Towards a Semantic-based Theory of Language Learning

Isabelle Tellier
Laboratoire d'Informatique Fondamentale de Lille, équipe Grappa

Abstract

The notion of Structural Example has recently emerged in the domain of grammatical inference. It allows to solve the old difficult problem of learning a grammar from positive examples but seems to be a very *had hoc* structure for this purpose. In this article, we first propose a formal version of the Principle of Compositionality based on Structural Examples. We then give a sufficient condition under which the Structural Examples used in grammatical inference can be inferred from sentences and their semantic representations, which are supposed to be naturally available in the environment of children learning their mother tongue. Structural Examples thus appear as an interesting intermediate representation between syntax and semantics. This leads us to a new formal model of language learning where semantic information play a crucial role.

1. Introduction

The problem of grammatical inference from positive examples consists in the design of algorithms able to identify a formal grammar from sentences it generates. It is the computational version of the problem of children language learning and is then of great cognitive interest.

But strings of words are not informative enough to specify a grammar : it has been proved that even the class of regular languages is not learnable from positive examples in usual models of learning ([4, 14]).

To overcome this difficulty, a recently investigated solution consists in providing *Structural Examples* to the learner instead of strings of words ([2, 6, 7, 10, 11]). A Structural Example is a more or less simplified version of the syntactic (or analysis) tree.

But this solution is not very satisfying from a cognitive point of view, as Structural Examples seem to be very unnatural species. The purpose of this article is to provide a new interpretation of Structural Examples, as a relevant intermediate level between syntax and semantics. This interpretation allows to formulate a simple rule-based definition of the Principle of Compositionality and a semantic-based model of natural language learning.

2. Structural Examples used in Grammatical Inference

Let us call a *composition* a tree whose leaves are taken among a finite vocabulary Σ and whose internal nodes are indexed by symbols belonging to a signature Γ. In the following, we will note $\Gamma=\{g_1, \ldots, g_m\}$, for some integer m.

For any (context-free) grammar G, let us partition the set of rules of G and associate a unique symbol taken among Γ with each class. A composition built on the vocabulary Σ of G is said to be a *Structural Example for G* iff there exists a syntactic tree generating the corresponding sentence in G so that the composition is obtained by replacing every non terminal symbol of this syntactic tree by the symbol class of the rule used to rewrite it.

Example 1 :

It is well know ([1]) that AB-Categorial Grammars are equivalent with context-free grammars, so the previous definitions can straightforwardly be adapted to this class. Let us define a basic AB-Categorial Grammar G' for the analysis of a small subset of English. Let Σ' ={a, man, John, Mary, runs, loves} be the vocabulary and {S, T, CN} the set of basic categories. In this set, S is the axiom, T stands for "term" and CN for "common nouns". The assignment function f is then defined by : f(John)=f(Mary)={T}, f(runs)={T\S}, f(loves)={(T\S)/T}, f(man)={CN} and f(a)={(S/(T\S))/CN}.

The only admitted reduction rules, as usual, are called R1 and R' 1 and are defined by : for any category A and B, R1[A/B, B]=A and R' 1[B, B\A]=A.

This grammar allows to recognize sentences like : "John runs", "a man runs" or "John loves Mary" as follows (with a little abuse of notation, for readability, rules R1 and R’1 are used as if they applied on couples (word, category) instead of on categories alone) :

R’1[(John, T).(runs, T\S)]=(John . runs, S)

R1[R1[(a, (S/(T\S))/CN).(man, CN)].(runs, T\S)]

=R1[(a . man, S/(T\S)).(runs, T\S)]=(a . man . runs, S)

R’1[(John, T).R1[(loves,(T\S)/T).(Mary, T)]]

=R’1[(John, T).(loves . Mary, T\S)]=(John . loves . Mary, S)

Let Γ' ={g_1 g_2}. In AB-Categorial Grammars, the most natural partitioning of the set of rules is based on the distinction between the two directions of functional application : let then R1 be indexed by g_1 and R' 1 by g_2 The Structural Examples for our grammar corresponding with the previous three analysis trees are then respectively : g_2(John, runs), g_1(g_1(a, man)), runs) and g_2(John, g_1(loves, Mary))

When Γ is reduced to a unique symbol, Structural Examples only display the branching of the syntactic trees without indication about the intermediate non terminal symbols, and are called *skeletons*.

