likes(other, skat) \qquad (4)$$

Using constraints, time and loaction related restrictions can also be modelled. For time related restrictions it's necessary to handle intervals to check temporal overlaps. Location related restrictions calculate and weight distances. The distances are used for ranking purposes. Matches which have a shorter distance are better matches as similar pairs of matches but with greater distance.

5.2 Matching on Constraints

Checking hard constraints can be done by comparing the *requires* and *provides* relations of both, activities and the user profiles. If a hard constraint is violated by an activity description or an involved user profile, the activity will not be considered further in this query. Matching hard constraints should be done before soft constraints are considered. In this way hard constraints are used as filters to omit activities that are being violated. Soft constraints have to be checked only on the remaining set of activities to calculate values of the matching quality.

Soft constraints have different weights, i. e. a value between 0.0 and 1.0 representing its importance to a user. These weights are either derived from the user profiles or from the user's query where the requester can specify the importance of each constraint.

If a soft constraint doesn't match, the matchmaker can

1. check the *severity* of the violation (e.g. the other's age is 38 while the claimed age is 40; this violation would not be as strong as if the other's age was 12). Note that this is only possible if a distance between the claimed and the actual value can be obtained (here difference in ages).
2. combine the severity of the violation with the weight of the constraint and find out how severe this violation is for the complete activity. Lower weights of constraints might qualify severe violations and vice versa.

If we assume that the severity can be normalized to a value between 0.0 and 1.0 where 0.0 means no violation and 1.0 represents the hardest possible violation, the *matching violation* V can obtained by a sum

$$V = \sum_{c \in C} s_c \cdot w_c \qquad (5)$$

where s_c represents the normalized severity of the violation of feature c and w_c the weight given by the user. C is the set of all relevant constraints.

In this way, it is possible to calculate for every remaining activity (after checking the hard constraints) a value of how well it fits to a query. A low V means a better fitting. According to these values, target activities can be ranked and presented in the corresponding order.

5.3 Knowledge from User Profiles and Missing Knowledge

We do not only distinguish hard and soft constraints, but also *profile constraints* and *on the fly constraints*. These constraints refer to where they are defined. Profile constraints are defined in user profiles and are used for recurring constraints only. If a user has defined constraints via the profile, the system will take them into account automatically when initiating a request. It's a way of constraining the search implicit. On the other hand, it should be possible to define constraints manually when doing a search. Those constraints are specified on the fly and are valid only for a specific request. Manually constraining the search should have higher priority as constraints in profiles. For this reason, different knowledge has different priority. Information given in profiles have higher priority as background knowledge. A request has in turn higher priority as profiles. So knowledge with higher priority overwrites lower priority. As a result, constraints defined in the profile influence the search results implicitly, whereas constraints defined on the fly influence it explicitly.

Suppose, a user has $ignore(cardgames)$ in his profile the constraint specified and searches for Backgammon. Skat, Poker, and chess are in the system as available activities. Because Backgammon isn't available, the only similar activities are Skat, Poker, and chess. The matching service just offers chess as an alternative activity and discards the card games Skat and Poker, because they're on the ignore list. Now suppose, the same user initiates an explicit request for Skat. The request has higher priority as the constraint in the profile and overwrites it. This approach allows users to find still activities explicit, even when the profile states otherwise, by overwriting constraints.

Further, it is also important not to treat unmatched constraints as fails because of missing information about the feasibility on the other side. Assume Zoey wants to attend a concert, but needs someone with a car to go there. So the car is a requirement that can be modelled as a (hard) constraint: $requires(car)$. Her neighbour Tim also wants to go to the concert. However, he doesn't mention in his stalled activity that he's going to drive with his own car. The problem here is, Zoey wouldn't find him although the activities would match. In this case, the matchmaker should identify the match and the missing fulfiller (car). Then, inform Zoey about the possible match and propose her to contact Tim to check, if the activity can be matched anyway. After contacting Tim, Zoey is able to go with Tim to the concert by car.

Missing information can be treated as *wild cards* which match everything. The matchmaker doesn't know if Tim possesses a car, but the requirement is assumed to be fulfilled. However, the activity is marked as *uncertain* until Tim confirms he's going to the concert by driving his own car.

5.4 Presentation of Results

The approach we're going to use here is to present *all* matching results to the user. For this purpose, the result list is divided into three subsets: matches with complete information, matches with incomplete (missing) information, followed by matches violated by hard constraints. The results within the first subset are ranked by violation of soft constraints. Matches with no violations come first, then matches with low violation and finally matches with high violation. To improve the subset of matches with missing values the user is asked to provide additional information.

6 CONCLUSION AND FUTURE WORK

In this paper we proposed a framework for matchmaking similar activities. It is part of a larger web-application called EMN-MOVES. The target group of the system are older people and the overall goal is to improve their mobility and their social life. We described two different kinds of scenarios the system might be confronted with to derive the requirements of the framework. We have evaluated the requirements against existing approaches and concluded none of these can fully support our needs for a platform based on neighbourly help.

The presented framework will be the starting point for the development of a general framework for matchmaking. Currently, we are designing an algorithm which allows to calculate both similarity matches and best fits and incorporates a goodness criterion for ranking the results.

ACKNOWLEDGEMENTS

This work is funded by BMBF grant 16SV5700K (Technology and Innovation), Cooperation project "Europäische Metropolregion Nürnberg macht mobil durch technische und soziale Innovationen für die Menschen in der Region" (EMN-MOVES). We thank members of the senior citizen councils of Bamberg, Erlangen, and Nürnberg. We also thank the reviewers for their helpful comments.

References

[1] Serge Abiteboul, 'Querying Semi-Structured Data', in *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, pp. 1–18, (1997).

[2] Nilanjan Banerjee, Dipanjan Chakraborty, Koustuv Dasgupta, Sumit Mittal, Seema Nagar, et al., 'R-U-In?-exploiting rich presence and converged communications for next-generation activity-oriented social networking', in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pp. 222–231. IEEE, (2009).

[3] Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore, 'A hybrid approach to semantic web services matchmaking', *International Journal of Approximate Reasoning*, **48**(3), 808–828, (2008).

[4] Dedre Gentner and Arthur B. Markman, 'Defining structural similarity', *The Journal of Cognitive Science*, **6**, 1–20, (2006).

[5] Robert L. Goldstone and Ji Yun Son, 'Similarity', *Psychological Review*, **100**, 254–278, (2004).

[6] Javier González-Castillo, David Trastour, and Claudio Bartolini, 'Description Logics for Matchmaking of Services', in *IN PROCEEDINGS OF THE KI-2001 WORKSHOP ON APPLICATIONS OF DESCRIPTION LOGICS*, (2001).

[7] Ian Horrocks and Peter Patel-Schneider, 'Reducing OWL Entailment to Description Logic Satisfiability', in *The Semantic Web - ISWC 2003*, eds., Dieter Fensel, Katia Sycara, and John Mylopoulos, volume 2870 of *Lecture Notes in Computer Science*, 17–29, Springer Berlin / Heidelberg, (2003).

[8] Sadaaki Miyamoto, 'Information clustering based on fuzzy multisets', *Inf. Process. Manage.*, **39**(2), 195–213, (March 2003).

[9] Rajesh Raman, Miron Livny, and Marvin Solomon, 'Matchmaking: Distributed resource management for high throughput computing', in *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on*, pp. 140–146. IEEE, (1998).

[10] Ute Schmid, Michael Siebers, Johannes Folger, Simone Schineller, Dominik Seuss, Marius Raab, Claus-Christian Carbon, and Stella Faerber, 'A cognitive model for predicting aesthetical judgements as similarity to dynamic prototypes'.

[11] D. Shaw, P.E. Newson, P.W. O'Kelley, and W.B. Fulton. Social matching of game players on-line, 2005.

[12] *Online matchmaking*, eds., Monica T. Whitty, Andrea J. Baker, and James A. Inman, Basingstoke, 2007.

Perceptual Similarity and Analogy in Creativity and Cognitive Development

Georgi Stojanov[1] and **Bipin Indurkhya**[2]

Abstract. We argue for the position that analogy represents the core mechanism in human cognitive development rather than being a special cognitive skill among many. We review some developmental psychology results that support this claim. Analogy and metaphor, on the other hand, are seen as central for the creative process. Whereas mainstream research in artificial creativity and computational models of reasoning by analogy stresses the importance of matching the structure between the source and the target domains, we suggest that perceptual similarities play a much more important role. We provide some empirical data to support these claims and discuss their consequences.

1 Introduction

Analogy, together with its close cousin metaphor, is considered, by some accounts, to be fundamental to thought itself [41, 43]. Similarities and analogies are also known to play a key role in creativity [e.g. 43, 15, 16, 18]. Here, although many studies of creativity emphasize that one of the underlying mechanisms is that of re-conceptualization [17, 55, 60 and, to some extent 33 through Karmiloff-Smith's notion of *representational-redescription* processes in human cognitive development] most artificial creativity systems focus on generating a product that is considered to be creative rather than viewing creativity as a re-conceptualization of a given object/situation [58]. To our knowledge, there have been only a few attempts to model the process of re-conceptualization itself, which lies at the heart of creativity [20, 21, 26, 44]; Davies and Goel [10] proposed to tackle analogical mapping problem by re-representing, in visuospatial domain, the knowledge of the source and the target, so that similarities that may not be easily noticeable in amodal (i.e. logical formuli) representations will pop-up and can be mapped. However, computational models of analogy largely consider analogy as a special cognitive skill or heuristic that is evoked in some situations on top of other cognitive processes.

The aim of this paper is to articulate a view that sees analogy as a fundamental mechanism of cognitive development, and examine the implications of this view for modeling creativity. The paper is organized as follows. We start in Sec. 2 with a discussion of analogy and creativity; in Sec. 3 we briefly review the computational approaches to modeling analogy and comment on their shortcomings. We continue with Sec. 4 by discussing the limitations of the existing artificial creativity systems. In Sec. 5, we present our framework by connecting analogy with Piaget's mechanisms of assimilation and accommodation. In Sec. 6, we discuss some implications of this framework for the role of similarity in analogy and creativity, and in Sec. 7 we summarize our main conclusions.

[1] Computer Science, Mathematics, and Science department, The American University of Paris, email: gstojanov@aup.fr

[2] Computer Science department, AGH University of Science and Technology, Cracow, Poland, email: bipin@agh.edu.pl

2 Analogy in Creativity

Analogy has been recognized as a key mechanism of creativity [6, 17, 20, 21, 36, 43, 47]. However, one must distinguish between two modes of analogy. On one hand, analogy refers to 'seeing one thing as another', and on the other hand it refers to the process whereby the structure and the attributes of one object or situation (the source) are mapped to another object or situation (the target). This latter mechanism seems contrary to creativity according to many accounts [29], and so it needs a little elaboration.

Every conceptualization (of objects, situations, visual scenes) involves loss of some potential information: potential differences are ignored between two objects that are put in the same category, and potential similarities are ignored between two objects that are put in different categories. The concepts and categories, and their underlying cognitive structures that naturally evolve through a cognitive agent's interaction with the environment reflect the priorities of the agent. The information that is retained in the conventional conceptualization is the one that has been useful to the agent in its phylogenetic and ontogenetic past. So, as long as one stays in the familiar domain (in which the conventional conceptualizations are very useful), and the problem at hand does not require the potentially lost information, reasoning from conventional operations and conceptualizations may be very efficient. However, as soon as the problem does require new information, the existing conceptualization stops being useful, and a new conceptualization becomes necessary. In such situations, analogy, as it is traditionally construed, becomes a hindrance, because it reinforces the existing conceptualization; and metaphor becomes a very useful heuristic. (See also [17, 25, 28]). We should note here that various researchers have characterized the relation between metaphor and analogy in different ways (see also [27]), but for our purpose here, we are taking the view that metaphor can be applied to virtually all comparisons between two objects/situations. Though this process may be triggered by noticing some similarities between them, but sometimes the very act of comparing the two creates the similarities. In this regard, we echo the position taken by Aubusson et al [1] who write: "While not always the case, there appears to be a tendency to use the term analogy when the comparison is extended highlighting a range of similarities and differences between two things. Thus, all analogies are metaphors but not all metaphors are extended into analogies."

If we follow this argument, analogy, in its traditional sense at least, which is based on structural similarities, turns out to be an anathema to creativity. The reason is that analogies are based on mapping the structure or attributes of the source, to the structure and attributes of the target. So an analogy, which is based on the existing conceptualization of the source, will retrieve targets with similar structure, thereby further strengthening the existing conceptualization of that very target. In some cases, this may be enough to solve the problem, e.g. by bringing to the forefront some of the less prominent attributes of the target. But if the problem could not be solved because of needing new information, then a structure-mapping based analogy approach will not be very useful.

3 Computational Models of Analogy

Much of the research on computational modeling of analogy has worked with what might be characterized as *mapping-between-existing-representations* paradigm, where there are given representations of the source and the target, and various algorithms are applied for mapping parts of the source to parts of target. Models differ from one another with respect to whether mapping between relations is preferred over attributes; or whether an incremental or a distributed approach is applied to compute the mapping [15, 13, 22, 23, 24]. All these approaches have severe limitations in that they cannot model emergence of new structure, which is very crucial as far as creativity is concerned. (For a good critical overview see [9]). Though these models do capture a certain aspect of creativity in noticing new connections between existing knowledge, and in importing novel hypotheses from the source to the target, they do not produce a paradigm shift of Kuhnian kind. In this regard, models based on corpus-based analyses and distributed representations seem more promising [59, 61], but so far they are limited to linguistic metaphors.

In contrast, some other approaches have focused on the process of representation building itself, notable among them being the work of Hofstadter and his colleagues. In this paradigm, the appropriate representations of the source and the target and the mapping between them evolve together by parallel processes that interact with each other [26, 41]. As another example, Yaner and Goel [63] propose an analogy-based problem solver that builds a representation of the target, but is different from Hofstadter's approach because the outcome of this representation-building process is deterministic. Such approaches come closer to being able to model creativity, for often creative insights emerge from applying a concept to an object (or a low-level representation of it) that is not habitually associated with it. In our earlier work, we have formalized this process [25], and have applied it to model creativity in legal reasoning [26], but clearly much more work remains to be done. Moreover, in real-life, a number of different cognitive processes may act in consort to generate a creative insight, modeling of which may require hybrid architectures [31, 43].

4 Artificial Creativity Systems

Research in creativity (usually conducted within psychology), since its inception, has shown a bias towards exceptional individuals, that is the big-C creativity. Consequences of these views were felt in the research in Artificial Creativity (AC) and Computational models of Metaphor and Analogy (CMA) where often the existence of creativity and analogy modules was hypothesized. Attempts to model those modules were made and virtually all of them aimed at big-C creativity (composing music, writing novels, painting…) as opposed to mundane creativity. In [58] we give an extensive review of the approaches in AC research. Below, we only mention the main conclusions.

What can be said about the vast majority of existing Artificial Creative systems?

-Virtually all of them focus on the product (a consequence of the ***product generating paradigm*** in which they are working) rather than on the process. Thus, we may call this approach ***top-down*** or ***product-first*** approach;

- Most of them are given, in advance, ***a detailed (hard-coded) description of the domain***. This can be: language syntactic rules, narrative structure, and some semantics for artificial prose writers; musical notation and rules for artificial composers and creative interpreters; basic drawing primitives for artificial painters; basic mathematical operations, a lot of search heuristics with evaluation functions, and big knowledge/fact base for artificial scientists);

-All these AC systems appear to be ***closed systems*** in the sense that there is no way to appreciate, and build upon, the feedback from naïve (or not) observers;

-None of these AC systems are ***socially embedded*** except, in a certain sense, via their designers who themselves, receive the feedback from the audience and eventually make the necessary changes in their programs;

-Finally, virtually all of the researchers within AC looked for inspiration into the existing theories of the domain in which their systems are supposed to be creative: literary and narrative theory, music theory, visual arts, etc. This goes counter to our intuitions and the empirical facts that many artists and scientists report that actually combining domains (in which they not need be widely recognized but simply familiar enough) has resulted in some of their most creative outputs.

On a more abstract level (and maybe with a bit of risk of oversimplification) we could say that the majority of AC systems to date quite resemble the generic architecture of a GOFAI expert system from the '70s and '80s of the last century. The aim of those expert systems was to replace human experts in some particular narrow domain like: general MD practitioner who would come up with a diagnose given the symptoms of the patient, or an operating system administrator who would know how to fine-tune the parameters and optimize the functioning of a complex operating system. Just like the AC systems, they usually contained a huge knowledge base, many heuristics, and representative cases (in case-based reasoning). The knowledge representation was mainly symbolic, and some systems also included probabilistic reasoning.

Given the dominant approaches to computational models of analogy (i.e. focusing on relational matches between two hardcoded representations), it is probably not surprising that although (as mentioned in section 2) analogy is often seen as a key factor to the creative process, and we rarely see AC systems that use analogy. (However, see [64], as well as 12 and 5 for recent counter-examples).

Research in modeling creativity in scientific discovery (from Lenat and Brown [40] and Langley at al [38] in the '80s) to King et al in 2009 [35] seem to address some of the issues raised in the section above. For example, Lenat's Automated Mathematician can build upon its own output an incorporate input from outside. But, we think that this is just an illusion just because the domains in which these systems operate is more constrained and it is relatively easy to adapt existing formalisms to describe their inputs and outputs. An excellent critical review of these systems is given in Chalmers et al [9].

5 Analogy in Cognitive Development

Reasoning by analogy is sometimes seen as a pinnacle of cognitive development. Goswami in [19], for instance, notes that in Piaget's account, analogical reasoning occurs only in adolescence during the formal operation stage. In contrast, Goswami goes on to review a number of research results that show that analogy is far more pervasive in cognitive development, and occurs much earlier, i.e. even in 3 and 4 year olds. Goswami argued that children in some of Piaget's experiments were not able to solve a relational analogy problem because they were not familiar with the causal relations among the objects (e.g. **'bicycle : handlebars :: ship : ?'**, handlebars are used to guide the bicycle in the *same way* the **ship's wheel** is used to steer the ship). However, a close look at Piaget's numerous studies reveals that he has also noted the onset of analogical thinking manifested as a sensorimotor schema at a very young age: "At 1;4(0) L. tried to get a watch chain out of a match-box when the box was not more than an eighth of an inch open. She gazed at the box with great attention, then opened and closed her mouth several times in succession, at first only slightly and then wider and wider. It was clear that the child, in her effort to picture to herself the means of enlarging the opening, was using as 'signifier' her own mouth, with the movements of which she was familiar tactually and kinesthetically as well as by analogy with the visual image of the mouths of others. It is possible that there may also have been an element of 'causality through imitation,' L. perhaps still trying, in spite of her age, to act on the box through her miming. But the essential thing for her, as the context of the behaviour clearly showed, was to grasp the situation, and to picture it to herself actively in order to do so." ([49, p. 65; see also 48]

We included this long quote here for it illustrates that Piaget was fully aware of the key role played by analogy and its various manifestations in cognitive development as early as sixteen months[2]. Accordingly, in the framework that we propose here, cognitive development ***is*** a series of small creative leaps where cognitive agent internalizes its interaction with the environment. Using the standard language of cognitive science or artificial intelligence, we can see these internal constructs as agent's representations of the environment, or, to be more accurate: representation of agent's embededness in that particular environment. Initially, these representations are entirely expressed in terms of the innate Piagetian sensory-motor schemas. That is, we can look at the innate schemas as the **source domain** for a metaphorical description of the agent environment (the unknown **target domain**). Through the processes of assimilation (current metaphors can explain new experiences) and accommodation (re-conceptualization of the source domain is needed in view of new experiences), as well as spontaneous reorganization of the internal schema space (for example, by finding similarities and connections among distant subspaces) cognitive agents change themselves and their environments (physical, social, linguistic). (See also [25]). They need to master motor skills, language, social conventions and norms etc. This maturation process is comprised of many creative acts, driven by our genetic heritage as well as the micro and macro social context. The growth continues throughout the lifetime of the individual and, in some cases, particularly creative individuals may question some of the norms and conventions of their culture, and may impact significantly some particular established domain (arts, sciences, religion) or even create entirely new domains. The point here is that creativity is understood as a continuum and not as a binary ('yes' or 'no') attribute. It is also the driving force behind our cognitive development and it relies on more basic cognitive processes described above. An initial outline of this approach can be found in [57] and its partial implementation in context of mobile robot learning can be found in [51].

If what we suggest is plausible, the dominant approaches in artificial creativity and computational modeling of analogical (metaphorical) reasoning will have to be re-evaluated and probably fundamentally changed.

6 Similarity and Analogy in Creativity

In cognitive development, it has been noted that younger children tend to focus on surface-level similarities, and only later they take into account relational and structural similarities. For example, Namy and Gentner [42] remark: "Children up to five years go for perceptually similar objects. Clearly, then, a large number of studies have converged to demonstrate that perceptual properties such as shape loom large in children's responses on categorization tasks. This evidence suggests that children rely on shape or other salient perceptual features—perhaps even to an extent that seems detrimental to their acquisition of conceptually coherent object categories." (p. 6)

Apart from Piaget's theory of cognitive development which we mention above, other theories can be re-casted too as series of small P-creative leaps during which conventional conceptualization arise. Karmiloff-Smith [32, 33] proposes the *representational redescription model* which comprises of an endogenously driven "process by which implicit information *in* the mind subsequently becomes explicit knowledge *to* the mind" [33 p. 18]. The process can be likened to Piaget's stages from early sensorimotor schemas

[2] These differences (Goswami vs Piaget) may stem from Piaget's idiosyncratic approach to research: he was not trying to study particular modules or faculties (such as reasoning by analogy, for example) but as an acute observer he was trying to offer the best explanation that may account for certain types of behavior at certain age groups. We would like to offer the following speculation: if we were to ask Piaget about analogical reasoning and how/when it develops he might say that there is no particular age when children begin to reason by analogy. What happens is a gradual progression that starts from objects being understood only in terms of the sensorimotor schemas in which it is involved; an object 'is for something' and there is no independent representation of them. Their properties remain contextual (and not fixed) and hence reasoning about relations among objects will neither be stable nor be consistent, especially at an early age of 4 or 5. Piaget thus, we might assume, hesitated to call this reasoning by analogy, reserving the term for his formal-operations stage when abstract object and relation representations are fully developed and available for conscious manipulation. Later in the paper we offer an alternative framework for interpretation of Piaget's theory by adopting a broader construal of what is meant by 'analogy'. Also see [7].

(which too are endogenously driven to be executed) to the final stage of formal operations where the subject deliberately/consciously manipulates different abstract schemas. Karmiloff-Smith's developing agent goes through four phases where the first one (I) is characterized by **I**mplicit/behavioral knowledge/skill representations applied to certain task. These are rather detailed and task specific, and are not available for conscious manipulation by the subject. The next three phases E1, E2, E3 represent the emergence of more and more explicit, abstract, and finally (E3) verbalizable representations. These representations lose many specific details compared to the first (**I)** phase representations, but become more flexible/reusable, and declarative. As such, they also become members of a huge library of source analogies which can be applied to different problems. (See [26] for a similar approach to modeling creativity in legal reasoning.)

Barsalou and Prinz's [3] theory of mundane creativity in perceptual symbol systems [4] also comes close to the picture we want to paint here. Their theory of cognitive development focuses on the formation of perceptual symbols, which originate from the perceptual input (across all modalities) during agent's sensorimotor interaction with the environment. By the processes of selective attention, the subject focuses on some aspects of the entire perceptual input, filtering out alternative potential aspects to a large extent. These perceptual aspects are transferred into the long-term memory and in essence can be seen as concepts that can be recalled by similar perceptual input. In the language that we have adopted here, this would correspond to the emergence of conventional conceptualization. A creative insight then would happen when a subject uses a non-conventional perceptual symbol or symbols to perceive the given object/situation/scene.

In creativity research, it has been widely recognized that similarities play a key role in the generation of new ideas [36, 54, 62]. Although surface similarities are often found to influence memory access and recall [2], most of the research has focused on semantic aspects of the similarity, like structural alignment, for these are considered to be more helpful in problem solving and learning. In fact, surface similarities are often thought to be distracting [14]. A number of other creativity researchers, however, point out that focusing on structural similarities reinforce conventional way of viewing a given situation, and the crux of creativity lies in breaking the conventional structure and conceptualize the situation in a new way [6, 8, 17, 53, 56]. In this process, surface similarities can act as cues to connect two (conventionally) unrelated objects in a new way.

Some of our recent empirical studies further support this view. In one set of studies ([30, 45, 46], we have found that low-level perceptual similarities — that is, similarities with respect to texture, shape, color etc. determined algorithmically — facilitate creation of conceptual features and conceptual similarities. In another study [31], we focused on the creative process involved in connecting two pictures by painting another picture in the middle. This technique was involved in four *Infinite Landscape* workshops conducted by a visual artist at Art Museums in Japan and Europe 2007-11. Based on the artist's verbal recollection of the ideas that occurred to him as he drew each of the connecting pictures, we identified the micro-processes and cognitive mechanisms underlying these ideas, we found that *surface features, contrast,* and *meaning deconstruction* play major roles in the generation of new ideas.

What can we conclude from the above? First, traditional models of analogical reasoning which prefer relation over attribute mapping may be useful when we have to solve a novel problem in a domain with high structural similarities to some familiar domain. The solutions that may result from this process will rarely be deemed creative and will reinforce traditional conceptualizations of both domains. On the other hand, we may have no or little knowledge of the deep structure of the target domain (as, for example, in the early stages of the cognitive development). In these cases, perceptual similarities may lead to novel conceptualizations (of both source and target domains) and highly creative solutions or products.

7 Conclusions

We have presented a view here where analogy represents a core process in human cognitive development. We have argued that creativity in human agents represents a continuum: from everyday/mundane/P-creativity to the big-C creativity. Accepting the view that analogy is crucial for creativity, we attempted to make the case that superficial (attribute) similarities may actually lead to more original solutions or products. Structural analogies only reinforce the conventional conceptualization, which may be a hindrance in case the problem at hand requires information that is not normally a part of this conceptualization. We have re-casted Piaget's theory of cognitive development by describing assimilation and accommodation as progressive reasoning by analogy starting from early analogizing in terms of sensory motor schemas, to analogies in mature cognitive agents who have developed object representations. Within this framework for creativity, we gave a critical overview of today's artificial creativity models, and provided some empirical support for the claim that surface-level or perceptual similarities may play a more central role in creativity than has been supposed so far.

Acknowledgements

G. Stojanov would like to thank Sandra Bruno for the stimulating discussion about Piaget's views on analogy.

REFERENCES

[1] P.J. Aubusson, A.G. Harrison, S.M. Rithcie, "Metaphor and Analogy: Serious Thought in Science Education", in P.J. Aubusson, A.G. Harrison, S.M. Rithcie (eds.) *Metaphor and Analogy in Science Education*, Springer, 2006.

[2] J. A. Barnden andK.J. Holyoak, K.J. (Eds.), *Analogy, Metaphor, and Reminding,* Intellect Books. 1994.

[3] L.W. Barsalou, & J.J. Prinz, "Mundane creativity in perceptual symbol systems," In T.B. Ward, S.M. Smith, & J. Vaid (Eds.), *Creative thought: An investigation of conceptual structures and processes* (pp. 267-307). Washington, DC: American Psychological Association. 1997.

[4] L.W. Barsalou, "Grounding symbolic operations in the brain's modal systems," In G.R. Semin & E.R. Smith (Eds.), *Embodied grounding: Social, cognitive, affective, and neuroscientific approaches* (pp. 9-42). New York: Cambridge University Press. 2008.

[5] A. G. Baydin, R. L. De Mantaras and S. Ontanon, "Automated Generation of Cross-Domain Analogies via Evolutionary Computation", ", in the proceedings of the 4th International Conference on Computational Creativity (ICCC), UCD, Dublin, Ireland, 2012.

[6] N. Bonnardel,, “Towards Understanding and Supporting Creativity in Design: Analogies in a Constrained Cognitive Environment,” *Knowledge-Based Systems 13*, 505–513. 2000.
[7] S. Bruno, "Raisonnement par analogie et conceptualisation" Thèse de doctorat, Université Paris 5 - René Descartes. 2000.
[8] de Bono, E., *New Think: The Use of Lateral Thinking in the Generation of New Ideas.* Basic Books, New York. 1975.
[9] D.J. Chalmers, R.M. French, & D.R. Hofstadter, “High-level perception, representation, and analogy: A critique of artificial intelligence methodology,” *Journal of Experimental & Theoretical Artificial Intelligence*, 4(3), 185–211. 1992.
[10] J. Davies and A. K. Goel, “Visuospatial Re-Representation in Analogical Reasoning”, *The Open Artificial Intelligence Journal*, 2, 11--20. 2008.
[11] D. P. O'Donoghue, M. T. Keane, “A Creative Analogy Machine: Results and Challenges”, in the proceedings of the 4th International Conference on Computational Creativity (ICCC), UCD, Dublin, Ireland, 2012.
[12] K. Dunbar, “How scientists think: On-line creativity and conceptual change in science,” In T.B. Ward, S.M. Smith and J. Vaid (Eds.), *Creative thought: An investigation of conceptual structures and processes.* Washington, DC: American Psychological Association. 1997.
[13] B. Falkenhainer, K.D. Forbus, D. Gentner, “The Structure-Mapping Engine: Algorithm and Examples,” *Artificial Intelligence* 41: 1–63. 1989.
[14] Faries, J.M., & Schlossberg, K.R., “The effect of similarity on memory for prior problems,” *Proceedings of the 16th annual conference of the Cognitive Science Society,* 278 – 282. 1994.
[15] D. Gentner, “Structure-mapping: A theoretical framework for analogy”, *Cognitive Science* 7:155-170. 1983.
[16] D. Gentner, S. Brem, R. W. Ferguson, A.B. Markman, B.B. Levidow, P. Wolff, and K. Forbus, "Analogical reasoning and conceptual change: A case study of Johannes Kepler", The Journal of the Learning Sciences 6(1):3-40. 1997.
[17] W.J.J. Gordon, *Synectics: The Development of Creative Capacity.* Harper & Row, New York. 1961.
[18] A. K. Goel, “Design, Analogy, and Creativity”, IEEE Expert 12(3), 62-70. 1997.
[19] U. Goswami, “Analogical reasoning in children” in J. Campione, K. Metz, A. Sullivan Palincsar (eds.), *Children's Learning in Laboratory and Classroom Contexts Essays in Honor of Ann Brown,* Routledge, 2007.
[20] D. Hofstadter, *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought.* New York: Basic Books. 1995.
[21] D. Hofstadter, "Analogy as the Core of Cognition", in D. Gentner, K. Holyoak, and B. Kokinov (eds.) *The Analogical Mind: Perspectives from Cognitive Science,* Cambridge, MA: The MIT Press/Bradford Book, pp. 499–538. 2001.
[22] K. Holyoak, P. Thagard, “Analogical Mapping by Constraint Evaluation”, Cognitive Science, 13, 295-355. 1989.
[23] J. E. Hummel & K. J. Holyoak, “Distributed representations of structure: A theory of analogical access and mapping”, *Psychological Review,* 104, 427-466. 1997.
[24] J.E. Hummel & K.J. Holyoak, “A symbolic-connectionist theory of relational inference and generalization”, Psychological Review, 110, 220-264. 2003.
[25] B. Indurkhya, *Metaphor and Cognition.* Kluwer Academic Publishers, Dordrecht, The Netherlands. 1992.
[26] B. Indurkhya, On modeling creativity in legal reasoning. *Proceedings of the Sixth International Conference on AI and Law*, 180–189. ACM, New York. 1997.
[27] B. Indurkhya, “Emergent representations, interaction theory, and the cognitive force of metaphor*,” New Ideas in Psychology, 24(*2), 133–162. 2006.
[28] B. Indurkhya, “Rationality and reasoning with metaphors. *New Ideas in Psychology 25* (1), 16–36. 2007.
[29] B. Indurkhya, “On the role of metaphor in creative cognition,” *Proceedings of the International Conference on Computational Creativity: ICCC-X,* Lisbon, Portugal. 2010.
[30] B. Indurkhya, K. Kattalay, A. Ojha, P. Tandon, Experiments with a creativity-support system based on perceptual similarity. In H. Fujita and I. Zualkernan (Eds.) *New Trends in Software Methodologies, Tools and Techniques,* Amsterdam: IOS Press. 2008.
[31] B. Indurkhya, S. Ogawa, “An Empirical Study on the Mechanisms of Creativity in Visual Arts,” *Proc. of the 34th Annual Conf. of the Cog. Sci. Society: CogSci 2012,* Sapporo, Japan. 2012.
[32] A. Karmiloff-Smith, “From metaprocess to conscious access: Evidence from children’s metalinguistic and repair data,” *Cognition*, 23, 95–147. 1986.
[33] Karmiloff-Smith, A., *Beyond modularity: A developmental perspective on cognitive science.* MIT Press. 1992.
[34] M.T. Keane, “Incremental analogising: Theory and model” In K.J. Gilhooly, M.T. Keane, R. Logie & G. Erdos (Eds), *Lines of Thinking: Reflections on the Psychology of Thought*. Vol. 1. Chichester: John Wiley. 1990.
[35] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan, and A. Clare, “The Automation of Science”, *Science* 324 (5923), 85. 2009.
[36] A. Koestler, *The Act of Creation.* Hutchinsons, London. 1964.
[37] Kokinov, B., Holyoak, K. & Gentner, D. (Eds.), *New Frontiers in Analogy Research,* Sofia: New Bulgarian University Press. 2009.
[38] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow, *Scientific discovery:* Computational explorations of the creative process, Cambridge, MA: MIT Press, 1987.
[40] D. B. Lenat and J. S. Brown, "Why AM and EURISKO appear to work." *Artificial Intelligence* 23(3):269—294. 1984.
[41] M. Mitchell, *Analogy-Making as Perception*. ISBN 0-262-13289-3, 1993.
[42] L.L. Namy, D. Gentner, “Making a Silk Purse Out of Two Sow’s Ears: Young Children’s Use of Comparison in Category Learning,” *Journal of Experimental Psychology,* 13(1), 5-15. 2002.
[43] N.J. Nersessian, S. Chandrasekharan, “Hybrid Analogies in Conceptual Innovation in Science,” *Cognitive Systems Research 10* (3), 178–188. 2009.
[44] S. Ogawa, B. Indurkhya, A. Byrski, “A meme-based architecture for modeling creativity,” *Proc. of the Int. Conf. on Computational Creativity,* Dublin, Ireland. 2012.
[45] A. Ojha, B. Indurkhya, “Perceptual vs. conceptual similarities and creation of new features in visual metaphor,” in B. Kokinov, K. Holyoak & D. Gentner (Eds.) *New Frontiers in Analogy Research,* Sofia: New Bulgarian University Press. 2009.
[46] A. Ojha, B. Indurkhya, “On the role of perceptual features in metaphor,” in E. Gola & F. Ervas (eds.) *Metaphor and Communication.* To appear.
[47] T. Okada, S. Yokochi, K. Ishibashi, K. Ueda, “Analogical Modification in the Creation of Contemporary Art,” *Cognitive Systems Research 10* (3), 189–203. 2009.
[48] J. Piaget, *The Origins of Intelligence in the Child* (M. Cook, Trans.) Penguine, New York. (Original work published 1936). 1977.
[49] J. Piaget, *Play, Dreams and Imitation in Childhood* (C. Gattegno & F.M. Hodgson, Trans.) W.W. Norton, New York. (Original work published 1945). 1962.
[50] G. Polya. *How to Solve It,* (2nd ed.), Princeton University Press, Princeton. 1957.
[51] V. Poprcova, G. Stojanov, A. Kulakov, “Inductive Logic Programming (ILP) and Reasoning by Analogy in Context of Embodied Robot Learning,” in Proceedings of IJATS, pp. 64-73. 2010.
[52] I.A. Richards, *The philosophy of rhetoric.* Oxford University Press, Oxford. 1936.

