In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 95–98, Lisbon, Portugal, 2000.

ALLiS: a Symbolic Learning System for Natural Language Learning

Hervé Déjean
Seminar für Sprachwissenschaft
Universität Tübingen
dejean@sfs.nphil.uni-tuebingen.de

1 Introduction

We present ALLiS, a learning system for identifying syntactic structures which uses theory refinement. When other learning techniques (symbolic or statistical) are widely used in Natural Language Learning, few applications use theory refinement (Abecker and Schmid, 1996), (Mooney, 1993). We would like to show that even a basic implementation of notions used in TR is enough to build an efficient machine learning system concerning the task of learning linguistic structures.

ALLiS relies on the use of background knowledge and default values in order to build up an initial grammar and on the use of theory refinement in order to improve this grammar. This combination provides a good machine learning framework (efficient and fast) for Natural Language Learning. After presenting theory refinement (Section 2) and a general description of ALLiS (Section 3), we will show how each step of TR is applying in the specific case of learning linguistic structures (non-recursive phrases).

2 About Theory Refinement

Theory refinement (hereafter TR) consists of improving an existing knowledge base so that it fits more with data. No work using theory refinement applied to the grammar learning paradigm seems to have been developed. We would like to point out in this article the adequacy between theory refinement and Natural Language Learning. For a more detailed presentation of TR, we refer the reader to (Abecker and Schmid, 1996), (Brunk, 1996). (Mooney, 1993) defines it as:

> Theory refinement systems developed in Machine Learning automatically modify a Knowledge Base to render it consistent with a set of classified training examples.

This technique thus consists of improving a given Knowledge Base (here a grammar) on the basis of examples (here a treebank). Some methods try to modify the initial knowledge base as little as possible. (Abecker and Schmid, 1996) presents the general algorithm as:

1. Build a more or less correct grammar *on the basis of background knowledge.*
2. Refine this grammar using training examples:
   (a) Identify the revision points
   (b) Correct them

The first step consists of acquiring an initial grammar (or more generally a knowledge base). In this work, the initial grammar is automatically induced from a tagged and bracketed corpus. The second step (refinement) compares the prediction of the initial grammar with the training corpus in order to, first, identify the *revision points*, i.e. points that are not correctly described by the grammar, and second, to correct these revision points.

3 ALLiS

ALLiS (Architecture for Learning Linguistic Structures) (Déjean, 2000a) is a symbolic machine learning system which generates categorisation rules from a tagged and bracketed corpus. These categorisation rules allow (partial) parsing. Unless (Brill, 1993), these rules cannot be directly used in order to parse a text. ALLiS uses an internal formalism in order to represent the grammar rules it has learned. This internal representation (Table 1) allows the use of different systems in order to parse the structures. Each system requires a conversion of theses rules into its formalism. This use of "intermediary" formalism allows the separation of two different problems: the generation of (linguistic) rules and the use of them. Unless Transformation-Based Learning (Brill, 1993) which modifies training data each time a rule is learned, ALLiS always uses the original training data. By this way you try to separate the problem of learning "linguistic" rules to the problem of parsing (the adequate use of these rules). The rules generated contains enough information (elements which compose the contexts, structures of these elements) so that we can correctly generate rules for a specific parser. We can note that, although rules have to be ordered during the parse, this order does not depend on the order used during the learning step, but depends on the category of the element.

	Context		in(1) or out(2)		
Tag	Left	Right	W	L	R
VBG	PRP$		1	1	
VBG	POS		1	1	
VBG	JJ		1	1	
VBG	DT		1	1	
VBG	TO		1	2	
VBG	IN		1	2	
VBG	VBG		1	2	

Table 1: Contexts generated for the categorisation of the category AL (NP).

Table 1 shows a part of the file generated concerning the categorisation of the tag *VBG*. The first line has to be read: when the tag VBG occurs after the tag PRP$ (left context) and when the tag PRP$ occurs in the structure (L=1(in)), the tag VBG is categorised as *AL* (left adjunct: see next section). In order to parse a text, a module automatically converts this formalism into appropriate formalisms which can be used by existing symbolic parsers. Several tools have been tried: the CASS parser (Abney, 1996), XFST (Karttunen et al., 1997)) and LT TTT (Glover et al., 1999). The TTT formalism seems to be the most appropriate (rules are easy to generate and the resulting parser is fast). The TTT rule corresponding to the first line of the table 1 is given table 2

```
<RULE name="AL" targ_sg="@[CAT='AL']">
<REL match="W[C='PRP$'
              m_mod='TEST'
              S='NP']">
</REL>
<REL match="W[C='VBG']">
</REL>
</RULE>
```

Table 2: TTT formalism.

