Analogical Learning and Formal Proportions: Definitions and Methodological Issues

Nicolas Stroppa, François Yvon
GET/ENST & LTCI, CNRS UMR 5141
46 rue Barrault - F-75013 Paris
{stroppa,yvon}@enst.fr

Apprentissage par analogie et proportions formelles : définitions et aspects méthodologiques

Résumé

L'apprentissage par analogie exploite un mécanisme inductif en deux étapes consistant à : (i) construire un appariement entre une nouvelle situation et des situations déjà connues ; (ii) transférer partiellement des propriétés de la situation analogue vers la situation nouvelle. Cette approche présuppose la capacité à rechercher et à exploiter de tels appariements, ce qui implique de donner un sens à la notion de relation analogique et d'implanter efficacement leur calcul.

Dans cet article, nous proposons une définition unifiée de la notion de proportion analogique valant pour un grand nombre de structures algébriques. Nous montrons que cette approche est adaptée à la manipulation de grandes bases de données d'objets structurés, rencontrées notamment dans les tâches de Traitement Automatiques des Langues. Nous discutons cette approche et la comparons à d'autres méthodes d'apprentissage à partir d'instances.

Abstract

Analogical learning is a two-step inference process: (i) computation of a mapping between a new and a memorized situation; (ii) transfer from the known to the unknown situation. This approach requires the ability to search for and exploit such mappings, which are based on the notion of analogical proportions, hence the need to properly define these proportions, and to efficiently implement their computation.

In this paper, we propose a unified definition of analogical proportions, which applies to a wide range of algebraic structures. We show that this definition is suitable for learning in domains involving large databases of structured data, as is especially the case of many Natural Language Processing applications. We finally discuss some issues this approach raises and relate it to other instance-based learning schemes.

1 Introduction

Analogical learning is a two-step inductive process. The first step consists in the construction of a mapping between a new instance of a problem and solved instances of the same problem. Once this mapping is established, solutions for the new instance can be induced, based on one or several analogs.

This general procedure has been developed in several directions: on the one hand, work on analogical reasoning Gentner *et al.* (2001) has focused on the definition and computation of complex *structural* mappings between instances; instance-based learners Aha (1997) such as k-NN, on the other hand, rely on a notion of analogy which is primarily based on similarity computations. While the former have mainly worked with small databases of highly structured objects, the latter have been able to cope with large databases of non-structured (typically vectors in $\mathbb{R}^n$) objects.

In several application domains, such as Natural Language Processing, both difficulties simultaneously occur: linguistic representations (e.g. strings, trees, feature-structures) are intrinsically complex and multidimensional; databases (lexicons, tree banks, etc.) often contain hundreds of thousands of instances. In this context, the use of instance-based learners has been considered (e.g. Daelemans *et al.* (1999)); this however requires flat (vectorized) representations of the data.

Alternative proposals aim at preserving the structural nature of linguistic representations (e.g. Pirrelli & Yvon (1999); Lepage (1999*b*,*a*)). This is achieved through the computation of *formal proportions*, which are often a good sign of deep similarities between objects. The main insight of these models is best understood on an example: the word strings *write* and *writer* are orthographical neighbors. Orthographic similarities can be misleading; in order to minimize such confusions, these models attempt to exploit the systemic organization and the redundancy of linguistic data. As a consequence, inference is directed by the finding of *recurrent similarity relationships*: other pairs such as *read-reader*, *review-reviewer*, which form analogical proportions with *write-writer* will serve as prime evidence of the deep similarity between these terms. These models have mainly focused on sequential representations of linguistic data, using restrictive definitions of proportions.

The instance-based learner introduced hereafter draws much inspiration from these works, and uses a similar inference procedure. Our contribution is twofold:

- to propose a general definition of formal analogical proportions for widely-used algebraic structures such as attribute-value vectors, words on finite alphabets, feature-structures and labeled trees. For each of these structures, we provide efficient algorithms able to process large instance databases.

- to show how these definitions can be used to perform effective learning for various linguistic analysis tasks, thus bridging the gap between a symbolic approach and a more statistically-oriented notion of analogical learning.

This paper is organized as follows. Section 2 introduces our interpretation of analogical learning and relates it to other models of analogical learning and reasoning. Section 3 presents a general definition of analogical proportions as well as its instantiation for various algebraic structures. Section 4 explores in more depth the relationships between our model and traditional instance-based learning; it also relates our approach with other algebraic definition of proportions. We conclude by discussing current limitations of this model and by suggesting possible extensions.