[53] G. Rodari, *The Grammar of Fantasy* (J. Zipes, Trans.) New York: Teachers & Writers Collaborative. 1996.
[54] K. Sawyer, *Explaining Creativity.* Oxford (UK): Oxford University Press. 2006.
[55] D. A. Schön, *Displacement of Concepts.* New York: Humanities Press. 1963.
[56] O. Shapira & N. Liberman N, "An Easy Way to Increase Creativity," *Scientific American (Mind Matters),* July 21, 2009.
[57] G. Stojanov, "Embodiment as Metaphor: Metaphorizing-in The Environment" in C. Nehaniv (ed.*) Lecture Notes in Artificial Intelligence*, Vol. 1562, pp. 88-101, Springer-Verlag, 1999.
[58] G. Stojanov, "Computational Models of Creativity: Taking Early Cognitive Development Seriously" *to appear* in Indyrkhya, B. and Manjalay, J. (eds.) Cognition, Experience and Creativity, Orient Blackswan Pvt. Ltd., 2012.
[59] R. Sun, "A microfeature based approach towards metaphor interpretation,*" Proceedings of the 14th IJCAI,* 424–429. Morgan Kaufman, San Francisco. 1995.
[60] Veale, T. "Re-Representation and Creative Analogy: A Lexico-Semantic Perspective," New Generation Computing, *24* (3), 223-240. 2006.
[61] T. Veale, Y. Hao, "A Fluid Knowledge Representation for Understanding and Generating Creative Metaphors," *Proceedings of COLING 2008*, 1–5. Manchester. 2008.
[62] T. B. Ward, "Analogies," In M.A. Runco & S.R. Pritzker (Eds.) Encyclopedia of Creativity, (2nd ed.), New York: Academic Press. 2011.
[63] P. W. Yaner, A. K. Goel, "Analogical recognition of shape and structure in design drawings", AI EDAM 22(2): 117-128. 2008.
[64] J. Zhu, S. Ontanon, "Towards Analogy-Based Story Generation ", in the Proceedings of ICCC-X, Lisbon, 2010.

Analogical proportions in a lattice of sets of alignments built on the common subwords in a finite language

Laurent Miclet[1] and **Nelly Barbot**[2] and **Baptiste Jeudy**[3]

Abstract. We define the locally maximal subwords and locally minimal superwords common to a finite set of words. We also define the corresponding sets of alignments. We give a partial order relation between such sets of alignments, as well as two operations between them. We show that the constructed family of sets of alignments has the lattice structure. The study of analogical proportion in lattices gives hints to use this structure as a machine learning basis, aiming at inducing a generalization of the set of words.

Keywords: Locally maximal subwords, alignments, algebraic structure of sets of alignments on a set of words (lattice), analogical proportion.

1 Introduction

Much has been done on finding maximal subwords and minimal superwords to a set of words, when the order relation is based on the length of words. We are interested in this paper in the same problem, but for the finer order relation based on the definition of a subword. Is there a manner to characterize the set of maximal subwords and that of minimal superwords, given a finite set U of words, according to this relation ? More than that, is there an algebraic relation between all these sets of subwords and superwords of U ? An answer to these questions would allow to give a precise definition to what the words of U share, and how this common core is organised.

The firsts parts of this paper gives a partial answer to these points. We define in section 2 a particular case of the notion of alignment, which will be useful for our construction. Actually, in section 3, we define two operations and an order relation on sets of alignments that leads to the construction of a lattice.

We are also interested in how this structure could be analysed in terms of analogical proportions, which could be used in machine learning. Since we start from a finite set of words, the convenient machine learning framework seems to be grammatical inference (from a positive set of positive samples, in our case). It seems that the lattice structure is particularly adapted to learning by analogy, since some natural analogical proportions can be observed in such a structure. We give in section 4 some hints on these points.

[1] IRISA-Dyliss, Rennes France. miclet@enssat.fr
[2] IRISA-Cordial, Lannion France. barbot@enssat.fr
[3] Université de Saint-Étienne, Laboratoire Hubert Curien. baptiste point jeudy at univ-st-etienne point fr

2 Maximal subword, minimal superword, alignment

2.1 Basics

Let Σ be an alphabet, *i.e.* a finite set of letters. A *word* u is a sequence $u_1 \dots u_n$ of letters in Σ. The length of u, denoted $|u|$ is n. The empty word, of null length, is ϵ. A *language* is a set of words. A *subword* of a word u is a word obtained by deleting letters of u at some (non necessarily adjacent) positions[4] in u. We denote $u \bullet v$ the *shuffle* of the two words u and v.

In $\Sigma^\star$, the set of all words on Σ, we use the order relation $\leq$ defined by: ($u \leq v \Leftrightarrow u$ is a subword of v). When u is a subword of v, v is called a *superword*[5] of u. For example: $abc \leq aabbcd$.

A word w is a *common subword* to u and v when $w \leq u$ and $w \leq v$. The word w is a *maximal* common subword to u and v if there does not exist any other common subword x to u and v such that $w \leq x$. For example, ab and c are maximal common subwords to $u = cadba$ and $v = fagbhc$, while a is a non maximal common subword. Defining a common maximal subword to a finite set of words is a straightforward extension.

A maximal common subword to two words and to a non empty finite set of words is defined in an analog way.

In a partially ordered set S, an antichain is a subset of S composed of pairwise incomparable elements. Any subset T of S can be reduced to a maximal antichain by removing from T every element of T lesser than another element of T.

2.2 Alignments

2.2.1 Definition

Definition 1 *An* alignment *is a finite set of pairs* (w, l) *where* w *is a word and* l *a set of indices between 1 and* $|w|$. *The set* l *defines a subword of* w *denoted* $w[l]$. *Moreover, an alignment* a *must satisfy the following properties for all* $(w, l) \in \mathsf{a}$ *and* $(w', l') \in \mathsf{a}$*:*

1. $w[l] = w'[l']$
2. $(w = w') \Rightarrow (l = l')$
3. $(w \leq w') \Rightarrow (w = w')$

The set of words on which the alignment is defined is called the *support* and is denoted $word(\mathsf{a}) = \{w \mid \exists l \subset \mathbb{N} \text{ with } (w, l) \in \mathsf{a}\}$.

[4] Other terms for *subword* are *subsequence* and *partial word*. A *factor*, or *substring* is a subword of u built by contiguous letters of u.

[5] A *superword* of u, also called a *supersequence* must not be confused with a *superstring* of u, in which the letters of u are contiguous. In other words, u is a factor (a substring) of any superstring of u. See [Gus97], pages 4, 309 and 426.

According to our definition[6], the support is an antichain of words for $\leq$.

The set of indices l will be called the *position* of the indexed subword of $w[l]$.

In the following, an alignment will be represented by a set of words in which some letters are boxed. For each element (w, l) of the alignment, the boxed letters represent the subword $w[l]$ (also called the boxed subword of the alignment).

For legibility, the n words can be displayed in such a manner that the corresponding letters of w in the n words are in the same column. Some blanks can be added freely to help the reading. For example:

$$\mathtt{a} = \begin{pmatrix} & \boxed{a} & & \boxed{c} & b & d & e & g & \\ & \boxed{a} & & \boxed{c} & & & e & & h \\ g & \boxed{a} & h & \boxed{c} & & d & & & \end{pmatrix}$$

denotes the alignment

$\mathtt{a} = \{(acbdeg, \{1, 2\}), (aceh, \{1, 2\}), (gahcd, \{2, 4\})\}$.

We can write also without ambiguity:

$$\mathtt{a} = (\boxed{a}\boxed{c}bdeg, \boxed{a}\boxed{c}eh, g\boxed{a}h\boxed{c}d).$$

2.2.2 *Locally maximal alignments and locally maximal subwords*

Generally speaking, two alignments on the same support $W = \{w_1, \ldots, w_n\}$ with the same boxed subword r can be different (having different set of indexes). We could define maximal alignments as those whose boxed letters are maximal subword of W.

However, all interesting alignments would not be maximal with this definition. Consider for example the two words $w_1 = abcd$ and $w_2 = dabcab$. The complete set of common subwords is $\{\epsilon, a, b, c, ab, ac, bc, abc, d\}$ and their set of maximal common subwords is $\{abc, d\}$.

But these two subwords are not sufficient to define the totality of the interesting alignments. Actually the alignment $(\boxed{a}\boxed{b}cd, dabc\boxed{a}\boxed{b})$ is somehow "maximal" since it is not comparable to the only alignment with the boxed subword abc, namely $(\boxed{a}\boxed{b}\boxed{c}d, d\boxed{a}\boxed{b}\boxed{c}ab)$.

This leads to define the following notion of *locally maximal alignment* and of *locally maximal subword*.

Definition 2 *An alignment* $\mathtt{a} = \{(w_1, l_1), \ldots, (w_n, l_n)\}$ *is* locally maximal *if there is no other alignment* $\mathtt{b} = \{(w_1, l'_1), \ldots, (w_n, l'_n)\}$ *on the same support such that for all i,* $l_i \subset l'_i$.

Notice that the empty alignment $\emptyset$ is locally maximal.

Definition 3 *The set of boxed subwords associated to all locally maximal alignments between a finite set of words* $W = \{w_1, \ldots, w_n\}$ *is called the set of* locally maximal subwords *to W and is denoted* $\sqcap(W)$.

For some $r \in \sqcap(W)$*, the set of locally maximal alignements associated to r is denoted* $A_r(W)$.

We also define: $A(W) = \bigcup_{r \in \sqcap(W)} A_r(W)$.

For example, let us consider $W = \{ababc, cabd\}$, its sets of locally maximal alignments are given by

$$\begin{aligned} A_{ab}(W) = \{ \quad & (\boxed{a}\boxed{b}abc, c\boxed{a}\boxed{b}d), \\ & (\boxed{a}ba\boxed{b}c, c\boxed{a}\boxed{b}d), \\ & (ab\boxed{a}\boxed{b}c, c\boxed{a}\boxed{b}d) \quad \} \end{aligned}$$

$A_c(W) = \{(abab\boxed{c}, \boxed{c}abd)\}$.

$A(W) = A_{ab}(W) \bigcup A_c(W)$.

Then, the set of locally maximal subwords of W is

$$\sqcap(W) = \{ab, c\}$$

2.3 Language associated with an alignment

Definition 4 *Let* $w = w_1 \cdots w_p$ *be a word, locally maximal subword of two words u and v at only one position (*i.e. $|A_w(\{u, v\})| = 1$*). Then there exists an unique set of factors of u, denoted* $(u^1, \ldots, u^{p+1})$*, and an unique set of factors of v, denoted* $(v^1, \ldots, v^{p+1})$*, such that* $u = u^1 w_1 \ldots u^p w_p u^{p+1}$ *and* $v = v^1 w_1 \ldots v^p w_p v^{p+1}$*. We define* $L(A_w(\{u, v\}))$ *as the following finite language:*

$$L(A_w(\{u, v\})) = (u^1 \bullet v^1) w_1 (u^2 \bullet v^2), \ldots, (u^p \bullet v^p) w_p (u^{p+1} \bullet v^{p+1})$$

The construction of $L(A_w(\{u, v\}))$ is shown in Figure 1, with straightforward graphic conventions.

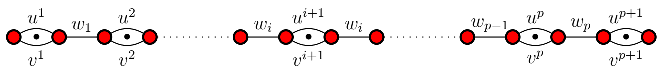


Figure 1. The construction of $L(A_w(u, v))$ when $|A_w(\{u\}, \{v\})| = 1$.

If $|A_w(\{u, v\})| > 1$, $L(A_w(\{u, v\})$ is defined as the union of all languages associated with all different positions of w as locally maximal subword of u and v. Finally, $L(A(\{u, v\}))$ is defined as the union of the languages $L(A_w(\{u, v\}))$, for all w locally maximal subwords of u and v.

Proposition 1 *Let w be a locally maximal subword common to two words u and v and* $L(A_w(\{u, v\}))$ *constructed as above. We have:*

1. *All words in* $L(A_w(\{u, v\}))$ *are (non necessarily minimal) common superwords of u and v.*
2. *For any word* $W \in L(A_w(\{u, v\}))$*, we have*[7] $|W| + |w| = |u| + |v|$.

Proof. We firstly give an axample to show that a word of $L(A_w(\{u, v\}))$ can be a non-minimal superword of u and v.

We take the two words $u = abcabb$ and $v = aabbc$. The associated alignment $(\boxed{a}\boxed{b}ca\boxed{b}b, \boxed{a}a\boxed{b}\boxed{b}c)$ is locally maximal. The language $L(A_{abb}(\{u, v\})$ contains the language $aabcab(b \cdot c)$ and, in particular, the word $w = abcabbc$. The word $w' = abcabbc$ is another superword of u and v, and $w' \leq w$. Thus, w is not an locally maximal superword of u and v.

Then we demonstrate the proposition.

Let us consider $W \in L(A_w(\{u, v\}))$. By definition of $L(A_w(\{u, v\}))$, there exists $(u^1, \ldots, u^{p+1})$ and $(v^1, \ldots, v^{p+1})$, respectively sets of factors of u and v, such that the word W can be written as $W = x^1 w_1 \ldots x^p w_p x^{p+1}$ where, for every $i \in \{1, \ldots p+1\}$, $x^i \in (u^i \bullet v^i)$.

[6] An alignment (regardless of the third point of our definition), is called a *trace* by Wagner and Fisher [WF74] for two words and a *threading scheme* in Maier [Mai78].

[7] A consequence of this assertion is : let $LCS(u, v)$ be a longest common subword to u and v and $SCS(u, v)$ be a shortest common superword to u and v. Then we have: $|LCS(u, v)| + |SCS(u, v)| = |u| + |v|$.

1. Therefore, for every $i \in \{1, \ldots p+1\}$, $x^i \geq u^i$ and $x^i \geq v^i$. We then have $W \geq u^1 w_1 \ldots u^p w_p u^{p+1} = u$ and $W \geq v^1 w_1 \ldots v^p w_p v^{p+1} = v$.
2.

$$\begin{aligned} |W| &= |x^1| + |w_1| + \ldots + |x^p| + |w_p| + |x^{p+1}| \\ &= |w| + \sum_{i=1}^{p+1} |x^i| = |w| + \sum_{i=1}^{p+1} \left(|u^i| + |v^i|\right) \\ &= |w| + (|u| - |w|) + (|v| - |w|) = |u| + |v| - |w| . \end{aligned}$$

□

2.4 Constructive algorithms

We have devised an algorithm producing a finite automaton $\mathcal{A}_{\sqcap(\{u,v\})}$ which exactly recognizes the language $\sqcap(\{u,v\})$, the set of locally maximal subwords common to two words u and v, due to lack of space, we do not describe it here. It is based on the transformation of an 2-d array displaying which letters are common to two words into a finite automaton recognizing $\sqcap(u,v)$ (see an example on figure 2(a)).

Starting from $\mathcal{A}_{\sqcap(\{u,v\})}$, it is then simple to produce a finite automaton that we call $\mathcal{A}_{\sqcup(\{u,v\})}$ which exactly recognizes the language $L(A(\{u,v\}))$ (also denoted $\sqcup(\{u,v\})$). We display an example at figure 2(b).

3 Order relation and operations between alignments

In this section, we are interested in a particular family of alignments, since we want to describe what have in common the subwords and superwords of a finite set U of sentences. We will consider alignments on U, *i.e.* alignments with a support *subset of* U. Moreover, we will assume that U is an antichain according to the order relation $\leq$.

3.1 Order relation

Definition 5 (Order on alignments on U) *Given two alignments on U $\mathsf{a} = \{(w_1, l_1), \ldots, (w_n, l_n)\}$ and $\mathsf{b} = \{(w'_1, l'_1), \ldots, (w'_m, l'_m)\}$, we write $\mathsf{a} \sqsubseteq \mathsf{b}$ if for all $i \in (1,n)$, it exists $j \in (1,m)$ such that*

1. $w_i = w'_j$
2. $l'_j \subseteq l_i$

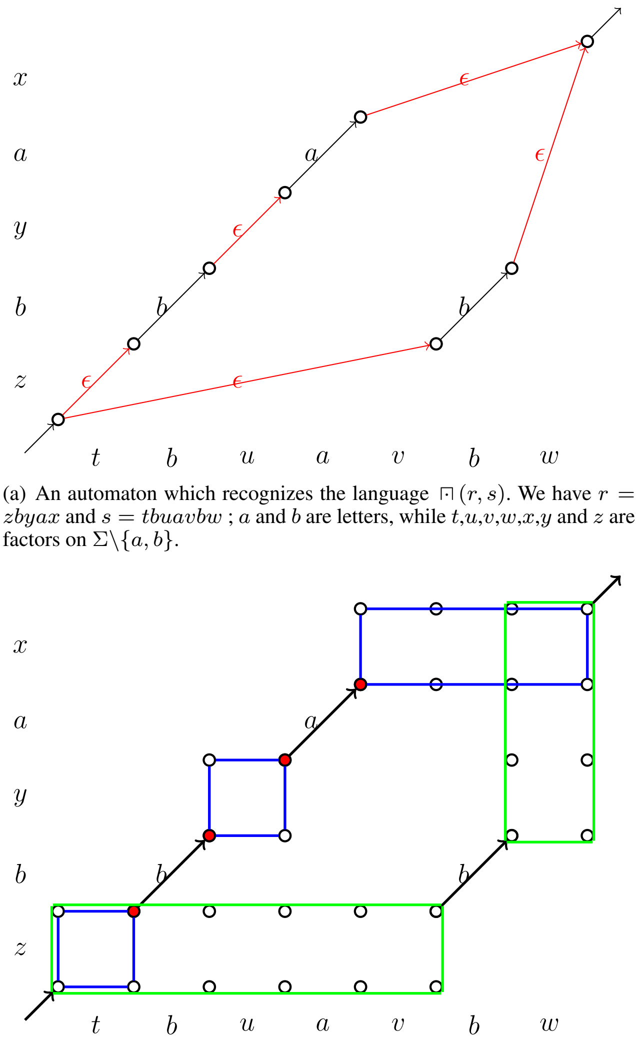
Therefore, if $\mathsf{a} \sqsubseteq \mathsf{b}$, *then* $word(\mathsf{a}) \subseteq word(\mathsf{b})$.

It is easy to check that $\sqsubseteq$ is a partial order relation on the set of alignments and that the empty alignment $\emptyset$ is smaller than every other alignment.

Definition 6 (Homogeneous sets of alignments) *A set of alignments is homogeneous if it is non empty and all its elements have the same support. The family of homogeneous sets of locally maximal alignments is denoted $\mathcal{A}_H$.*

In order to link this definition with definition 3, we can notice that, for any subset W of U, $A(W) \in \mathcal{A}_H$.

Definition 7 (Order on homogeneous sets of alignments on U) *Let A and B be two homogeneous sets of alignements. We have $A \sqsubseteq B$ if for all $\mathsf{b} \in B$, there is $\mathsf{a} \in A$ such that $\mathsf{a} \sqsubseteq \mathsf{b}$.*



(a) An automaton which recognizes the language $\sqcap(r,s)$. We have $r = zbyax$ and $s = tbuavbw$; a and b are letters, while t,u,v,w,x,y and z are factors on $\Sigma \backslash \{a,b\}$.

(b) An automaton which recognizes $\sqcup(r,s) = (z \bullet t)b(u \bullet y)a(vbw \bullet x) \cup (tbuav \bullet z)b(w \bullet yax)$. A rectangle holds for the shuffle of the factors on its sides

Figure 2. Constructing the languages $\sqcap(r,s)$ and $\sqcup(r,s)$

Proposition 2 *$\sqsubseteq$ is a partial order on $\mathcal{A}_H$ and the smallest element is $\{\emptyset\}$.*

Proof.
Reflexivity and transitivity are immediate. In order to check the antisymmetry, let us consider two homogeneous sets of locally maximal alignments, denoted A and B, such that: $A \sqsubseteq B$ and $B \sqsubseteq A$. Since A and B are homogeneous, all alignments in A have the same support, denoted $word(A)$, and the same holds for B, with the support denoted $word(B)$. From the definition of $\sqsubseteq$, we easily check that $word(A) = word(B)$. Let us consider $\mathsf{b}_1 = \{(w_1, l'_1), \ldots, (w_n, l'_n)\} \in B$: since $A \sqsubseteq B$ and $B \sqsubseteq A$, it exists $\mathsf{a} \in A$ and $\mathsf{b}_2 \in B$ such that $\mathsf{a} \sqsubseteq \mathsf{b}_1$ and $\mathsf{b}_2 \sqsubseteq \mathsf{a}$. By transitivity, we have $\mathsf{b}_2 \sqsubseteq \mathsf{b}_1$. At last, b_1 and b_2 having the same support and being locally maximal, it implies that $\mathsf{b}_1 = \mathsf{b}_2$ and then $\mathsf{a} \in B$. Hence, $A \subseteq B$. Similarly, we can check that $B \subseteq A$.

□

3.2 Definition and properties of $\uplus$

Definition 8 *Let* $\mathtt{a} \in A_r(\{u_1, \cdots, u_n\})$ *and* $\mathtt{b} \in A_s(\{v_1, \cdots, v_m\})$, *where* $\mathtt{a} = \{(u_1, l_1), \ldots, (u_n, l_n)\}$ *and* $\mathtt{b} = \{(v_1, l'_1), \ldots, (v_m, l'_m)\}$. *Firstly, we construct* $\mathtt{a} + \mathtt{b}$, *the finite set of alignments* $\mathtt{c} = \{(w_1, L_1), \ldots, (w_p, L_p)\}$ *such that*

1. $\{w_1, \ldots, w_p\} = word(\mathtt{a}) \cup word(\mathtt{b})$
2. *for all* (i, k), *if* $(w_k = u_i)$ *then* $(L_k \subseteq l_i)$
3. *for all* (j, k), *if* $(w_k = v_j)$ *then* $(L_k \subseteq l'_j)$

Secondly, we denote $\mathtt{a} \uplus \mathtt{b}$ *the set of minimal elements of* $\mathtt{a} + \mathtt{b}$ *according to* $\sqsubseteq$.

As consequence, if $\sqcap(\{r, s\}) \neq \emptyset$, then the boxed word in $\mathtt{c} \in \mathtt{a}+\mathtt{b}$ is a subword of r and s, else, no letter is boxed in $\mathtt{c}$. In addition, if $\mathtt{a}$ and $\mathtt{b}$ contains an identical word $u_i = v_j$ such that $l_i \cap l'_j = \emptyset$, no letter is then boxed in $\mathtt{c}$.

The operation $\uplus$ is extended to homogeneous sets of alignments by the following definition.

Definition 9 *Let* A *and* B *be two homogeneous sets of alignments. We define* $A \uplus B$ *as the set of the minimal elements of* $A+B$ *according to* $\sqsubseteq$ *where*

$$A + B = \bigcup_{\substack{\mathtt{b} \in B \\ \mathtt{a} \in A}} (\mathtt{a} + \mathtt{b})$$

Proposition 3 *The operation* $\uplus$ *is internal to* $\mathcal{A}_H$, *commutative and idempotent.*

Proof. Let us consider $A \in \mathcal{A}_H$ and $B \in \mathcal{A}_H$.

1. All the alignments in $A \uplus B$ are locally maximal by definition and have the same support, namely $word(A) \cup word(B)$.
2. The commutativity is straightforward.
3. Let $\mathtt{a}$ be an element of A, it is immediate that $\mathtt{a} \in (\mathtt{a} + \mathtt{a}) \subseteq A + A$. Moreover, since $A \in \mathcal{A}_H$, $\mathtt{a}$ is a locally maximal alignment, and so $\mathtt{a} \in A \uplus A$. Consequently, $A \subseteq A \uplus A$. Reciprocally, let $\mathtt{c}$ be an element of $A \uplus A$. Then it exists a couple $(\mathtt{a}, \mathtt{b}) \in A^2$ such that $\mathtt{c} \in \mathtt{a} + \mathtt{b}$. Since $A \in \mathcal{A}_H$ and $word(\mathtt{c}) = word(\mathtt{a}) \cup word(\mathtt{a})$, $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$ have the same support. Moreover, from definitions 5 and 8, $\mathtt{a} \sqsubseteq \mathtt{c}$ and $\mathtt{b} \sqsubseteq \mathtt{c}$. $\mathtt{c}$ being a minimal element of $A + A$ according to $\sqsubseteq$, and a and b belonging to $A + A$, it turns out that $\mathtt{a} = \mathtt{b} = \mathtt{c}$. At last, $\mathtt{c} \in A$. Hence $A \uplus A \subseteq A$. $\sqsubseteq$ is then idempotent on $\mathcal{A}_H$.

□

3.3 Construction of $\capdot$

Definition 10 *Let* $\mathtt{a} \in A_r(\{u_1, \cdots, u_n\})$ *and* $\mathtt{b} \in A_s(\{v_1, \cdots, v_m\})$ *where* $\mathtt{a} = \{(u_1, l_1), \ldots, (u_n, l_n)\}$ *and* $\mathtt{b} = \{(v_1, l'_1), \ldots, (v_m, l'_m)\}$. *We construct* $\mathtt{a} \capdot \mathtt{b}$, *the finite set of alignments* $\mathtt{c} = \{(w_1, L_1), \ldots, (w_p, L_p)\}$ *such that*

1. $\{w_1, \ldots, w_p\} = word(\mathtt{a}) \cap word(\mathtt{b})$
2. *Either, for all* (i, k) *such that* $w_k = u_i$ *we have* $l_i \subseteq L_k$, *or for all* (j, k) *such that* $w_k = v_j$ *we have* $l'_j \subseteq L_k$.
3. $\mathtt{c}$ *is a locally maximal alignment.*

An alignment in $\mathtt{a} \capdot \mathtt{b}$ is thus based either on a restriction of $\mathtt{a}$ to the support $word(\mathtt{a}) \cap word(\mathtt{b})$ or on a restriction of $\mathtt{b}$ to the same support. For instance, if $\mathtt{a} = \{(\boxed{a}cd, ab\boxed{a}c, \boxed{a}ba)\}$ and $\mathtt{b} = \{(a\boxed{c}d, aba\boxed{c}, \boxed{c}a)\}$, then $\mathtt{a} \capdot \mathtt{b} = \{(\boxed{a}\boxed{c}d, \boxed{a}ba\boxed{c}), (\boxed{a}\boxed{c}d, ab\boxed{a}\boxed{c})\}$.

Definition 11

$$A \capdot B = \bigcup_{\substack{\mathtt{b} \in B \\ \mathtt{a} \in A}} (\mathtt{a} \capdot \mathtt{b})$$

Proposition 4 *The operation* $\capdot$ *is internal to* $\mathcal{A}_H$, *commutative and idempotent.*

Proof. The commutativity is straightforward (definition 10 is symetric wrt $\mathtt{a}$ and $\mathtt{b}$). For idempotence, we use the fact (direct consequence of the definition) that if $\mathtt{a}$ and $\mathtt{b}$ are locally maximal alignements on the same support, then $\mathtt{a} \capdot \mathtt{b} = \{\mathtt{a}, \mathtt{b}\}$. Let us consider $A \in \mathcal{A}_H$: if $\mathtt{a} \in A$ then $\mathtt{a} \in (\mathtt{a} \capdot \mathtt{a}) \subseteq (A \capdot A)$ and therefore $A \subseteq A \capdot A$. If $\mathtt{c} \in A \capdot A$, then there exists $(\mathtt{a}, \mathtt{b}) \in A^2$ such that $\mathtt{c} \in \mathtt{a} \capdot \mathtt{b}$. Since $\mathtt{a}$ and $\mathtt{b}$ have the same support, either $\mathtt{c} = \mathtt{a}$ or $\mathtt{c} = \mathtt{b}$, therefore $\mathtt{c} \in A$ and $A \capdot A \subseteq A$. □

3.4 Structure of homogeneous sets of alignments on U

We define $\sup_{\sqsubseteq}(A, B)$ as the minimal set of alignments larger than A and B (if it exists) according to $\sqsubseteq$. Similarly, $\inf_{\sqsubseteq}(A, B)$ is the maximal set of alignments smaller than A and B.

Proposition 5 *Let* A *and* B *be finite homogeneous sets of alignments. Then* $\sup_{\sqsubseteq}(A, B)$ *exists and:*

$$\sup_{\sqsubseteq}(A, B) = A \uplus B$$

Proof.

- First, we show that $A \uplus B$ is greater than A and B for $\sqsubseteq$. Let $\mathtt{c} \in A \uplus B$. By construction, there exist $\mathtt{a} \in A$ and $\mathtt{b} \in B$ such that $\mathtt{c} \in \mathtt{a} \uplus \mathtt{b} \subseteq \mathtt{a} + \mathtt{b}$. By the first item of definition 8, $word(\mathtt{a}) \subseteq word(\mathtt{c})$ and by the two other items, we can conclude that $\mathtt{a} \sqsubseteq \mathtt{c}$. Thus for every $\mathtt{c} \in C$ there is $a \in A$ such that $\mathtt{a} \sqsubseteq \mathtt{c}$. Thus $A \sqsubseteq A \uplus B$ and $B \sqsubseteq A \uplus B$.
- Let C be a set of alignments greater than A and B, and let $\mathtt{c} \in C$. There are $\mathtt{a} \in A$ and $\mathtt{b} \in B$ such that $\mathtt{a} \sqsubseteq \mathtt{c}$ and $\mathtt{b} \sqsubseteq \mathtt{c}$. We need to find $\mathtt{c}' \in A \uplus B$ such that $\mathtt{c}' \sqsubseteq \mathtt{c}$. Remove from the support of $\mathtt{c}$ all words not in the support of $\mathtt{a}$ or $\mathtt{b}$. The obtained alignment may not be locally maximal, so we add more boxed letters to make it locally maximal. The result alignment $\mathtt{c}'$ satisfies all conditions of Definition 8, thus $A \uplus B \sqsubseteq C$ and therefore $\sup_{\sqsubseteq}(A, B) = A \uplus B$. □

There is no equivalent relation between $\capdot$ and inf for all homogeneous sets of alignments, we must restrict to sets of all alignments built on a given set of words.

Definition 12 *If* U *is a finite collection of words, we define the collection of sets of alignments* $\mathcal{A}(U) = \{A(V) \mid V \subseteq U\}$.

Proposition 6 *Let* A *and* B *be sets of alignments in* $\mathcal{A}(U)$. *Then, in* $\mathcal{A}(U)$, $\inf_{\sqsubseteq}(A, B)$ *exists and:*

$$\inf_{\sqsubseteq}(A, B) = A \capdot B$$

Proof.

- First, we show that if $A = A(V)$ and $B = A(W)$ with $V \subseteq U$ and $W \subseteq U$ then $A \capdot B = A(W \cap V)$. Let $\mathtt{c} \in A \capdot B$. $\mathtt{c}$ is a locally maximal alignment on its support $word(A) \cap word(B) = W \cap V$, thus $\mathtt{c} \in A(W \cap V)$. Let $\mathtt{c} \in A(W \cap V)$. Let $\mathtt{a}$ be an alignment on W such that $\mathtt{c} \sqsubseteq a$, then $\mathtt{c}$ is obtained from $\mathtt{a} \in A$ using the definition of $A \capdot B$ and $\mathtt{c} \in A \capdot B$.

- Let $C \in \mathcal{A}(U)$ be a set of alignments smaller than A and B. We show that C is smaller than $A \Cap B$. Some alignments of C are smaller than alignments of A and others are smaller than alignments of B. Since C is homogeneous, its support $word(C)$ must be included in $word(A) \cap word(B)$ and since $C = A(T)$ for some $T \subseteq U$, then $T \subseteq V \cap W$. Therefore $A(T) \sqsubseteq A(V \cap W)$ which is exactly $C \sqsubseteq A \Cap B$.

$\square$

Proposition 7 *Let $U = \{u_1, u_2, \cdots, u_n\}$ be a finite set of words, the operations $\Cap$ and $\uplus$ are internal to $\mathcal{A}(\mathcal{U})$.*

Proof. For $\Cap$ it is a consequence of the previous definition. For $\uplus$, it is not difficult to see it from the definition of $\uplus$.

$\square$

Proposition 8 *Let $U = \{u_1, u_2, \cdots, u_n\}$ be a finite set of words, antichain for $\leq$. Then $\mathcal{U} = (\mathcal{A}(U), \uplus, \Cap)$ is a lattice. This lattice is said to be built on the finite language U.*

Proof. This is a direct consequence of the three previous propositions.

$\square$

4 Analogical proportions in the lattice $\mathcal{U}$

4.1 The axioms of analogical proportion

Definition 13 (Analogical proportion) *An analogical proportion on a set $\mathbb{E}$ is a relation in $\mathbb{E}^4$ such that, for all 4-tuples A, B, C et D in relation in this order (denoted $A : B :: C : D$):*

1. $A : B :: C : D \Leftrightarrow C : D :: A : B$
2. $A : B :: C : D \Leftrightarrow A : C :: B : D$

For every 2-tuple, one has : $A : B :: A : B$

It is easy to show that five other proportions are equivalent:

$B : A :: D : C \quad D : B :: C : A \quad D : C :: B : A$
$B : D :: A : C \quad C : A :: D : B$

These requirements are often called the *axioms* of analogical proportion (see [Lep03]).

4.2 Analogical proportions between words

A first definition using factorization. According to Yvon and Stroppa [SY05] a general definition of analogical proportion, conform to the axioms, can be given in many different cases thanks to the notion of *factorization*. We show here how it applies in $\Sigma^\star$, and we will come back later to its use in general lattices.