4 The Generation of the Initial Grammar

The first step is to assign to each tag of the corpus a default category corresponding to its most frequent behaviour regarding the structure we want to learn. The result of its operation is a set of rules which assign a default category to each tag.

In general, the baseline is computed by giving an element its most frequent tag. ALLiS uses an initial grammar which is a little more sophisticated: it uses the same principle with the exception that the default tag depends on contexts. Generally the chunk tagset is composed of three tags: B,I, and O. ALLiS uses a subcategorisation of the I category. It considers that a structure is composed of a Nucleus (tag N) with optional left and right adjuncts (AL and AR). These three classes (AL, N, AR) possess an attribute B[1] with the value $+/-$. Furthermore, an element is considered as AL/AR iff it occurs before/after a nucleus. For this reason, a tag such as *JJ*[2] can be categorised as AL or O(outside) according to its context. Precision and recall of this initial grammar are around 86%. An example of NP analyse provided by the initial grammar is:

(1) [It_PRP_N] 's_VBZ_O [traders_NNS_O] squaring_VBG_O [positions_NNS_N] .

(2) The_DT_0 operating_VBG_O [chief_NN_N 's_POS_AL$_{B+}$] [post_NN_N] is_VBZ_O new_JJ_O .

The initial grammar categorises the tag **VBG** as occurring by default outside an NP, which is mainly the case (as in example (1)). But in

[1] Introduction of a break with the preceding adjacent structure (this property simulates the B tag).

[2] JJ: adjective (Penn treebank tagset).

some cases this default categorisation is wrong (example 2). Since the default structure is defined as: S $\rightarrow$ [AL* N AR*]+, the phrase *the operating chief* can not be correctly parsed by the initial grammar. Such an error can be fixed during the refinement step as explained in the next section.

5 The Refinement

Once this initial grammar is built, we confront it to the bracketed corpus, and apply the refinement step. The general theory refinement algorithm given by (Abecker and Schmid, 1996) is:

- Find revision points in theory
- Create possible revisions
- Choose best revision
- Revise theory

until no revision improves theory

The next sections now show how these operations are performed by ALLiS.

5.1 Revision Points

Revision points correspond to errors generated by the initial grammar. In the example (2), the word *operating* does not belong to the NP since the tag **VBG** is categorised as O(outside NP). This is thus a revision point. During the refinement, ALLiS finds out all the occurrences of a tag whose categorisation in the training corpus does not correspond to the categorisation provided by the initial grammar. Once revision points are identified, ALLiS disposes of two kinds of operators in order to fix errors: the *specialisation* and the *generalisation*. We just use basic implementation of these operators, but it is nevertheless enough to get efficient results comparable to other systems (Table 5).

5.2 The Specialisation

The specialisation relies on two operations: the *contextualisation* and the *lexicalisation*. The contextualisation consists of specifying contexts in which a rule categorises with a high accuracy an element. The table 1 provides examples of contexts for the tag *VBG* in which this tag occurs in an NP, and thus which fix the revision point of example (2). The lexicalisation consists

		Context				
Tag	Word	Left	Right	W	L	R
VBG	operating		NN	1		1
VBG	recurring		NNS	1		1
VBG	continuing		NNS	1		1

Table 3: Lexicalisation of the tag VBG.

of replacing[3] a tag by a specific word (Table 3). Some words in some contexts can have a behaviour which can not be detected at the tag level. If contextualisation is rather corpus independent, lexicalisation generates rules which depend of the type of the training corpus. More details about these two operations can be found in (Déjean, 2000b).

5.3 The Generalisation

After specialisation, some structures are still not recognised. If some revisions points can not be fixed using only local contexts, a generalisation (by relaxing constraints) in the definition of the structure can improve parsing. A structure is composed of a nucleus and optional adjuncts (Section 3). Such a structure can not recognise all the sequences categorised as NP in the training corpus. These unrecognised sequences are composed of elements without nucleus. In example (3), the sequence *the reawakening* composes a NP although it is tagged as *AL AL* by ALLiS.