2 Principles of analogical learning

2.1 Analogical reasoning

The ability to identify analogical relationships between what looks like unrelated situations, and to use these relationships to solve complex problems, lies at the core of human cognition Gentner *et al.* (2001). A number of models of this ability have been proposed, based on symbolic (e.g. Falkenhainer *et al.* (1989); Thagard *et al.* (1990); Hofstadter & Mitchell (1995)) or subsymbolic (e.g. Plate (2000); Holyoak & Hummel (2001)) approaches. The main focus of these models is the dynamic process of analogy making, which involves the identification of a structural mapping between a memorized and a new situation, which, while looking very different, share a set of high-level relationships. Such mappings rely on the complex internal structure of the representations that are manipulated: good analogies involve several subparts of these representations and the relationships between them.

Analogy-making seems to play a central role in our reasoning ability; it is also invoked to explain some human skills which do not involve any sort of conscious reasoning. This is the case for many tasks related to the perception and production of language: lexical access, morphological parsing, word pronunciation, etc. In this context, analogical models are primarily seen as an alternative to rule-based approaches: the focus lies more on the generalization performance in the face of large amount of data than on the subtlety of the matching procedure. Various implementation of these low-level analogical processes have been proposed: decision trees, neural networks or instance-based learners (see e.g. Skousen (1989); Daelemans *et al.* (1999)), which have mainly considered flattened representations of linguistic data.

The model we propose lies somehow between these two approaches: we try to remain faithful to the idea of systemic, structural analogies, which prevails in the AI literature, and which is well suited for structured data, while also trying to exploit the main intuitions of large-scale, instance-based learning models.

2.2 Analogical learning

We consider the following supervised learning problem: a learner is given a set $\mathcal{S}$ of training instances $\{X_1, \ldots, X_n\}$. Each instance X_i is a vector of the form $\langle X_{i1}, \ldots, X_{im} \rangle$, whose components (also called features) may take various forms. Given $\mathcal{S}$, the learning task consists in predicting the features of partially informed new instances. Put in more standard terms, the set of known (resp. unknown) features for a new instance X forms the *input space* (resp. *output space*). This supervised learning setting is more general than the classification task, in which only one feature (a class label) is induced.

In our model, training instances are simply stored for future use: no generalization (abstraction) of the data is performed, which is characteristic of *lazy* *learning* Aha (1997); Mitchell (1997). Given a new instance X, we identify formal analogical proportions involving X in the input space, in order to reconstruct its missing features using analogical proportions in the output space. In the sequel, we describe the inference process in more details.

An analogical proportion is a relation involving four objects A, B, C and D, denoted by $A : B :: C : D$ and which reads *A is to B as C is to D*. The definition and computation of these proportions are studied in Section 3. For the moment, we contend that it is possible to construct analogical proportions between (possibly partially informed) vectors in $\mathcal{S}$. When a vector A is partially informed, $I(A)$ (resp. $O(A)$) denotes its known (resp. unknown) part.

Let $I(X)$ be the known part of a new instance X. The inference process is formalized as follows:

1. Construct the set $\mathcal{T}(X) \subset \mathcal{S}^3$ defined as:

$$\mathcal{T}(X) = \{(A, B, C) \in \mathcal{S}^3 \; |I(A) : I(B) :: I(C) : I(X)\}$$

2. For each $(A, B, C) \in \mathcal{T}(X)$, compute hypotheses $\widehat{O(X)}$ by solving the equation:

$$\widehat{O(X)} = O(A) : O(B) :: O(C) :?$$

This inference procedure shows lots of similarities with the k-nearest neighbors classifier (k-NN) which, given a new instance, (i) searches the training set for close neighbors, (ii) compute the unknown class label according to the neighbors' labels. However, in our setting, no metric is required: we only rely on the definition of analogical proportions, which reveal systemic, rather than superficial, similarities. Implementing this procedure requires to address two specific issues.

- When exploring $\mathcal{S}^3$, an exhaustive search strategy evaluates $|\mathcal{S}|^3$ triples, which can prove to be intractable. Moreover, objects in $\mathcal{S}$ may be unequally relevant, and we might expect the search procedure to treat them accordingly.

- Whenever several competing hypotheses exist in $\widehat{O(X)}$, a ranking of these hypotheses must be performed. In our current implementation, hypotheses are ranked by a majority voting procedure.

These issues are well-known problems for k-NN classifiers. The second one does not seem critical, and is usually solved based on frequency counts. In contrast, a considerable amount of effort has been devoted to reduce and optimize the search process, via editing and condensing methods, as studied e.g. in Dasarathy (1990); Wilson & Martinez (2000). Proposals for solving this problem are discussed in Section 4.