Definition 14 (Analogical proportions between words.) *$(x, y, z, t) \in \Sigma^\star$ are in analogical proportion, which is denoted $x : y : z : t$, if and only if there exists a positive integer n and two sets of words $(\alpha_i)_{i\in[1,n]}$ and $(\beta_i)_{i\in[1,n]} \in \Sigma^\star$ such that:*

$x = \alpha_1 \ldots \alpha_n,\ t = \beta_1 \ldots \beta_n,\ y = \alpha_1\beta_2\alpha_3 \ldots \alpha_n,\ z = \beta_1\alpha_2\beta_3 \ldots \beta_n$
ou
$x = \alpha_1 \ldots \alpha_n,\ t = \beta_1 \ldots \beta_n,\ y = \beta_1\alpha_2\beta_3 \ldots \alpha_n,\ z = \alpha_1\beta_2\alpha_3 \ldots \beta_n$

and $\forall i, \alpha_i\beta_i \neq \epsilon$.

Example. *reception* : *refection* :: *deceptive* : *defective* is an analogical proportion between sequences, with $n = 3$ and the factors : $\alpha_1 = re$, $\alpha_2 = cept$, $\alpha_3 = ion$, $\beta_1 = de$, $\beta_2 = fect$, $\beta_3 = ive$.

	β_1	α_1	β_2	α_2	β_3	α_3
x :		re		cept		ion
y :		re	fect			ion
z :	de			cept	ive	
t :	de		fect		ive	

The authors have shown that this definition is conform to the axioms.

Another definition using alignments. This second definition, with the associated algorithms, is given in [MBD08]. The axioms of analogical proportion are verified as well.

Definition 15 *Let u, v, w and x four words in $\Sigma^\star$. We assume that an analogical proportion is defined on Σ. We extend this relations to $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$, adding the proportions $a : \epsilon :: a : \epsilon$ for all $a \in \Sigma$. Then u, v, w and x are in* analogical proportion *in $\Sigma^\star$ if there exists an alignment between the four words such that every column of the alignment is an analogical proportion in Σ_ϵ.*

Example Let $\Sigma = \{a, b, c, A, B, C\}$ an alphabet with the analogical proportions $a : b :: A : B$, $a : c :: A : C$, $c : b :: C : B$. The following alignment shows that there is an analogical proportion in $\Sigma^\star$ between the four words $CaCA$, $CcbBA$,bAc and $bCbb$.

$$\begin{pmatrix} C & a & & C & A \\ C & c & b & B & A \\ b & A & & c & \\ b & C & b & b & \end{pmatrix}$$

Note that there is no boxed letter in this alignment. It can happen anyway in the case of a column such that $a : a :: a : a$.

Links between the two definitions. The second definition using alignments is shown to imply the first one (not the reverse). However, a straightforward modification of the first one lead to a complete equivalence [Has11].

4.3 Analogical proportions in a lattice

Using the factorization technique, Stroppa and Yvon [SY05] have found that a general definition of an analogical proportion can be given in a lattice. Unfortunately, his definition was uncomplete. We give here the complete one.

Definition 16 *For four elements $(x, y, z, t) \in (L, \vee, \wedge)^4$, the analogical proportion denoted $(x : y :: z : t)$ is true if and only if:*

$$x = (x \wedge y) \vee (x \wedge z) \text{ and } x = (x \vee y) \wedge (x \vee z)$$
$$y = (x \wedge y) \vee (t \wedge y) \text{ and } y = (x \vee y) \wedge (t \vee y)$$
$$z = (t \wedge z) \vee (x \wedge z) \text{ and } t = (t \vee z) \wedge (t \vee y)$$
$$t = (t \wedge z) \vee (t \wedge y) \text{ and } z = (t \vee z) \wedge (x \vee z)$$

The geometry of this definition is displayed in figure 3.

A simple example of proportion in a lattice is given by the following property:

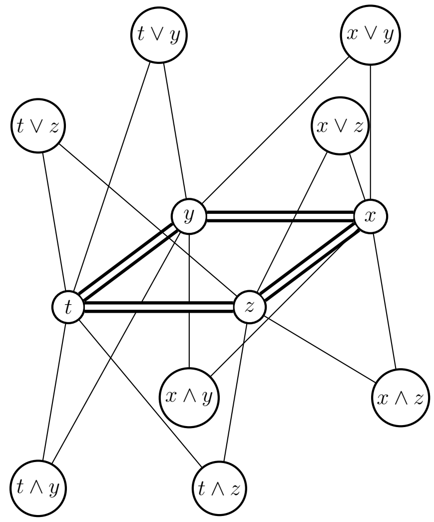


Figure 3. The general analogical proportion in a lattice.

Proposition 9 *Let y and z be two elements of a lattice. Then the following analogical proportion holds:*

$$(y \ : \ y \vee z \ :: \ y \wedge z \ : \ z)$$

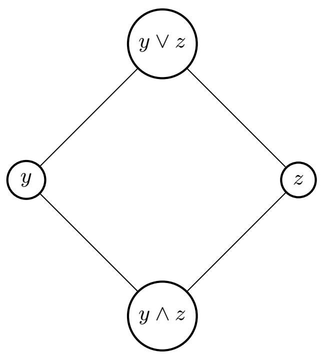


Figure 4. A canonical proportion in a lattice: $(y \ : \ y \vee z \ :: \ y \wedge z \ : \ z)$.

4.4 Learning from $\mathcal{U}$

After having given the basis in the previous sections, we give preliminary here remarks and hints concerning some possible extensions of this work to applications, via machine learning, in connexion with analogical proportions and lattice structure.

Firstly, when investigating the connexions between locally maximal subwords, locally minimal superwords and analogical proportions, a first property is easy to show from definition 15 and proposition 1.

Proposition 10 *Let $w = w_1 \cdots w_p$ be a locally maximal subword of two words u and v. Then:*

$$\forall t \in L(A_w(\{u,v\})), \ \exists w \in \dot{\sqcap}(u,v) \quad \textit{such that} \quad t \ : \ u \ :: \ v \ : \ w$$

$$\forall w \in \dot{\sqcap}(u,v), \ \exists t \in L(A_w(\{u,v\})), \quad \textit{such that} \quad t \ : \ u \ :: \ v \ : \ w$$

Take $u = abcabb$ and $v = aabbc$ with the maximal subword $y = abb$. The alignment $(\boxed{a}\,\boxed{b}\,ca\,\boxed{b}\,b, \boxed{a}\,a\,\boxed{b}\,\boxed{b}\,c)$ is locally maximal. The language $L(A_{abb}(\{u,v\})$ contains the word $w = abcabbc$. The facing figure displays the analogical proportion $w : u :: v : y$

$$\begin{pmatrix} a & \boxed{a} & \boxed{b} & c & a & \boxed{b} & b & c \\ & \boxed{a} & \boxed{b} & c & a & \boxed{b} & b & \\ a & \boxed{a} & \boxed{b} & & & \boxed{b} & & c \\ & \boxed{a} & \boxed{b} & & & \boxed{b} & & \end{pmatrix}$$

However, what we are really interested in is to find how using the lattice $\mathcal{U}$ and its analogical properties to generalize U. As a second remark, we note that any homogeneous set of alignments A in $\mathcal{U}$ represents an intensional definition of the finite language $\sqcup(A)$, the set of locally minimal superwords common to all the words in the support[8] of A. We can also construct, as indicated in section 2.4, a finite automaton as an intensional representation of this language, with the syntactic analysis facility. Therefore, we have potentially at our disposal a lattice of finite automata, in connection with the lattice of subsets of U: each automaton recognizes a finite language which is a particular generalization of the associated support, itself a subset of U.

We denote hereafter $\leq$ the order relation between finite set of words derived from the subword relation $\leq$, defined by: $M \leq N$ iff $\forall m \in n, \ \exists \ n \in N$ such that $m \leq n$. For example, $\{ab, c\} \leq \{abcd, e\}$. There is an partial inclusion relation between the languages recognized by this lattice of automata, compatible with that of the subsets, since the following property holds.

Proposition 11 *For any subsets J and K of U, the three following relations are equivalent: $L(A(J)) \leq L(A(K))$, $J \subset K$ and $\dot{\sqcap}(K) \subset \dot{\sqcap}(J)$.*

Note that the exploration of such a lattice of automata, constructed on a finite set of positive examples, is the basis of the efficient finite automata inference, see [dlH10]. This could be one basis for the use of our lattice in machine learning.

Another threads to follow could be the idea of analogical closure of a finite language, as described in [Lep03] and that of analogical generation, see [BMMA07]. In both, a triple of words is taken in the learning sample and a fouth sentence is generated, under the constraint that the four sentences are in analogical proportion. It is not yet clear to the authors how this technique can be combined with the lattice structure, but this could be a connection with the area of machine learning on the basis of formal concepts, as in [Kuz01].

5 Conclusion and related work

The problem of finding one longest common subsequence (subword) or one shortest common supersequence (superword) to two or more words has been well covered (see e.g. [Gus97], pp 287-293 and 309, [IF92]). However, to the best of our knowledge, the problem of finding an intentional definition to the sets of maximal subwords and minimal superwords of a set of words has not been explored yet. In this, we have produced, via the construction of a lattice of alignement sets, an interesting subset of minimal superwords and maximal subwords to a set of words. We have not worked yet neither on the theoretical complexity of the construction of the lattice of alignments,

[8] Remember that this support, that we have denoted $word(A)$, is a subset of U.

nor on its practical complexity and applications. Hereafter we give some bibliographical hints to this problem.

A complexity result (sometimes misinterpreted) is given by Maier [Mai78] who has demonstrated that the "yes/no longest common subsequence problem" and the "yes/no shortest common supersequence problem" are NP-complete for alphabets of sufficient size. These problems are defined as follows: "Given an integer k and a set of sequences R, is $|LCS(R)| \leq k$? Is $|SCS(R)| \geq k$?" where $|LCS(R)|$ and $|SCS(R)|$ are the length of a longest common subsequence and the length of a shortest common supersequence of R.

It is also true that finding the length of a shortest (longest) super(sub)sequence common to a set of k sequences is in[9] $\mathcal{O}(m_1 \ldots m_k)$, with m_i the size of the i-th of the k sequences, hence exponential in k.

The works of Fraser and Irving [FIM96] have produced algorithms to find the *longest* minimal common supersequence (superword) and the *shortest* maximal common subsequence, according to the order relation $\leq$.

Yvon and Stroppa [SY05] give a definition of an analogical proportion between words and also within lattices. Our objective is to use the properties of the lattice structure on alignement sets to solve the associated analogical equations.

REFERENCES

[BMMA07] S. Bayoudh, H. Mouchère, L. Miclet, and E. Anquetil. Learning a classifier with very few examples: analogy based and knowledge based generation of new examples for character recognition. In *European Conference on Machine Learning, Springer LNAI 4701*, 2007.

[dlH10] C. de la Higuera. *Grammatical Inference*. Cambridge University Press, 2010.

[FIM96] C. Fraser, R. Irving, and M. Middendorf. Maximal common subsequences and minimal common supersequences. *Information and Computation*, 124:145–153, 1996.

[GHE89] D. Gentner, K. Holyoak, and B. Kokinov (Editors). *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press, 1989.

[Gus97] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge Univ. Press, 1997.

[Has11] A. Ben Hassena. *Apprentissage par analogie de structures d'arbres*. PhD thesis, Université de Rennes 1, 2011.

[IF92] R. Irving and C. Fraser. Two algorithms for the longest common subsequence of three (and more) strings. In *Proc. 3rd Symp. on Combinatorial Pattern Matching. Springer LCNS 644*, pages 214–229, 1992.

[IF94] R. Irving and C. Fraser. Maximal common subsequences and minimal common supersequences. In *Proc. 5rd Symp. on Combinatorial Pattern Matching. Springer LCNS 807*, pages 173–183, 1994.

[Kuz01] O. Kuznetsov. Machine learning on the basis of formal concept analysis. *Automation and Remote Control*, 62, Issue 10:1543 – 1564, 2001.

[Lep03] Y. Lepage. *De l'analogie rendant compte de la commutation en linguistique*. Université de Grenoble, Grenoble, 2003. Habilitation à diriger les recherches.

[Mai78] D. Maier. The complexity of some problems on subsequences and supersequences. *JACM*, 25:332–336, 1978.

[MBD08] L. Miclet, S. Bayoudh, and A. Delhay. Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *journal of Artificial Intelligence Research*, 32:793–824, 2008.

[MBJ12] L. Miclet, N. Barbot, and B. Jeudy. The construction of a finite automaton recognizing exactly the maximal subwords common to two (and more) words. *In submission.*, 2012.

[Mit97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[9] The elementary operation is the comparison.

[SY05] N. Stroppa and F. Yvon. Analogical learning and formal proportions: Definitions and methodological issues. Technical Report ENST-2005-D004, École Nationale Supérieure des Télécommunications, June 2005.

[WF74] R. Wagner and M. Fisher. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

Belief revision-based case-based reasoning

Julien Cojan[1] and **Jean Lieber**[234]

Abstract. Adaptation is a task of case-based reasoning (CBR) that aims at modifying a case to solve a new problem. Now, belief revision deals also about modifications. This paper studies how some results about revision can be applied to formalize adaptation and, more widely, CBR. Revision operators based on distances are defined in formalisms frequently used in CBR and applied to define an adaptation operator that takes into account the domain knowledge and the adaptation knowledge. This approach to adaptation is shown to generalize some other approaches to adaptation, such as rule-based adaptation.

1 INTRODUCTION

Case-based reasoning and belief revision are two domains in which the notions of similarity and modification play an important role.

Case-based reasoning (CBR [19]) is a reasoning process using a case base, where a case is a representation of a problem-solving episode, in general, in the form of a problem-solution pair. CBR aims at solving a *target problem* and generally consists in a retrieval step (selection of one or several case(s) from the case base that is/are similar to the target problem), an adaptation step (modification of the retrieved case(s) to propose a solution to the target problem), and a possible storage of the case formed by the target problem and its solution.

Belief revision is the process of changing a belief base about a static world by incorporating new beliefs while keeping the belief base consistent. When the old beliefs are inconsistent with the new beliefs, the formers have to be modified in order to restore consistency with the latters. Usually, belief revision is based on the minimal change principle [1]: most of the old beliefs should be kept. One way to measure change (so that it is minimal) is to use a similarity metric (to be maximized) or a distance (to be minimized).

Thus, the question raised is whether the modification performed during CBR could be performed by a belief revision operator. This question has been addressed in several publications and this paper gives a synthesis of some of them.

The paper is organized as follows. Some preliminaries about CBR are given in section 2. Section 3 introduces belief revision. In CBR, the modifications are performed during the adaptation step, section 4 is the core of the paper and describes revision-based adaptation. More globally, belief revision can be applied to CBR as a whole as section 5 shows. Finally section 6 concludes the paper.

[1] INRIA Sophia-Antipolis, Edelweiss Project, Julien.Cojan@inria.fr
[2] Université de Lorraine, LORIA, UMR 7503 — 54506 Vandœuvre-lès-Nancy, France, Jean.Lieber@loria.fr
[3] CNRS — 54506 Vandœuvre-lès-Nancy, France
[4] Inria — 54602 Villers-lès-Nancy, France

2 PRELIMINARIES

2.1 Formalism

The approach to CBR presented in this paper can be applied to a variety of representation languages. It is assumed that there exists a representation language $\mathcal{L}$: a formula is an element of $\mathcal{L}$. The semantics of $\mathcal{L}$ is given by a (possibly infinite) set $\mathcal{U}$ and by a function $\text{Mod} : \varphi \in \mathcal{L} \mapsto \text{Mod}(\varphi) \in 2^{\mathcal{U}}$, defining, in a model-theoretical manner, the semantics of $\mathcal{L}$: $\mathtt{a}$ is a model of φ if $\mathtt{a} \in \text{Mod}(\varphi)$; φ_1 entails φ_2 ($\varphi_1 \models \varphi_2$) if $\text{Mod}(\varphi_1) \subseteq \text{Mod}(\varphi_2)$; φ_1 and φ_2 are equivalent ($\varphi_1 \equiv \varphi_2$) if $\text{Mod}(\varphi_1) = \text{Mod}(\varphi_2)$. A subset A of $\mathcal{U}$ is *representable* in $\mathcal{L}$ if there exists a formula φ such that $\text{Mod}(\varphi) = A$.

It is assumed that $\mathcal{L}$ is closed under conjunction, which means that for every $\varphi_1, \varphi_2 \in \mathcal{L}$ there exists a formula denoted by $\varphi_1 \wedge \varphi_2$ such that $\text{Mod}(\varphi_1, \varphi_2) = \text{Mod}(\varphi_1) \cap \text{Mod}(\varphi_2)$.

Some formalisms are closed under negation (or complement), which means that for every $\varphi \in \mathcal{L}$, there exists a formula denoted by $\neg\varphi$ such that $\text{Mod}(\neg\varphi) = \mathcal{U} \setminus \text{Mod}(\varphi)$. For such formalisms, $\varphi_2 \vee \varphi_2$ is an abbreviation for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \Rightarrow \varphi_2$ is an abbreviation for $\neg\varphi_1 \vee \varphi_2$ and $\varphi_1 \Leftrightarrow \varphi_2$ is an abbreviation for $(\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)$.

Propositional logic with n variables is an example of such a formalism: $\mathcal{U}$ denotes the set of interpretations on the variables. Every $A \subseteq \mathcal{U}$ is representable in this logic.

2.2 Case-based reasoning: principles and notations

For CBR, $\mathcal{U}$ is called the *case universe*. A *case instance* $\mathtt{a}$ is, by definition, an element of $\mathcal{U}$: $\mathtt{a} \in \mathcal{U}$. A *case* $\mathtt{C}$ is a class of case instances: $\mathtt{C} \in 2^{\mathcal{U}}$ (in this paper, a case represents a class of experiences, it is what is called an ossified case in [19] and a generalized case in [18]). For instance, when the formalism is propositional logic with n variables, $\mathcal{U}$ is the set of the 2^n interpretations and a case $\mathtt{C}$ is represented by a formula φ: $\mathtt{C} = \text{Mod}(\varphi)$.

A *source case* is denoted by `Source`: it is a case of `CaseBase` (the case base). The *target case* is denoted by `Target`: it is the input of the CBR system. In many applications, the source cases `Source` are *specific*: each of them represents a single case instance $\mathtt{a}$ ($\mathtt{Source} = \{\mathtt{a}\}$). By contrast, the target case specifies only its "problem part" and needs to be completed by a "solution part". The aim of the CBR process is to perform this completion:

$$\mathtt{CBR} : (\mathtt{CaseBase}, \mathtt{Target}) \mapsto \mathtt{ComplTarget}$$
$$\text{with } \mathtt{ComplTarget} \subseteq \mathtt{Target}$$

Usually, this inference is decomposed into two steps:

$$\mathtt{Retrieval} : (\mathtt{CaseBase}, \mathtt{Target}) \mapsto \mathtt{Source} \in \mathtt{CaseBase}$$
$$\mathtt{Adaptation} : (\mathtt{Source}, \mathtt{Target}) \mapsto \mathtt{ComplTarget}$$

In many CBR applications, a case instance $\mathtt{a}$ can be decomposed into a problem part x and a solution part y: $\mathtt{a} = (x, y)$. Let $\mathcal{U}_{\mathtt{pb}}$ and $\mathcal{U}_{\mathtt{sol}}$ be the universes of problem and solution instances: $x \in \mathcal{U}_{\mathtt{pb}}$, $y \in \mathcal{U}_{\mathtt{sol}}$, $\mathcal{U} = \mathcal{U}_{\mathtt{pb}} \times \mathcal{U}_{\mathtt{sol}}$. A source case $\mathtt{Source}$ is decomposed in a *source problem* $\mathtt{srce} \in 2^{\mathcal{U}_{\mathtt{pb}}}$ and its solution $\mathtt{Sol}(\mathtt{srce}) \in 2^{\mathcal{U}_{\mathtt{sol}}}$, thus $\mathtt{Source} = \mathtt{srce} \times \mathtt{Sol}(\mathtt{srce})$ that is interpreted as: for all $x \in \mathtt{srce}$ there exists $y \in \mathtt{Sol}(\mathtt{srce})$ such that $\mathtt{a} = (x, y)$ is a licit case (i.e., y solves x). The solution part of the target problem is unknown, thus $\mathtt{Target} = \mathtt{tgt} \times \mathcal{U}_{\mathtt{sol}}$, where $\mathtt{tgt}$ is called the *target problem*. When cases are decomposed in problems and solutions, CBR aims at solving the target problem $\mathtt{tgt}$, thus $\mathtt{ComplTarget} = \mathtt{tgt} \times \mathtt{Sol}(\mathtt{tgt})$ where $\mathtt{Sol}(\mathtt{tgt}) \in 2^{\mathcal{U}_{\mathtt{sol}}}$. In general, a target problem is specific: it is a singleton, i.e., $\mathtt{tgt} = \{x^t\}$ with $x^t \in \mathcal{U}_{\mathtt{pb}}$. This problem-solution decomposition is not needed to present revision-based adaptation but it is a prerequisite for other approaches to adaptation mentioned in the paper. When cases are decomposed in problem and solution parts, it is common to consider the adaptation problem as an analogical problem represented by the following diagram:

$$\begin{array}{ccc} \mathtt{srce} & \text{———} & \mathtt{tgt} \\ \downarrow & & \downarrow \\ \mathtt{Sol}(\mathtt{srce}) & \text{—} & \mathtt{Sol}(\mathtt{tgt}) \end{array} \qquad (1)$$

that can be read as "$\mathtt{Sol}(\mathtt{tgt})$ is to $\mathtt{Sol}(\mathtt{srce})$ as $\mathtt{tgt}$ is to $\mathtt{srce}$" (transformational analogy [3]) or "$\mathtt{Sol}(\mathtt{tgt})$ is to $\mathtt{tgt}$ as $sol(\mathtt{srce})$ is to $\mathtt{srce}$" (derivational analogy [4]).

The domain knowledge $\mathtt{DK}$ is a knowledge base giving a necessary condition for a case instance to be licit. Thus, the domain knowledge can be represented by a subset $\mathtt{DK}$ of $\mathcal{U}$ and for each $\mathtt{a} \in \mathcal{U}$, $\mathtt{a} \notin \mathtt{DK}$ involves that $\mathtt{a}$ is not licit. When the case universe is decomposed in $\mathcal{U}_{\mathtt{pb}} \times \mathcal{U}_{\mathtt{sol}}$, $\mathtt{a} = (x, y) \notin \mathtt{DK}$ means that y is not a solution of x or that x and/or y are meaningless (i.e., they are objects represented in the language that have no correspondence in the real world, e.g., in the domain of zoology, a cat that is not a mammal). Having no domain knowledge (or not taking it into account) amounts to $\mathtt{DK} = \mathcal{U}$.

Each source case is assumed to be consistent with the domain knowledge, i.e., $\mathtt{DK} \cap \mathtt{Source} \neq \emptyset$. Similarly, if a target case is inconsistent with the domain knowledge, it has not to be considered for the CBR inference (the CBR system has to reject it). Thus, if $\mathtt{Target}$ is an input of the adaptation procedure, it is required that $\mathtt{DK} \cap \mathtt{Target} \neq \emptyset$. The result of adaptation must also be consistent with $\mathtt{DK}$, therefore: $\mathtt{DK} \cap \mathtt{ComplTarget} \neq \emptyset$.

2.3 Distances and metric spaces

A distance on a set $\mathcal{U}$ is defined in this paper as a function $d : \mathcal{U} \times \mathcal{U} \to [0; +\infty]$ such that $d(\mathtt{a}, \mathtt{b}) = 0$ iff $\mathtt{a} = \mathtt{b}$ (the properties of symmetry and triangle inequality are not required in this paper, unless explicitly specified). Let $\mathtt{b} \in \mathcal{U}$ and $A, B \in 2^{\mathcal{U}}$. $d(A, \mathtt{b})$ is a notation for $\inf_{\mathtt{a} \in A} d(\mathtt{a}, \mathtt{b})$, $d(\mathtt{b}, A)$ is a notation for $\inf_{\mathtt{a} \in A} d(\mathtt{b}, \mathtt{a})$, and $d(A, B)$ is a notation for $\inf_{\mathtt{a} \in A, \mathtt{b} \in B} d(\mathtt{a}, \mathtt{b})$. A is said to be closed under d if $\{\mathtt{b} \in \mathcal{U} \mid d(A, \mathtt{b}) = 0\} = \{\mathtt{b} \in \mathcal{U} \mid d(\mathtt{b}, A) = 0\} = A$. By convention, the infimum on an empty set is $+\infty$, e.g., $d(A, \emptyset) = +\infty$.

If $\mathcal{U} = \mathbb{R}^n$ (where $\mathbb{R}$ is the set of the real numbers), a L_1-distance is a distance d parametrized by a base $\mathcal{B}$ of the vector space $\mathcal{U}$ such that $d(\mathtt{a}, \mathtt{b}) = \sum_{i=1}^{n} |v_i - u_i|$ where $(u_1, u_2, \ldots, u_n)$ (resp., $(v_1, v_2, \ldots, v_n)$) is the representation of $\mathtt{a}$ (resp., of $\mathtt{b}$) in $\mathcal{B}$. When $\mathcal{B}$ is the canonical base, $\mathtt{a}_i = u_i$ and $\mathtt{b}_i = v_i$ for each $i \in \{1, 2, \ldots n\}$. By extension, if $\mathcal{U}$ is a subset of $\mathbb{R}^n$, a distance d on $\mathcal{U}$ that is the restriction of a L_1 distance on $\mathbb{R}^n$ is also called a L_1 distance on $\mathcal{U}$ (for example, if $\mathcal{U} = \mathbb{Z}^n$ where $\mathbb{Z}$ is the set of integers).

3 BELIEF REVISION

3.1 Belief revision in propositional logic

In [1], postulates of belief revision are proposed in a general logical setting. These postulates are based on the idea of minimal change. They are applied to propositional logic in [13] which presents 6 postulates that a belief operation $\dotplus$ has to verify in this formalism: given ψ and μ, two belief bases, $\psi \dotplus \mu$ is a revision of ψ by μ. One of these postulates states that $\dotplus$ is independent to syntax:

$$\text{if } \psi_1 \equiv \psi_2 \text{ and } \mu_1 \equiv \mu_2 \text{ then } \psi_1 \dotplus \mu_1 \equiv \psi_2 \dotplus \mu_2 \qquad (2)$$

where ψ_1, ψ_2, μ_1, and μ_2 are formulas representing beliefs. As a consequence of (2), a formula φ can be assimilated to $\mathrm{Mod}(\varphi)$, the set of its models: in the rest of the paper, formulas and subsets of $\mathcal{U}$ are used indifferently. Then, the other 5 postulates can be rewritten as follows (using subsets of $\mathcal{U}$, the set of interpretations, instead of propositional formulas):

($\dotplus$1) $A \dotplus B \subseteq B$.
($\dotplus$2) If $A \cap B \neq \emptyset$ then $A \dotplus B = A \cap B$.
($\dotplus$3) If $B \neq \emptyset$ then $A \dotplus B \neq \emptyset$.
($\dotplus$4) $(A \dotplus B) \cap C \subseteq A \dotplus (B \cap C)$.
($\dotplus$5) If $(A \dotplus B) \cap C \neq \emptyset$ then $A \dotplus (B \cap C) \subseteq (A \dotplus B) \cap C$.

(A, B, and C are subsets of $\mathcal{U}$.) The interpretation of these postulates is made further, for their application to revision-based adaptation.

Intuitively, to revise A by B, the idea is to modify minimally A into A' so that $A' \cap B \neq \emptyset$, and then $A \dotplus B = A' \cap B$. Now, there are many ways to model minimal modifications. Among them, there is the modification based on a distance d on $\mathcal{U}$: given $\lambda \geq 0$, $G^d_\lambda(A)$ is the generalization (a kind of modification) of $A \subseteq \mathcal{U}$ defined by

$$G^d_\lambda(A) = \{\mathtt{b} \in \mathcal{U} \mid d(A, \mathtt{b}) \leq \lambda\}$$

Then, the revision operator $\dotplus^d$ is defined by

$$A \dotplus^d B = G^d_\delta(A) \cap B \qquad \text{where } \delta = \inf\{\lambda \mid G^d_\lambda(A) \cap B \neq \emptyset\}$$

Note that the infima on $\mathcal{U}$ are always reached when $\mathcal{U}$ is finite, which is the case when $\mathcal{U}$ is the set of interpretations over n propositional variables. This kind of revision operators is a direct generalization of the Dalal revision operator [9], which is based on the Hamming distance between propositional interpretations. The following equivalent definition can be given:

$$A \dotplus^d B = \{\mathtt{b} \in B \mid d(A, \mathtt{b}) = \delta\} \qquad \text{where } \delta = d(A, B)$$

(the δ in the two definitions are the same).

3.2 Belief revision in a metric space

The ($\dotplus$1-5) postulates can be straightforwardly generalized to other formalisms where $\mathcal{U}$ is (a priori) any set and each formula φ of such a formalism is assimilated to a subset $\mathrm{Mod}(\varphi)$ of $\mathcal{U}$.

Given a distance d on $\mathcal{U}$, $\dotplus^d$ can be defined as above but it must be noticed that $\dotplus^d$ may not satisfy the ($\dotplus$1-5) postulates. This issue is considered further, in section 4.4.1.

The representability issue must also be addressed in this generalization, thus we propose the following postulate:

($\dotplus$6) For any $\psi, \mu \in \mathcal{L}$, there exists $\rho \in \mathcal{L}$ such that $\mathrm{Mod}(\rho) = \mathrm{Mod}(\psi) \dotplus \mathrm{Mod}(\mu)$.

In propositional logic, for any operator $\dotplus$, this postulate holds, since every subset of the set $\mathcal{U}$ of the interpretations of propositional variables is representable in this logic.

3.3 Integrity constraint belief merging

Let $\psi_1, \psi_2, \ldots, \psi_k$, and μ be $k+1$ belief bases. Merging $\psi_1, \psi_2, \ldots, \psi_k$, given the integrity constraint μ consists in building a belief base φ such that $\varphi \models \mu$ and φ keeps "as much as possible" from the ψ_i's. A merging operator $\triangle : \left(\mu, \{\psi_i\}_{1 \leq i \leq k}\right) \mapsto \varphi = \triangle_\mu\left(\{\psi_i\}_{1 \leq i \leq k}\right)$ is assumed to satisfy some postulates similar to the postulates for a revision operator (in [15], such postulates are defined in propositional logic but can be easily generalized to metric spaces). Actually, the notion of integrity constraint extends the notion of belief revision in the sense that if $\triangle$ is such a merging operator, then $\dotplus$ defined by $\psi \dotplus \mu = \triangle_\mu(\{\psi\})$ satisfies the ($\dotplus$1-5) postulates.

4 REVISION-BASED ADAPTATION

This section defines revision-based adaptation (§4.1), presents an example in propositional logic (§4.2), studies its properties (§4.3), describes with details revision-based adaptation in metric spaces (§4.4), mentions briefly an extension to multiple case adaptation (§4.5), relates this approach to adaptation with rule-based adaptation (§4.6) and gives pointers on other work related to revision-based adaptation (§4.7).

4.1 Definition

Let $\mathcal{U}$ be the case universe and $\dotplus$ be a revision operator on $\mathcal{U}$. The $\dotplus$-adaptation is defined as follows [16]:

$$\mathtt{ComplTarget} = (\mathtt{DK} \cap \mathtt{Source}) \dotplus (\mathtt{DK} \cap \mathtt{Target}) \quad (3)$$

($\mathtt{DK}\cap\mathtt{Source}$) (resp., ($\mathtt{DK}\cap\mathtt{Target}$)) is the source (resp., target) case interpreted within the domain knowledge (i.e., case instances known to be not licit are removed). Thus (3) can be interpreted as a minimal modification of the source case to satisfy the target case, given the domain knowledge, knowing that the minimality of modification is the one associated with the operator $\dotplus$.

4.2 Example in Propositional Logic

Let us consider the following story. Léon is about to invite Thècle and wants to prepare her an appropriate meal. His target problem can be specified by the characteristics of Thècle about food. Let us assume that Thècle is vegetarian (denoted by the propositional variable v) and that she has other characteristics (denoted by o) not detailed in this example:

$$\mathtt{Target} = v \wedge o$$

From his experience as a host, Léon remembers that he had invited Simone some times ago and he thinks that Simone is very similar to Thècle according to food behavior, except that she is not a vegetarian ($\neg v \wedge o$). He had proposed to Simone a meal with salad (s), beef (b), and a dessert (d), and she was satisfied by the two formers but has not eaten the dessert, thus Léon has retained the source case

$$\mathtt{Source} = (\neg v \wedge o) \wedge (s \wedge b \wedge \neg d)$$

Besides that, Léon has some general knowledge about food: he knows that beef is meat, that meat and tofu are protein foods, and that vegetarians do not eat meat. Moreover, the only protein food that he is willing to cook, apart from meat, is tofu. Thus, his domain knowledge is

$$\mathtt{DK} \quad = \quad b \Rightarrow m \quad \wedge \quad m \vee t \Leftrightarrow p \quad \wedge \quad v \Rightarrow \neg m$$

where b, m, t, and p are the propositional variables for "some beef/meat/tofu/protein food is appreciated by the current guest". According to $\dotplus$-adaptation, what meal should be proposed to Thècle? If $\dotplus$ is the Dalal revision operator, the $\dotplus$-adaptation of the meal for Simone to a meal for Thècle is

$$\mathtt{ComplTarget} \equiv \mathtt{DK} \wedge \mathtt{Target} \wedge (s \wedge t \wedge \neg d)$$

In [16], this adaptation is qualified as conservative: the salad and the absence of dessert is reused for the target case and, though the beef is not kept (to ensure a consistent result), the consequence of b that is consistent with DK, i.e., p, is kept, and thus, t is proposed instead of beef (since $v \wedge p \models_{\mathtt{DK}} t$; in other words, some protein food is required, the only vegetarian protein that Léon is willing to cook is tofu, thus there will be tofu in the meal).

4.3 Properties

The ($\dotplus$1-6) postulates entail some properties of revision-based adaptation.

($\dotplus$1) applied to $\dotplus$-adaptation gives $\mathtt{ComplTarget} \subseteq \mathtt{DK}\cap\mathtt{Target}$, i.e., $\mathtt{ComplTarget} \subseteq \mathtt{Target}$ (that is a property required by an adaptation process: cf. section 2.2) and $\mathtt{ComplTarget} \subseteq \mathtt{DK}$ (no instance case $\mathtt{a}$ known to be illicit —$\mathtt{a} \in \mathcal{U} \setminus \mathtt{DK}$— is in the result).

Let us assume that $\mathtt{DK}\cap\mathtt{Source}\cap\mathtt{Target} \neq \emptyset$. Then, ($\dotplus$2) entails that $\mathtt{ComplTarget} = \mathtt{DK}\cap\mathtt{Source}\cap\mathtt{Target}$. Thus, $\mathtt{ComplTarget} = \mathtt{DK}\cap\mathtt{Source}\cap\mathtt{Target}$: if the target case is consistent with the source case, given the domain knowledge, then it can be inferred by $\dotplus$-adaptation that Source solves Target. This is consistent with the principle of this kind of adaptation: ComplTarget is obtained by keeping from Source as much as possible, and if no modification is needed then no modification is applied.

