Various Types of Learning with Types

Isabelle Tellier ISABELLE.TELLIER@UNIV-LILLE3.FR
LIFL, Université de Lille, F-59653 Villeneuve d'Ascq Cedex

1. Introduction

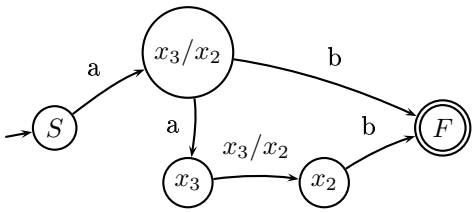
In recent papers (Tellier, 2005a; Tellier, 2005b; Tellier, 2006), we have shown that there exist close connexions between grammatical inference techniques used to infer languages represented by finite state automata (FSA), and grammatical inference techniques used to infer languages represented by Categorial Grammars (CGs). This connexion was established from a simple translation from FSA into unidirectional CGs, and conversely. In this paper, we propose to deepen the parallelism. First, two learning techniques operating by specialisation are compared. Then, we consider the notion of typing, introduced in both contexts. We show that both notions partly coincide.

2. Recursive Automata and Unidirectional Categorial Grammars

In (Tellier, 2005a; Tellier, 2005b), it is shown that every unidirectional CG can be represented by an extension of a FSA called a Recursive Automaton (RA). Unidirectional CGs are a lexicallized grammatical formalism, in which categories taking the form of "fractions" are assigned to the members of the final vocabulary and where the rules are reduced to a "functional application". This formalism has the expressivity of CF grammars (Bar Hillel et al., 1960). A RA is a special case of Recursive Transition Network (Woods, 1970). It looks very much like a FSA, except that some of its transitions can be labelled by states. To use a transition labelled by a state q, you have to produce a string belonging to the language of q, i.e. a string leading from q to a final state. In the following is given an example of a unidirectional CG recognizing the language $a^n b^n$, $n \geq 1$, and the corresponding RA. This CG is can be obtained from the sample of positive examples $\{ab, aabb\}$, each associated with a flat right-branching tree, by applying the learning algorithm proposed by (Kanazawa, 1998).

unidirectional CG :
a: $S/(x_3/x_2),\ (x_3/x_2)/x_3$
b: $x_3/x_2,\ x_2$

corresponding RA:



3. Learning by Specialization

A first parallelism not yet noticed can be made between the "state fission" technique used by (Fredouille, 2000) and the learning algorithm described in (Moreau, 2004). The previous figure shows that the notion of *category*, assigned in a CG to the members of the finite vocabulary, corresponds to the notion of *state* in a FSA. In brief, to learn a (rigid) CG from positive examples, Moreau suggests to initially assign a basic category (corresponding to a distinct variable) to every member of the vocabulary, and then to define substitutions over these variables to allow a correct syntactic analysis of the examples. For example, if the letters a and b are initially assigned basic categories x_1 and x_2 respectively, then, when the positive example ab is provided, the variable x_1 is substituted by: $x_1 = S/x_2$. This substitution is best seen as a *constraint* that variables must satisfy. This substitution operation can be compared to the fission of a state labelled by x_1 into two states : one labelled by S, the other one labelled by x_2, the transition being labelled by S/x_2. Both are *specialization algorithms*.

4. Typed Automata and Typed Data

The notion of typing was first introduced in (Kermorvant & de la Higuera, 2002), to introduce domain knowledge into the grammatical inference process. The idea was to assign types to the states of the prefix tree FSA, referring to a "typing FSA", and to forbid the fusion of states of different types. In (Coste

et al., 2004), it is shown that the help comes from the fact that the typing FSA relies to the target one (by a morphism).