3 An algebraic framework for analogical proportions

We have assumed, in Section 2.2, that a mechanism for computing analogical proportions was available. In this Section, we propose an algebraic framework for defining analogical proportions between structured data. After giving the general definition (Section 3.1), we present its instantiations for various structures such as lattices, free monoids and trees.

The computation of analogical proportions has been implemented in a generic analogical solver based on Vaucanson, an automata manipulation library using generic programming Lombardy *et al.* (2003).

3.1 Analogical proportions

The notion of an analogical proportion involves two key ideas: the *decomposition* of objects into smaller parts bound to satisfy *alternation constraints.* In the sequel, $(U, \oplus)$ will denote a semigroup, i.e. a set provided with an associative internal composition law $\oplus$. To formalize the idea of decomposition, we first introduce the notion of *factorization* of an element u of U, defined as follows.

Definition 1.
A factorization *of $u \in U$ is a sequence $u_1 \ldots u_n$, with $\forall i, u_i \in U$, such that: $u_1 \oplus \ldots \oplus u_n = u$.*

This definition is consistent with the classical notion of factorization in $(\mathbb{N}, \times)$: $2 \times 3 \times 5$ is a factorization of 30.

Integrating the alternation constraint between terms in a decomposition yields a general definition of analogical proportions:

Definition 2 (Analogical proportion).
$(x, y, z, t) \in U$ form an analogical proportion*, denoted by $x : y :: z : t$ if and only if there exists some factorizations*

$$x_1 \oplus \ldots \oplus x_n = x,\ y_1 \oplus \ldots \oplus y_n = y,\ z_1 \oplus \ldots \oplus z_n = z,\ t_1 \oplus \ldots \oplus t_n = t$$

such that $\forall i,\ (y_i, z_i) \in \{(x_i, t_i), (t_i, x_i)\}$.

This definition is valid for any semigroup, and *a fortiori* for any richer algebraic structure. Thus, it readily applies to the case of groups, vector spaces, free monoids, and structures widely used in knowledge representation such as sets and recursive feature structures. When the algebraic structure has additional properties (commutativity, identity element, unique inverse for $\oplus$), this definition is greatly simplified. For instance, if $(U, \oplus)$ is a group, definition (2) yields:

Theorem 1 (Analogical proportion in a group). *If $(U, \oplus)$ is a group, $x : y :: z : t$ iff $x \oplus t = y \oplus z$.*

This is consistent with traditional visions of analogy: in $(\mathbb{R}^\star, \times)$, it corresponds to the classical proportionality relation ($\frac{x}{y} = \frac{z}{t}$); in a vector space, it expresses the relation between the summits of a parallelogram.

3.2 Words over Finite Alphabets

3.2.1 Analogical Proportions between Words

Let Σ be a finite alphabet. $\Sigma^\star$ denotes the set of finite sequences of elements of Σ, called *words* over Σ. $\Sigma^\star$, provided with the concatenation operation . is a monoid, i.e. a semigroup plus an identity element, the empty word ε. Definition (2) readily applies, and allows to capture analogies between words such as:

$$\textit{viewing} : \textit{reviewer} :: \textit{searching} : \textit{researcher}$$

where the alternating subparts are: $x_1 = \epsilon$, $x_2 = \textit{view}$, $x_3 = \textit{ing}$ and $t_1 = \textit{re}$, $t_2 = \textit{search}$, $t_3 = \textit{er}$. This definition properly generalizes the proposal of Lepage (2001). It does not ensure the existence of a solution to analogical equations, nor does it ensure the uniqueness of a solution when one exists.

3.2.2 A Finite-state Solver

In the case of words, definition (2) yields an efficient procedure for solving analogical equations, based on the formalism of finite-state transducers. The main steps of this procedure are only sketched here, we refer to Yvon *et al.* (2004) for additional details and proofs. To start with, let us introduce the notions of *complementary set* and *shuffle product.*

Complementary set A *subword* of $w \in \Sigma^\star$ is a word which can be derived from w by deleting some letters. If v is a subword of w, the *complementary set* of v with respect to w, denoted by $w\backslash v$ is the set of subwords obtained by deleting in w all the letters in v. For example, $\{eea\}$ is the complementary set of *xmplr* with respect to *exemplar*. If v is not a subword of w, $w\backslash v$ is empty. This notion generalizes to regular languages. The complementary set of v with respect to w is a regular set: it is the output language of the transducer T_w (see Figure 1) for the input v.