($\dotplus$3) gives: if $\mathtt{DK} \cap \mathtt{Target} \neq \emptyset$ then $\mathtt{ComplTarget} \neq \emptyset$. Since $\mathtt{ComplTarget} \subseteq \mathtt{DK}$ (cf. ($\dotplus$1)), $\mathtt{DK} \cap \mathtt{ComplTarget} \neq \emptyset$, which is a property required by an adaptation operator (cf. section 2.2).

According to [13], ($\dotplus$4) and ($\dotplus$5) capture the minimality of modifications. Thus they express the minimality of modification made by a $\dotplus$-adaptation. This can be interpreted as follows. The conjunction of ($\dotplus$4) and ($\dotplus$5) can be reformulated as:

$$\begin{cases} \text{Either } (A \dotplus B) \cap C = \emptyset, \\ \text{Or } (A \dotplus B) \cap C = A \dotplus (B \cap C). \end{cases} \quad (4)$$

Let F represent some additional features about the target problem: the new target case is $\mathtt{Target}_2 = \mathtt{Target} \cap F$. If ComplTarget is consistent with F, then ($\dotplus$4) and ($\dotplus$5) entails that the adaptation of Source to $\mathtt{Target}_2$ gives $\mathtt{ComplTarget}_2 = \mathtt{ComplTarget} \cap F$. In other words, if F does not involve needs on modifications (corresponding to an inconsistency) then the result of the $\dotplus$-adaptation can be reused straightforwardly.

($\dotplus$6) involves that ComplTarget is representable in $\mathcal{L}$.

4.4 Revision-Based Adaptation in Metric Spaces

In this section, $\dotplus^d$-adaptation is considered on a metric space $(\mathcal{U}, d)$.

4.4.1 The ($\dotplus$1-6) postulates in metric spaces

This section studies the revision postulates for $\dotplus^d$. Some of these postulates are not satisfied by $\dotplus^d$ and some additional assumptions on the representation language $\mathcal{L}$ and on d are proposed that are sufficient conditions for their satisfaction (provided that the subsets A, B, and C of $\mathcal{U}$ involved in postulates ($\dotplus$1-5) can be represented in $\mathcal{L}$).

($\dotplus$1) is always satisfied by $\dotplus^d$ (cf. the definitions).

($\dotplus$2) is not always satisfied by $\dotplus^d$ as the following counterexample shows. Let $\mathcal{U} = \mathbb{R}$ and let $\mathcal{L}$ be the language of intervals of $\mathbb{R}$, e.g., $[0; 1[= \{\mathtt{a} \in \mathcal{U} \mid 0 \leq \mathtt{a} < 1\}$. Let $d : (\mathtt{a}, \mathtt{b}) \mapsto |\mathtt{b} - \mathtt{a}|$. It can be shown that $[0; 1[\dotplus^d [0; 1[= [0; 1] \not\subseteq [0; 1[$. Now, let us consider the following additional assumption:

($\mathcal{L}$1) Every subset A of $\mathcal{U}$ that can be represented in $\mathcal{L}$ is closed under d.

Under this assumption, ($\dotplus$2) is satisfied as proven hereafter. Let A and B be two subsets of $\mathcal{U}$ such that A is closed and $A \cap B \neq \emptyset$. Thus, $d(A, B) = 0$ and

$$\begin{aligned} A \dotplus^d B &= \{\mathtt{b} \in B \mid d(A, \mathtt{b}) = 0\} \\ &= \{\mathtt{b} \in \mathcal{U} \mid d(A, \mathtt{b}) = 0\} \cap B \\ &= A \cap B \qquad \text{since } A \text{ is closed} \end{aligned}$$

Therefore $A \dotplus^d B = A \cap B$ and ($\dotplus$2) is satisfied.

($\dotplus$3) is not always satisfied by $\dotplus^d$ as the following counterexample shows. Let $\mathcal{U}$ and d be the same as in the counterexample of ($\dotplus$2). Let $A = [0; 1]$ and $B =]2; 3]$. $B \neq \emptyset$ but $A \dotplus^d B = \emptyset$. This suggests that A and B should be closed but even if the ($\mathcal{L}$1) assumption was made, ($\dotplus$3) may be not satisfied. For example, if $\mathcal{U} = \mathbb{R}^2$ and d is a L_1 distance on $\mathcal{U}$, $A = \{(x, y) \in \mathcal{U} \mid y = 0\}$ and $B = \{(x, y) \in \mathcal{U} \mid x > 0 \text{ and } y = 1/x\}$, we have $B \neq \emptyset$ and $A \dotplus^d B = \emptyset$ though A and B are closed. Now, let us consider the following assumption:

($\mathcal{L}$2) For every A and B non empty subsets of $\mathcal{U}$ representable in $\mathcal{L}$, the distance between A and B is always reached: there exist $\mathtt{a} \in A$ and $\mathtt{b} \in B$ such that $d(A, B) = d(\mathtt{a}, \mathtt{b})$.

Under this assumption, ($\dotplus$3) is satisfied as proven hereafter. Let $A, B \in 2^{\mathcal{U}}$ such that $B \neq \emptyset$. If $A = \emptyset$, $d(A, B) = d(A, \mathtt{b}) = +\infty$ for every $\mathtt{b} \in B$, thus $A \dotplus^d B = B \neq \emptyset$. If $A \neq \emptyset$ and if A and B are representable in $\mathcal{L}$ then ($\mathcal{L}$2) entails that $d(A, B) = d(\mathtt{a}, \mathtt{b})$ for some $(\mathtt{a}, \mathtt{b}) \in A \times B$.) Since $d(\mathtt{a}, \mathtt{b}) \geq d(A, \mathtt{b}) \geq d(A, B) = d(\mathtt{a}, \mathtt{b})$, $d(A, \mathtt{b}) = d(\mathtt{a}, \mathtt{b})$. Therefore, $\mathtt{b}$ is such that $d(A, \mathtt{b}) = d(A, B)$ thus $\mathtt{b} \in A \dotplus^d B$ and so, $A \dotplus^d B \neq \emptyset$. So ($\dotplus$3) is satisfied.

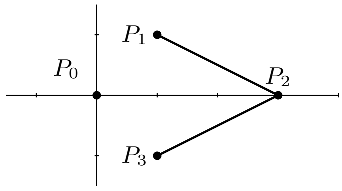
If $\mathcal{U} = \mathbb{R}^n$, if d is a L_1 distance on $\mathcal{U}$, and if each A representable in $\mathcal{L}$ is closed and bounded, then ($\mathcal{L}$2) is satisfied. More generally, if d is a distance in the classical mathematical sense (it verifies separation, symmetry, triangle inequality, and $d(\mathtt{a}, \mathtt{b}) < +\infty$ for every $\mathtt{a}, \mathtt{b} \in \mathcal{U}$), and if every A representable in $\mathcal{L}$ is a compact space, then ($\mathcal{L}$2) is satisfied.

($\dotplus$4) and ($\dotplus$5) are always satisfied by $\dotplus^d$ as proven hereafter. The conjunction of these postulates is equivalent to (4). Let $A, B, C \in 2^{\mathcal{U}}$. If $(A \dotplus^d B) \cap C = \emptyset$ then (4) is verified. Now, assume that $(A \dotplus^d B) \cap C \neq \emptyset$ and let $\mathtt{b} \in (A \dotplus^d B) \cap C$. Then $\mathtt{b} \in (A \cap B)$ and $d(A, \mathtt{b}) = d(A, B)$. Thus, the following chain of relations can be established:

$$d(A, B) \leq d(A, B \cap C) \leq d(A, \mathtt{b}) = d(A, B)$$

(cf. the infimum appearing in the definition of $d(A, \cdot)$ and the fact that $\mathtt{b} \in B \cap C$). Therefore, these real numbers are all equal and $d(A, B) = d(A, B \cap C)$. Hence $A \dotplus^d (B \cap C) = \{\mathtt{b} \in B \cap C \mid d(A, \mathtt{b}) = d(A, B)\} = (A \dotplus^d B) \cap C$.

($\dotplus$6) is not always satisfied by $\dotplus^d$ as the following counterexample shows. Let $\mathcal{U} = \mathbb{R}^2$, d be the L_1 distance on $\mathbb{R}^2$ (with the canonical base), and a formula of $\mathcal{L}$ represent a polygonal line. Let us consider $\psi, \mu \in \mathcal{L}$ such that $\mathrm{Mod}(\psi) = \{P_0\}$ with $P_0 = (0, 0)$ and $\mathrm{Mod}(\mu)$ is the polygonal line $P_1 - P_2 - P_3$ such that $P_1 = (0, 1)$, $P_2 = (0, 3)$, and $P_3 = (0, -1)$:



Then $\mathrm{Mod}(\psi) \dotplus^d \mathrm{Mod}(\mu) = \{P_1, P_3\}$ which is not a polygonal line thus cannot be expressed in $\mathcal{L}$.

4.4.2 Attribute-constraint formalisms

Definitions. In this section, it is assumed that $\mathcal{U} = V_1 \times V_2 \times \ldots \times V_n$ where each V_i is a "simple value" space, i.e. either $\mathbb{R}$ (the set of real numbers), $\mathbb{Z}$ (the set of integers), any interval of $\mathbb{R}$ or $\mathbb{Z}$, $\mathbb{B} = \{\mathtt{true}, \mathtt{false}\}$, or another set defined in extension. For $i \in \{1, 2, \ldots, n\}$, the attribute a_i is the i^{th} projection:

$$a_i : (\mathtt{a}_1, \mathtt{a}_2, \ldots, \mathtt{a}_n) \in \mathcal{U} \mapsto \mathtt{a}_i \in V_i$$

A formula φ of the representation language is a constraint, i.e., a Boolean expression based on the attributes a_i: $\varphi = P(a_1, a_2, \ldots, a_n)$. The semantics of φ is

$$\mathrm{Mod}(\varphi) = \{\mathtt{a} \in \mathcal{U} \mid P(a_1(\mathtt{a}), a_2(\mathtt{a}), \ldots, a_n(\mathtt{a}))\}$$

These formalisms contain propositional logic with n variables: $V_i = \mathbb{B}$ (for each $i \in \{1, 2, \ldots, n\}$), knowing that the Boolean expressions are based on the Boolean operations `and`, `or`, and `not`. For example, if $n = 3$, and $a_1 = o$, $a_2 = t$ and $a_3 = v$:

$$\mathrm{Mod}(\neg v \vee o) = \{(\mathtt{a}_1, \mathtt{a}_2, \mathtt{a}_3) \in \mathcal{U} \mid \mathtt{or}(\mathtt{not}(\mathtt{a}_3), \mathtt{a}_1)) = \mathtt{true}\}$$

These formalisms also contain the attribute-value formalisms often used for representing cases in CBR [14]: a specific case $\mathtt{C}$ is defined by $\mathtt{C} = (a_1 = v_1) \wedge (a_2 = v_2) \wedge \ldots \wedge (a_n = v_n)$ and thus $\mathtt{C} = \{(v_1, v_2, \ldots, v_n)\}$. When problem-solution decomposition is made, in general, the attributes are split in problem attributes (a_1, $\ldots$, a_p) and solution attributes (a_{p+1}, $\ldots$, a_n). Classically, the distance used on $\mathcal{U}$ for the retrieval is the weighted sum of distances on each problem attribute.

Application to the numerical case with linear constraints. Now, it is assumed that each V_i is either $\mathbb{R}$ or $\mathbb{Z}$ and each formula is a conjunction of linear constraints on the attributes. A linear constraint is an expression of the form $\sum_{i=1}^{n} \alpha_i \cdot a_i \leq \beta$ where $\alpha_1, \ldots, \alpha_n, \beta \in \mathbb{R}$.

Let d be the L_1 distance on $\mathcal{U}$ parametrized by a base $\mathcal{B}$. It can be shown that $\dotplus^d$ satisfies all the ($\dotplus$1-6) postulates (where A, B, and C are defined thanks to conjunctions of linear constraints). $\dot{+}^d$-adaptation amounts to solve the following optimization problem:

$$\mathtt{a} \in \mathtt{DK} \cap \mathtt{Source} \tag{5}$$

$$\mathtt{b} \in \mathtt{DK} \cap \mathtt{Target} \tag{6}$$

$$\text{minimize} \sum_{i=1}^{n} |v_i - u_i| \tag{7}$$

where $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$ are the respective representations of $\mathtt{a}$ and $\mathtt{b}$ in $\mathcal{B}$. $\mathtt{ComplTarget}$ is the set of the $\mathtt{b}$ that solve this optimization problem.

In this optimization problem, (5) and (6) are linear constraints, but the function to be minimized in (7) is not linear. However, this optimization problem can be solved thanks to the solving of the following linear problem (introducing the new variables $z_1, \ldots, z_n$):

$$\begin{array}{ll} \mathtt{a} \in \mathtt{DK} \cap \mathtt{Source} & \\ \mathtt{b} \in \mathtt{DK} \cap \mathtt{Target} & \\ v_i - u_i \leq z_i & (1 \leq i \leq n) \\ u_i - v_i \leq z_i & (1 \leq i \leq n) \\ \text{minimize} \sum_{i=1}^{n} z_i & \end{array}$$

It can be shown that the optimal values of $\mathtt{a}$ and $\mathtt{b}$ in the two optimization problems are the same. Therefore, in this formalism, $\dot{+}^d$-adaptation amounts to a linear programming problem, which is NP-complete if some $V_i = \mathbb{Z}$ but is polynomial when all $V_i = \mathbb{R}$ [12].

More details about this process can be found in [5].

A cooking application. This principle has been applied to a CBR system called Taaable (`http://taaable.fr`) that has been a contestant of the CCC (*Computer Cooking Contest*, organized during the ICCBR conferences). The CCC provides a recipe base. A contestant of the CCC is a system that has to solve cooking problems using these recipes (a case of this application is a recipe). These problems are specified by a set of desired ingredients or dish types, and undesired ones (e.g., "I'd like a pear pie but I don't like cinnamon."). Taaable has won the main challenge and the adaptation challenge of this contest in 2010 [2]. The adaptation of ingredient quantities was made possible thanks to a reduction to linear programming as mentioned before. Details can be found in [2] but the idea, explained on a simplified example, is as follows. Suppose that the user wants a recipe of a pear pie and that Taaable retrieves an apple pie. The domain knowledge is expressed by linear constraints on these properties, such as:

$$\mathtt{mass}_{\mathtt{fruit}} = 120 \cdot \mathtt{nb}_{\mathtt{apple}} + 100 \cdot \mathtt{nb}_{\mathtt{pear}}$$

$$\mathtt{mass}_{\mathtt{sweet}} = \mathtt{mass}_{\mathtt{sugar}} + 10 \cdot \mathtt{nb}_{\mathtt{apple}} + 15 \cdot \mathtt{nb}_{\mathtt{pear}}$$

(these knowledge can be found in a free nutritional database). Each $\mathtt{mass}_{\bullet}$ is an attribute on $\mathbb{R}$ and each $\mathtt{nb}_{\bullet}$ is an attribute on $\mathbb{N}$ (non negative integers). The source case is a singleton $\{\mathtt{a}\}$ such that $\mathtt{nb}_{\mathtt{apple}}(\mathtt{a}) = 4$ and $\mathtt{mass}_{\mathtt{sugar}}(\mathtt{a}) = 40$. The target case corresponds to the constraint $\mathtt{nb}_{\mathtt{apple}} = 0$ (the substitution of apples by pears is inferred by a previous step similar to a $\dot{+}$-adaptation in propositional logic). The $\dot{+}^d$-adaptation leads to a maximal preservation of the attributes $\mathtt{mass}_{\mathtt{fruit}}$ and $\mathtt{mass}_{\mathtt{sugar}}$ and since the pears contain more sweet than the apples, the mass of added sugar is lowered (there is a sort of "compensation effect"). More precisely, the $\dot{+}^d$-adaptation (at least for some base $\mathcal{B}$) gives $\mathtt{ComplTarget} = \{\mathtt{b}\}$ with $\mathtt{nb}_{\mathtt{pear}}(\mathtt{b}) = 5$ (the total fruit mass from $\mathtt{Source}$ to $\mathtt{ComplTarget}$ is modified from 480 to 500) and $\mathtt{mass}_{\mathtt{sugar}}(\mathtt{b}) = 5$ (the total sweet mass is unchanged).

4.5 Multiple case adaptation

Some CBR systems retrieve several cases and then adapt them in order to solve the target case:

$$\begin{array}{rl} \mathtt{Retrieval}: & (\mathtt{CaseBase}, \mathtt{Target}) \\ & \mapsto \{\mathtt{Source}_i\}_{1 \leq i \leq k} \subseteq \mathtt{CaseBase} \\ \mathtt{Adaptation}: & \left(\{\mathtt{Source}_i\}_{1 \leq i \leq k}, \mathtt{Target}\right) \mapsto \mathtt{ComplTarget} \end{array}$$

This adaptation is called multiple case adaptation and is also known as case combination. Multiple case adaptation extends single case adaptation (which is a case combination with $k = 1$) in the same way as integrity constraint belief merging extends belief revision (cf. section 3.3), hence the idea[5] to use a merging operator $\triangle$ on $\mathcal{U}$ to define a multiple case adaptation process:

$$\mathtt{ComplTarget} = \triangle_{\mathtt{DK} \cap \mathtt{Target}} \left(\{\mathtt{DK} \cap \mathtt{Source}_i\}_{1 \leq i \leq k}\right)$$

which generalizes (3).

This approach to multiple case adaptation is studied in [5].

4.6 Revision-based adaptation and rule-based adaptation

Other approaches to adaptation have been defined in the CBR literature. This section compares revision-based adaptation to one of them.

Rule-based adaptation is the adaptation based on a set of *adaptation rules*. Following the formalization of [17], an adaptation rule is an ordered pair $(\mathtt{r}, \mathcal{A}_\mathtt{r})$ where $\mathtt{r}$ is a binary relation on $\mathcal{U}_{\mathtt{pb}}$ and $\mathcal{A}_\mathtt{r}$ is such that, for $x^s, x^t \in \mathcal{U}_{\mathtt{pb}}$ and $y^s \in \mathcal{U}_{\mathtt{sol}}$ ($\mathtt{Source} = \{(x^s, y^s)\}$, $\mathtt{Target} = \{x^t\} \times \mathcal{U}_{\mathtt{sol}}$):

if $x^s \; \mathtt{r} \; x^t$ **then** $\mathcal{A}_\mathtt{r}(x^s, y^s, x^t) = y^t$ *probably* solves x^t

The rule is not certain (hence the "probably").

The adaptation rules can be composed as explained hereafter. Let $\mathtt{AK}$ be the finite set of adaptation rules that are available to the CBR system. Let $\mathtt{AK}_{\mathtt{pb}} = \{\mathtt{r} \mid (\mathtt{r}, \mathcal{A}_\mathtt{r}) \in \mathtt{AK}\}$.

$\mathtt{AK}_{\mathtt{pb}}$ provides a structure on $\mathcal{U}_{\mathtt{pb}}$. A similarity path from $x^s \in \mathcal{U}_{\mathtt{pb}}$ to $x^t \in \mathcal{U}_{\mathtt{pb}}$ is a path in $(\mathcal{U}_{\mathtt{pb}}, \mathtt{AK}_{\mathtt{pb}})$: it is a sequence of relations $\mathtt{r}^i \in \mathtt{AK}_{\mathtt{pb}}$ such that there exist $x^0, x^1, \ldots, x^q \in \mathcal{U}_{\mathtt{pb}}$ with $x^0 = x^s$, $x^q = x^t$, and $x^{i-1} \; \mathtt{r}^i \; x^i$ ($1 \leq i \leq q$). Given such a similarity path, $y^t \in \mathcal{U}$ that probably solves x^t can be computed by applying successively the rules $(\mathtt{r}^1, \mathcal{A}_{\mathtt{r}^1}), \ldots, (\mathtt{r}^q, \mathcal{A}_{\mathtt{r}^q})$: $y^i = \mathcal{A}_{\mathtt{r}^i}(x^{i-1}, y^{i-1}, x^i)$ for i taking the successive values 1, 2, …, q. Finally, $y^t = x^q$ probably solves x^t. This can be graphically represented by the following diagram, composed of q diagrams like the one of (1), section 2.2:

$$\begin{array}{ccccccccc} x^s = x^0 & \overset{\mathtt{r}^1}{\text{———}} & x^1 & \overset{\mathtt{r}^2}{\text{———}} & x^2 & \cdots\cdots & x^{q-1} & \overset{\mathtt{r}^q}{\text{———}} & x^q = x^t \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ y^s = y^0 & \overset{\mathcal{A}_{\mathtt{r}^1}}{\text{———}} & y^1 & \overset{\mathcal{A}_{\mathtt{r}^2}}{\text{———}} & y^2 & \cdots\cdots & y^{q-1} & \overset{\mathcal{A}_{\mathtt{r}^q}}{\text{———}} & y^q = y^t \end{array}$$

[5] Once suggested by Pierre Marquis. Thanks Pierre!

There may be several similarity paths from x^s to x^t. The choice between them is usually based on a cost function such that if SP_1 and SP_2 are two similarity paths from x^s to x^t and $\mathtt{cost}(SP_1) < \mathtt{cost}(SP_2)$ then SP_1 is preferred to SP_2, which is interpreted as "SP_1 is more likely to lead to an appropriate solution to x^t than SP_2." The function `cost` is usually assumed to be additive, that is $\mathtt{cost}(SP)$ is the sum of $\mathtt{cost}(\mathtt{r})$ for $\mathtt{r}$ a relation of SP. To each $(\mathtt{r}, \mathcal{A}_\mathtt{r}) \in \mathtt{AK}$, $\mathtt{cost}(\mathtt{r}) > 0$ is an information associated with this adaptation rule.[6]

Let $d_{\mathtt{AK}}$ be the distance on $\mathcal{U}$ defined by

$$d_{\mathtt{AK}}((x^s, y^s), (x^t, y^t)) = \min \left\{ \mathtt{cost}(SP) \;\middle|\; \begin{array}{l} SP\text{: similarity path from } x^s \text{ to } y^t \\ \text{such that the application of } SP \\ \text{on } \{(x^s, y^s)\} \text{ gives } y^t \end{array} \right\}$$

with the convention $\min \emptyset = +\infty$. Let $\mathtt{ComplTarget} = \mathtt{tgt} \times \mathtt{Sol(tgt)}$ be the result of $\dotplus^{d_{\mathtt{AK}}}$-adaptation without domain knowledge ($\mathcal{U} = \mathtt{DK}$). If there is no similarity path from x^s to x^t, then $\mathtt{ComplTarget} = \mathtt{Target}$ (the adaptation fails: it does not add any information to the target case). Else, $\mathtt{b} = (x^t, y^t) \in \mathtt{ComplTarget}$ iff y^t is obtained by application of a similarity path of minimal cost. Therefore, revision-based adaptation includes rule-based adaptation. Moreover, DK can be taken into account in $\dotplus^{d_{\mathtt{AK}}}$-adaptation, thus this enable to specify a rule-based adaptation taking into account the domain knowledge. Conversely, if some adaptation knowledge AK in the form of rules has been acquired (e.g., by means of knowledge discovery and data-mining techniques [8, 10]), this can be useful to specify a relevant revision operator. Indeed, there are many possible revision operators and the adaptation knowledge enables to make some choices among them.

A limitation of rule-based reasoning is that it can fail (i.e., $\mathtt{ComplTarget} = \mathtt{Target}$) and this is particularly true when there are few adaptation rules (if $\mathtt{AK} \subseteq \mathtt{AK}'$ then $d_{\mathtt{AK}}(\mathtt{a}, \mathtt{b}) \geq d_{\mathtt{AK}'}(\mathtt{a}, \mathtt{b})$ so if $\dotplus^{d_{\mathtt{AK}'}}$-adaptation fails then $\dotplus^{d_{\mathtt{AK}}}$-adaptation fails). One way to overcome this limitation is to combine this kind of adaptation with another approach to adaptation and the principle of revision-based adaptation can be used to formalize this combination. This idea is formalized as follows. Let us assume that the other approach to adaptation that has to be combined with rule-based adaptation can be formalized as a revision-based adaptation and let d_0 be a distance on $\mathcal{U}$ such that this adaptation coincides with the $\dotplus^{d_0}$-adaptation (intuitively, this adaptation is a "novice" adaptation, hence the 0 in d_0). The $\dotplus^{d}$-adaptation with d defined below combines rule-based adaptation with $\dotplus^{d_0}$-adaptation (for $\mathtt{a}, \mathtt{b} \in \mathcal{U}$):

$$d(\mathtt{a}, \mathtt{b}) = \inf_{\mathtt{c} \in \mathcal{U}} (W_{\mathtt{AK}} \cdot d_{\mathtt{AK}}(\mathtt{a}, \mathtt{c}) + W_0 \cdot d_0(\mathtt{c}, \mathtt{b}))$$

where $W_{\mathtt{AK}}$ and W_0 are two positive constants. When $\mathtt{AK} = \emptyset$, $d = d_0$ (intuitively: with no adaptation knowledge, the adaptation process is a novice). If the infimum above is reached on $\mathtt{c} = (x^c, y^c)$, the $\dotplus^{d}$-adaptation consists in a rule-based adaptation of $\mathtt{Source} = \{\mathtt{a}\}$ to solve $\{x^c\}$ and then a $\dotplus^{d_0}$-adaptation of $\mathtt{c}$ to solve the target case.

4.7 Other studies related to revision-based adaptation

Some other studies related to revision-based adaptation have been carried out.

[6] A coarse modeling of this cost is $\mathtt{cost}(\mathtt{r}) = -\log P$ where P is the probability that y^t is a licit solution of x^s. Thus, the additivity of the cost corresponds to an independence assumption of the q adaptation steps.

An algorithm based on revision-based adaptation principles has been described for the description logic $\mathcal{ALC}$ [6]. In this work, each case is represented as an instance associated with some assertions and the domain knowledge is represented by a set of terminological axioms.

Belief revision has been studied in qualitative algebras [7] and thus, it is natural to apply this work on revision-based adaptation to these formalisms. This has been studied in [11], with an application to the adaptation of the procedural part of a cooking recipe (using a temporal algebra) and to the adaptation of crops spatial allocation (using a spatial algebra).

5 REVISION-BASED CBR

Let SOURCE be the union of all the cases from the case base:

$$\mathtt{SOURCE} = \bigcup_i \mathtt{Source}_i \text{ where } \mathtt{CaseBase} = \{\mathtt{Source}_i\}_i$$

The following question can be raised: according to what conditions can the CBR process with SOURCE as only source case be equivalent to the CBR process with CaseBase? This question is addressed below with a $\dotplus^{d}$-adaptation.

Let $A_1, A_2, \ldots, A_n$, and B be $n+1$ subsets of $\mathcal{U}$. Let $\delta_i = d(A_i, B)$ and $\Delta = \min_i \delta_i$. The following equation holds:

$$\left(\bigcup_i A_i\right) \dotplus^{d} B = \bigcup_{i, \delta_i = \Delta} (A_i \dotplus^{d} B)$$

Indeed $d(\bigcup_i A_i, \mathtt{b}) = \min_i d(A_i, \mathtt{b})$ for any $\mathtt{b} \in B$, and so $(\bigcup_i A_i) \dotplus B = \{\mathtt{b} \in \mathcal{U} \mid \min_i d(A_i, b) = \Delta\} = \bigcup_{i, \delta_i = \Delta}(A_i \dotplus B)$.

From this equation applied to $A_i = \mathtt{DK} \cap \mathtt{Source}_i$ and $B = \mathtt{DK} \cap \mathtt{Target}$, it comes that the $\dotplus^{d}$-adaptation of SOURCE to solve Target gives COMPL_TARGET such that

$$\mathtt{COMPL\_TARGET} = \bigcup_{i, \delta_i = \Delta} \mathtt{ComplTarget}_i$$

where $\mathtt{ComplTarget}_i$ is the result of the $\dotplus^{d}$-adaptation of $\mathtt{Source}_i$ to solve Target.

First, let us consider that there is only one i such that $\delta_i = \Delta$. Then $\mathtt{COMPL\_TARGET} = \mathtt{ComplTarget}_i$. Therefore if the retrieval process aims at selecting the source case $\mathtt{Source}_i \in \mathtt{CaseBase}$ that minimizes $d(\mathtt{DK} \cap \mathtt{Source}_i, \mathtt{DK} \cap \mathtt{Target})$ then

$$\mathtt{CBR}(\{\mathtt{SOURCE}\}, \mathtt{Target}) = \mathtt{CBR}(\mathtt{CaseBase}, \mathtt{Target}) \quad (8)$$

Now, let us consider that there are ex aequo source cases for such a retrieval process: there are several source cases Source such that $d(\mathtt{DK} \cap \mathtt{Source}, \mathtt{DK} \cap \mathtt{Target}) = \Delta$. Then the equation (8) still holds if the two following modifications are made:

- Retrieval returns the set S of source cases Source minimizing $d(\mathtt{DK} \cap \mathtt{Source}, \mathtt{DK} \cap \mathtt{Target})$;
- Adaptation first performs a $\dotplus^{d}$-adaptation of each $\mathtt{Source} \in S$ and then takes the union of the results.

Therefore, if the distance d used for $\dotplus^{d}$-adaptation is also used for the retrieval process as it is described above, the whole CBR inference can be specified by $\dotplus^{d}$ and DK: DK is the "static" knowledge (stating that some case instances are not appropriate) and d is the "dynamic" knowledge about the modification from a case instance to another one. This can be linked to the general principle of "adaptation-guided retrieval" [20] stating that the adaptation knowledge should be used during retrieval: a source case must be preferred to another one if the former requires less “adaptation effort” (this adaptation effort being measured thanks to d for $\dotplus^d$-based CBR).

6 CONCLUSION

Case-based reasoning systems use similarity (usually in the form of a similarity measure or a distance). This is obvious for the retrieval of a case similar to the target case but this paper shows how it can be used for adaptation: an important class of revision operators is based on distances. Indeed, $\dotplus^d$-based adaptation can be reformulated as the process of selecting the case instances that are the closest ones to the source case, in the metric space $(\mathcal{U}, d)$, with constraints given by DK.

This approach to adaptation has a good level of generality since it captures some other approaches to adaptation (as shown for rule-based adaptation). However, even if revision-based adaptation would capture all the approaches to adaptation (and we do not claim that it is the case —not yet), it would not close the investigations about adaptation in CBR. Indeed, a revision operator is parametrized by a topology (usually a distance) and the issue of the choice of an appropriate topology, is far from being completely addressed. In fact, the choice of this topology is an adaptation knowledge acquisition issue and an important aspect of the research on revision-based adaptation is to have related this topology of the case universe with the adaptation knowledge.

ACKNOWLEDGEMENTS

The authors wish to thank the reviewers for their helpful remarks that have been useful for improving the quality of this paper.

REFERENCES

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson, ‘On the Logic of Theory Change: partial meet functions for contraction and revision’, *Journal of Symbolic Logic*, **50**, 510–530, (1985).
[2] A. Blansché, J. Cojan, V. Dufour Lussier, J. Lieber, P. Molli, E. Nauer, H. Skaf Molli, and Y. Toussaint, ‘TAAABLE 3: Adaptation of ingredient quantities and of textual preparations’, in *18h International Conference on Case-Based Reasoning - ICCBR 2010, "Computer Cooking Contest" Workshop Proceedings*, (2010).
[3] J. G. Carbonell, ‘Learning by analogy: Formulating and generalizing plans from past experience’, in *Machine Learning, An Artificial Intelligence Approach*, ed., R. S. Michalski and J. G. Carbonell and T. M. Mitchell, chapter 5, 137–161, Morgan Kaufmann, Inc., (1983).
[4] J. G. Carbonell, ‘Derivational analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition’, in *Machine Learning*, volume 2, chapter 14, 371–392, Morgan Kaufmann, Inc., (1986).
[5] J. Cojan and J. Lieber, ‘Belief Merging-based Case Combination’, in *Case-Based Reasoning Research and Development (ICCBR 2009)*, pp. 105–119, (2009).
[6] Julien Cojan and Jean Lieber, ‘An Algorithm for Adapting Cases Represented in $\mathcal{ALC}$’, in *22th Internationational Joint Conference on Artificial Intelligence*, Barcelone Espagne, (07 2011).
[7] J.-F. Condotta, S. Kaci, P. Marquis, and N. Schwind, ‘A Syntactical Approach to Qualitative Constraint Networks Merging’, in *Proc. of the 17th LPAR (Logic for Programming, Artificial Intelligence and Reasoning)*, pp. 233–247, (2010).
[8] S. Craw, N. Wiratunga, and R. C. Rowe, ‘Learning adaptation knowledge to improve case-based reasoning’, *Artificial Intelligence*, **170**(16-17), 1175–1192, (2006).
[9] M. Dalal, ‘Investigations into a theory of knowledge base revision: Preliminary report’, in *AAAI*, pp. 475–479, (1988).
[10] M. d’Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary, ‘Case Base Mining for Adaptation Knowledge Acquisition’, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI’07)*, ed., M. M. Veloso, pp. 750–755. Morgan Kaufmann, Inc., (2007).
[11] V. Dufour-Lussier, J. Lieber, F. Le Ber, and L. Martin, ‘Adapting spatial and temporal cases’, in *International Conference on Case-Based Reasoning (ICCBR)*, Lyon, France, (2012). To appear.
[12] N. Karmarkar, ‘A new polynomial-time algorithm for linear programming’, *Combinatorica*, **4**(4), 373–396, (1984).
[13] H. Katsuno and A. Mendelzon, ‘Propositional knowledge base revision and minimal change’, *Artificial Intelligence*, **52**(3), 263–294, (1991).
[14] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, Inc., 1993.
[15] S. Konieczny, J. Lang, and P. Marquis, ‘DA^2 merging operators’, *Artificial Intelligence*, **157**(1-2), 49–79, (2004).
[16] J. Lieber, ‘Application of the Revision Theory to Adaptation in Case-Based Reasoning: the Conservative Adaptation’, in *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR-07)*, Lecture Notes in Artificial Intelligence 4626, 239–253, Springer, Belfast, (2007).
[17] J. Lieber and A. Napoli, ‘Correct and Complete Retrieval for Case-Based Problem-Solving’, in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98), Brighton, United Kingdom*, ed., Henri Prade, pp. 68–72, (1998).
[18] K. Maximini, R. Maximini, and R. Bergmann, ‘An investigation of generalized cases’, in *Proceedings of the 5th International Conference on Case Base Reasoning (ICCBR’03)*, eds., K. D. Ashley and D. Bridge, volume 2689 of *LNAI*, pp. 261–275, Trondheim, Norway, (June 2003). Springer.
[19] C. K. Riesbeck and R. C. Schank, *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.
[20] B. Smyth and M. T. Keane, ‘Using adaptation knowledge to retrieve and adapt design cases’, *Knowledge-Based Systems*, **9**(2), 127–135, (1996).