The problem of grammatical inference from Structural Examples consists in identifying a formal grammar from Structural Examples. It has been recently studied and partly solved when the set of rules is partitioned into one class (i. e. with skeletons) in [10, 11] or, like in the example, into two classes for Classical Categorial Grammars (or AB-Categorial Grammars) in [2, 6, 7].

Some of the algorithms providing a solution to this new problem are computationally efficient. But, when provided with sentences, trying every possible composition based on these sentences is computationally highly expensive in space and time and the result is a set of many compositions among which the Structural Example(s) is(are) indistinguishable.

3. The Principle of Compositionality

Since Montague's work ([8]), the Principle of Compositionality is often formally stated as a rule-based correspondence between syntactic and semantic trees ([5]). The interesting point to notice is that this correspondence between *rules* is independent of the nature of the intermediate non terminal symbols (or categories) appearing at the nodes of the syntactic tree. This means that the full syntactic tree is not necessary to obtain the semantic tree : the corresponding Structural Example is enough, *provided that the partition of the syntactic rules made to define it coincide with the distinct useful semantic functions.*

To formalize this idea, let a Compositional Set <G, Γ, K, H> be composed of a (context-free) grammar G, a signature Γ, a mapping K associating each rule of G with a symbol of Γ (defining the partition of the set of rules of G) and a compositional meaning assignment H=<L, t, T> defined as follows :

- L is a semantic representation language ;
- t is a mapping associating each member of the vocabulary Σ of G with a unique meaning in L : as, in Structural Examples, we give up the categories of the analysis tree, t *only depends on the vocabulary*, so we have to admit here a non ambiguous meaning assignment for words ;
- T is a *bijective* mapping associating each symbol g_j in Γ with a semantic function noted h_j, $1 \leq j \leq m$.

H is a morphism applying on *compositions*. For any sentence $w=u_1...u_n$ generated by G and any composition on w noted g*(w), H is defined by : $H[g^*(u_1...u_n)]=T(g^*)[t(u_1)...t(u_n)]$, obtained from $g^*(u_1...u_n)$ by replacing each u_i by $t(u_i)$, $1 \leq i \leq n$ and each g_j by $T(g_j)=h_j$, $1 \leq j \leq m$. For every *Structural Example* g*(w), *the evaluation of the expression H(g*(w)) represents the (or, in case of syntactic ambiguity, one of the) meaning(s) of w.*

Example 2 :

Let <G' ,Γ' , K' , H' > be a Compositional Set assigning meaning representations to the sentences generated by the grammar G' of Example 1. K' associates R1 and R' 1 respectively with g_1 and g_2 and H' =<L' , t' , T' >. For sake of simplicity, L' is a typed (the typing system cannot be developed here) first order predicate logic augmented with lambda-calculus (i.e. an unintensional version of Montague's intensional logic). Furthermore :

- t' associates each word u in Σ with a logical formula t' (u) in L' (respecting the types). The logical translations of individual words are :

  * $t'(a)=\lambda P_1 \lambda Q_1 \exists x[P_1(x) \wedge Q_1(x)]$

  where x and y are individual variables and P_1 and Q_1 variable predicates of arity 1 (as indicated by the indexes).

  * every other word u in Σ is translated into a logical constant noted $t'(u)=u_i'$ where i is the arity, only noted when i≥1 (conjugated verbs are first reduced to their infinitive form).