On the other hand, the notion of *type* was also used to infer CGs in (Dudau-Sofronie et al., 2001; Dudau-Sofronie et al., 2003). In this case, types where interpreted as semantic information derived from categories (by a morphism) and assigned to the vocabulary. Do these notions coincide ? The situation is more complex than it seems. Let us illustrate it on a simple example inspired by natural language, where the target is a unidirectional CG. The provided typed examples is (for details about this algorithm, refer to the references) :

a	man	runs
$(tx_1(tx_2e))x_3(tx_4e)$	tx_5e	tx_6e

Here, t and e are the only possible basic types (corresponding to the type of entity and of truth value respectively in a logical model), and the variables stand for a binary operator: they can either stay undefined or take the value $x_i = /$. The types are terms, they can be represented by states of a RA. The learning algorithm proposed in this case is similar to the one evoked in the previous section: it consists in trying to perform a syntactic analysis of the sentence, by defining substitutions on the variables. In our example:

- in a first step, the only way to apply a functional application between two consecutive categories is to define the following substitution: $x_3 = /$. This, as already seen, is some kind of state fission. But it is not enough: the functional application relying on this operator can apply only if $tx_4e = tx_5e$, that is if $x_4 = x_5$ which specifies a state fusion between "compatible" (in the sense of *unifiable*) states of the RA. The partial analysis resulting from this step is:

a man	runs
$(tx_1(tx_2e))$	tx_6e

- In a second step, similar to the previous one, we define: $x_1 = /$ and $x_2 = x_6$. The resulting type t ensures the grammaticality.

5. conclusion

This paper does not provide any new algorithm or theoretical result, but proposes another look on already known techniques. We see that the learning algorithm of CGs from typed data exposed in (Dudau-Sofronie et al., 2001; Dudau-Sofronie et al., 2003) is in fact a combination of specialization and of generalization techniques. The initial types associated with the elements of the vocabulary specify some kind of maximal bound on the possible fissions to be performed. The underlying notion of type associated with states is richer than the one proposed in (Kermorvant & de la Higuera, 2002; Coste et al., 2004) .

References

Bar Hillel, Y., Gaifman, C., & Shamir, E. (1960). On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel, 9F.*

Coste, F., Fredouille, D., Kermovant, C., & de la Higuera, C. (2004). Introducing domain and typing bias in automata inference. *proceedings of the 7th ICGI* (pp. 115–126). Springer Verlag.

Dudau-Sofronie, D., Tellier, I., & Tommasi, M. (2001). From logic to grammars via types. *proceedings of LLL 2001, Learning Language in Logic* (pp. 35–46).

Dudau-Sofronie, D., Tellier, I., & Tommasi, M. (2003). Une classe de grammaires catégorielles apprenable à partir d'exemples typés. *5ème Conférence francophone sur l'apprentissage automatique* (pp. 169–184). Presses Universitaires de Grenoble.

Fredouille, D. (2000). Expériences sur l'inférence de langage par spécialisation. *proceedings of CAP'2000* (pp. 117–130).

Kanazawa, M. (1998). *Learnable classes of categorial grammars.* The European Association for Logic, Language and Information. CLSI Publications.

Kermorvant, C., & de la Higuera, C. (2002). Learning language with help. *6th International Colloquium on Grammatical Inference* (pp. 161–173). Springer Verlag.

Moreau, E. (2004). Apprentissage partiel de grammaires catégorielles. *TALN 2004* (pp. 299–308).

Tellier, I. (2005a). Inférence grammaticale et grammaires catégorielles : vers la grande unification ! *7ème Conférence en Apprentissage* (pp. 63–78). Presses Universitaires de Grenoble.

Tellier, I. (2005b). When categorial grammars meet regular grammatical inference. *5th International Conference on Logical Aspects of Computational Linguistics* (pp. p.317–332). Springer Verlag.

Tellier, I. (2006). Learning recursive automata from positive examples. *Revue d'Intelligence Artificielle, New Methods in Machine Learning*, 775–804.

Woods, W. A. (1970). Transition network grammars of natural language analysis. *Communications of the ACM*, 591–606.