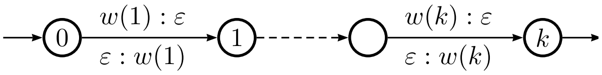

Figure 1: The transducer computing the complementary set wrt w. $w(i)$ denotes the i^{th} letter in w.

Shuffle The *shuffle* $u \bullet v$ of two words u and v is introduced e.g. in Sakarovitch (2003) as follows:

$$u \bullet v = \{u_1 v_1 u_2 v_2 \dots u_n v_n, \text{ st. } u_i, v_i \in \Sigma^\star, \\ u_1 \dots u_n = u, v_1 \dots v_n = v\}$$

The shuffle of two words u and v contains all the words w which can be composed using all the symbols in u and v, subject to the condition that if a precedes b in u (or in v), then it must precede b in w. Taking, for instance, $u = abc$ and $v = def$, the words $abcdef$, $abdefc$, $adbecf$ are in $u \bullet v$; this is not the case with $abefcd$, in which d occurs after, rather than before, e. The shuffle operation generalizes straightforwardly to languages. As noted, for instance in Sakarovitch (2003), the shuffle of two regular languages is also a regular language, and is recognized by a finite-state automaton derived from u and v.

The notions of complementary set and shuffle are related through the following property, which is a direct consequence of the definitions.

$$w \in u \bullet v \Leftrightarrow u \in w\backslash v$$

Solving analogical equations The notions of shuffle and complementary sets enable us to provide an alternative characterization of analogical proportion between words, based on the following theorem. This theorem gives a necessary and sufficient condition for an analogical proportion to hold.

Theorem 2.

$$x : y :: z : t \Leftrightarrow x \bullet t \cap y \bullet z \neq \emptyset$$

The intuition behind this theorem is that an analogical proportion is verified if the symbols in x and t are also found in y and z, and appear in the same relative order. From this proposition, we derive the following corollary:

Theorem 3.

$$t \textit{ is a solution of } x : y :: z :? \Leftrightarrow t \in y \bullet z \backslash x$$

This result states that the set of solutions of an analogical equation $x : y :: z :?$ is a regular set, which can be constructed with a finite-state transducer. Moreover, it can be shown that (i) this analogical solver generalizes the approach based on edit distance proposed by Lepage (1998), and more importantly that (ii) it generalizes straightforwardly to situations where the alphabet Σ is itself a structured set, equipped with an arbitrary notion of analogical proportions between individual symbols. It is thus possible to define and compute, for instance, analogies between strings of feature-structures, between strings of strings, etc (again, refer to Yvon *et al.* (2004) for details).

3.3 Lattices, feature structures and sets

A *lattice* is non-empty algebra with two binary operations (denoted by $\vee$ and $\wedge$) which are idempotent, associative, commutative and which satisfy the absorption law. The powerset of a set, equipped with the union and intersection operations is a lattice, and so is the set of feature structures, equipped with the unification and generalization operations. Both are commonly used in traditional representations of linguistic data. It is, for instance, usual to represent each English phoneme as a finite set of binary features: under this representation, the consonant /p/ is represented as: $\{consonant, anterior\}$. Feature-structures are also common, and constitute, for instance, the basic representation in many modern syntactic theories such as LFG, HPSG, etc.

If $(U, \vee, \wedge)$ is a lattice, $(U, \vee)$ is a semigroup: $\vee$ is the operation used to build factorization of elements in U. The existence of another operation allows to greatly simplify definition (2). Simple algebraic manipulations yield the following theorem:

Theorem 4 (Analogical proportion in a lattice). *If* $(U, \vee, \wedge)$ *is a lattice,* $x : y :: z : t$ *if and only if*

$$\begin{aligned}
x &= (x \wedge y) \vee (x \wedge z) \\
y &= (x \wedge y) \vee (t \wedge y) \\
z &= (t \wedge z) \vee (x \wedge z) \\
t &= (t \wedge z) \vee (t \wedge y)
\end{aligned}$$

This result is again in accordance with the intuition, and allows, for instance, to capture well-known oppositions between pairs of phonemes such as

$/p/ : /b/ :: /t/ : /d/$. Defining these phonemes as:

$$\begin{aligned} /p/ &= \{consonant, anterior\} \\ /b/ &= \{consonant, anterior, voice\} \\ /t/ &= \{consonant, coronal\} \\ /d/ &= \{consonant, coronal, voice\}, \end{aligned}$$

one can readily check that the four conditions in Theorem (4) are satisfied.