- T' is defined by :

  * $T'(g_1)=h_1$ where for every couple (a, b) in L' , $h_1(a, b)=a(b)$;

  * $T'(g_2)=h_2$ where for every couple (a, b) in L' , $h_2(a, b)=b(a)$;

The application of H' to the Structural Examples of Example 1 gives :

$H'[g_2(John, runs)]=T'(g_2)[t'(John), t'(runs)]$

$=h_2[John', run_1']=run_1'(John')$

$H'[g_1(g_1(a, man)), runs)]=T'(g_1)[(T'(g_1)[t'(a), t'(man)]), t'(runs)]$

$=h_1[h_1(\lambda P_1 \lambda Q_1 \exists x[P_1(x) \wedge Q_1(x)], man_1'), run_1']$

$=(\lambda P_1 \lambda Q_1 \exists x[P_1(x) \wedge Q_1(x)](man_1'))(run_1')$

$=\exists x[man_1'(x) \wedge run_1'(x)]$

$H'[g_2(John, g_1(loves, Mary))]=T'(g_2)[t'(John), T'(g_1)[t'(loves), t'(Mary)]]$

$=h_2[John', h_1(love_2', Mary')]$

$=love_2'(Mary')(John')$.

Note that skeletons may not be precise enough to specify compositional semantics because as $\Gamma=\{g_1\}$, only one semantic function $T(g_1)=h_1$ is allowed. But, in the vast domain of computational linguistics, compositional logical-based meaning are classical and extensions of the basic version of Example 2 -to Lambek grammars for instance ([12])- can be defined. In some cases, the Curry-Howard correspondence specifies the mapping T.

Structural Examples can thus be considered as the *minimal basis necessary for the definition of compositional semantics*. This link between Structural Examples and semantics can now help us interpreting where Structural Examples come from in the learning domain.

4. A new learning model

It is natural to suppose that when a child learns a language, she has at her disposal (heard) syntactically correct sentences together with their *meaning*, available in the environment (and pointed by the speaker). The corresponding computational situation is an algorithm which takes as input both syntactically correct sentences and (one of) their semantic representation(s).

Let us suppose that the underlying linguistic system is a Compositional Set <G, Γ, K, H> where H=<L, t, T> and that the innate knowledge includes the semantic language L, the set $\Gamma=\{g_j\}_{1\leq j\leq m}$ and the corresponding set of semantic functions $\{h_j\}_{1\leq j\leq m}$. The rule translation mapping T is then considered as universal and independent of the language to be learned. As in usual semantic-based methods of learning ([3]), word meanings (i.e. the mapping t) are also supposed to be already known when the grammatical inference starts. Only G and K remain to be learned.

But to make use of the input data, we need more than the Principle of Compositionality : we need a property that we suggest to call *Fully Compositionality*. A Compositional Set <G, Γ, K, H> will be said *Fully Compositional* if for every sentence w generated (or recognized) by G and every *composition* g*(w) built on w, we have : if there exists a Structural Example g' *(w) for G satisfying eval(H(g*(w)))=eval(H(g' *(w))) then g*(w) is also a Structural Example.

To help intuition, this new definition can be considered as stating that if w is a syntactically correct sentence then any composition based on w and translated by H into a correct meaning for w (i.e. any compositionallyobtained meaning) is also a Structural Example for G. In other words, in a Fully Compositional framework, the evaluation of the image by H of a composition can be used as a *criterion* for deciding if this composition is in fact a Structural Example.

In this case, from input couples made of a sentence and (one of) its meaning(s), it is possible to infer Structural Examples. Figure 3 shows two strategies for this, called *forward* and *backward inference* because of their similarity with usual forward and backward chaining in deduction theory.

for any <w, m> where $w=u_1...u_n$ *is a sentence and m (one of) its meaning representation(s) do :* • *found <- false ;* • *while NOT (found) do* * *try a composition e=g*(w) based on w ;* * *if eval(H(e))=m then found <-true ;* • *return(e).*	*for any <w, m> where* $w=u_1...u_n$ *is a sentence and m (one of) its meaning representation(s) do :* • *found <- false ;* • *while NOT (found) do :* * *try a composition on the semantic translations :* $k=h^*(t(u_1)...t(u_n))$ *;* * *if eval(k)=m then found <- true ;* • *return*$((T^{-1}(h))^*(w))$

Figure 3 : forward and backward inference of Structural Examples

Both algorithms are *in the worst case* exponential in time but only *linear in space*, since only one candidate Structural Example is stored and checked at the same time. The total number of different possible compositions g*(w) defined from a signature $\Gamma=\{g_j\}_{1\leq j\leq m}$ where each g_j is of arity 2 based on a sentence $w=u_1...u_n$ composed of n words with $n\geq 1$ equals : $C(n-1)*m^{n-1}$ where C(k) is the Catalan number defined by C(k)=(2k)!/(k!(k+1)!) for any $k\in\mathbb{N}$ (the Catalan number C(k) counts the number of binary trees with k internal nodes, that is with k+1 leaves).