Cautious analogical-proportion based reasoning using qualitative conceptual relations

Steven Schockaert[1] and **Henri Prade**[2]

Abstract. Propositional rule bases may be incomplete in the sense that some situations of interest are not explicitly covered by any of their rules. While logical deduction does not produce meaningful results in such a case, a variety of methods have been proposed to derive plausible conclusions about a given situation, by comparing it with similar or analogous situations that are explicitly covered by available rules. Most of these methods, however, rely on the availability of quantitative information which may be difficult to obtain and/or justify. In this paper, we therefore propose a form of commonsense reasoning which remains at the qualitative level. In particular, we use qualitative spatial relations between geometric representations of properties to encode how they are conceptually related, essentially corresponding to a weaker version of analogical proportions. A commonsense inference relation is then obtained by identifying a rule base with a mapping between two geometric spaces, and making assumptions about the regularity of this mapping.

1 Introduction

Many domains make use of a large number of labels to categorize instances. In the domain of music, for example, labels for describing different genres abound, ranging from coarse labels such as *classical music* or *pop music*, to fine-grained labels such as *lo-fi*, *doom metal*, or *vocal jazz*. As another example, consider the domain of wines, and the following rules:

$$\textit{chianti} \rightarrow \textit{low-tannins} \wedge \textit{medium-body} \quad (1)$$

$$\textit{merlot} \rightarrow (\textit{low-tannins} \vee \textit{mid-tannins}) \wedge \textit{medium-body} \quad (2)$$

Given the large number of available labels, rule bases about domains such as music genres or wines are not likely to be complete. For example, assume that we have no rules about *barbera* wine. In such a case, logical deduction cannot tell us anything about the amount of tannins in *barbera*. On the other hand, if we know from experience that *barbera* tastes quite similar to *chianti*, we may conclude that the amount of tannins in *barbera* is not likely to be high. To formalize this form of commonsense reasoning, a variety of similarity based reasoning have already been proposed [2, 15, 7, 11, 3, 17]. The intuition is usually that given a rule $\alpha \rightarrow \beta$ and a fact α^*, the more similar α is to α^*, the more likely it is that a situation similar to β holds.

Although the idea of similarity based reasoning is important in understanding human reasoning, and although it has enabled a large number of applications (including the work on fuzzy rule based systems and case based reasoning), it does not offer a fully satisfactory solution to the problem at hand, because of problems such as:

- Where do the similarity degrees come from and what do they mean?
- How similar should α and α^* be to say something meaningful about the similarity between the conclusion β^*, and the rule consequent β?
- What exactly can we derive about the similarity between β and β^*?

While specific applications may give specific answers to these questions, we believe that a more qualitative method is needed to handle the problem of incomplete rule bases in general. In [9], we proposed such a qualitative method for completing rule bases based on the notion of analogical proportion. An analogical proportion $a : b :: c : d$ expresses that "a is related to b as c is related to d". Possible examples are:

chick : *chicken* :: *kitten* : *cat*

loft : *penthouse* :: *cottage* : *mansion*

hard-rock : *progressive-rock* :: *heavy-metal* : *progressive-metal*

More formally, if A, B, C and D are the set of features exhibited by a, b, c and d, the analogical proportion $a : b :: c : d$ is said to hold when $A \setminus B = C \setminus D$ and $B \setminus A = D \setminus C$ [1]. In [9], we suggested to complete rule bases using the assumption that that when each of the corresponding arguments of 4 rules are in an analogical proportion then also the conclusions of these 4 rules should form an analogical proportion. More precisely, given three rules $a_1 \wedge ... \wedge a_k \rightarrow a$, $b_1 \wedge ... \wedge b_k \rightarrow b$, $c_1 \wedge ... \wedge c_k \rightarrow c$ and the premises $d_1, ..., d_k$ such that the analogical proportions $a_i : b_i :: c_i : d_i$ holds for every i, [9] suggests to add the rule $d_1 \wedge ... \wedge d_k \rightarrow d$ to the knowledge base, where $X = d$ is the unique solution which makes $a : b :: c : X$ an analogical proportion, if one exists.

Analogical-proportion based reasoning eliminates the need for degrees, and can often suggest answers in situations where similarity based reasoning cannot. However, from a practical point of view, in many domains it may be hard to find four-tuples of properties which form a perfect analogical proportion, and when analogical proportions only hold approximately, the method from [9] may not be sufficiently cautious. To make analogical-proportion based reasoning more robust in such a case, in this paper we further develop the ideas from [13] on *interpolating and extrapolating rules* using qualitative knowledge.

The paper is structured as follows. First, in Section 2, we focus on the idea of using *betweenness* to interpolate rules, essentially providing a qualitative counterpart to similarity based reasoning. Then,

[1] Cardiff University, UK, email: s.schockaert@cs.cardiff.ac.uk
[2] Université Paul Sabatier, IRIT, CNRS, Toulouse, France, email: prade@irit.fr

in Section 3, we turn our attention to the idea of *analogical change* for extrapolating rules. Subsequently, in Section 4, we discuss some of the limitations of the present approach, as well as some ideas to address them in future work. In particular, we explore some ideas to obtain the required information about betweenness and analogical change, we discuss the problem of introducing inconsistencies when making interpolative or extrapolative inferences, and we touch on the idea of applying interpolative and extrapolative reasoning in a non-monotonic setting.

2 Interpolation

Syntax

At the syntactic level, the idea of interpolating rules corresponds to the following inference rule:

$$\frac{\begin{array}{l}\alpha_1 \rightarrow \beta_1\\ \alpha_2 \rightarrow \beta_2\end{array}}{\alpha_1 \bowtie \alpha_2 \rightarrow \beta_1 \bowtie \beta_2} \tag{3}$$

where we write $\alpha_1 \bowtie \alpha_2$ for the disjunction of all formulas that are conceptually between α_1 and α_2. In the example from the introduction, for instance, *chianti* $\bowtie$ *merlot* would be the disjunction of all wines whose taste is between that of *chianti* and *merlot*, whereas we may consider that

$$(\mathit{lt} \wedge \mathit{mb}) \bowtie ((\mathit{lt} \vee \mathit{mt}) \wedge \mathit{mb}) \equiv (\mathit{lt} \vee \mathit{mt}) \wedge \mathit{mb} \tag{4}$$

where we have abbreviated the labels for the ease of presentation (e.g. *lt* stands for *low-tannins*).

In practice, it may be difficult or even impossible to characterize $\alpha_1 \bowtie \alpha_2$ and $\beta_1 \bowtie \beta_2$ using the available labels and the usual propositional connectives. For example, while we may know that *barbera* is between *chianti* and *merlot*, we may not necessarily be able to enumerate all such wines. Even worse, sometimes the available labels make it impossible to exactly characterize $\alpha_1 \bowtie \alpha_2$. For instance, let α_1 = *3-bedroom-apartment* and α_2 = *penthouse*, then we may wonder whether a *loft* should be included in the disjunction $\alpha_1 \bowtie \alpha_2$, i.e. whether *loft* $\rightarrow$ *3-bedroom-apartment* $\bowtie$ *penthouse* holds. A loft with three bedrooms can be considered intermediate between a 3 bedroom apartment and a 3 bedroom penthouse, but for a loft with fewer bedrooms this is harder to justify. In other words, while some lofts are conceptually between 3 bedroom apartments and penthouses, this does not hold for all lofts. If the language does not contain a label for 3 bedroom lofts, we may therefore not be able to precisely characterize *3-bedroom-apartment*$\bowtie$*penthouse*.

Because of this observation, in practice we are left with approximating $\alpha_1 \bowtie \alpha_2$. In particular, we assume that we have access to rules of the form

$$\alpha^* \rightarrow \alpha_1 \bowtie \alpha_2 \qquad\qquad \beta_1 \bowtie \beta_2 \rightarrow \beta^*$$

which indicate, respectively, that at least all situations covered by α^* are conceptually between α_1 and α_2, and that all situations which are conceptually between β_1 and β_2 satisfy β^*.

Example 1 *From* (1) *and* (2) *we derive using interpolation:*

$$\mathit{chianti} \bowtie \mathit{merlot} \rightarrow (\mathit{lt} \wedge \mathit{mb}) \bowtie ((\mathit{lt} \vee \mathit{mt}) \wedge \mathit{mb})$$

Considering the equivalence in (4)*, and the rule*

$$\mathit{barbera} \rightarrow \mathit{chianti} \bowtie \mathit{merlot}$$

we find using classical deduction that

$$\mathit{barbera} \rightarrow (\mathit{lt} \vee \mathit{mt}) \wedge \mathit{mb}$$

Notice how the symbol $\bowtie$ is essentially treated as a binary modality. We assume this modality to be reflexive and symmetric in the sense that

$$\alpha \bowtie \alpha \equiv \alpha$$
$$\alpha \bowtie \beta \equiv \beta \bowtie \alpha$$

for any propositional formulas α and β. We moreover assume that α and β themselves are "between α and β", i.e.

$$\alpha \vee \beta \rightarrow \alpha \bowtie \beta$$

In practice, we may only have information about the betweenness of atoms (i.e. individual labels) and not about the betweenness of more complex propositional formulas. We may consider, for instance, the following inference rule to lift betweenness for atoms to betweenness for formulas:

$$\frac{\begin{array}{l}\alpha \rightarrow \alpha_1 \bowtie \alpha_2\\ \beta \rightarrow \beta_1 \bowtie \beta_2\end{array}}{(\alpha \vee \beta) \rightarrow (\alpha_1 \vee \beta_1) \bowtie (\alpha_2 \vee \beta_2)} \tag{5}$$

$$\frac{\alpha_1 \bowtie \alpha_2 \rightarrow \alpha}{(\alpha_1 \vee \beta) \bowtie (\alpha_2 \vee \beta) \rightarrow (\alpha \vee \beta)} \tag{6}$$

$$\frac{\begin{array}{l}\alpha \rightarrow \alpha_1 \bowtie \alpha_2\\ \alpha, \alpha_1 \text{ and } \alpha_2 \text{ are "logically independent" from } \beta\end{array}}{(\alpha \wedge \beta) \rightarrow (\alpha_1 \wedge \beta) \bowtie (\alpha_2 \wedge \beta)} \tag{7}$$

$$\frac{\begin{array}{l}\alpha_1 \bowtie \alpha_2 \rightarrow \alpha\\ \beta_1 \bowtie \beta_2 \rightarrow \beta\end{array}}{(\alpha_1 \wedge \beta_1) \bowtie (\alpha_2 \wedge \beta_2) \rightarrow (\alpha \wedge \beta)} \tag{8}$$

Note that because $\bowtie$ is symmetric, from the premises of (5) we can also derive

$$(\alpha \vee \beta) \rightarrow (\alpha_1 \vee \beta_2) \bowtie (\alpha_2 \vee \beta_1)$$
$$(\alpha \vee \beta) \rightarrow (\alpha_2 \vee \beta_1) \bowtie (\alpha_1 \vee \beta_2)$$
$$(\alpha \vee \beta) \rightarrow (\alpha_2 \vee \beta_2) \bowtie (\alpha_1 \vee \beta_1)$$

and similar for (6)–(8).

To see why these inference rules make sense, and to elucidate the informal requirement of logical independence, it is useful to consider the notion of interpolation at the semantic level.

Semantics

We characterize interpolation at the semantic level using the idea of conceptual spaces [4]. Specifically, we assume that the meaning of every label can be represented as a convex region in some geometric space, whose dimensions correspond to elementary cognitive features; they are usually called "quality dimensions" in this context.

In the case of labels referring to wines, for instance, there would be quality dimensions corresponding to colour (e.g. three dimensions, encoding hue, saturation and intensity), dimensions corresponding to the texture of the wine, its taste, smell, etc. The points of the conceptual space would then correspond to specific instances, while regions correspond to categories. Essentially, the region representing a category (e.g. *chianti*) corresponds to the points which are closest to the prototypes of that category [5], which is why such regions are naturally convex. For simplicity, we assume that conceptual spaces are Euclidean spaces.

Given this geometric setting, betweenness can naturally be characterized: we say that a category b is between a and c, if some point of the region corresponding with b is between some point of the region corresponding with a and some point of the region corresponding with c. To make this more precise, let us write $\textit{reg}(\alpha)$ for the conceptual space representation of a propositional formula α, where $\textit{reg}(\alpha \wedge \beta) = \textit{reg}(\alpha) \cap \textit{reg}(\beta)$ and $\textit{reg}(\alpha \vee \beta) = \textit{reg}(\alpha) \cup \textit{reg}(\beta)$. In this paper, we do not explicitly consider negation, and rather assume that propositions are grouped in domains of pairwise disjoint attributes. In the wine example, we may for instance consider the domain $A = \{\textit{low-tannins}, \textit{mid-tannins}, \textit{high-tannins}\}$. We assume there are implicit constraints that enforce the pairwise disjointness of attributes from the same domain. Sometimes, we do use the notation $\neg a$ as a shorthand for domains $A = \{a, \neg a\}$ with only two elements.

We have $\alpha \rightarrow \beta \bowtie \gamma$ iff

$$\forall q \in \textit{reg}(\alpha) \,.\, \exists p \in \textit{reg}(\beta), r \in \textit{reg}(\gamma), \lambda \in [0,1] \,.\, \overrightarrow{pq} = \lambda \cdot \overrightarrow{pr}$$

and $\beta \bowtie \gamma \rightarrow \alpha$ iff

$$\forall p \in \textit{reg}(\beta), r \in \textit{reg}(\gamma), \lambda \in [0,1] \,.\, p + \lambda \cdot \overrightarrow{pr} \in \textit{reg}(\alpha)$$

where the points $p + \lambda \cdot \overrightarrow{pr}$ for $\lambda \in [0,1]$ are exactly the points which are between p and r. From these characterizations it is easy to verify that inference rules (5), (6) and (8) are indeed valid. The argument for (7) is a bit more subtle. The intuition is that because of the assumption of logical independence, we can see the underlying conceptual space as a Cartesian product $\mathcal{C}_1 \times \mathcal{C}_2$ such that $\textit{reg}(\alpha)$, $\textit{reg}(\alpha_1)$ and $\textit{reg}(\alpha_2)$ are all of the form $X \times \mathcal{C}_2$ (i.e. the dimensions in $\mathcal{C}_2$ are irrelevant for describing the categories α, α_1 and α_2), whereas $\textit{reg}(\beta)$ is of the form $\mathcal{C}_1 \times Y$ (i.e. the dimensions in $\mathcal{C}_1$ are irrelevant for describing the category β).

If α is inconsistent (e.g. because it is the conjunction of two pairwise disjoint attributes), then $\textit{reg}(\alpha) = \emptyset$. In such a case, no point is between a point of $\textit{reg}(\alpha)$ and a point of any other region $\textit{reg}(\beta)$. Accordingly, we assume that

$$\bot \bowtie \beta \equiv \beta \bowtie \bot \equiv \bot$$

To describe the interpolation process itself, i.e. inference rule (3), assume that a propositional rule base R is available, containing negation-free rules as before. The antecedent α of a rule corresponds to a region $\textit{reg}_1(\alpha)$ in some conceptual space $\mathcal{C}_1$ (typically corresponding to the Cartesian product of more elementary conceptual spaces). Similarly, the consequent β of a rule corresponds to a region $\textit{reg}_2(\beta)$ in a conceptual space $\mathcal{C}_2$. We can thus view the rule base R as a mapping f from regions of $\mathcal{C}_1$ to regions of $\mathcal{C}_2$. A rule $\alpha^* \rightarrow \beta^*$ can then be derived from the rule base R using classical deduction iff $f(\textit{reg}_1(\alpha^*)) \subseteq \textit{reg}_2(\beta^*)$.

By making certain meta-assumptions about the relationship between the conceptual spaces $\mathcal{C}_1$ and $\mathcal{C}_2$, we may be able to refine the mapping f. In particular, in many cases $\mathcal{C}_2$ will be a subspace of $\mathcal{C}_1$ (i.e. the quality dimensions that are needed to describe the labels in the consequents of rules are a subset of those needed to describe the antecedents). In such a case, f is the approximation of a linear mapping from points of $\mathcal{C}_1$ to points of $\mathcal{C}_2$. We can then refine f to a mapping $\widehat{f}$ such that $f(X) \setminus \widehat{f}(X)$ are all points from $\mathcal{C}_2$ that could never be obtained by a linear mapping from $\mathcal{C}_1$ which is consistent with f. It can then be shown that $\widehat{f}(\textit{reg}_1(\alpha^*)) \subseteq \textit{reg}_2(\beta^*)$ iff $\alpha^* \rightarrow \beta^*$ can be derived from R using inference rule (3), and (5)-(8) together with classical deduction. *In other words, at the semantic level, the interpolative inference rule* (3) *corresponds to an assumption of regularity.* We refer to [13] for more details.

Finally, note that interpolative reasoning can also be formalized in terms of analogical proportions, by considering that b is between a and c iff the analogical proportion $a : b :: b : c$ holds. For interpolative inferences, the method outlined in this paper yields more cautious conclusions than the method from [9], which is intuitively due to the fact that $a : b :: b : c$ means that b is between a and c *and the distance between a and b is identical to the distance between b and c*. In the next section, we present a method for extrapolative reasoning, which is again more cautious than the method from [9].

3 Extrapolation

Geometrically, analogical proportions intuitively correspond to the idea of a parallelogram, indicating that the direction of change to go from a to b is parallel to the direction of change from c to d, and that the amount of change is identical. As this latter amount is difficult to quantify, we will use a more qualitative approach, and restrict our attention to the direction of change. In particular, we write $\gamma \rhd \langle \alpha, \beta \rangle$ for the disjunction of all propositional formulas which correspond to situations that differ from some situation satisfying γ in the way as some situation satisfying β differs from a situation satisfying α. More precisely, at the semantic level we have $\delta \rightarrow \alpha \rhd \langle \beta, \gamma \rangle$ iff for every s in $\textit{reg}(\delta)$ there exist p in $\textit{reg}(\alpha)$, q in $\textit{reg}(\beta)$, r in $\textit{reg}(\gamma)$ and $\lambda \geq 0$ such that

$$\overrightarrow{rs} = \lambda \cdot \overrightarrow{pq}$$

Example 2 *Let a partitioning of house sizes be given by* $\{\textit{very-small}, \textit{small}, \textit{medium}, \textit{large}, \textit{very-large}\}$*, then we have*

$$\textit{medium} \rhd \langle \textit{very-small}, \textit{large} \rangle \equiv \textit{medium} \vee \textit{large} \vee \textit{very-large}$$

Indeed, the change from very-small *to* large *denotes an increase in size. Therefore the house sizes compatible with medium* $\rhd$ $\langle \textit{very-small}, \textit{large} \rangle$ *are those that are at least as large as* medium*.*

As in the case of betweenness, when we move from unidimensional to multi-dimensional domains, it is often not possible to provide precisely characterize formulas of the form $\gamma \rhd \langle \alpha, \beta \rangle$*. For example, we may assume*

$$\textit{prog-metal} \rightarrow \textit{heavy-metal} \rhd \langle \textit{hard-rock}, \textit{prog-rock} \rangle$$

without being able to precisely define all music genres that differ from heavy-metal *as* prog-rock *differs from* hard-rock *using the labels that are available to us.*

We obtain a form of extrapolative reasoning by assuming that analogical changes in the antecedent of rules should lead to analogical changes in the consequent:

$$\frac{\begin{array}{l}\alpha_1 \rightarrow \beta_1\\ \alpha_2 \rightarrow \beta_2\\ \alpha_3 \rightarrow \beta_3\end{array}}{\alpha_3 \rhd \langle \alpha_1, \alpha_2 \rangle \rightarrow \beta_3 \rhd \langle \beta_1, \beta_2 \rangle} \qquad (9)$$

To lift information about analogical changes between atomic labels to analogical changes of propositional formulas, the following inference rules are available to us:

$$\frac{\begin{array}{l}\alpha \rightarrow \alpha_3 \rhd \langle \alpha_1, \alpha_2 \rangle \\ \beta \rightarrow \beta_3 \rhd \langle \beta_1, \beta_2 \rangle\end{array}}{(\alpha \vee \beta) \rightarrow (\alpha_3 \vee \beta_3) \rhd \langle (\alpha_1 \vee \beta_1), (\alpha_2 \vee \beta_2) \rangle} \tag{10}$$

$$\frac{\begin{array}{l}\alpha_3 \rhd \langle \alpha_1, \alpha_2 \rangle \rightarrow \alpha \\ \beta_3 \rhd \langle \beta_1, \beta_2 \rangle \rightarrow \beta\end{array}}{(\alpha_3 \vee \beta_3) \rhd \langle (\alpha_1 \vee \beta_1), (\alpha_2 \vee \beta_2) \rangle \rightarrow (\alpha \vee \beta)} \tag{11}$$

$$\frac{\begin{array}{l}\alpha \rightarrow \alpha_3 \rhd \langle \alpha_1, \alpha_2 \rangle \\ \alpha, \alpha_1 \text{ and } \alpha_2 \text{ are "logically independent" from } \beta \text{ and } \gamma\end{array}}{(\alpha \wedge \gamma) \rightarrow (\alpha_3 \wedge \gamma) \rhd \langle (\alpha_1 \wedge \beta), (\alpha_2 \wedge \beta) \rangle} \tag{12}$$

$$\frac{\begin{array}{l}\alpha_3 \rhd \langle \alpha_1, \alpha_2 \rangle \rightarrow \alpha \\ \beta_3 \rhd \langle \beta_1, \beta_2 \rangle \rightarrow \beta\end{array}}{(\alpha_3 \wedge \beta_3) \rhd \langle (\alpha_1 \wedge \beta_1), (\alpha_2 \wedge \beta_2) \rangle \rightarrow (\alpha \wedge \beta)} \tag{13}$$

The inference rules (10)–(13) are justified using a geometric argument, similar as for betweenness. The extrapolation principle (9) itself can again be shown to be valid when the rule base approximates a linear mapping between conceptual spaces.

Example 3 *Consider the following rule base about houses:*

$$\textit{large} \wedge \textit{detached} \rightarrow \textit{comf} \vee \textit{lux} \tag{14}$$
$$\textit{large} \wedge \textit{row-house} \rightarrow \textit{comf} \tag{15}$$
$$\textit{small} \wedge \textit{detached} \rightarrow \textit{bas} \vee \textit{comf} \tag{16}$$

which defines the comfort level (basic, comfortable, luxurious) of a house, based on its size (small, medium, large) and type (detached, row-house, semi-detached). From the extrapolation principle (9) *we find*

$$\big((\textit{small} \wedge \textit{det}) \rhd \langle (\textit{large} \wedge \textit{det}), (\textit{large} \wedge \textit{rh}) \rangle\big) \rightarrow \big((\textit{bas} \vee \textit{comf}) \rhd \langle (\textit{comf} \vee \textit{lux}), (\textit{comf} \vee \textit{comf}) \rangle\big) \tag{17}$$

Using (12) *and the fact that $\beta \rightarrow \alpha \rhd \langle \alpha, \beta \rangle$ for any α and β, we find*

$$(\textit{small} \wedge \textit{rh}) \rightarrow (\textit{small} \wedge \textit{det}) \rhd \langle (\textit{large} \wedge \textit{det}), (\textit{large} \wedge \textit{rh}) \rangle \tag{18}$$

where we have again abbreviated some labels. From

$$\textit{bas} \rhd \langle \textit{comf}, \textit{comf} \rangle \rightarrow \textit{bas}$$
$$\textit{comf} \rhd \langle \textit{lux}, \textit{comf} \rangle \rightarrow (\textit{bas} \vee \textit{comf})$$

we find using (11)*:*

$$(\textit{bas} \vee \textit{comf}) \rhd \langle (\textit{comf} \vee \textit{lux}), (\textit{comf} \vee \textit{comf}) \rangle \rightarrow (\textit{bas} \vee \textit{comf}) \tag{19}$$

Combining (17)–(19)*, we find*

$$(\textit{small} \wedge \textit{row-house}) \rightarrow (\textit{bas} \vee \textit{comf}) \tag{20}$$

Intuitively, from the rule base (14)–(16) *we derive that detached houses are more comfortable than row houses, hence a small row house can not be more comfortable than a small detached house.*

If we know consider that

$$\textit{semi-detached} \rightarrow \textit{row-house} \bowtie \textit{detached}$$

Using the interpolation principle (3) *we can derive from* (16) *and* (20) *that*

$$(\textit{small} \wedge \textit{semi-detached}) \rightarrow (\textit{bas} \vee \textit{comf})$$

i.e. since semi-detached houses are intermediate between detached houses and row houses, their comfort level should be intermediate as well.

4 Discussion

In this section, we discuss a number of obstacles to implementing the ideas of interpolation and extrapolation in practice, and provide some ideas on how to circumvent them.

4.1 Obtaining conceptual relations

Regarding the applicability of our approach, an important question is how the required relational knowledge about conceptual spaces can be obtained. Depending on the specific application, different options may be available.

In some domains, it is feasible to manually encode a complete qualitative description of a conceptual space. Most notably, this is the case for conceptual spaces that are unidimensional, for which it suffices to provide a ranking of the labels of interest. For instance, a conceptual space of housing sizes may be described by encoding that

$$\textit{very-small} < \textit{small} < \textit{medium} < \textit{large} < \textit{very-large}$$

From this description, we immediately obtain that e.g. $\textit{small} \rightarrow \textit{very-small} \bowtie \textit{medium}$.

A second possibility is to extract conceptual relations from natural language. In [16], for instance, the idea of latent relational analysis was introduced, with the aim of finding analogical proportions. The main idea is that two pairs of words are likely to be related analogously, i.e. form an analogical proportion, when the lexical contexts in which they co-occur are similar. For example, the words *kitten* and *cat* are found in sentences such as "a kitten is a young cat", while the words *chick* and *chicken* are found in sentences such as "a chick is a young chicken". From such observations, the analogical proportion *kitten* : *cat* :: *chick* : *chicken* can be discovered. Another technique for discovering analogical proportions from the web was proposed in [8], estimating the strength of analogical proportions by converting co-occurrence statistics using Kolmogorov information theory.

If sufficient information is available about instances of concepts or properties, several data-driven approaches can be used, which directly take advantage of the geometric nature of the relations of interest. For instance, [5] suggests to start from pairwise similarity judgements between instances, and use multi-dimensional scaling to obtain coordinates for them in a Euclidean space. Representations of concepts can then be obtained by determining the corresponding Voronoi tessellation, after which the conceptual relations of interest can be evaluated by straightforward geometric calculations. In [12], the feasibility of such an approach was demonstrated in the domain of music genres, using similarity judgements that were obtained indirectly using user-contributed meta-data from the website last.fm[3]. Rather than starting from similarity judgements, [14] suggests an approach based on singular value decomposition (SVD), which is a form of dimensionality reduction. Translated to our setting, the approach would start from a matrix where rows correspond to instances and columns correspond to binary features that these instances may or may not have. Instances are then represented in a high-dimensional space with one dimension for each feature, and coordinates are either 0 or 1, depending on whether the instance has the corresponding feature. Using SVD, a linear transformation is then determined which maps this high-dimensional space onto a space of lower dimension, with real-valued coordinates.

Note that these data-driven approaches essentially use quantitative information to obtain a qualitative representation. One reason for not using a purely quantitative approach is that the available data is not likely to be sufficiently informative to build accurate conceptual space representations, but still allows to discover information about qualitative relations between regions. A second reason is that geometric calculations, such as determining convex hulls or Voronoi tessellations, are computationally expensive in high-dimensional spaces. Even the space required for representing polytopes is exponential in the number of dimensions. When all we are interested in are spatial relations such as betweenness and parallelism, we can avoid to actually build the conceptual space, using a linear programming approach that was proposed in [12].

4.2 Handling inconsistencies

As mentioned in Sections 2 and 3, we see a rule base as an incomplete approximation of a mapping between two conceptual spaces (or between two Cartesian products of conceptual spaces), and the interpolative and extrapolative inference principles are tied to assumptions on the regularity of this mapping. In particular, both principles are valid when this mapping is linear. For interpolation, it even suffices that the mapping is monotonic. If these regularity assumptions are met, we are guaranteed that interpolation and extrapolation will never introduce logical inconsistencies.

Relaxing the linearity assumption

In practice, on the other hand, the regularity assumptions may not hold. For example, consider the following rules, which contain information about the amount of traffic (*light*, *moderate*, *heavy*) at different times during the day:

$$\textit{morning} \rightarrow \textit{heavy-traffic} \tag{21}$$

$$\textit{mid-day} \rightarrow \textit{moderate-traffic} \tag{22}$$

$$\textit{evening} \rightarrow \textit{heavy-traffic} \tag{23}$$

Together with the constraint that *light-traffic*, *moderate-traffic* and *heavy-traffic* are mutually exclusive properties. Using interpolation, and the assumption that

$$\textit{mid-day} \rightarrow \textit{morning} \bowtie \textit{evening}$$

we then derive the rule

$$\textit{mid-day} \rightarrow \textit{heavy-traffic}$$

[3] http://www.last.fm

which is in conflict with (22). This can be explained due to a failure of the monotonicity assumption. In particular, the underlying mapping from different times of the day to different amounts of traffic is not a projection to a lower-dimensional conceptual space, but rather expresses an observation about the world. We can contrast such *phenomenological rules* with *conceptual rules*, which link concepts to their inherent properties (as well as super-concepts). For the latter type of rules, the mapping between conceptual spaces usually is a projection from one space onto a lower-dimensional sub-space, which trivially satisfies the linearity assumption.

In the case of (21)–(23) the underlying mapping is not even deterministic, in the sense that the exact amount of traffic at e.g. 9 am may vary from day to day (even if we assume that the rule base talks about weekdays in a specific city). Nonetheless, even for rules where the linearity assumption fails, interpolation may still be useful. For instance, suppose we introduce the labels *mid-morning* and *mid-afternoon*, which are between *morning* and *mid-day*, and between *mid-day* and *evening* respectively. From (21) and (22) we may derive

$$\textit{mid-morning} \rightarrow \textit{moderate-traffic} \vee \textit{heavy-traffic}$$

Indeed, while the mapping underlying the rule base may, in principle, be arbitrary, it seems natural to assume that more regular mappings would be more likely, i.e. *we could make the assumption that any completion of the knowledge base should not introduce additional irregularities*. In particular, by identifying irregularities with violations of the monotonicity assumption, this leads to the assumption that the conceptual space $\mathcal{C}_1$ corresponding with the antecedent of the rules can be partitioned in a minimal number of segments, such that the mapping is monotonic over these segments. In the traffic example, we would thus assume that the amount of traffic is monotonically decreasing throughout the morning and monotonically increasing throughout the afternoon. While such conclusions would not be valid in general, they are reasonable to make in absence of any other information. Depending on how the rule base (21)–(23) was obtained, we may also argue that the absence of a rule for *mid-morning* suggests that this case is not special, i.e. that those cases which are irregular in some sense would be more likely to be contained in the rule base.

To avoid inconsistencies, the above view suggests that from a rule base R we should try to identify subsets $R_1, ...R_k$ of rules, such that no inconsistencies arise as long as interpolation is applied to two rules from the same set R_i. To be compatible with the above view, we should moreover insist that when $\alpha \rightarrow \alpha_1 \bowtie \alpha_2$, $(\alpha_1 \rightarrow \beta_1) \in R_i$, $(\alpha_2 \rightarrow \beta_2) \in R_i$ and $(\alpha \rightarrow \beta) \in R$, then we should have that $(\alpha \rightarrow \beta) \in R_i$. In other words, the sub-bases R_i should contain all rules that apply to a given (convex) segment of the conceptual space $\mathcal{C}_1$. In this way, we can ensure that when a new rule $\alpha^* \rightarrow \beta^*$ is derived by interpolation from a sub-base R_i, the rules in R_i are indeed the most relevant ones, i.e. that they are the ones whose antecedent is closest to α^* in some sense. In a similar, but slightly less cautious fashion, we may assume that the mapping underlying the rule base R is piecewise linear, and apply extrapolation locally to the sub-bases $R_1, ..., R_k$.

Restricting to the most salient properties

Another reason why inconsistencies may arise is because the information about betweenness or analogical change is not accurate, or, more fundamentally, because it only takes the most salient properties of objects in the account. For example, when we derive betweenness information for wines from wine-food pairings, it will mainly reflect the taste of the wine, and to a much lesser extent properties such as price. As an additional example, we may consider that *coffeehouses* are conceptually between *bars* and *restaurants*, as both *coffeehouses* and *bars* emphasise drinking rather than eating, while *coffeehouses* generally do serve some food (sandwiches, cakes) as well. Nonetheless, we may consider that

$$\textit{bar} \rightarrow \textit{serves-wine} \tag{24}$$

$$\textit{coffeehouse} \rightarrow \neg\textit{serves-wine} \tag{25}$$

$$\textit{restaurant} \rightarrow \textit{serves-wine} \tag{26}$$

Using interpolation and the assumption

$$\textit{coffeehouse} \rightarrow \textit{bar} \bowtie \textit{restaurant}$$

we derive the rule

$$\textit{coffeehouse} \rightarrow \textit{serves-wine}$$

which is in conflict with the rule base. In this case, the inconsistency is mainly do the fact that the property of serving wine was not considered when asserting that coffeehouses are between bars and restaurants. The most natural way to avoid inconsistencies would then be to avoid applying interpolation to derive conclusions from the domain $A = \{\textit{serves-wine}, \neg\textit{serves-wine}\}$. In absence of any conflicts about attributes from a given domain, we then assume that interpolative and extrapolative conclusions are valid for that domain, an assumption which may need to be revised if additional knowledge became available.

4.3 Typicality

The ideas of interpolation and extrapolation, explored in this paper, and the ideas of non-monotonic reasoning in the sense of [6] serve two rather complementary goals. Whereas the former is concerned with handling missing generic knowledge (i.e. the absence of rules that allow us to derive meaningful conclusions about the situation at hand), the latter allows us to deal with missing factual knowledge (i.e. the absence of a complete description of the situation at hand). Thus it is natural to try to combine both ideas, as illustrated by the next example.

Example 4 *Consider the following set of default rules:*

$$\textit{bird} \mid\!\sim \textit{flies}$$

$$\textit{penguin} \mid\!\sim \textit{bird}$$

$$\textit{penguin} \mid\!\sim \neg\textit{flies}$$

$$\textit{aptenodytes} \mid\!\sim \textit{penguin}$$

$$\textit{eudyptula} \mid\!\sim \textit{penguin}$$

and assume that we also know that

$$\textit{pygoscelis} \rightarrow \textit{aptenodytes} \bowtie \textit{eudyptula} \tag{27}$$

Then we may want to combine the interpolation principle with a form of non-monotonic reasoning to conclude

$$\textit{pygoscelis} \mid\!\sim \neg\textit{flies}$$

The intuition underlying the semantics of default rules such as $\textit{bird} \mid\!\sim \textit{flies}$ is that typical birds fly, but there may be birds that are exceptional and which may not fly. When taking a geometric view, we may assume that each label is represented by two nested regions, where the inner region contains the typical instances of the corresponding concept. Let us write $\textit{typ}(a_1 \wedge ... \wedge a_n)$ for the typical instances of the concept $a_1 \wedge ... \wedge a_n$, where *typ* is treated as a modality. A default rule such as $\textit{bird} \mid\!\sim \textit{flies}$ is then interpreted as the classical rule $\textit{typ}(\textit{bird}) \rightarrow \textit{flies}$. The modality *typ* is assumed to at least satisfy the following axiom

$$\textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow a_1 \wedge ... \wedge a_n$$

expressing that all typical instances of a concept are instances, i.e. at the semantic level $\textit{reg}(\textit{typ}(a_1 \wedge ... \wedge a_n)) \subseteq \textit{reg}(a_1 \wedge ... \wedge a_n)$.