An example of analogical proportion between feature structures is given in Figure 2. In this example, the analogical proportion involves four identical syntactic structures, with an alternation of the number (NUM) and person ($PERS$) values of the subject ($SUBJ$) sub-structure.

$$\begin{array}{ccc} \begin{bmatrix} SUBJ: \begin{bmatrix} [1]\textbf{agr} \\ PERS: \left[\,\textbf{1st}\,\right] \\ NUM: \left[\,\textbf{sing}\,\right] \end{bmatrix} \\ PRED: \left[\,[1]\,\right] \end{bmatrix} & : & \begin{bmatrix} SUBJ: \begin{bmatrix} [1]\textbf{agr} \\ PERS: \left[\,\textbf{3rd}\,\right] \\ NUM: \left[\,\textbf{sing}\,\right] \end{bmatrix} \\ PRED: \left[\,[1]\,\right] \end{bmatrix} \\ & :: & \\ \begin{bmatrix} SUBJ: \begin{bmatrix} [1]\textbf{agr} \\ PERS: \left[\,\textbf{1st}\,\right] \\ NUM: \left[\,\textbf{plur}\,\right] \end{bmatrix} \\ PRED: \left[\,[1]\,\right] \end{bmatrix} & : & \begin{bmatrix} SUBJ: \begin{bmatrix} [1]\textbf{agr} \\ PERS: \left[\,\textbf{3rd}\,\right] \\ NUM: \left[\,\textbf{plur}\,\right] \end{bmatrix} \\ PRED: \left[\,[1]\,\right] \end{bmatrix} \end{array}$$

Figure 2: Analogical proportion between feature structures.

Theorem (4) readily yields an efficient procedure for computing analogical proportions in lattices: computing an analogical proportion only requires 8 atomic operations: 4 unions and intersections for sets, 4 unifications and generalizations for feature structures. The case of multi-sets, i.e. sets which can contain multiple instances of an element, yields a similar definition.

3.4 Trees

Labelled trees are very common structures in Natural Language Processing tasks: they can represent syntactic structures, or terms in a logical representation of a concept or of a sentence. To express the definition of analogical proportion between trees, we introduce the additional notion of substitution.

Definition 3 (Substitution).
A substitution *is a pair* $(v \leftarrow t')$*, where* v *is a variable and* t' *is a tree. The application of the substitution* $(v \leftarrow t')$ *to a tree* t *consists in replacing each leaf of* t *labelled by* v *by the tree* t'*. The result of this operation is denoted* $t \triangleleft_v t'$*.*

It is now possible to extend definition 2 as:

Definition 4 (Analogical proportion (trees)).
$(x, y, z, t) \in U$ *form an* analogical proportion*, denoted by* $x : y :: z : t$ *if and only if there exists some variables* $(v_1, \ldots, v_{n-1})$ *and some factorizations*

$$\begin{gathered} x_1 \triangleleft_{v_1} \ldots \triangleleft_{v_{n-1}} x_n = x, y_1 \triangleleft_{v_1} \ldots \triangleleft_{v_{n-1}} y_n = y, \\ z_1 \triangleleft_{v_1} \ldots \triangleleft_{v_{n-1}} z_n = z, t_1 \triangleleft_{v_1} \ldots \triangleleft_{v_{n-1}} t_n = t, \end{gathered}$$

such that $\forall i,\ (y_i, z_i) \in \{(x_i, t_i), (t_i, x_i)\}$*.*

An example of such a proportion is illustrated on Figure 3 with syntactic trees. In this example, the analogical proportion relates the parse trees of the active and passive versions of two sentences; the altering chunks are subtrees which alternate in the four complete trees.

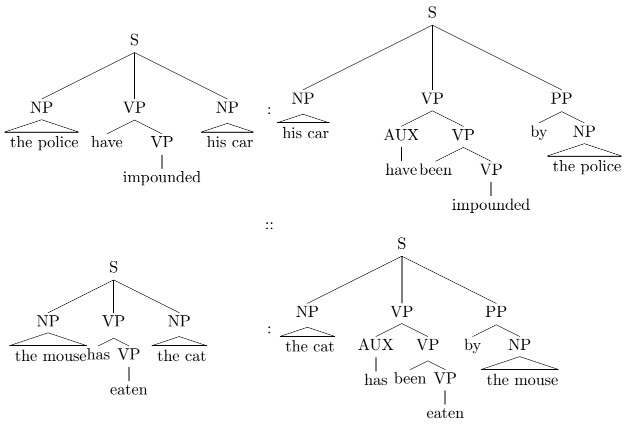

Figure 3: Analogical proportion between trees.