But much better efficiency can be obtained by taking into account the *arities* of word meanings in the backward inference algorithm, as follows.

Example 3 :

A natural simple heuristic is the following : "when trying a composition on a sequence of semantic expressions, first try the functional applications between two logical expressions whose arities are in decreasing order". Let us apply it to our examples sentences :

• from the sequences of word meanings : "John', run'_1", and "$\lambda P_1\lambda Q_1\exists x[P_1(x)\wedge Q_1(x)]$, man_1', run_1'", the only semantic compositions respecting the heuristic are the only ones which are also Structural Examples.

• from the sequence : "John', $love_2$', Mary'", the heuristic selects two semantic compositions among eight possible, including the correct one.

5. Interpretation

The learning strategy for natural languages proposed here applies in two steps. The first step, where semantic information play a crucial role, is *the* *inference of Structural Examples*. Although it remains to be proved, usual compositional syntactico-semantic frameworks seem to satisfy the Fully Compositionality property. In this case, Structural Examples can be inferred from strings of words and semantic representations. The second step is the *inference of a grammar from Structural Examples* which has already received interesting partial solutions. Structural Examples thus appear as a crucial intermediate representation between syntax and semantics.

Note that when the full target is reached, the system applying this strategy will be able not only to parse a syntactically correct sentence with G, but also to associate a meaning with it, which is the exact cognitive definition of *learning to understand* ([13]). This new conception of language learning is a formal alternative to usual purely syntactic theories and it meets the psycholinguistic opinion that : " knowing a language is knowing how to translate mentalese into strings of words and vice-versa " ([9]).

6. References

[1] Y. Bar Hillel, C. Gaifman, E. Shamir, "On Categorial and Phrase Structure Grammars", Bulletin of the Research Council of Israel, 9F, p1-16, 1960.

[2] W. Buszkowski, G. Penn, "Categorial grammars determined from linguistic data by unification", *Studia Logica* 49, p431-454, 1990.

[3] J. A. Feldman, "Real Language Learning", proceedings of ICGI 98, LNAI 1433, p114-125, Springer-Verlag, 1998.

[4] E. M. Gold, "Language Identification in the Limit", *Information and Control* 10, P447-474, 1967.

[5] T. M. V. Janssen, "Compositionality", in : *Handbook of Logic and Language*, Elsevier, Amsterdam and MIT Press, Cambridge, J. Van Benthem & A. ter Meulen (Eds), p417-473, 1997.

[6] M. Kanazawa, "Identification in the Limit of Categorial Grammars", *Journal of Logic, Language & Information*, vol 5, n°2, p115-155, 1996.

[7] M. Kanazawa, "Learnable classes of Categorial Grammars", CSLI Publications and FoLLI, *Studies in Logic, Language and Information*, 1998.

[8] R. Montague, *Formal Philosophy; Selected papers of Richard Montague*, Yale University Press, New Haven, 1974.

[9] S. Pinker, *The Language Instinct*, Penguin Press, London, 1994.

[10] Y. Sakakibara, "Learning context-free grammars from structural data in polynomial time", *Theoretical Computer Science* 76, p223-242, 1990.

[11] Y. Sakakibara, "Efficient Learning of Context-Free Grammars from Positive Structural Examples", *Information and Computation* 97, p23-60, 1992.

[12] I. Tellier, "Meaning helps learning syntax", proceedings of ICGI 98, LNAI 1433, p25-36, Springer-Verlag, 1998.

[13] I. Tellier, "Learning to Understand", research report IT-320 of the LIFL, 1998 : ftp://ftp.grappa.univ-lille3.fr/pub/reports/tellierml.ps

[14] L. G. Valiant, "A theory of the learnable", *Communication of the ACM*, p1134-1142, 1984.