To obtain meaningful inferences, some additional assumptions need to be made on how the formulas $\textit{typ}(a_1 \wedge ... \wedge a_n)$ relate to the corresponding formulas $a_1 \wedge ... \wedge a_n$. To obtain inferences in the spirit of System P [6], we need to assume that the following are valid inferences for the modality *typ*:

$$\frac{\begin{array}{l}\textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow b_1 \wedge ... \wedge b_m \\ \textit{typ}(a_1 \wedge ... \wedge a_n \wedge b_1 \wedge ... \wedge b_m) \rightarrow \gamma\end{array}}{\textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow \gamma}$$

and

$$\frac{\begin{array}{l}\textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow b_1 \wedge ... \wedge b_m \\ \textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow \gamma\end{array}}{\textit{typ}(a_1 \wedge ... \wedge a_n \wedge b_1 \wedge ... \wedge b_m) \rightarrow \gamma}$$

corresponding to the cut rule and rational monotony rule from System P respectively. When $\textit{typ}(a_1 \wedge ... \wedge a_n) \rightarrow b_1 \wedge ... \wedge b_m$, it should be possible to find geometric models in which $\textit{reg}(\textit{typ}(a_1 \wedge ... \wedge a_n)) = \textit{reg}(\textit{typ}(a_1 \wedge ... \wedge a_n \wedge b_1 \wedge ... \wedge b_m))$. Such models are simpler in the sense that a smaller number of distinct regions is needed to explain the semantics of the rules. In other words, as for interpolation and extrapolation, we find that the underlying principle relates to a preference for simpler, or more regular models.

To obtain inferences in the spirit of System Z, we can add Reiter defaults [10] of the form

$$M(\textit{typ}(a_1 \wedge ... \wedge a_n) \equiv a_1 \wedge ... \wedge a_n) \vdash \textit{typ}(a_1 \wedge ... \wedge a_n) \equiv a_1 \wedge ... \wedge a_n$$

i.e. if it is consistent to assume that all instances of $a_1 \wedge ... \wedge a_n$ are typical, then we should do so. Such defaults again express a preference for models with a minimal number of distinct regions, i.e. a preference for simpler models.

Example 5 *Consider again the rule base from Example 4. The only equivalence of the form* $\textit{typ}(\alpha) \equiv \alpha$ *that could introduce inconsistencies is the equivalence* $\textit{typ}(\textit{bird}) \equiv \textit{bird}$ *which would entail both* $\textit{typ}(\textit{penguin}) \rightarrow \neg\textit{flies}$ *and* $\textit{typ}(\textit{penguin}) \rightarrow \textit{flies}$*. Hence, the knowledge base corresponding to the rules from Example 4 is*

$$\textit{typ}(\textit{bird}) \mid\!\sim \textit{flies}$$

$$\textit{penguin} \rightarrow \textit{bird}$$

$$\textit{penguin} \rightarrow \neg\textit{flies}$$

$$\textit{aptenodytes} \rightarrow \textit{penguin}$$

$$\textit{eudyptula} \rightarrow \textit{penguin}$$

from which we can entail

$$\textit{aptenodytes} \rightarrow \neg\textit{flies}$$

$$\textit{eudyptula} \rightarrow \neg\textit{flies}$$

Using the interpolation principle, (27) *and* $\neg\textit{flies} \equiv \neg\textit{flies} \bowtie \neg\textit{flies}$*, we find*

$$\textit{pygoscelis} \mid\!\sim \neg\textit{flies}$$

5 Conclusions

In this paper, we have proposed the use of interpolative and extrapolative inference to complete propositional rule bases, as a more cautious form of analogical-proportion based reasoning. At the semantic level, these forms of inference are based on the assumption that rule bases approximate mappings between conceptual spaces which are regular in some sense. This regularity imposes some constraints on what approximations are possible, which in turn translate to additional rules at the syntactic level. In practical applications, this assumption of regularity may only be partially valid. For this reason, we have discussed a number of ways to avoid the introduction of logical inconsistency when making interpolative or extrapolative inferences. We have also sketched how our ideas could be extended to deal with default rules, by exploiting a geometric view on nonmonotonic reasoning. The geometric nature of our semantics, based on Gärdenfors' idea of conceptual spaces, stands in contrast with traditional models for commonsense reasoning based on possible worlds (and preference orders between them). It can be exploited by data-driven techniques, opening the door for an automated acquisition of commonsense knowledge.

REFERENCES

[1] Meriam Bayoudh, Henri Prade, and Gilles Richard, 'A kolmogorov complexity view of analogy: From logical modeling to experimentations', in *Proceedings of the Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 93–106, (2010).

[2] Didier Dubois and Henri Prade, 'What are fuzzy rules and how to use them', *Fuzzy Sets and Systems*, **84**(2), 169 – 185, (1996).

[3] Didier Dubois, Henri Prade, Francesc Esteva, Pere Garcia, and Lluís Godo, 'A logical approach to interpolation based on similarity relations', *International Journal of Approximate Reasoning*, **17**(1), 1 – 36, (1997).

[4] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*, MIT Press, 2000.

[5] P. Gärdenfors and M.A. Williams, 'Reasoning about categories in conceptual spaces', in *International Joint Conference on Artificial Intelligence*, pp. 385–392, (2001).

[6] Sarit Kraus, Daniel Lehmann, and Menachem Magidor, 'Nonmonotonic reasoning, preferential models and cumulative logics', *Artificial Intelligence*, **44**(1-2), 167–207, (1990).

[7] Jesús Medina, Manuel Ojeda-Aciego, and Peter Vojtás, 'Similarity-based unification: a multi-adjoint approach', *Fuzzy Sets and Systems*, **146**(1), 43 – 62, (2004).

[8] Henri Prade and Gilles Richard, 'Testing analogical proportions with Google using Kolmogorov information theory', in *Proceedings of the 22nd International FLAIRS Conference*, pp. 272–277, (2009).

[9] Henri Prade and Steven Schockaert, 'Completing rule bases in symbolic domains by analogy making', in *Proceedings of the EUSFLAT-LFA Conference*, pp. 928–934, (2011).

[10] R. Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**(1-2), 81 – 132, (1980).

[11] E.H. Ruspini, 'On the semantics of fuzzy logic', *International Journal of Approximate Reasoning*, **5**, 45–88, (1991).

[12] Steven Schockaert and Henri Prade, 'Interpolation and extrapolation in conceptual spaces: A case study in the music domain', in *Proceedings of the 5th International Conference on Web Reasoning and Rule Systems*, pp. 217–231, (2011).

[13] Steven Schockaert and Henri Prade, 'Qualitative reasoning about incomplete categorization rules based on interpolation and extrapolation in conceptual spaces', in *Proceedings of the Fifth International Conference on Scalable Uncertainty Management*, pp. 303–316, (2011).

[14] Robert Speer, Catherine Havasi, and Henry Lieberman, 'Analogyspace: reducing the dimensionality of common sense knowledge', in *Proceedings of the 23rd AAAI Conference on Artificial intelligence*, pp. 548–553, (2008).

[15] R. Sun, 'Robust reasoning: integrating rule-based and similarity-based reasoning', *Artificial Intelligence*, **75**(2), 241–295, (1995).

[16] Peter D. Turney, 'Measuring semantic similarity by latent relational analysis', in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 1136–1141, (2005).

[17] Thomas Vetterlein, 'A logic of the similarity with prototypes and its relationship to fuzzy logic', in *Proceedings of the EUSFLAT-LFA Conference*, pp. 196–202, (2011).

Issues in Analogical Learning over Sequences of Symbols: a Case Study with Named Entity Transliteration.

Philippe Langlais[1]

Abstract. Formal analogies, that is, proportional analogies involving relations at a formal level (e.g. *cordially* is to *cordial* as *appreciatively* is to *appreciative*) have a long history in Linguistics [18]. They can accommodate a wide variety of linguistic data without resorting to *ad hoc* representations [26] and are inherently good at capturing long dependencies between data. Unfortunately, applying analogical learning on top of formal analogy to nowadays large Natural Language Processing (NLP) tasks is very challenging. In this paper, we draw on previous works we conducted and identify some issues that remain to be addressed for formal analogy to stand by itself in the landscape of NLP. As a case study, we monitor our current implementation of analogical learning on a task of transliterating English proper names into Chinese.

1 INTRODUCTION

A proportional analogy is a relationship between four objects [x : y :: z : t], which reads as "x is to y as z is to t". While some works have been proposed for handling semantic relationships [32, 8], we focus in this study on formal proportional analogies (hereafter formal analogies or simply analogies), that is, proportional analogies involving relationships at the formal level, such as [$miracle$: $miraculeux$ :: $fable$: $fabuleux$].

Early work on formal analogies for NLP was devoted to propose computational definitions of proportional analogies. Yvon [34] proposed a definition where a prefixation or a suffixation operation was allowed between forms. In [22], Lepage proposed a richer model allowing at the same time, prefixation, suffixation, as well as infixation operations. His model is characterized in terms of the edit-distance that must verify 4 entities in (formal) analogical relation. Later on, Yvon et al. [36] proposed a model of analogy which generalizes the model of [22] thanks to finite-state machines. In particular, this model can account for inversions (i.e. *Paul gave an apple to Mary* is to *Mary received an apple from Paul* as *Paul gave an letter to Mary* is to *Mary received an letter from Paul*). Stroppa [31] further extended this model to various algebraic structures, among which trees which are ubiquitous in NLP. Also, Miclet et al. [24] built on the definition of [36] and defined the notion of *analogical dissimilarity* on forms. Presumably, allowing near analogies might be of interest in several AI applications. An extension of analogical dissimilarity to tree structures has been recently proposed in [2] .

Another thread of studies is devoted to applications of analogical learning to NLP tasks. Lepage [22] early proposed an analogical model of parsing which uses a treebank (a database of syntactically analyzed sentences). He conducted proof-of-concept experiments. Yvon [34] addressed the task of grapheme-to-phoneme conversion, a problem which continues to be studied thoroughly (e.g. [3]). In [17], the authors address the task of identifying morphologically related word forms in a lexicon, the main task of the MorphoChallenge evaluation campaign [13]. Their approach, which capitalizes on formal analogy to learn relations between words proved to be competitive with state-of-the-art approaches (e.g. [5]) and ranked first on the Finnish language according the EMMA metric (see [28]) which is now the official metric since Morphochallenge 2010. Stroppa and Yvon [31] applied analogical learning to computing morphosyntactic features to be associated with a form (lemma, part-of-speech, and additional features such as number, gender, case, tense, mood, etc.). The performance of the analogical device on the Dutch language was as good as or better than the one reported in [33].

Lepage and Denoual [19] pioneered the application of analogical learning to Machine Translation. Different variants of the system they proposed have been tested in a number of evaluation campaigns (see for instance [21]). Langlais and Patry [14] investigated the more specific task of translating unknown words, a problem simultaneously investigated in [7]. In [16], the authors applied analogical learning to translating terms of the medical domain in different language directions, including some that do not share the same scripts (e.g. Russian/English). The precision of the analogical engine was higher than the one of a state-of-the-art phrase-based statistical engine [12] trained at the character level, but the recall was lower. A simple combination of both systems outperformed significantly both engines. See [27] for a technical discussion of those works. Very recently, Gosme et Lepage [11] investigate the use of formal analogy for smoothing n-gram language models. They report improvements over fair baselines in different languages, but for small training corpora only.

Analogical learning has also been applied to various other purposes, among which terminology management [4], query expansion in Information Retrieval [25], classification of nominal and binary data, as well as handwritten character recognition [24]. All these studies witness that analogical learning based on formal analogies can lead to state-of-the-art performance in a number of applications. Still, it encompasses a number of issues that seriously hinder its widespread use in NLP [27]. This motivates the present paper.

2 PRINCIPLE

In order to understand the methodology we first clarify the process of analogical learning. Let $\mathcal{L} = \{(i(x_k), o(x_k))_k\}$ be a training set gathering pairs of input $i(x_k)$ and output $o(x_k)$ representations of elements x_k. We call input set, and note it $\mathcal{I} = \bigcup_k i(x_k)$, the set of input-space representations in the training set. Given an element t for which only $i(t)$ (or alternatively $o(t)$) is known, analogical learning

[1] University of Montreal, Canada, email: felipe@iro.umontreal.ca

works by:

1. building $\mathcal{E}_i(t) = \{(x, y, z) \in \mathcal{L}^3 \mid [i(x) : i(y) :: i(z) : i(t)]\}$, the set of triplets in the training set that define with t a proportional analogy in the input space,
2. building $\mathcal{E}_o(t) = \{u \mid [o(x) : o(y) :: o(z) : u]$ and $(x, y, z) \in \mathcal{E}_i(t)\}$, the set of solutions to the *analogical equations* obtained in the output space,
3. aggregating the solutions in $\mathcal{E}_o(t)$ in order to select $o(t)$.

In this description, $[x : y :: z : t]$ is our notation for a (formal) proportional analogy;[2] and $[x : y :: z :?]$ is called an analogical equation and represents the set of it solutions. In the sequel, we call x-form, y-form, z-form and t-form the first, second, third and fourth forms respectively of $[x : y :: z : t]$. Also, we sometime refer the 2-first steps of the inference as the *generator*, while we call the third one the *aggregator*.

Let's illustrate this on a tiny example where the task is to associate a sequence of part-of-speech (POS) tags to any given sentence considered as a sequence of words. Let $\mathcal{L} = \{$(*he loves her*, PRP VBZ PRP), (*she loved him*, PRP VBD PRP), (*he smiles at her*, PRP VBZ IN PRP)$\}$ be our training set which maps sequences of words (input) to sequences of POS tags (output). Tagging a (new) sentence such as *she smiled at him*, involves: (i) identifying analogies in the input space: [*he loves her* : *she loved him* :: *he smiles at her* : *she smiled at him*] would be found, (ii) solving the corresponding equations in the output space: [PRP VBZ PRP : PRP VBD PRP :: PRP VBZ IN PRP : ?] would be solved, and (iii) selecting the solution. Here, PRP VBD IN PRP would be the only solution produced.

There are three important aspects to consider when deploying the above learning procedure. First, the search stage (step-1) has a time complexity which is prohibitive in most applications of interest (cubic in the size of $\mathcal{I}$). Second, the aggregation (step-3) of the possibly numerous spurious[3] solutions produced during step-2 is difficult. Last, it might happen that the overall approach does not produce any solution at all, simply because no source analogy is identified during step-1, or because the source analogies identified do not lead to analogies in the output space (failure of the inductive bias).

3 ISSUES WITH ANALOGICAL LEARNING

3.1 Formal Analogy

We mentioned that several definitions of formal analogy have been proposed. There are two of them that stand above the others in the sense that they can account for a larger variety of relations than the others: the one defined in [22] and the one defined in [36]; the latter generalizing the former. The choice of the definition to work with has some practical impact, since simpler relations (such as prefixation) are easier to recognize than more complex ones. Although we normally work with the second definition (because it is the most general one we know of), the discussion in this paper generally applies for all sensible definitions we know.

3.2 Searching in the Input Space

Identifying analogies in the input space (step-1) is a process cubic in the size of $\mathcal{I}$. Clearly, a brute-force approach would be manageable for toy problems only. This is why several authors have worked out some strategies we discuss in this section.

[2] We also use $[x : y :: z : t]$ as a predicate.

[3] A solver typically produces several analogical solutions, among which a few are valid.

3.2.1 A quadratic search procedure

The search for input analogies can be transformed into a quadratic number of equation solving [19] thanks to the symmetry property of analogical relations ($[x : y :: z : t] \Leftrightarrow [y : x :: t : z]$). Unfortunately, this solution barely scales to sets of a few thousands of representatives (a typical vocabulary in an NLP application has in the order of 10^5 words). Therefore, sampling has to be performed.

More precisely, for an element t to be treated, we solve analogical equations $[y : x :: i(t) :?]$ for some pairs $\langle x, y\rangle$ belonging to the neighborhood of $i(t)$. Those solutions that belong to the input space are the z-forms we are interested in. This strategy reduces the search procedure to the resolution of a number of analogical equations which grows quadratically with the size of the neighborhood set $\mathcal{N}$:

$$\mathcal{E}_{\mathcal{I}}(t) = \{ \langle x, y, z\rangle \mid \langle x, y\rangle \in \mathcal{N}(i(t)) \times \mathcal{N}(i(t)), [y : x :: i(t) : z] \}$$

For instance, in [14] the authors deal with an input space in the order of tens of thousand forms by sampling x and y among the closest forms, in terms of edit-distance, to the form $i(t)$.

3.2.2 Exhaustive tree-count search

In [15], the authors developed algorithms for scaling up the search procedure. The main idea is to exploit a property of formal analogies [22]:

$$[x : y :: z : t] \Rightarrow |x|_c + |t|_c = |y|_c + |z|_c \quad \forall c \in \mathcal{A} \qquad (1)$$

where $\mathcal{A}$ is the alphabet on which the forms are built, and $|x|_c$ stands for the number of occurrences of character c in x. In the sequel, we denote $\mathcal{C}(\langle x, t\rangle) = \{\langle y, z\rangle \in \mathcal{I}^2 \mid |x|_c + |t|_c = |y|_c + |z|_c \quad \forall c \in \mathcal{A}\}$ the set of pairs satisfying the count property with respect to $\langle x, t\rangle$.

Their strategy, called tree-count, consists in first selecting an x-form in the input space. This enforces a set of necessary constraints on the counts of characters that any two forms y and z must satisfy for $[x : y :: z : t]$ to hold. By considering all forms x in turn[4], we collect a set of candidate triplets for t. A verification of those that actually define with t an analogy must then be carried out. Formally, they build:

$$\mathcal{E}_{\mathcal{I}}(t) = \{ \langle x, y, z\rangle \mid x \in \mathcal{I}, \langle y, z\rangle \in \mathcal{C}(\langle x, i(t)\rangle), [x : y :: z : i(t)] \}$$

This strategy will only work if (i) the number of quadruplets to check is much smaller than the number of triplets we can form in the input space (which happens to be the case in practice), and if (ii) we can efficiently identify the pairs $\langle y, z\rangle$ that satisfy a set of constraints on character counts. To this end, the authors proposed to organize the input space thanks to a data structure they call a *tree-count* (hence the name of the search procedure), which is easy to build and supports efficient runtime retrieval.[5]

3.2.3 Sampled tree-count search

The tree-count search strategy allows to *exhaustively* solve step 1 for reasonably large input spaces (tens of thousands of forms). However,

[4] Anagram forms do not have to be considered separately.

[5] Possibly involving filtering.

computing analogies in very large input space (hundreds of thousand of forms) remains computationally demanding, as the retrieval algorithm must be carried out $o(\mathcal{I})$ times. In this case, in [15], the authors proposed to sample the x-forms:

$$\mathcal{E}_{\mathcal{I}}(t) = \{ \langle x, y, z \rangle \mid \begin{array}{l} x \in \mathcal{N}(i(t)), \\ \langle y, z \rangle \in \mathcal{C}(\langle x, i(t) \rangle), \\ [x : y :: i(t) : z] \ \} \end{array}$$

The authors proposed a sampling strategy which selects x-forms that share with t some sequences of symbols. To this end, input forms are represented in a k-dimensional vector space, whose dimensions are frequent symbol n-grams, where $n \in [\mathtt{min};\mathtt{max}]$[6]. A form is thus encoded as a binary vector of dimension k, in which the ith coefficient indicates whether the form contains an occurrence of the ith n-gram. At runtime, we select the N forms that are the closest to a given form t, according to a distance (i.e. cosine).

3.2.4 Checking for analogies

For all the aforementioned search strategies, we need to verify that 4 forms are indeed in analogical relation. Stroppa [29] proposed a dynamic programming algorithm for checking $[x : y :: z : t]$ when the definition in [36] is being used. The complexity of this algorithm is in $o(|x| \times |y| \times |z| \times |t|)$. Since a large number of calls to the analogy checking algorithm must be performed during step 1 of analogical learning. The following property may come at help [15]:

$$\begin{array}{l} [x : y :: z : t] \Rightarrow \\ \quad (x[1] \in \{y[1], z[1]\}) \ \vee \ (t[1] \in \{y[1], z[1]\}) \\ \quad (x[\$] \in \{y[\$], z[\$]\}) \ \vee \ (t[\$] \in \{y[\$], z[\$]\}) \end{array} \qquad (2)$$

where $s[\$]$ indicates the last symbol of s. A simple trick consists in calling for the verification of an analogy only for the quadruplets that pass this test.

3.2.5 Open issues

One can already go a long way with the sampled tree-count approach we described. Still, it is unclear which sampling strategy should be considered for a given application. The vector space model proposed in [15] seems to work well in practice, but more experiments should confirm this.

More fundamentally, none of the search procedures proposed so far take into account the fact that many analogies might be redundant. For instance, to relate the masculine French noun *directeur* to its feminine form *directrice*, it is enough to consider $[recteur : rectrice :: directeur : directrice]$. Other analogies (i.e. $[fondateur : fondatrice :: directeur : directrice]$) would simply confirm this relation. In [29], Stroppa formalizes this redundancy by the concept of *analogical support set*. Formally, A is an analogical support set of E iff:

$$\{[x : y :: z :?] : \langle x, y, z \rangle \in A^3\} \supseteq E$$

This raises the question of whether it would be possible to identify a minimal subset of the training set, such that analogical learning would perform equally well in this subset. Determining such a subset would reduce computation time drastically. Also, it would be invaluable for modelling how forms in an input system are related to forms in an output one. We are not aware of studies working on this.

[6] Typical values are min=max=3 and k=20 000.

3.3 Solving Equations

Algorithms for solving analogical equations have been proposed for both definitions of interest we mentioned. For the definition of [36], it can be shown [35] that the set of solutions to an analogical equation is a rational language, therefore we can build a finite-state machine for encoding those solutions. In practice however, the automaton is non deterministic, and in the worst case, enumerating the solutions can be exponential in the length of the sequences being involved in the equation. The solution proposed in [16] consists in sampling this automaton without building it. The more we sample this automaton the more solutions we produce. In our implementation, we call *sampling rate* (ρ) the number of samplings considered.[7] It is important to note that typically, a solver produces several solutions to an equation, many being simply spurious, which means that they obey the definition of formal analogy, but are not valid forms.

To illustrate this, Figure 1 reports the solutions produced to the equation $[even : usual :: unevenly :?]$ by our implementation of the solver defined in [16]. Clearly, many solutions are not valid forms in English, although they define proper solutions according to the definition of formal analogy proposed in [36]. Indeed, this definition recognizes no less than 72 different legitimate solutions, which we were able to produce with enough sampling ($\rho \geq 2000$) in less than a few tenth of milliseconds.

Figure 1. 3-most frequent solutions to $[even : usual :: unevenly :?]$ along with their frequency, as produced by our solver, as a function of the sampling rate ρ. nb stands for the total number of solutions produced.

ρ	nb	solutions		
20	12	*usuaunlly* (3)	***unusually*** (2)	*usunually* (2)
100	34	***unusually*** (6)	*usuaunlly* (6)	*uunsually* (4)
1000	67	***unusually*** (57)	*uunsually* (23)	*usuunally* (19)
2000	72	***unusually*** (130)	*uunsually* (77)	*usunually* (43)

The problem of multiple solutions to an equation is exacerbated when we deal with longer forms. In such cases, the number of spurious solutions can become quite large. As a simple illustration of this, consider the equation $e = [this\ guy\ drinks\ too\ much : this\ boat\ sinks :: those\ guys\ drink\ too\ much :?]$ where forms are considered as strings of characters (the space character does not have a special meaning here). Figure 2 reports the number of solutions produced as a function of sampling rate. For small values of ρ, the solution might be missed by the solver (i.e. $\rho \leq 20$). For larger sampling rates, the expected solution typically appears (with frequent exceptions) among the most frequently generated ones. Note that the number of solutions generated also increases quite drastically. Clearly, enumerating all the solutions is not a good idea (too much solutions, too time consuming).

The fact that a solver can (and typically does) produce spurious solutions means that we must devise a way to distinguish ”good” solutions from spurious ones. We defer this issue to the next section. Yet, we want to stress that currently, our sampling of the automaton that recognizes the solutions to an equation is done entirely randomly. It would be much more efficient to learn to sample the automaton, such that more likely solutions are enumerated first. Several algorithms might be applied for this task, among which the Expectation-Maximization algorithm for transducers described in [9].

[7] We leave this notion unspecified, read [16] for details.

Figure 2. 3-most frequent solutions produced by our solver at different sampling rates for the equation *e*. r indicates the position of the expected solution in the list if present (ϕ otherwise). nb indicates the number of solutions produced, and t the time counted in seconds taken by the solver. For readability, spaces are represented with the symbol _.

$\rho = 20$	$nb = 8$
$t = 0.0003$	$r = \phi$
thos_boate_sinks (2)	
tho_boatse_sinks (2)	
thoboatse__sinks (2)	

$\rho = 100$	$nb = 28$
$t = 0.001$	$r = 13$
thoboatse__sinks (2)	
tho_boatse_sinks (2)	
those_sboat_sink (2)	

$\rho = 1000$	$nb = 28$
$t = 0.009$	$r = 2$
those_boat_ssink (5)	
those_boats_sink (5)	
thoes_tboa_sinks (5)	

$\rho = 10^6$	$nb = 19\,796$
$t = 3.82$	$r = 10$
thoes_boat_sinks (2550)	
thoses_boat_sink (1037)	
those_boat_ssink (999)	

3.4 Aggregating Solutions

Step-3 of analogical learning consists in aggregating all the solutions produced. We saw in the previous section that the number of solutions to an analogical equation can be rather large. Also, there might be quite a large number of analogical equations to solve during step-2, which simply increases the number of solutions gathered in $\mathcal{E}_o(t)$. In many works we know, this problem is not discussed, why our experiments indicate this is a important issue. In [20], Lepage and Lardilleux filter out solutions which contain sequences of symbols not seen in the output space of the training set. This typically leaves many solutions alive, including spurious ones. In [19], Lepage and Denoual propose to keep the most frequently generated solution. The rationale being that forms that are generated by various analogical equations are more likely to be good ones. Also, Ando and Lepage [1] show that the closeness of objects in analogical relations is another interesting feature for ranking solutions generated.

In [16], the authors investigate the use of a classifier trained in a supervised way to recognize good solutions from bad ones. This approach improved the selection mechanism over several baselines (such as selecting the most frequently generated solution), but proved to be difficult to implement, in part because many examples have to be classified, which is time consuming, but also because most of the solutions in $\mathcal{E}_o$ are spurious ones, leaving us with a very unbalanced task, which is challenging. Last but not least, the best classifiers trained were using features computed on the whole set $\mathcal{E}_o$, such as the frequency with which a solution is proposed. This means that it cannot be used to early filter the unlikely solutions generated.

Improving the classifier paradigm deserves further investigations. Notably, in [16], only a small number of features have been considered. Better feature engineering, as well as more systematic tests on different tasks must be carried out for better understanding the limits of the approach.

As discussed in [1], it is intuitively more suited to see the problem of separating good from spurious solutions as a ranking problem. Ranking is an active research topic in machine learning. We refer the reader to the LETOR (LEarning TO Rank) website for an extensive list of resources on this subject.[8] Ranking the solutions proposed by the two-first steps of analogical learning must be investigated as a replacement of the classification solution proposed in [16].

[8] http://research.microsoft.com/en-us/um/beijing/projects/letor//

3.5 Dealing with Silence

In most experiments we conducted, we faced the problem that the learning mechanism we described might produce no solution for a given entity. This might happen because no source analogy has been identified, or because the source analogies identified do not lead to target equations that have a solution. Depending on the nature of the input space and the training material available, this problem can be rather important.

On a task of translating medical terms [16], the authors submitted the silent cases to another approach (in their case a statistical translation engine). Combining analogical learning with statistical machine translation has also been investigated in [6]. In [19], the authors proposed to split the form to treat in two parts and apply analogical learning to solve those two subforms. This raises a number of issues which do not seem to have received attention. Knowing where to split the input form in order to maximize the chance of being able to solve the two new sub-problems is one of those.

3.6 Learning over Tree Structures

Few authors have discussed the possibility of manipulating tree structures instead of sequences of symbols in analogical learning. Stroppa [29] proposed a definition of formal analogies on trees, based on the notion of factorization of trees, very much in line with the definition of formal analogies between sequences of symbols defined in [36]. Based on this definition, the authors of [30] described an exact algorithm for solving an analogical equation on trees which complexity is at least exponential in the number of nodes of the largest tree in the equation. They also proposed two approximate solvers by constraining the type of analogies captured (notably, passive/active alternations are not anymore possible). Ben Hassena [2] proposed a solution for reasoning with trees based on tree alignment. The constraints imposed over the possible alignments are much more restrictive than the ones of [30], but the author reports a solver (a dynamic programing algorithm) which has a polynomial complexity. Unfortunately, none of the aforementioned approaches scale to even medium-sized corpora of trees. For instance in [2] the author applied analogical learning on a training set of less than 300 tree structures, a very small corpus by today's standards. See also the work of Ando and Lepage [1] for a very similar setting.

4 CASE STUDY

4.1 Settings

In order to illustrate some of the elements we discussed in the previous section, we applied analogical learning to the task of transliterating English proper names into Chinese. The task we studied is part of the NEWS evaluation campaign conducted in 2009 [23]. Transliteration is generally defined as phonetic translation of names across languages and is often thought as a critical technology in many domains, such as machine translation and cross-language information retrieval or extraction [23]. Examples of transliteration from English proper names into Chinese are reported in Table 4.

The organizers of the NEWS campaign kindly provided us with the data that was distributed to the participants of the task. Its main characteristics are reported in Table 1, after the English letters have been lowercased. The distribution of Chinese characters is typically Zipfian, and 116 out of the 370 different characters seen in the training set appear less than 10 times (30 characters appear only once).

In order to transliterate the English proper names of the test set, we gathered a training set $\mathcal{L}_1 = train + dev$ by concatenating the training set and the development set that were released, that is, 34 857 pairs of English and Chinese proper names. Including the development set in the training material is fine, since there is no training involved when generating the set of solutions. In parallel to this, we also generated solutions for the development set (dev), using the released training material only ($\mathcal{L}_2 = train$); the solutions produced were used for training a classifier to recognize good from spurious solutions. This classifier was then applied to the solutions produced for the test set ($test$) thanks to $\mathcal{L}_1$.

Table 1. Main characteristics of the English-Chinese data provided in NEWS 2009.

	$train$	dev	$test$
examples	31 961	2 896	2 896
EN symbols	26	26	26
CH symbols	370	275	283

examples	
Emission	埃米申
Blagrove	布格夫
Aposhian	阿 波 希 安

We ran two configurations of our *generator*: FULL-TC corresponds to the exhaustive tree-count setting described in Section 3.2.2, while SAMP-TC corresponds to the sampled version described in Section 3.2.3.[9] Since the number of source analogies identified can be quite large for some test forms, we enforced a timeout of 1 minute per English proper name for accomplishing step-1 of the inference in the FULL-TC setting and a timeout of 20 seconds for the SAMP-TC configuration. In both cases, the solver was run with a sampling rate of $\rho = 200$.

Regarding the classifier, we followed [16] and trained a voted-perceptron [10]. We computed a total of 19 features including the frequency of a solution, its rank in the list, input and output degrees (a notion defined for instance in [29]), language models likelihoods, etc. A greedy search over the feature set revealed that a handful of features only where useful. We trained the classifier over 5 000 epochs. The same classifier was used for both the FULL-TC and the SAMP-TC configurations we tested.

4.2 Monitoring Analogical Inference

We describe in the following the FULL-TC configuration, while Table 2 reports the figures of interest for both configurations. For the exhaustive configuration, the average time spent on step-1 per English form is 17 seconds. For 327 forms, the timeout applied, which means that we likely missed useful source analogies involving those forms. Most of the time spent during step-1 was devoted to check candidate analogies, that is, the quadruplets that pass the test in Equation 1. The trick we mentioned in Equation 2 avoided 63.8% of the verifications, a very nice speed up.

An average of 4 517 input analogies were identified per test form (with a maximum of 32 016); for 18 of them however, we could not identify any source analogy, leading to no response in those cases. Out of the 2878 test forms for which we could identify at least one source analogy, 2838 of them lead to an average of 487 output equations, the other 50 were left without answer. Solving all those equations led to an average of 405 solutions per test form (minimum 2, maximum 2221). Note that many equations solved did not lead to any solution, which explains why on average, the number of solutions is lower than the number of equations solved. The average time for solving the equations per form was 0.22 seconds (maximum 1.5s). In the end, we decided to keep up to the 100 most frequently generated solutions for a given test form (a solution is typically generated by several equations).

[9] The 1000-closest input forms to each English test forms where considered, based on a vector space representing the $k = 1000$ most frequent 3-grams of characters observed in $\mathcal{I}$, and the cosine distance.

Table 2. Main characteristics of the two configurations tested.

	FULL-TC	SAMP-TC
avg. time (step-1)	17s	2
avg. time (step-2)	0.22s	0.01s
number of timeouts	327	1
avg. input analogies	4517	158
avg. output equations	487	18
avg. number of solutions	405	37.5
silence (step-1)	18	76
silence (step-2)	50	249

It is interesting to note the discrepancy between the number of source analogies identified and the number of target equations effectively solved, which is much lower. This indicates either that the source analogies were in large part fortuitous, or that the inference bias (one analogy in the input space corresponds to an analogy in the output space) does not apply well for this task.

4.3 Evaluation

Table 3. Number of reference solutions among the 100-top frequent solutions proposed by the FULL-TC configuration. Read the text for more.

rank	nb	r_{2374}%	r_{all}%	nb	r_{1659}%	r_{all}%
1	1093	46.0	37.7	1410	85.0	48.7
2	1418	59.7	48.9	1627	98.1	56.2
3	1582	66.6	54.6	1657	99.9	57.2
4	1699	71.6	58.7	1659	100.0	57.3
5	1796	75.7	62.0	.	.	.
⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	2374	100.0	82.0	1659	100.0	57.3

The left part of Table 3 reports the number of reference transliterations identified in the first $rank$ positions of the list of solutions proposed by the generator. We note that we could treat at most 2374 test forms correctly if we consider the 100-most frequently generated solutions produced. This represents only 82% of the test forms. Looking only at the most frequently generated solution[10], we observe that 37.7% of the test forms were transliterated correctly. This represents an accuracy of 46% (see r_{2374}) if we only consider the 2374 test forms where the reference transliteration was identified correctly among the first 100 solutions. These figures clearly show that being able to distinguish good from spurious solutions has the potential to improve the overall approach by more than 30 absolute points.