To solve analogical proportions between trees, we have considered a heuristic approach, consisting in using tree linearization procedures and the analogical solver for words detailed above (see Stroppa & Yvon (2005*b*) for more details). This approach yields a fast, albeit approximative algorithm, which makes analogical inference tractable even for large tree databases.

4 Characterization of the inductive model

In this section, some aspects of the inductive model presented in Section 2.2 are discussed and related to other instance-based learning procedures.

4.1 Learning from structured and unstructured data

In the sequel, we consider the simple task of learning the conjugation of English verbs. English conjugation is quite simple: each verb has at most five different forms: Present (except 3rd person, P), Present (3rd person, 3), Past (S), Participle (D), and Gerund (G). Training instances are thus triples such as $(search, P, search)$, $(search, 3, searches)$, $(search, S, searched)$, etc. Given such a training set, the task is to infer the 3rd component based on the first two: $(search, D, ?)$.

Instance-based approaches to this task (e.g. Skousen (1989)) are usually performed by turning it into a series of classification problems, which requires a substantial recoding of the data. In fact, instance-based learners are well suited for processing unstructured data (see however Ramon & Raedt (1999); Horváth *et al.* (2001) for attempts at using instance-based learners to infer relations or functions). A possible vectorization of the input uses fixed-length vector, where components encode the letters and the tense. Various representations of the output as class(es) can be entertained: as a series of letters (yielding one classification task for each output letter); or as a category label, which encodes the related transformation of the input. Under the first scheme, the training instance $(search, 3, searches)$ is turned into: $([s, e, a, r, c, h, -, -, -, -, 3], s)$ (for the first output letter classification task, since s is the first letter of *search*) and $([s,e,a,r,c,h,-,-,-,-,3],e)$ (second output letter), etc. Under the second one, the output looks like $([s,e,a,r,c,h,-,-,-,-,3],+es)$. This kind of approach is efficient and scales up well. However, the design of the encoding scheme plays here a critical role: in fact, reversing the order of the input letters makes the problem much easier, as it allows the learner to detect regularities between a verb's ending and the inflected form.

In comparison, our analogical learner can directly process the original representations. Furthermore, as inputs and outputs are treated symmetrically, it can also be used to perform the reverse task by exchanging the input and output spaces (see Stroppa & Yvon (2005*a*) for a detailed report of experiments on this reverse problem). Our learner can also readily accommodate more complex learning situations, where the output takes the form of a tree or a feature structure; turning such tasks into a series of standard classification problems can be less obvious.

4.2 Generalizing by analogy

While most learners are able to generalize to any new input, this is not the case for our analogical learner. Although this may look like a major flaw, such a behaviour actually makes sense for a lot of Natural Language Processing tasks: for instance, it is a nice feature of the system not to output any inflected form when the input is too remote from the training instances, as would be a "verb" like $cfgcffgw$. In this section, we discuss the generalization capabilities of analogical learners.

Definition 5 (Analogical extension).
The analogical extension $A_E(S)$ *of* S *contains the objects which are analogically related to three objects in* S*, i.e.:*

$$A_E(S) = \{t \mid x : y :: z : t \text{ with } (x,y,z) \in S^3\}$$

Assuming that S denotes the training set of some learning task, and that X is a new instance, the set $\mathcal{T}(X)$ will be non empty if and only if $X \in A_E(S)$. The generalization capability of the model is thus limited to $A_E(S)$. For example, given the training set

$$S = \{x_1 = (read, 3, reads), x_2 = (read, G, reading), x_3 = (view, 3, views),\\ x_4 = (view, G, viewing), x_5 = (eat, 3, eats)\}$$

and the input $x_6 = (eat, G, ?)$, the inference process will find two triples

$$\mathcal{T}(X) = \{(x_1, x_2, x_5); (x_3, x_4, x_5)\}$$

and output the form *eating*, which solves the analogical equation:

$$views : viewing :: eats :?.$$

In contrast, no triple (hence no output) is found for the input $x_7 = (absvd, G, ?)$ since $x_7 \notin A_E(T)$. This result illustrates the fact that the generalization capability strongly depends on the way proportions are defined. For example, if we adopt a restricted definition such as: $x : y :: z : t \Rightarrow x = y = z = t$, then $A_E(S)$

is equal to S and no generalization occurs. Conversely, if $x : y :: z : t$ is true for any (x, y, z, t), then $A_E(S)$ is the entire space of possible objects. For k-NN classifiers, a similar control of the generalization ability is performed through the parameter k: setting $k = 0$ corresponds to rote learning, while taking $k = \infty$ allows to output any of the class labels. However, while the choice of k is often problematic, we keep here a greater control over the definition of analogical proportions, which are chosen to reflect the structural organization of the data. This model appears to be well suited for many Natural Language Processing tasks since it is often possible to take advantage of existing linguistic knowledge. Consider the following task, where we need to compute the syntactic tree for the passive form of a sentence (cf. Figure 3, which represents an analogy in the output space for this task): in this context, subject/verb agreement could be enforced by imposing alternating subtrees to have the same number and person.