(3) [**the_DT reawakening_VBG**] of_IN [the_DT abortion-rights_NNS movement_NN]

Generalisation consists of accepting some sequences of elements which do no correspond to a whole structure (S $\rightarrow$ AL* N AR* | AL+ | AR+). The technique we use for this generalisation is just the deletion of the element N in the rule describing a structure. More generally, this step allows the correct parse of sequences where ellipsises occur. The most frequent partial structures correspond to the sequences: *DT JJ*, *DT VBG* and *DT*.

5.4 The Selection of Rules

During the operations of specialisation and generalisation, rules are generated in order to improve the initial grammar. But combination

[3]The lexicalisation can be considered as a replacement of a variable by a constant.

of both lexicalisation and contextualisation can yield rules which are redundant. In the table 4, the two last rules are learned whereas the first is enough.

Tag	Word	left	right
VBG	operating		NN
VBG	operating	IN	NN
VBG	operating	VBD	NN

Table 4: Superfluous rules.

The purpose of its step is to reduce the number of rules ALLiS generated. In fact the number of rules can be reduced during the specialisation step. But a simplest way is to select some rules after specialisation and generalisation according to heuristics.

The heuristic we used consists first of selecting the most frequent rules and then among them, those having the richest (longest) context (several rules can be obtained using only the criterion of frequency). In our case (learning linguistic structures), this heuristic provides good result, but a more efficient algorithm might may consist of parsing the corpus with the candidate rules and to select the most frequent rules providing the best parse.

We can note that these superfluous rules do not generally produce wrong analyses, even if some are not linguistically motivated. The fact that we try to get the minimal revised theory is computationally interesting since the reduction of rules eases parsing.

6 Results

ALLiS was used in order to learn several structures (Déjean, 2000b). The table 5 shows results for VP and NP and results obtained by other systems[4]. The best result is obtained by (Tjong Kim Sang, 2000) using a combination of NP representation. ALLiS offers the best score for the symbolic systems.

		P/R	F
NP	TKS00	93.63/92.89	93.26
	MPRZ99	92.4/93.1	92.8
	ALLiS	92.38/92.71	92.54
	XTAG99	91.8/93.0	92.37
	RM95	92.27/91.80	92.03
VP	**ALLiS**	92.48/92.92	92.70

Table 5: Results for NP and VP structures (precision/recall).

7 Conclusion

We showed that even a simple implementation of TR provides good results (comparable to other systems) for learning non-recursive structures from bracketed corpora. The next steps concern two directions. First the improvement of algorithms used by ALLiS (especially the selection of rules). The second step consists of applying ALLiS on other structures, especially the clause so that ALLiS can provide a more complete parsing.

[4]More complete results are shown in (Déjean, 2000a) and (Déjean, 2000b).

References

Andreas Abecker and Klaus Schmid. 1996. From theory refinement to kb maintenance: a position statement. In *ECAI'96*, Budapest, Hungary.

Steven Abney. 1996. Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.

Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Clifford Alan Brunk. 1996. *An investigation of Knowledge Intensive Approaches to Concept Learning and Theory Refinement*. Ph.D. thesis, University of California, Irvine.

Hervé Déjean. 2000a. Theory refinement and natural language learning. In *COLING'2000*, Saarbrücken.

Hervé Déjean. 2000b. A use of xml for machine learning. In *Proceeding of the workshop on Computational Natural Language Learning, CONLL'2000*.

Claire Glover, Andrei Mikheev, and Colin Matheson, 1999. *LT TTT version 1.0: Text Tokenisation Software*. http://www.ltg.ed.ac.uk/software/ttt/.

Lauri Karttunen, Tamás Gaál, and André Kempe. 1997. Xerox finite-state tool. Technical report, Xerox Research Centre Europe, Grenoble.

Raymond J. Mooney. 1993. Induction over the unexplained: Using overly-general domain theories to aid concept learning. *Machine Learning*, 10:79.

Erik F. Tjong Kim Sang. 2000. Noun phrase representation by system combination. In Morgan Kaufman Publishers, editor, *Proceedings of ANLP-NAACL 2000, Seattle*.