The right part of Table 3 indicates the performance of the FULL-TC inference after the aggregation step. Out of the 2374 test forms for which the correct solution was identified in the first 100 positions, only 1659 (70%) ones now receive a good solution. This shows that the classifier is too aggressive. On the other hand, 48.7% of the test forms now have the correct solution in the first position. This represents an increase of 11 absolute points over keeping the most-frequent solution. Actually, we can observe that for most of the test forms, the reference solution is either in the 2-first positions, either not present at all. Considering the fact that we did not spend much time for engineering features for the task, this is rather encouraging.

[10] Ties are broken randomly.

Table 4. Random excerpt of analogical transliterations produced by FULL-TC. r_c (resp. r) indicates the rank of the correct transliteration in the candidate list after (resp. before) the aggregation step. nb indicates the number of solutions generated. We replaced the Chinese characters we could not print correctly with our LaTeX processor by roman letters.

EN forms	reference	solutions	r_c	r	nb
auchter	x 克 特	x 克 特 (218)	1	1	380
sundell	森 德 y	森 德 y (692)	1	5	664
fannin	范 宁	范 妮 恩 (54)	ϕ	5	104
frere	弗 里 y	弗 里 y (6113)	1	1	630
shurkin	舒 金	舒 y 金 (237) 舒 金 (208)	2	3	386

Table 4 provides a random excerpt of the output produced by the FULL-TC configuration. Table 5 reports the results of our system as measured by the official metrics that were used to evaluate the different participating systems [23]. Clearly, our system is not among the leading ones. In fact, we would have ended up at the 14th rank according to accuracy (ACC); 18 systems participated to the 2009 exercise. Since our major goal was to monitor analogical learning, we did not put efforts yet into improving those figures, although there are straightforward things that could be done, such as always providing 10 candidate solutions, even if the classifier filtered in much less (except for accuracy, the other metrics are assuming a list of 10 candidates). Also, we did not attempt anything for dealing with silent test forms. In [6], the authors show that combining in a simple way analogical learning with statistical machine translation can improve upon the performance of individual systems. Last, it is shown in [6] that representing examples as sequences of syllables instead of characters (as we did here) leads to a significant improvement of analogical learning on a English-to-Indi transliteration task.

Table 5. Metrics used at the NEWS 2009 evaluation campaign. For comparisons, $1st$ and $last$ indicates respectively the first and last performing systems, as reported in [23].

metric	FULL-TC	SAMP-TC	$1st$	$last$
ACC:	0.486	0.308	0.731	0.199
F-score	0.772	0.612	0.895	0.606
MRR	0.527	0.330	0.812	0.229
MAP_{ref}	0.486	0.308	0.652	0.199

5 CONCLUSION

We presented a number of works on formal analogy dedicated to various NLP tasks. We discussed a number of issues that we feel remain to be investigated for the approach to meet higher acceptance among the NLP community. We presented a case study, transliteration of proper names, for which we reported encouraging results. More importantly, we used this case study for illustrating some of the issues behind the scene of analogical learning.

ACKNOWLEDGEMENTS

We thank the reviewers for their detailed and enriching comments.

REFERENCES

[1] Shinichi Ando and Yves Lepage, 'Linguistic structure analysis by analogy: Its efficiency', in *NLPRS*, pp. 401–406, Phuket, Thailand, (1997).

[2] Anouar Ben Hassena, *Apprentissage analogique par analogie de structures d'arbres*, Ph.D. dissertation, Univ. de Rennes I, France, 2011.

[3] Aditya Bhargava and Grzegorz Kondrak, 'How do you pronounce your name? improving g2p with transliterations', in *49th ACL/HLT*, pp. 399–408, Portland, USA, (2011).

[4] Vincent Claveau and Marie-Claude L'Homme, 'Structuring terminology by analogy-based machine learning', in *7th International Conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark, (2005).

[5] M. Creutz and K. Lagus, 'Inducing the morphological lexicon of a natural language from unan- notated text', in *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pp. 106–113, Espoo, Finland, (2005).

[6] Sandipan Dandapat, Sara Morrissey, Sudip Kumar Naskar, and Harold Somers, 'Mitigating problems in analogy-based ebmt with smt and vice versa: a case study with named entity transliteration', in *PACLIC*, Sendai, Japan, (2010).

[7] Étienne Denoual, 'Analogical translation of unknown words in a statistical machine translation framework', in *MT Summit XI*, pp. 135–141, Copenhagen, Denmark, (2007).

[8] Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka, 'Cross-language latent relational search: Mapping knowledge across languages', in *AAAI'11*, pp. 1237 – 1242, (2011).

[9] Jason Eisner, 'Parameter estimation for probabilistic finite-state transducers', in *40th ACL*, pp. 1–8, Philadelphia, USA, (2002).

[10] Y. Freund and R. E. Schapire, 'Large margin classification using the perceptron algorithm', *Mach. Learn.*, **37**(3), 277–296, (1999).

[11] Julien Gosme and Yves Lepage, 'Structure des trigrammes inconnus et lissage par analogie', in *18e TALN*, Montpellier, France, (2011).

[12] Philipp Koehn, 'Pharaoh: a beam search decoder for phrase-based statistical machine translation models', in *6th AMTA*, Washington DC, (2004).

[13] M. Kurimo, S. Virpioja, V.T. Turunen, G.W. Blackwood, and W.J. Byrne, 'Overview and results of morpho challenge 2009', in *10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009)*, Lecture Notes in Computer Science, pp. 578–597, (2009).

[14] Philippe Langlais and Alexandre Patry, 'Translating Unknown Words by Analogical Learning', in *EMNLP*, pp. 877–886, Prague, Czech Republic, (2007).

[15] Philippe Langlais and François Yvon, 'Scaling up analogical learning', Technical report, Paritech, INFRES, IC2, Paris, France, (Oct. 2008).

[16] Philippe Langlais, François Yvon, and Pierre Zweigenbaum, 'Improvements in Analogical Learning: Application to Translating multi-Terms of the Medical Domain', in *12th EACL*, pp. 487–495, Athens, (2009).

[17] Jean-François Lavallée and Philippe Langlais, 'Moranapho: un système multilingue d'analyse morphologique basé sur l'analogie formelle', *TAL*, **52**(2), 17–44, (2011).

[18] Yves Lepage. De l'analogie rendant compte de la commutation en linguistique. Habilitation à diriger des recherches, Université Joseph Fourier, Grenoble I, France, 2003.

[19] Yves Lepage and Étienne Denoual, 'Purest ever example-based machine translation: Detailed presentation and assesment', *Mach. Translat*, **19**, 25–252, (2005).

[20] Yves Lepage and Adrien Lardilleux, 'The greyc translation memory for the iwslt 2007 evaluation campaign', in *4th IWSLT*, pp. 49–54, Trento, Italy, (2008).

[21] Yves Lepage, Adrien Lardilleux, and Julien Gosme, 'The greyc translation memory for the iwslt 2009 evaluation campaign: one step beyond translation memory', in *6th IWSLT*, pp. 45–49, Tokyo, Japan, (2009).

[22] Yves Lepage and Ando Shin-ichi, 'Saussurian analogy: A theoretical account and its application', in *7th COLING*, pp. 717–722, (1996).

[23] Haizhou Li, A. Kumaran, Vladimir Pervouchine, and Min Zhang, 'Report of news 2009 machine transliteration shared task', in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pp. 1–18, (2009).

[24] Laurent Miclet, Sabri Bayroudh, and Arnaud Delhay, 'Analogical dissimilarity: Definitions, algorithms and two experiments in machine learning', *Journal of Artificial Intelligence Research*, 793–824, (2008).

[25] Fabienne Moreau, Vincent Claveau, and Pascale Sébillot, 'Automatic morphological query expansion using analogy-based machine learn-

ing', in *29th European conference on IR research (ECIR'07)*, pp. 222–233, Berlin, Heidelberg, (2007).

[26] Vito Pirrelli and François Yvon, 'The hidden dimesion: a paradigmatic view of data-driven nlp', *Journal of Experimental & Theoretical Artificial Intelligence*, **11**, 391–408, (1999).

[27] Harold Somers, Sandipan Sandapat, and Sudip Kumar Naskar, 'A review of ebmt using proportional analogies', in *3rd Workshop on Example-Based Machine Translation*, pp. 53–60, Dublin, Ireland, (2009).

[28] Sebastian Spiegler, 'Emma: A novel evaluation metric for morphological analysis - experimental results in detail', Technical Report CSTR-10-004, University of Bristol, Bristol, (2010).

[29] Nicolas Stroppa, *Définitions et caractérisations de modèles à base d'analogies pour l'apprentissage automatique des langues naturelles*, Ph.D. dissertation, Telecom Paris, ENST, Paris, France, 2005.

[30] Nicolas Stroppa and François Yvon, 'Formal Models of Analogical Proportions'. Available on HAL Portal, 2007.

[31] Nicolas Stroppa and François Yvon, 'An analogical learner for morphological analysis', in *9th Conf. on Computational Natural Language Learning (CoNLL)*, pp. 120–127, Ann Arbor, USA, (2005).

[32] P.D. Turney and M.L. Littman, 'Corpus-based learning of analogies and semantic relations', in *Machine Learning*, volume 60, pp. 251–278, (2005).

[33] Antal van den Bosch and Walter Daelemans, 'Data-oriented methods for grapheme-to-phoneme conversion', in *EACL*, pp. 45–53, Utrecht, Netherlands, (1993).

[34] François Yvon, 'Paradigmatic cascades: a linguistically sound model of pronunciation by analogy', in *In Proceedings of 35th ACL*, pp. 429–435, (1997).

[35] François Yvon, 'Finite-state machines solving analogies on words', Technical Report D008, École Nationale Supérieure des Télécommunications, (2003).

[36] François Yvon, Nicolas Stroppa, Arnaud Delhay, and Laurent Miclet, 'Solving analogies on words', Technical Report D005, École Nationale Supérieure des Télécommuncations, Paris, France, (2004).

(Re-)discovering the graphical structure of Chinese characters

Yves Lepage[1]

Abstract. The purpose of this paper is to show how it is possible to efficiently extract the structure of a set of objects by use of the notion of proportional analogy. As a proportional analogy involves four objects, the very naïve approach to the problem, has basically a complexity of $O(n^4)$ for a given set of n objects. We show, under some conditions on proportional analogy, how to reduce this complexity to $O(n^2)$ by considering an equivalent problem, that of enumerating analogical clusters that are informative and not redundant. We further show how some improvements make the task tractable. We illustrate our technique with a task related with natural language processing, that of clustering Sino-Japanese characters. In this way, we re-discover the graphical structure of these characters.

1 INTRODUCTION

1.1 Background

Analogy is defined in various ways by different recent authors [1, 7, 21]. Referring back to the most ancient definitions, one can reach an agreement on the following definition of *proportional analogy*:

> Four objects A, B, C and D, are in analogical relation (proportional analogy) if the first object is to the second object in the same way as the third object is to the fourth object. Proportional analogy is noted $A : B :: C : D$.

In all generality, if the relation between two objects (noted by the colon :) is called a *ratio* and the relation between the two pairs of objects (noted by the two colons ::) is called a *conformity,* then proportional analogy is a conformity of ratios between two pairs of objects.

Proportional analogy can be seen between words on the level of form or on the level of meaning or on both at the same time (see [2] for abnormal cases).

Form but not meaning:

to walk : *walked* :: *he* : *heed*

Meaning but not form:

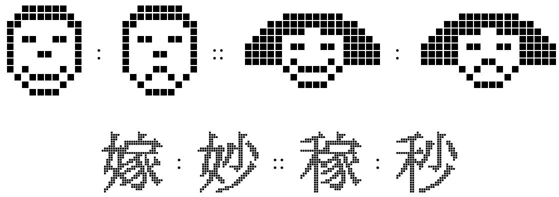
to walk : *walked* :: *to be* : *was*

Form and meaning:

to walk : *walked* :: *to work* : *worked*

Proportional analogies on the levels of form and meaning at the same time are called true analogies. Between chunks or short sentences, their number has been shown to be quite important [10, 12, 13]. Many studies, too many to cite here, address the efficiency of analogy for segmenting words or grouping them according to word families (as for Chinese, see for instance [19]). Forms which depart from declension or conjugation paradigms (groups of proportions in [14]) were called anomalies in Classical grammar [18]. Recently, analogies between word meanings (*water* : *riverbed* :: *traffic* : *road*) have been shown to be reproduceable on computers using large corpora and vector space models [17, 16, 15].

Proportional analogies are not only verbal. They may hold between any kind of objects provided, generally, that the objects be of the same kind, a point Aristotle, among other ancient and recent authors, insists on. The general principle, viewed as a cognitive process, is based on iconicity [3]. Taking the term to its restrained graphical sense, the following two examples illustrate proportional analogies between icons of black and white pixels.



The latter example is a graphical proportional analogy between four Sino-Japanese characters. By explicitly decomposing into constitutive elements, as in [20], to compute similarity between Sino-Japanese characters, it is understandable that the left and the right parts of the characters can be exchanged to give rise to the four different characters. The above analogy does not apply on the level of meaning, as the character meanings are unrelated: 'spouse', 'odd', 'to earn' and 'second (measure of time)'. It does not make an analogy on the level of pronunciation either.

The particular and practical problem which we tackle in a broader research concerned with ease of learning of Chinese characters, is to re-discover the graphical structure of Chinese characters in an automatic way by relying on the notion of proportional analogy. The general and theoretical problem that we thus tackle in the following sections is to rely on the properties of proportional analogies so as to automatically visualize the structure of a set of objects.

1.2 The problem

The naïve approach to the problem of the enumeration of all proportional analogies between a set of n objects consists in examining all possible quadruples of objects and checking for analogy. This naïve approach has a complexity of $O(n^4)$.

[1] IPS, Waseda University, Japan. Email: yves.lepage@waseda.jp

Without changing the complexity, the computation time may be reduced. For a given proportional analogy, there exists seven other equivalent forms (see Theorem 2.1 in [9]). This is implied by the basic properties of exchange of the means (exchanging objects B and C in the analogy $A : B :: C : D$, second line below) and symmetry of conformity (exchanging the terms on both sides of the :: sign, sixth line below). In this way, the following eight analogies are shown to be equivalent:

$A : B :: C : D$	
$A : C :: B : D$	exch. means
$B : A :: D : C$	exch. means + sym. :: + exch. means
$B : D :: A : C$	exch. means + sym. ::
$C : A :: D : B$	sym. :: + exch. means
$C : D :: A : B$	sym. ::
$D : B :: C : A$	sym. :: + exch. means + sym. ::
$D : C :: B : A$	exch. means + sym. :: + exch. means + sym. ::

Because of these eight equivalent forms, the enumeration time can be divided by a factor of 8, but the complexity remains $O(n^4)$.

To make our point clear, consider the following naïve estimation. In a preliminary experiment, we estimated the average time needed for the verification of one analogy between four Sino-Japanese characters using 36 features (see Section 5.3 for a description of the features). An average time of 0.8 ms was measured. For almost fifteen thousand Sino-Japanese characters (see Section 5.2 for a description of the data), and knowing that there are approximately 3.2×10^7 seconds in a year, the time needed to compute all possible analogies would exceed a million years.[2]

In order to reduce the complexity of this problem, we propose to modify our goal. Rather than aiming at individual analogies, we compute all possible ratios between all possible objects at hand. This computation is basically $O(n^2)$. The result of this computation allows us to cluster pairs of objects according to their ratios. These clusters summarize all possible analogies between all objects in a non-redundant way that still provides the total amount of information (see Section 2). The sequel of the paper shows how to compute such clusters and presents some of the actual results of such a computation on a set of Sino-Japanese characters.

The paper is structured as followed: Section 2 shows how the problem can be transformed into a problem of quadratic complexity and introduces the notion of analogical clusters for this purpose. Section 3 gives our proposed method to output analogical clusters. Section 4 mentions some improvements that can reduce computational time. Section 5 describes the application of the proposed method to the problem of structuring Sino-Japanese characters, and gives the results obtained in our experiments.

2 ANALOGICAL CLUSTERS

2.1 Objects as feature vectors

In this work, we represent an object by a vector of features with numerical values. We also impose that the feature space be the same for all objects, so that it is trivially possible to define a ratio between two objects as the vector of their difference. In such a setting, conformity is trivially reduced to equality between vectors.

The following equation illustrates a possible case of proportional analogy between vectors in a four-dimensional space.

$$\begin{pmatrix}3\\6\\10\\7\end{pmatrix} - \begin{pmatrix}2\\6\\2\\3\end{pmatrix} = \begin{pmatrix}10\\8\\9\\5\end{pmatrix} - \begin{pmatrix}9\\8\\1\\1\end{pmatrix}$$

Vector difference as a ratio, and equality between vectors as conformity, consistently define analogies that meet the intuitive notions about proportional analogies. Among other properties, the eight forms of equivalence for the same proportional analogy (see above Section 1.2) always hold.

2.2 Transitivity of conformity: analogical clusters

It is not always the case that conformity verifies transitivity. For instance, [8] shows that the intuitive notions of proportional analogy between strings of characters imply that there is no transitivity for conformity.[3]

In our setting with conformity being an equality, i.e., an equivalence relation, transitivity naturally holds in addition to reflexivity and symmetry. For proportional analogies, transitivity of conformity implies that:

$$A : B :: C : D \text{ and } C : D :: E : F \Rightarrow A : B :: E : F$$

For our present task of enumerating all possible proportional analogies between all objects in a given set, a transitive conformity can lead to an enormous economy in representation. To illustrate this point, consider the following three proportional analogies.

$$\begin{array}{c} A : B :: C : D \\ C : D :: E : F \\ A : B :: E : F \end{array}$$

They can be represented in a more economical way by the following list of equal ratios:

$$\begin{array}{c} A : B \\ C : D \\ E : F \end{array}$$

All ratios being equal, any possible proportional analogy formed by taking any two ratios holds.

From the above example, it is clear that, provided conformity is transitive, a list of n pairs of objects with the same ratio stands for a list of $n \times (n-1) \ / \ 2$ non-trivial proportional analogies (see Section 2.5 for trivial analogies). Consequently, under the assumption of transitivity for conformity, the problem of enumerating all possible proportional analogies between all possible objects in a given set can be transformed into a problem of enumerating all possible pairs of objects with the same ratio. The former problem has a complexity of $O(n^4)$ while the latter one has a complexity of $O(n^2)$.

From now on, we shall call a list of ratios of objects with the same value, an *analogical cluster*.

[2]
$$\begin{aligned} 14,655^4 \times 0.8 \text{ ms} &> 14^4 \times 10^{12} \times 0.8 \text{ ms} \\ &> 48 \times 10^{12} \text{ s} \\ &> 48 \times 10^{12} \ / \ (3.2 \times 10^7) \text{ years} \\ &> 1.5 \times 10^6 \text{ years} \end{aligned}$$

[3] This comes from the fact that some analogies between strings of characters admit multiple solutions. When this is the case, then, there is not transitivity for :: in the general case for the objects considered (see [8, p. 113]).

2.3 Equivalent forms of analogy: redundancy of clusters

Each analogical cluster stands for a different ratio, i.e., a vector that represents the difference between any two feature vectors each representing an object.

Because the order in analogical clusters is not relevant, an analogy extracted from an analogical cluster stands for two equivalent forms, obtained by symmetry of conformity (sixth line in the eight equivalent forms of proposition analogy in Section 1.2).

$$A : B :: C : D \;\Leftrightarrow\; C : D :: A : B$$

By inversion of ratios (third line in the eight equivalent forms of proposition analogy in Section 1.2), a proportional analogy involves two different ratios. And by exchange of the means, (second line in the eight equivalent forms of proposition analogy in Section 1.2), another two different ratios.

$$A : B :: C : D \;\Leftrightarrow\; B : A :: D : C \;\Leftrightarrow\; A : C :: B : D$$

Consequently, in total, the eight different forms of the same proportional analogy are to be found in four different analogical clusters (and only four clusters) among all the possible clusters that are output by a method yielding all the possible clusters standing for differences between all feature vectors representing all the objects in a given set.

Figure 1 shows such four analogical clusters for the proportional analogy $A : B :: C : D$. These four clusters are redundant because of the eight equivalent forms for the same proportional analogy, as we have just stated:

- In any cluster, the order of appearance of the pairs of objects being irrelevant, each cluster encapsulates two equivalent forms of the same proportional analogy. This is symmetry of conformity.
- Analogical clusters (1) and (2) together contain the same information as clusters (3) and (4) together. The relation between (1) and (2) (and between (3) and (4)) is the exchange of the means.
- Analogical clusters (1) and (3) are indeed the same up to an exchange of the objects on the left and the right of the : sign. This is actually inversion of ratios. The same is also true for clusters (2) and (4).

Cluster number			
(1)	(2)	(3)	(4)
$A : B$	⋮	⋮	⋮
⋮	$B : D$	$B : A$	$C : A$
⋮	⋮	⋮	⋮
$C : D$	$A : C$	⋮	$D : B$
⋮	⋮	$D : C$	⋮

Figure 1. For a given proportional analogy $A : B :: C : D$, the set of analogical clusters output by a method that looks for all possible vector differences between all possible feature vectors representing objects in a given set should include four clusters.

It is trivially possible to eliminate the redundancy between clusters (1) and (2) and clusters (3) and (4). This can be done by avoiding the computation of the difference between two vectors and its opposite value (the same two vectors in the reverse order). For that, it suffices to sort all the vectors in some predefined increasing order, and to compute only the differences between two vectors ranked in that order. In this way, a particular proportional analogy will appear in two, and only two, different analogical clusters among the set of all clusters. As a result, globally, the set of all clusters will contain no redundant information.

2.4 Equality of feature vectors: separation of space

By definition of the ratio as a vector difference, the case where $A : B$ and $A : C$ belong to the same analogical cluster is only possible if the vectors representing B and C are the same. This can only happen if the feature vectors do not separate the space of objects into each individual object. Reciprocally, if the feature vectors are unique for each different objects in the given set, the two ratios $A : B$ and $A : C$ for different B and C will be different. For our proposed method, this implies to check for the *separation of the space of objects* before proceeding to clustering.

2.5 Trivial analogies: informativity of clusters

Finally, we must mention a particular case of no interest as it does not bring any information. This is the special case of the cluster for the null vector; i.e., null ratio. It has the following form.

$$\begin{array}{c} A : A \\ B : B \\ C : C \\ \vdots \end{array}$$

It represents the set of all *trivial proportional analogies,* i.e., proportional analogies of the form: $A : A :: B : B$. As our interest is the enumeration of informative analogical clusters we simply avoid to produce this cluster.

By exchange of the means, trivial analogies are equivalent to analogies of the form $A : B :: A : B$. Enumeration of pairs of objects in a predefined sorting order trivially ensures that the difference between two objects is never computed twice. However, it does not prevent from outputting clusters that would contain only one pair of objects. This happens when two objects have a unique vector difference. This problem will be tackled in Section 4.1.

3 INFORMATIVE AND NON-REDUNDANT ENUMERATION OF ANALOGICAL CLUSTERS

3.1 Feature tree and quadratic exploration of the feature tree

Each object is represented by a vector of features. An order can be imposed on the features. In this way, each vector is considered as a list with a recursive structure of a head (the first feature value) and a tail (the remaining features). The lexicographic order, relying on the order on integers, can be applied to such a set of lists. In this way, it is possible to sort the feature vectors representing all objects.

A tree structure underlies such an ordered list. For the first feature, each different value can be encoded in one node. Each such node can be assigned the interval that represents the span over the sorted list of objects. This can be recursively applied to each interval with the tail

	[1;2]:2	[3;3]:3	[4;4]:9	[5;5]:10
[1;2]:2	[1;2]x[1;2]:0	[1;2]x[3;3]:1	[1;2]x[4;4]:7	[1;2]x[5;5]:8
[3;3]:3			[3;3]x[4;4]:6	[3;3]x[5;5]:7
[4;4]:9				[4;4]x[5;5]:1
[5;5]:10				

	[4;4]:9	[5;5]:8
[1;1]:6	[1;1]x[4;4]:3	
[2;2]:7	[2;2]x[4;4]:2	
[3;3]:6		[3;3]x[5;5]:2

Figure 3. On the left, computation of the value differences on the first level for the feature tree of Figure 2. The blank cells are not computed to avoid redundancy (opposite values) or trivial analogies (diagonal cells where intervals are reduced to one object). On the right, recursive computation of the value differences on the second level for a difference of 7 on the first level. The corresponding list of pairs of intervals is [1;2] x [4;4] + [3;3] x [5;5] (refer to table on the left). Two new lists of pairs of intervals are obtained: [1;1] x [4;4] + [3;3] x [5;5] for value 2 and [2;2] x [4;4] for value 3. The latter list of pairs of intervals is deleted as it contains only one pair of intervals, each interval reducing to one object (degenerated cluster).

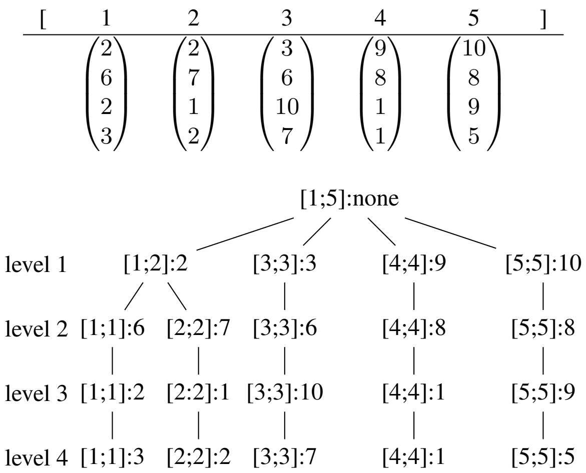


Figure 2. The feature tree (below) corresponding to a set of five objects represented as feature vectors (above). The intervals, noted with brackets, are followed by the value of the feature on that level.

of the feature vectors considered as lists (thus for the second feature and so on) to build a tree structure where the levels stand for each different feature and where each node holds the interval of the sorted objects with the same value for that feature, given all values above are equal. On the last level, each interval should be reduced to one object if the space is well separated. Such a tree structure can be traversed in breadth-first order. Figure 2 illustrates such a data structure for a set of 5 feature vectors.

This data structure[4] is quite different from the one used in [5] to search a space of strings of characters for analogies. Firstly, the geometry is different. In [5], the nodes on the same level may correspond to different characters (i.e. features). This is not the case here. Each level must correspond to exactly the same feature. Consequently, on the contrary to the structure in [5] no intermediate node can stand for an object. All the objects are to be found on the leaves. Secondly, the labels borne by the nodes are different. In our tree, the nodes bear the spanning interval in the sorted list of objects and the value of the feature; the name of the feature, being useless, is forgotten. This is of primary importance for the parallel traversal in sorted order of objects with the first interval never overtaking the second interval so as to avoid redundancy (see Section 2.3 and see below). Thirdly, our use is different as we aim at a complete enumeration of all possible ratios, which compelled the design of this data structure.

The computation of all ratios between all feature vectors simply consists in traversing the same tree in parallel in breadth-first order (a kind of a Cartesian self-product), and computing the differences between the values on each pair of nodes. For the same difference at a given local level, all the pairs of intervals are memorized in a list. This procedure is recursively applied down to the last level for each different value at a local level. Figure 3 illustrates this process for the feature tree given in Figure 2.

Sections 4.1 and 4.2 show that it is possible to terminate the exploration by checking for some structural conditions on the list of pairs of intervals memorized.

In the parallel traversal, we impose that for two lists of pairs of intervals to be processed, the first list be strictly before the second list. This is tantamount to explore only the upper corner of a matrix excluding its diagonal. This avoids redundancy and non-informativity. when computing all possible analogies: the ratio of two vectors is computed once, its opposite is not (Section 2.3); intervals that are reduced to one object on the diagonal are checked to avoid trivial clusters (Section 2.5).

3.2 Sketch of the method

The following gives a sketch of the proposed method.

- Convert each object into a feature vector;
- check for separation of space;
- define an order on the feature vectors (we use least correlations of values among features);
- sort the feature vectors according to lexicographic order in the defined order of features;
- build a feature tree for the sorted feature vectors;
- traverse the feature tree in parallel in breadth-first order to compute the differences between the feature vectors by blocks;
- output the list of pairs of intervals (on the last level, each interval should be reduced to one object if the space is well separated) that corresponds to each vector difference.

By construction and by definition, each list of pairs of objects, that share the same feature vector difference, is an analogical cluster.

[4] The tree structure described here is the same as the one used in two of our previous works: for the complete enumeration of all analogies between sentences contained in corpora of 100,000 short sentences in Chinese, Japanese and English [10] (with sequences of bits as features and various ratios for various features and automatic sorting of the features for early detection of useless zones in the cluster space so as to speed up the overall process); and for the enumeration of clusters reflecting linguistic oppositions among 40,000 short sentences in English and Japanese [11]. In these two works, respecting the equality of edit distance for analogies of commutation between strings of characters implied extra processing.

4 IMPROVEMENTS

4.1 Elimination of clusters reduced to one ratio

We call degenerated clusters those clusters which contain only one ratio, $A : B$ i.e., one pair of objects. Obviously, such clusters do not give rise to any analogy other than the trivial analogy $A : B :: A : B$ and are thus not worth to output. An early detection of such cases leads to an important reduction in processing time.

The implementation of the early detection of such degenerated clusters relies on the data structure of feature tree. After the computation of all possible differences between all possible vectors down to a certain level in the tree, it is easy to scan all the differences and look at the intervals they represent. If a set of pairs of intervals contains only one pair of intervals, each of which being reduced to one object, this is a case of a degenerated cluster. Such a cluster may be immediately deleted so as to stop any further computation on the lower levels.

A comparison of two runs of the programs with or without early detection of degenerated clusters is given in Table 1. It shows that, for our special case of structuring Sino-Japanese characters, a reduction of one third of the computational time can be achieved. There exists some overhead as is shown by the fact that an increase of 55% in computational time is observed for 1,000 characters.

Table 1. Comparison of runtimes with or without detection of degenerated clusters (clusters reduced to one ratio).

number of chars processed	runtimes in seconds without	runtimes in seconds with	time reduction in percentage
1,000	9	14	+55 %
2,000	39	36	-7 %
3,000	92	82	-10 %
4,000	173	142	-17 %
5,000	277	219	-20 %
6,000	426	313	-26 %
7,000	605	438	-27 %
8,000	739	557	-24 %
9,000	944	702	-25 %
10,000	1204	836	-30 %
11,000	1517	1123	-25 %
12,000	1864	1302	-30 %
13,000	2265	1342	-40 %
14,000	2646	1791	-32 %
14,655	2873	1889	-34 %

4.2 Conditional elimination of clusters reduced to one proportional analogy

In Section 2.5, it was shown that an analogy appears in only two analogical clusters. For economy of description, we would like to eliminate redundant information as most as possible. When an analogy belongs to two clusters that contain a large number of pairs of objects, it is a priori impossible, without loss of information, to remove those lines that correspond to this analogy from one of the cluster. This is not the case when one of the analogy is reduced to a cluster that contains only one analogy, i.e., exactly those two lines corresponding to the analogy at hand. This situation is illustrated below:

$$\begin{array}{ccc} A & : & B \\ C & : & D \end{array} \qquad \Bigg| \qquad \begin{array}{ccc} & \vdots & \\ A & : & C \\ & \vdots & \\ B & : & D \\ & \vdots & \end{array}$$

In this case, it is possible to delete the cluster reduced to one analogy. This can be performed during the enumeration of analogical clusters, level by level, using the feature tree. In this case, clusters reduced to one analogy should be memorized on each level. At the end of the exploration of each level of the feature tree, such clusters can be removed from the list of clusters to explore further. This should lead to a reduction in the total computational time. Our current implementation does make use of this possibility and performs the deletion of clusters reduced to one analogy after complete enumeration of analogical clusters in a post-processing phase.

5 EXPERIMENTS

In the frame of a larger study concerned with measuring the ease with which learners can remember Chinese characters along with their pronunciation, we are interested in studying the regularities and the correspondences between the Chinese graphical forms of characters and their pronunciation.

It is known that Chinese characters exhibit some structure and are made of graphical elements which reflect either some iconic meaning or some pronunciation. As a first step in this study, we extracted all the possible analogies between Sino-Japanese characters using a fixed-sized font. We report hereafter some of the results obtained.

5.1 The structure of Chinese characters

A large number of Chinese characters exhibit some structure concealed in their components. The most known structure consists of two elements, one being a pronunciation clue and the second one being a meaning clue, usually called semantic key. An illustration is given in Figure 4.

京:先	identical left part (semantic key)
涼:洗	氵 [water]
凉:冼	冫 [ice]
倞:侁	亻 [human]

冫:亻	identical right part (pronunciation)	
冸:伴	半	PÀN
凉:倞	京	LIÀNG
冼:侁	先	ĒN

Figure 4. On the left, on each line, the characters share the same semantic key. On the right, the characters share a same pronunciation indicated by the right part.

This structure, although being quite common, is not valid for all characters. It is also believed that, because of phonetic changes, many characters that exhibited such kind of structure in ancient times cannot be interpreted in this way anymore.

In this paper, we are not interested with the relationship between graphical form and pronunciation. Our goal is limited to the extraction of the graphical structure of Chinese characters by automatic means.

5.2 Characters in monospace fonts

Monospace (or fixed-width or fixed-size) fonts are lists of characters described as black and white icons of fixed height and width. The font we use in our experiments is knj10B.bdf[5]. We use the 14,655 Sino-Japanese characters available in this font in the range between the Unicode codepoints 13,312 (㐀) and 40,891 (龻). Figure 5 shows a sample of these characters. As shown in this figure, the characters in this font have a fixed height of 18 lines and a fixed width of 24 pixels. The actual width used is 18 pixels, so that this font is a 18 × 18 pixel font, the value of 24 comes from the encoding of each line of the icons on 3 bytes that we use without change. Figure 5 visualizes the representation of three randomly selected characters from this font.

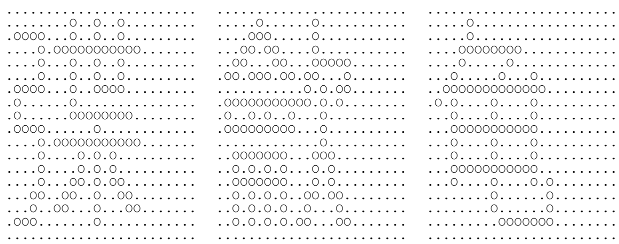


Figure 5. Visualization of some characters from the font knj10B as icons. White pixels are visualized as a dot, black pixels as a small circle.