4.3 Heuristic search procedures

As pointed out in Section 2.2, scanning all triples in S^3 might turn out to be intractable when working with very large databases, thus fostering the need for faster, approximate, search techniques.

The *analogical support* is, to some extent, the dual notion of the analogical extension. It consists in a minimal subset of S capable of reconstructing S by analogical learning. It may be not unique.

Definition 6 (Analogical support).
A_S is a (minimal) analogical support *of S iff*

$$A_E(A_S) \supseteq S \textit{ and } \forall A \subseteq A_S, A_E(A) \not\supseteq S$$

The analogical support of a dataset forms a kind of structural core. Limiting the training set S to A_S is a way to reduce the database which can be compared to other instance-based condensing techniques Wilson & Martinez (2000). For instance, an analogical support of the training set T (see above) is (x_1, x_2, x_3, x_5).

A support may be constructed by incremental or decremental methods. However, it is possible to avoid its construction when the data are sufficiently redundant. This is the case when lots of analogical triples are found for the greater part of the instances. In this situation, a reduction technique (which reduces search time but not storage requirements) is described below.

This methodology involves a pre-processing step, consisting in grouping training instances into bins. Two instances x and y are grouped together if they are considered as good candidates for the two first terms of an analogical proportion. For example, instances in the conjugation task may be grouped by lemmas. During the inference process, the search procedure only considers a number of randomly selected bins and tries to build proportions involving at least two instances in the same bin (in our toy problem, two forms of the same verb). Our hypothesis is that, given the intrinsic redundancy of linguistic data, these heuristics might be effective, which has been confirmed experimentally (again, refer to Stroppa & Yvon (2005*a*)).

4.4 Related work

Formal proportions of structured objects, including but not limited to the case of strings, have also been studied e.g. in Hofstadter & Mitchell (1995); Dastani *et al.* (2003); Schmid *et al.* (2003). The long term goal of these studies is to devise a general model of analogical perception and production. In the model of Schmid *et al.* (2003), four objects (a, b, c, d) form an analogical proportion if and only if $a = P\sigma_1$, $b = Q\sigma_1$, $c = P\sigma_2$, $d = Q\sigma_2$ where P and Q are generic (second-order logic) terms made up with non constrained operators and σ_1, σ_2 are substitutions acting on terms. Based on this definition, these authors are able to model complex structural mappings in domains involving small databases of highly-structured data. From a mathematical point of view, their underlying principles are consistent with ours, involving similar ideas of decomposition and alternations. We believe that both approaches could be recast within a unified framework: their use of arbitrary operators allows to accommodate a wider range of phenomena, at the cost of a significant increase in the computational complexity of the solving procedure.

5 Discussion and future work

In this paper, we have presented a generic analogical inference procedure, which applies to a wide range of actual learning tasks involving structured data , and we have detailed its instantiation for a wide range of common representations of training instances: sets, feature structures, strings and trees. We have also related our inference procedure with more traditional instance-based learners, and discussed its relationships with alternative definition of proportions in structured domains. The analogical learner we propose is a first step towards taking advantage of the statistical distribution of analogical proportions in the input space, without resorting to a surface similarity between objects. It thus seems to bridge the gap between these two approaches. Experiments have been conducted on several morphological analysis subtasks using analogies on words and on trees, and show promising generalization performance Stroppa & Yvon (2005*a*). These results suggest that the main hypotheses underlying this approach are valid: (i) searching for triples is tractable even with databases containing several hundred of thousands instances; (ii) formal analogical proportions are a reliable sign of deeper analogies between linguistic entities; they can thus be used to devise flexible and effective learning strategies for natural language processing tasks.

This work is being developed in various directions: first, we are gathering additional experimental results on several Natural Language Processing tasks, to get a deeper understanding of the generalization capabilities of the analogical learner. One interesting issue, only alluded to in this paper, is the integration of linguistic knowledge in the definition of analogical proportions, or in the specification of the search procedure. We are also considering alternative heuristic search procedures, which could improve or complement the ones presented in this paper. We finally believe that this approach might also prove useful in other domains involving large databases of structured instances, and are eager to experiment with other kinds of data.