5.3 Features used

The features we used are simply the number of black pixels on each line and each column. We use 18 lines and 18 columns; making up a sum of 36 features per characters. As an illustration, the first features for the leftmost character in Figure 5 are:

0, 3, 7, 12, 4, 4, 9, 2, 9, 5, etc.

We checked that the space of objects is completely separated by these features, *i.e.*, each vector of feature values represent only one object in the set of objects. The above 36 features discriminate each character among the 14,655 characters used in our experiments.

5.4 Analogical clusters obtained

We applied out method to extract the graphical structure of the 14,655 characters in our selected fixed-point font. The program, written in Python, needed less than 30 minutes to terminate on a machine with 4 Gb memory and equipped with an Intel Core i5 processor with a 1.7 GHz frequency.

Figure 6 shows a sample of 15 clusters output by the method. Visual inspection reveals the typical structure of characters decomposed into a left and a right part. This is in no way surprising given the features that we used and the frequency of this structure.

Figure 7 shows two examples of less usual patterns that consist in the addition of some elements in the middle of characters or a longer stroke in the central part of the characters.

Experiments performed with a lesser number of features show examples of clusters where the differentiation between the characters

[5] Font designed by Nagao Sadakazu (snagao@tkb.att.ne.jp). We use version 1.1 of 1999.

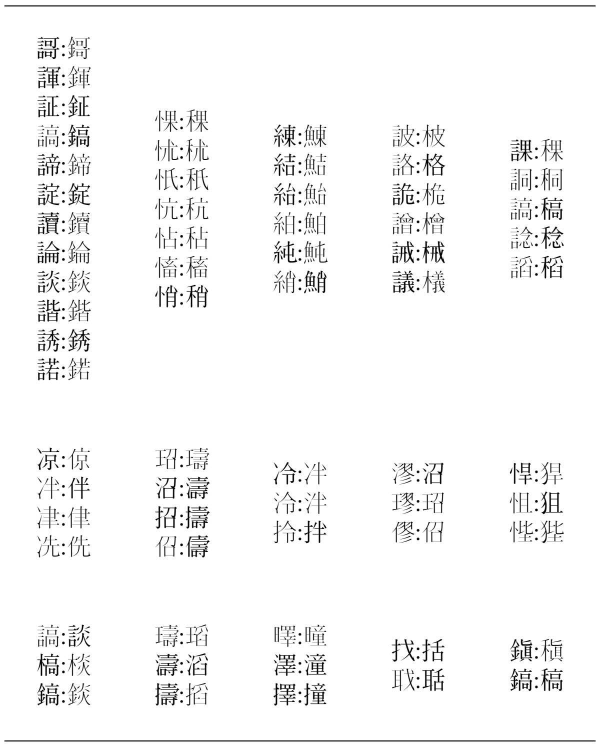


Figure 6. A sample of 15 analogical clusters output by our clustering method on 14,655 characters from the font knj10B. For the first six ones, the tenth one and the last one, both characters on the same line share the same right part (radical). The left parts (keys) are different but common to all lines. Conversely, in the other clusters the right part of the characters is the same in each column of the cluster. These clusters show commutations of various keys (four, three or two only) with only two different radicals.

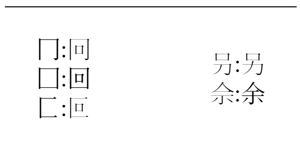


Figure 7. Two examples of less typical cases of analogical clusters output by our clustering method. The first one on the left shows the insertion of a square in the middle of the character. The second one captures a longer stroke in the center of the characters on the right.

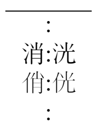


Figure 8. A cluster output by our clustering method using only the number of horizontal black pixels as features. Taking into account the columns eliminates this analogy.

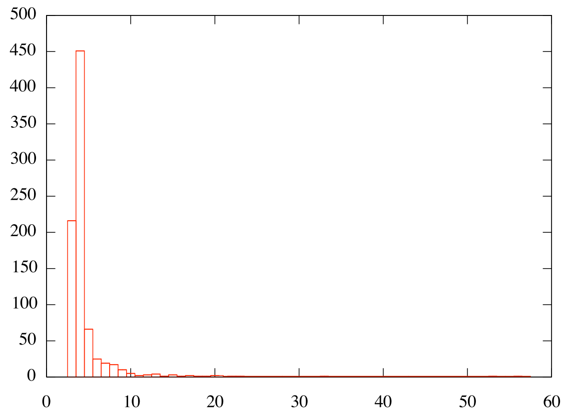


Figure 9. Distribution of clusters by number of pairs. In abscissae, number of pairs in the clusters. In ordinates, number of clusters with the same number of pairs. The largest cluster contains 55 pairs. There are only 16 clusters between 13 and 55 on the horizontal axis.

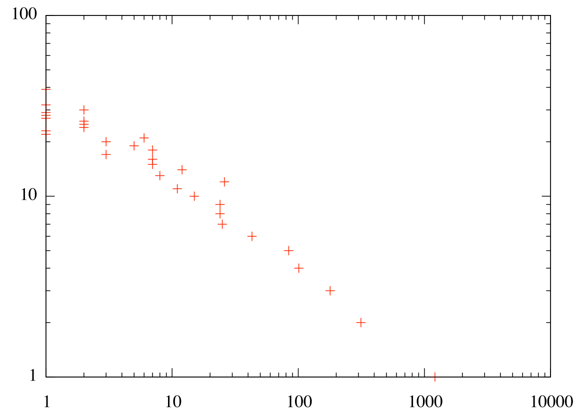


Figure 10. Distribution of characters by number of clusters they appear in. In abscissae, number of characters that appear in the same number of clusters. In ordinates, number of clusters. Logarithmic scales.

is not sufficient. An example of this is given in Figure 8. In this example, only the number of black pixels on the lines have been used (18 features). In this case, the vertical directions of the strokes in the left characters on the first and second lines have not been distinguished so that the method concluded to a proportional analogy that may be questionable (or not for reasons of equivalence between various forms of writing).

The distribution of clusters by number of pairs of objects (36 features) is plotted on Figure 9. This distribution exhibits a long tail. Few clusters are very large while short clusters are more numerous The fact that the number of clusters with 3 pairs of characters (451) is greater than the number of clusters with only two pairs of objects (216) is explained by the elimination of redundant clusters reduced to one analogy. The largest cluster contains 55 pairs of characters.

5.5 Number of clusters per character

The total number of characters that appear in all analogical clusters was 5,982. This represents 41% of the total number of characters used (14,655). We *a priori* expected a higher number.

We also measured the number of (non-redundant) clusters each character appears in. Figure 10 plots of the distribution of characters per number of clusters they appear in. The use of logarithmic scales suggests a Zipfian distribution that needs more enquiry. This measure gives an estimation of the complexity of a character by the number of oppositions it has with all other characters. This reflects its degree of freedom in the overall graphical system. A character which does not appear in any cluster is somehow free relatively to the overall system. In vision of our future experiments, we hypothetize that characters that appear in more clusters may be easier to remember if the learner has access to the global view given by the clusters. In addition, of course, the number of strokes should be taken into consideration.

6 RELATED WORKS AND CONCLUSION

This paper presented a method to automatically extract all possible proportional analogies from a given set of objects represented as feature vectors. In previous works, we showed how to do this for short sentences [10, 11] but extra computation was required to check for the edit distance constraint necessary in proportional analogies of commutation between strings of symbols[6].

Relying on specific properties of our formalization of objects as feature vectors, we defined the ratio between objects as a difference between vectors, and conformity as equality between vectors. This particular setting allowed us to reformulate the problem, which has a complexity of $O(n^4)$, in an equivalent problem with a quadratic complexity, that of enumerating analogical clusters, i.e., lists of pairs of objects with the same ratio.

We proposed an adequate data structure this problem and, by further exploiting the properties of proportional analogies, we showed how to avoid redundancy in the enumeration and non-informative clusters. With all this, we showed that the problem becomes tractable as as to solve our problem at hand: extracting all analogies between Sino-Japanese characters in their graphical form.

Although the iconicity of proportional analogies has already been stressed in a broader sense of the word [3, 4], this is the first attempt at solving analogies between icons of black and white pixels using their graphical form directly. This problem had already been mentioned in [8] but without a solution. Our formalization and application to Sino-Japanese characters shows that an explicit description of characters in terms of their constituents (keys or radicals) as is proposed in [20] can be avoided.

Our proposed method does not exhaust the subject of proportional analogies between icons made of black and white pixels. There remains a number of problems. One problem is the necessary fixed size of the icons to compare, hence our use of monospace fixed-size fonts. Firstly, any shift of a character by one or several lines (or columns) would disrupt the analogical relations that are made possible to compute with our feature vectors because almost all characters are well lined up with the first line and column. Secondly, analogical relations between characters of different sizes cannot obviously be captured with the method proposed here.

The work presented here is a preprocessing step in a larger study

[6] [6] is the first mention of the edit distance constraint in terms of similarities; [7] gives the equivalent expression with edit distances; [9] is the published form of the proceeding in which [7] appeared, with few years delay. The edit distance constraint is necessary between strings of symbols to avoid too many spurious analogies that would be formed without it. Experiments with several hundreds of thousands of short sentences in Chinese collected from the Web confirm this point.

of proportional analogies of graphical form and pronunciation among Chinese or Sino-Japanese characters. We perform the same kind of analogical clustering on the level of pronunciation and compute the intersection between analogical clusters on the graphical and on the pronunciation levels. We hypothesize that knowing analogical correspondences between the graphical and pronunciation levels of Chinese or Sino-Japanese characters would ease their memorization by learners. We intend to test this hypothesis with subjects.

REFERENCES

[1] Dedre Gentner, 'Structure mapping: A theoretical model for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).

[2] Robert R. Hoffman, 'Monster analogies', *AI Magazine*, **11**, 11–35, (1995).

[3] Esa Itkonen, 'Iconicity, analogy, and universal grammar', *Journal of Pragmatics*, **22**(1), 37–53, (1994).

[4] Esa Itkonen, 'Analogy as structure and process: Approaches in linguistics, cognitive psychology and philosophy of science', in *Human cogntive processing*, eds., Marcelo Dascal, Raymond W. Gibbs, and Jan Nuyts, volume 14, 250 p., John Benjamins Publishing Company, Amsterdam / Philadelphia, (2005).

[5] Philippe Langlais and François Yvon, 'Scaling up analogical learning', in *Coling 2008: Companion volume: Posters*, pp. 51–54, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[6] Yves Lepage, 'Languages of analogical strings', in *Proceedings of COLING-2000*, volume 1, pp. 488–494, Saarbrücken, (July-August 2000).

[7] Yves Lepage, 'Analogy and formal languages', in *Proceedings of FG/MOL 2001*, pp. 1–12, Helsinki, (August 2001).

[8] Yves Lepage, *Of that kind of analogies capturing linguistic commutations (in French)*, Habilitation thesis, Joseph Fourier Grenoble University, May 2003.

[9] Yves Lepage, 'Analogy and formal languages', *Electronic notes in theoretical computer science*, **53**, 180–191, (April 2004).

[10] Yves Lepage, 'Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus', in *Proceedings of COLING-2004*, volume 1, pp. 736–742, Genève, (August 2004).

[11] Yves Lepage and Chooi-ling Goh, 'Towards automatic acquisition of linguistic features', in *Proceedings of the 17th Nordic Conference on Computational Linguistics (NODALIDA 2009)*, eds., Kristiina Jokinen and Eckard Bick, pp. 118–125, Odense, (May 2009).

[12] Yves Lepage, Julien Migeot, and Erwan Guillerm, 'A corpus study on the number of true proportional analogies between chunks in two typologically different languages', in *Proceedings of the seventh international Symposium on Natural Language Processing (SNLP 2007)*, pp. 117–122, Pattaya, Thailand, (December 2007). Kasetsart University, ISBN 978-974-623-062-9.

[13] Yves Lepage, Julien Migeot, and Erwan Guillerm, 'A measure of the number of true analogies between chunks in Japanese', *Lecture Notes in Artificial Intelligence*, **5603**, 154–164, (2009).

[14] Hermann Paul, *Prinzipien der Sprachgeschichte*, Niemayer, Tübingen, 1920.

[15] Peter Turney, 'A uniform approach to analogies, synonyms, antonyms, and associations', in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 905–912, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[16] Peter D. Turney, 'Similarity of semantic relations', *Computational Linguistics*, **32**(2), 379–416, (2006).

[17] Peter D. Turney and Michael L. Littman, 'Corpus-based learning of analogies and semantic relations', *Machine Learning*, **60**(1–3), 251–278, (2005).

[18] Marcus Terentius Varro, *De lingua latina*, Coll. Belles-lettres, Paris, 1954. Trad. J. Collart.

[19] Tony Veale and Shanshan Chen, 'Learning to extract semantic content from the orthographic structure of Chinese words', *proceedings of the 17th Irish conference on Artificial Intelligence and Cognitive Science (AICS2006)*, (sept 2006).

[20] Lars Yencken and Timothy Baldwin, 'Measuring and predicting orthographic associations: Modelling the similarity of Japanese kanji', in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 1041–1048, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[21] François Yvon, Nicolas Stroppa, Laurent Miclet, and Arnaud Delhay, 'Solving analogical equations on words', Rapport technique ENST2004D005, ENST, (July 2004).

Arguing by analogy – Towards a formal view

A preliminary discussion

Leila Amgoud and **Youssef Ouannani** and **Henri Prade**[1]

Abstract. Although arguing by analogy is a current practice, little attention has been devoted to the study of this form of argumentation, especially at a formal level. This research note provides a preliminary study of what could be done in that direction. The discussion relies in particular on a logical modeling of analogical proportions (i.e., statements of the form "A is to B as C is to D"), in terms of similarity and dissimilarity, which has been recently proposed.

1 Introduction

The use of analogies plays an important role in many reasoning tasks, and analogical reasoning is usually recognized as a powerful, although heuristic, way to look for solutions by adaptation of existing ones, to jump to plausible conclusions, or to boost creativity in various areas [5] (including ancient [29] as well as modern mathematics [21]).

Analogical reasoning has been extensively studied in cognitive psychology [10, 8] and has been implemented in computational models [7, 17, 28]. The use of analogies in argumentation is often encountered, since arguments based on analogies are easy to grasp, are intuitively appealing, and may be especially convincing in public uses. However, analogical argumentation has been little studied if we except some works by philosophers [13, 2, 31] or linguists [20], or studies in legal reasoning [12, 14, 19]. Although argumentation has been extensively studied in artificial intelligence in the last two decades (see, e.g., [26], analogical arguments have almost not been considered (an exception is [3, 4]).

A reason for this state of fact might be related to the difference of nature between deductive reasoning and analogical reasoning. Deductive reasoning relies on a well known formal apparatus developed for a long time, and provides conclusions that are as much reliable as the premises are. While deductive reasoning handles generic knowledge as well as pieces of factual evidence, analogical reasoning rather considers particular cases or situations, and is much more brittle since it only provides tentative conclusions. Moreover, the formal studies of analogical reasoning, even if there has been a number of proposals, remain less developed and somewhat scattered, and roughly speaking, analogical reasoning is often thought as something which is beyond logic. This probably contributes to make more difficult a formal theory of argumentation able to handle analogical arguments.

[1] University of Toulouse, IRIT-CNRS, France, email: amgoud, youssef.ouannani, prade@irit.fr

In this short note, we take advantage of the existence of a propositional logic modeling of analogical proportions, (i.e., statements of the form "A is to B as C is to D") that has been recently developed, for offering some analysis of analogical arguments and for suggesting a formal view of their treatment. The rest of the paper is organized in two main parts. We first present an introductory overview of analogical reasoning based on analogical proportions, and then propose a preliminary study of analogical arguments.

2 A brief introduction to formal analogical reasoning

Analogy is currently understood as a weak form of similarity. For many authors, when comparing two objects S and T, one has to distinguish between identity, resemblance, and analogy. Resemblance is strictly weaker than identity. The fact that S resembles T if they belong to the same domain and have common features (which are easily observable), while S is analogous to T rather means that S and T may belong to different domains, and that S has the same relation with an object U as T has with another object V [11]. For instance, taking a famous example from Aristotle, "*Fish* (S) breathe through their *gills* (U), *mammal* (T) breathe through their *lungs* (V)". This idea of viewing analogy as making a parallel between two system of objects, each related by similar relations, or even equations, has been investigated for a long time (see, e.g., [33]), and is at the core of the structure-mapping model [9, 7].

Case-based reasoning [1] also relies on the comparison between two pairs, which may be denoted $(Prob_1, Sol_1)$ and $(Prob_2, Sol_2)$, where $Prob_1$ and $Prob_2$ are the multiple-features descriptions of two problems, whose solutions Sol_1 and Sol_2 are respectively known and unknown. Case-based reasoning then amounts to suggest that Sol_2 may be obtained by adapting Sol_1 on the basis of the similarities and differences between $Prob_1$ and $Prob_2$. Indeed analogy is as much a matter of dissimilarity as a matter of similarity. This what has been also put in evidence in the logical definition of an analogical proportion, which is now recalled.

2.1 A propositional logic view of an analogical proportion

An analogical proportion is a statement of the form "A is to B as C is to D", often denoted as $A : B :: C : D$, where A, B, C, D stand for objects, or situations. They may be described by means of sets of features. We assume here for simplicity that these features are binary. Thus, each of A, B, C, and D may be viewed as sets of properties (possessed by the corresponding items). Then, one may say that the analogical proportion $A : B :: C : D$ holds if we have both

$$A \cap \overline{B} = C \cap \overline{D} \text{ and } \overline{A} \cap B = \overline{C} \cap D,$$

where the overbar denotes set complementation. This means that "A differs from B as C differs from D and B differs from A as D differs from C". This definition [18] refines previous proposals [15, 30]. A logical counterpart of this idea [18] amounts to state that for each binary feature viewed as a Boolean variable, the corresponding analogical proportion, denoted $a : b :: c : d$, is specified by the following pair of constraints:

$$((a \to b) \equiv (c \to d)) \wedge ((b \to a) \equiv (d \to c)) \text{ is true}$$

Thus, the proportion $a : b :: c : d$ can now be viewed as a Boolean formula. It can checked that it takes the truth value 1 only for the 6 following 4-tuples (among 16 possible patterns) that are shown in Table 1. For all other valuations of (a, b, c, d), the formula $a : b :: c : d$ is false. It is easy to check that such a formal definition satisfies the properties that are usually assumed for an analogical proportion, namely:

Table 1. Truth table for analogical proportion

a	b	c	d
0	0	0	0
1	1	1	1
0	0	1	1
1	1	0	0
0	1	0	1
1	0	1	0

- $a : b :: a : b$ and $a : a :: b : b$ hold, but $a : b :: b : a$ does not hold in general;
- if $a : b :: c : d$ holds then $a : c :: b : d$ should hold (central permutation);
- if $a : b :: c : d$ holds then $c : d :: a : b$ should hold (symmetry).

The reader is referred to [18] for a study of other properties with respect to connectives (e.g., $a : b :: \neg b : \neg a$ holds), to [24] for an overview of related notions and potential applications, and to [25] for illustrations and the relation to case-based reasoning.

2.2 Analogical-proportion based inference

An analogical equation $a : b :: c : x$ where the value of x is unknown, is solvable iff $(a \equiv b) \vee (a \equiv c)$ holds (e.g., $0 : 1 :: 1 : x$ has no solution). *When it exists*, the unique solution is given by $x = c \equiv (a \equiv b)$. This can be stated in terms of propositional logic entailments such as $a : b :: c : d, a, b, c \vdash d$, together with 5 other similar entailments (e.g., $a : b :: c : d, \neg a, b, \neg c \vdash d$) [24]. It provides a basis for the following inference pattern

$$\frac{\forall i \in [1, m],\ \ a_i : b_i :: c_i : d_i}{\forall j \in [m+1, n],\ \ a_j : b_j :: c_j : d_j}$$

In simple terms, this means that *if the known part of the vector encoding D, say $(d_1, \cdots, d_m)$ is componentwise in analogical proportion with the corresponding parts $(a_1, \cdots, a_m)$, $(b_1, \cdots, b_m)$, $(c_1, \cdots, c_m)$ of the vectors encoding A, B, and C, then it should be also true for the unknown part $(d_{m+1}, \cdots, d_n)$ of the vector encoding D with respect to the corresponding parts of A, B, and C.* Thus, if $(a_{m+1}, \cdots, a_n)$, $(b_{m+1}, \cdots, b_n)$, $(c_{m+1}, \cdots, c_n)$ are known, then $(d_{m+1}, \cdots, d_n)$ can be obtained by equation solving. This extrapolation is exactly what analogical reasoning is about: we *transfer* the knowledge we have on the pair (A, B) to the pair (C, D) to predict the missing information about D, assuming a kind of regularity property. This is has been applied to classification problems, see [24] for references. This is obviously a form of reasoning that is not sound, but which may be useful for trying to guess unknown values.

A basic pattern considered when trying to formalize analogical reasoning in the setting of first order logic (see, e.g., [27]) is the following: We have two objects represented by terms s and t, we observe that they share a property P, and knowing that another property Q also holds for s, we are tempted to infer that it holds for t as well (a conclusion that may turn to be plausible especially if some dependency is suspected between P and Q). This "analogical jump" corresponds to the following simple inference pattern:

$$\frac{P(s),\ P(t),\ Q(s)}{Q(t)}$$

The above pattern may be directly related to the idea of analogical proportion: One may consider that "P(s) is to P(t) as Q(s) is to Q(t)" (indeed they are similar changing s into t), or by central permutation that "P(s) is to Q(s) as P(t) is to Q(t)" (changing P into Q), the above pattern may be restated as

$$\frac{\begin{array}{c}P(s) : P(t) :: Q(s) : Q(t)\\ P(s), P(t), Q(s)\end{array}}{Q(t)}$$

which is a valid pattern of inference, from the propositional logic view of the analogical proportion. Similarly, on may consider richer patterns involving n-ary predicates, such as from $P(s), R(s, s'), Q(s'), P(t), R(t, t')$ infer $Q(t')$, which may correspond to the analogical proportion $P(s) \wedge R(s, s') : P(t) \wedge R(t, t') :: Q(s) : Q(t)$, itself possibly extrapolated from $P(s) : R(s, s') :: P(t) : R(t, t')$ and $R(s, s') : Q(s') :: R(t, t') : Q(t')$.

All the above patterns are quite different at first glance from a pattern of analogical reasoning proposed by Polya [22], which is now recalled.

2.3 Polya's pattern of analogical reasoning

Polya [21] advocates the idea that analogical reasoning plays an important role when trying to solve problems in mathematics. Later, in [22] he proposed patterns of plausible reasoning in order to provide a more accurate view of reasoning mechanisms at work in problem solving. One of these patterns reads:

$$\frac{\begin{array}{c}a \text{ and } b \text{ are analogous}\\ a \text{ is true}\end{array}}{b \text{ true is more credible}}$$

In [23], a modeling of "a and b are analogous", denoted $a \sim b$, has been proposed using a preferential nonmonotonic consequence relation $|\!\!\sim$, as $a \sim b$ iff $|\!\!\sim a \equiv b$. Clearly, $a \sim b$ iff $\neg a \sim \neg b$ holds. Semantically speaking, it amounts to state that $\Pi(a \equiv b) > \Pi(\neg(a \equiv b))$, where Π is a possibility measure based on a possibility distribution that rank-orders the interpretations. Viewing a and b

as compound descriptions of situations, and using a possibility distribution on the features for assessing their importance, another more intuitive view would amount to say that a and b are analogous as soon as they only differ on *non important* features. The following patterns have been established (among others) [23]

$$\frac{\mid\!\sim a : b :: c : d \qquad a \sim b}{c \sim d}$$

$$\frac{a \sim b \quad c \sim d}{\mid\!\sim a : b :: c : d}$$

This shows a good agreement between the analogical proportion view and the relation $\sim$. The first inference pattern may be illustrated by an example mentioned by Aristotle [6]: Iphicrates, an Athenian general, provided the following argument about his son for whom one wanted that he serves in a public position, "if one deals with adults as tall children, are we going to deal with short adult as children?". Indeed, it can be checked that $tall\ child : adult :: child : short\ adult$ holds (considering that child and adult are normally short and tall respectively. Then considering that $tall\ child \sim adult$ leads to admit that $child \sim short\ adult$.

3 Analogical argumentation

Let us start by quoting [2]: "An *analogy* is a comparison between two objects, or systems of objects, that highlights respects in which they are thought to be similar. *Analogical reasoning* is any type of thinking that relies upon an analogy. An *analogical argument* is an explicit representation of analogical reasoning that cites accepted similarities between two systems in support of the conclusion that some further similarity exists." This well summarizes the basic issues.

In the previous section, we have recalled different patterns of analogical inference, and pointed out how they are underlain by the notion of analogical proportion, which itself puts in balance the ideas of dissimilarity and similarity. These different patterns provide a formal basis for discussing different issues regarding analogical arguments:

- how analogical argumentation differs from argumentation based on deductive reasoning,
- what kinds of attack exist against such arguments,
- how to evaluate analogical arguments.

An argument by analogy involves at least one premise which refers to an analogy, and as such departs from deductive (as well as inductive, or abductive) arguments [13]. An analogy may be a simple statement relating two objects "a is analogous to b" (or "a is like b"), or the statement of an analogical proportion. Thus, Polya's pattern of plausible reasoning provides the simplest form of argument by analogy, which departs from a deductive argument "a is true" and "a implies b", then "b is true".

Note also that rather stating "a is like b", one may use premises of the form "Objects A and B are similar in having properties P_1, ..., P_n", making explicit in what respects the objects are analogous. For instance, given that "Peter is like Paul, they like good life", and that "Paul spoilt his fortune in a few years", one may argue that "Peter (who is presently rich) will do the same". An example of argument involving an analogical proportion is the following: "credit rating agencies are useful", since "credit rating agency is to crisis as thermometer is to fever" and "thermometers are useful".

Analogical argument, as any argument may be attacked, or used in attacks against other arguments (which may have or not an analogy form). An example of this latter case, is provided by the Iphicrates example, where the analogical proportion is not challenged. On the contrary, it is used to show that given this analogical proportion, as soon as one accepts to consider $a =$ tall child and $b =$adult as analogous, one is led to accept an absurd conclusion, namely considering $c =$ child and $s =$ short adult as analogous.

An analogical argument may be attacked by

- *disputing the relevance of the similarities* that are pointed out (in terms of features or relations) with respect to the conclusion. This amounts in the "analogical jump" pattern of the previous section to say that properties P and Q are in fact unrelated. This may be done by providing a kind of *counterexample* by pointing out an object for which property P is true, but for which property Q is false.
- *disputing the alleged similarity* between two objects, or *challenging an analogical proportion* by pointing out that the two objects are in fact dissimilar with respect to another (relevant) property, or by exhibiting another (relevant) feature where the analogical proportion fails to hold. Thus, if we take the "credit rating agency" example, the analogy can becomes debatable once we remark that "credit rating agencies have an effect on the crisis" while "thermometers have no effect on the fever".
- *pointing out undesirable consequences*. A well-known example is given by the philosopher David Hume who attacked the teleological argument according to which since a complex object like a watch requires an intelligent designer, a (more) complex object like the universe should also have an intelligent designer. Apart from attacks of the two previous types, Hume argued for instance that since watches are often the result of the work of several people, the reasoning support polytheism as well.

Besides, it is also of interest to notice that a sequence of analogical arguments may be also lead to consider analogical proportions. Typically in a debate, a discussant d may state that situation $S2$ is like situation $S1$ and that what took place in $S1$ will happen in $S2$ as well. The opponent, discussant d', will argue that in fact there is an (important) feature where they differ, and that what took place in $S1$ may not happen in $S2$. Then d may produce another pair of situations $S3, S4$, where the same difference can be observed without affecting the conclusion advocated by d for $S2$. Then d' may counter-argue if he knows another pair of situations $S'3, S4'$ where the same difference does lead to a different conclusion. Thus this kind of exchange can be analyzed in terms of analogical proportions. Indeed, depending if we consider $S3 : S4 :: S1 : S2$, where the same effects have been observed for $S1, S3, S4$, or if we consider $S'3 : S'4 :: S1 : S2$ where different effects have been reported, on may conclude in opposite ways about $S2$ (using the transfer pattern of the previous section for inferring new analogical proportions). It suggests that analogical proportions should play a role in the analysis of analogical arguments.

4 Concluding remarks

Analogical argumentation, although it is currently used in practice, and has been discussed by philosophers, has received very little attention in artificial intelligence until now. The study of [3, 4] based on the structure-mapping model appears to be an exception. This research note has tried to provide some formal basis for the analysis of analogical arguments, by emphasizing the role played by analogical proportions in providing a logical view of analogical reasoning. What has been presented is clearly preliminary and much remains to be done for developing a formal model for analogical argumentation.

REFERENCES

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

[2] P. F. A. Bartha. *By Parallel Reasoning: The Construction and Evaluation of Analogical Arguments*. Oxford University Press, 2009.

[3] A. Bohan and M. T. Keane. Towards a computational model of analogical arguments. *Proc. of 4th Workshop on Computational Models of Natural Argument (CMNA IV)*, at ECAI'04, (F. Grasso, C. Reed and G. Carenini, eds.), 65–67, 2004.

[4] A. Bohan and M. T. Keane. Boosting analogical arguments: The effects of goodness & complexity on everyday arguments. *Proc XXVIIth Ann. Conf. Cognitive Sci. Soc. (CogSci05)*, Stresa, Italy, July 21-23, 304–309, 2005.

[5] E. de Bono. *Lateral Thinking. Creativity Step by Step*. Harper & Row, Publ., New York, 1970.

[6] M. Dorolle. *Le Raisonnement par Analogie*. PUF, Paris, 1949.

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

[8] R. M. French. The computational modeling of analogy-making. *Trends in Cognitive Sciences* 6(5),200 – 205, 2002.

[9] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2), 155–170, 1983.

[10] D. Gentner, K. J. Holyoak, and B. Kokinov (Eds.). *The Analogical Mind: Perspectives from Cognitive Sciences*, MIT Press, 2001.

[11] J. - B. Grize. Logique, analogie et identité. In: *La Logique naturelle. Enjeux et Perspectives*, Actes du colloque de Neuchâtel, 12-13 Sept. 2008 Travaux du Centre de Recherches Sémiologiques 68, Université de Neuchâtel, 91–98, 2010.

[12] M. Guarini. A defense of non-deductive reconstructions of analogical arguments. *Informal Logic*, 24 (2), 153–168, 2004.

[13] A. Juthe. Argument by analogy. *Argumentation*, 19, 1–27, 2005.

[14] H. Kloosterhuis. The normative reconstruction of analogy argumentation in judicial decisions: A pragma-dialectical perspective. In: *Practical Reasoning, Proc. Inter. Conf. on Formal and Applied Practical Reasoning (FAPR '96)*, (D. M. Gabbay, H. J. Ohlbach, eds.), Bonn, June 3-7, 1996, Springer, LNCS 1085, 375–383, 1996.

[15] Y. Lepage. Analogy and formal languages. *Electr. Notes Theor. Comput. Sci.*, 53, 2001.

[16] G. Marchant, J. Robinson, U. Anderson, M. Schadewald. The use of analogy in legal argument: Problem similarity, precedent, and expertise. *Organizational Behavior and Human Decision Processes*, 55 (1), 95–119, 1993.

[17] E. Melis and M. Veloso. Analogy in problem solving. In: Handbook of Practical Reasoning: Computational and Theoretical Aspects. Oxford Univ. Press, 1998.

[18] L. Miclet and H. Prade. Handling analogical proportions in classical logic and fuzzy logics settings. *Proc. 10th Europ. Conf. on Symb. and Quantit. Appr. to Reasoning with Uncertainty (ECSQARU09)*, (C. Sossai, G. Chemello, eds.), Verona, Jul. 1-3, Springer, LNCS 5590, 638–650, 2009.

[19] Y. Okubo and M. Haraguchi. Attacking legal argument by examining stability of case citation with goal-dependent abstraction. *Proc. 6th Inter. Conf. on Artificial Intelligence and Law (ICAIL '97)*, June 30 - July 3, Melbourne, ACM, 191–197, 1997.

[20] C. Plantin. Analogie et métaphore argumentatives. In: L'Argumentation au Carrefour des Disciplines : Sciences du Langage et Sciences Sociales, (R. Micheli, ed.), *A contrario*, 2, n°16, 110–130, 2011.

[21] G. Polya. *How to Solve it*. Princeton University Press, (2nd ed. 1957), 1945.

[22] G. Polya, *Mathematics and Plausible Reasoning*. Vol.2: Patterns of Plausible Inference. Princeton Univ. Press, 2nd ed. 1968, 1954.

[23] H. Prade and G. Richard. Cataloguing / analogizing: A nonmonotonic view. *Inter. J. of Intelligent Systems*, 26, 117–1195, 2011.

[24] H. Prade and G. Richard. Logical handling of analogical proportions in commonsense and transductive reasoning. *Proc. 3rd Inter. Conf. Soft Computing and Pattern Recognition (SoCPaR'11)*, IEEE, 561– 566, 2011.

[25] H. Prade and G. Richard. Analogy-making for solving IQ tests: A logical view. In: *Case-Based Reasoning Research and Development, Proc. 19th Inter. Conf. on Case-Based Reasoning (ICCBR'11)*, (A. Ram, N. Wiratunga, eds.), London, Sept.12-15, LNCS 6880, Springer, 241–257, 2011.

[26] I. Rahwan (Ed.). *Argumentation in Artificial Intelligence*. Springer, 2009.

[27] S. J. Russell. *The Use of Knowledge in Analogy and Induction*. Pitman, London, 1989.

[28] J. F. Sowa and A. K. Majumdar. Analogical reasoning. In: *Conceptual Structures for Knowledge Creation and Communication, Proc. 11th Inter. Conf. on Conceptual Structures (ICCS'03)*, (A. de Moor, W. Lex, and B. Ganter, eds.), Dresden, July 21-25, Springer, LNAI 2746, 16–36, 2003.

[29] G. Stamatellos. Argument by analogy in Thales and Anaximenes. In: *Just the Arguments: 100 of the Most Important Arguments in Western Philosophy*, (M. Bruce, S. Barbone, eds.), Wiley-Blackwell, 196–199, 2011.

[30] N. Stroppa, and F. Yvon. 2006. Du quatrième de proportion comme principe inductif : une proposition et son application à lapprentissage de la morphologie. *Traitement Automatique des Langues*, 47(2), 1–27.

[31] D. Walton. Similarity, precedent and argument from analogy. *Informal Logic*, 32 (2), 190–218, 2012.

[32] P. R. Wilson. On the argument by analogy. *Philosophy of Science*, 31 (1), 34–39, 1964.

[33] P. H. Winston. Learning and reasoning by analogy. *Commun. of ACM*, 23 (12), 689703, 1980.