References

Aha D.W. (1997). Editorial. *Artificial Intelligence Review*, 11(1-5):7–10, special Issue on Lazy Learning.

Daelemans W., van den Bosch A. & Zavrel J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–41, special issue on natural language learning.

Dasarathy B.V. (ed.) (1990). *Nearest neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA.

Dastani M., Indurkhya B. & Scha R. (2003). Analogical projection in pattern perception. *Journal of Experimental and Theoretical Artificial Intelligence*, 15(4):489–511.

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

Gentner D., Holyoak K.J. & Kokinov B. (eds.) (2001). *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press, Cambridge, MA.

Hofstadter D.R. & Mitchell M. (1995). The copycat project: A model of mental fluidity and analogy-making. In D.R. Hofstadter & the Fluid Analogies Research group (eds.), *Fluid Concepts and Creative Analogies*, chap. 5, p. 205–267, Basic Books, New York, NY.

Holyoak K.J. & Hummel J.E. (2001). Understanding analogy within a biological symbol system. In D. Gentner, K.J. Holyoak & B.N. Konikov (eds.), *The Analogical Mind: Perspectives from Cognitive Science*, p. 161–195, MIT Press, Cambridge, MA.

Horváth T., Wrobel S. & Bohnebeck U. (2001). Relational instance-based learning with lists and terms. *Machine Learning*, 43(1-2):53–80, special issue on inducive logic programming.

Lepage Y. (1998). Solving analogies on words: an algorithm. In *Proceedings of COLING-ACL 1998*, vol. 1, p. 728–735, Montréal, Canada.

Lepage Y. (1999*a*). Analogy + tables = conjugation. In *Proceedings of NLDB 1999*, p. 197–201, Klagenfurt, Germany.

Lepage Y. (1999*b*). Open set experiments with direct analysis by analogy. In *Proceedings of the 5thNatural Language Processing Pacific Rim Symposium (NLPRS 1999)*, p. 363–368, Beijing, China.

Lepage Y. (2001). Analogy and formal languages. In *Proceedings of FG/MOL 2001*, p. 373–378, Helsinki, Finland.

Lombardy S., Poss R., Régis-Gianas Y. & Sakarovitch J. (2003). Introducing Vaucanson. In *Proceedings of the 8thInternational Conference on Implementation and Application of Automata (CIAA 2003)*, vol. 2759 of *Lecture Notes in Computer Science Series*, p. 96–107, Springer-Verlag, Santa Barbara, CA.

MITCHELL T.M. (1997). *Machine Learning*. McGraw-Hill.

PIRRELLI V. & YVON F. (1999). The hidden dimension: paradigmatic approaches to data-driven natural language processing. *Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Memory-Based Language Processing*, 11:391–408.

PLATE T.A. (2000). Analogy retrieval and processing with distributed vector representations. *Expert systems*, 17(1):29–40.

RAMON J. & RAEDT L.D. (1999). Instance based function learning. In S. Džeroski & P. Flach (eds.), *Proceedings of the 9th International Workshop on Inductive Logic Programming*, vol. 1634 of *Lecture Notes in Artificial Intelligence*, p. 268–278, Springer-Verlag, London, UK.

SAKAROVITCH J. (2003). *Éléments de théorie des automates*. Vuibert, Paris.

SCHMID U., GUST H., KÜHNBERGER K.U. & BURGHARDT J. (2003). An algebraic framework for solving proportional and predictive analogies. In F. Schmalhofer, R. Young & G. Katz (eds.), *Proceedings of the European Conference on Cognitive Science (EuroCogSci 2003)*, p. 295–300, Lawrence Erlbaum, Osnabrück, Germany.

SKOUSEN R. (1989). *Analogical Modeling of Language*. Kluwer Academic Publishers, Dordrecht.

STROPPA N. & YVON F. (2005*a*). An analogical learner for morphological analysis. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)*, Ann Arbor, MI.

STROPPA N. & YVON F. (2005*b*). Formal models of analogical proportions. Technical report, École Nationale Supérieure des Télécommunications, Paris, France.

THAGARD P., HOLYOAK K.J., NELSON G. & GOCHFELD D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence*, 46(3):259–310.

WILSON D.R. & MARTINEZ T.R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286.

YVON F., STROPPA N., DELHAY A. & MICLET L. (2004). Solving analogies on words. Technical report D005, École Nationale Supérieure des Télécommunications, Paris, France.