

Lexicalizing a shallow parser

Roberto Basili, Maria Teresa Pazienza and Fabio Massimo Zanzotto

University of Rome Tor Vergata,
Department of Computer Science, Systems and Production,
00133 Roma (Italy),
{basili,pazienza,zanzotto}@info.uniroma2.it

Résumé

Current NL parsers are expected to run with throughput rate suitable to satisfy "time constraints" in real applications. The aim of the present work is, on the one hand, to investigate the effects of lexical information in a shallow parsing environment, on the other hand, to study the limits of a bootstrapping architecture that, automatically learning the lexical information in an unsupervised fashion, guarantees the reliability and portability of the parser to different domains. The investigated parser is *Chaos* (*Chunk analysis oriented system*), a robust parser based on stratification and lexicalization. Large scale evaluation over a standard tree bank is discussed.

1. Introduction

Advanced Natural Language (NL) parsers are expected to recognize grammatical phenomena with a throughput suitable to satisfy "time constraints" in real applications. Shallow and robust analyzers have been proposed in recent years to improve this throughput in contrast with top-down full-sentence parsers (PSG (Chomsky, 1957), LFG (Darlymple *et al.*, 1995), HPSG (Pollard & Sag, 1994)): higher processing speed, lower costs in the design and maintenance of grammars are the main benefits, with a corresponding reduction of the set of output information.

Shallow parsing techniques (e.g. (Appelt *et al.*, 1993; Basili *et al.*, 1992; Aït-Mokhtar & Chanod, 1997)) are thought to increase the throughput and reduce costs of grammar design. They are usually based on efficient representations and algorithms (e.g. finite state automaton) and are focused on very specific phenomena (e.g. noun phrases parsing) (Appelt *et al.*, 1993), or are dedicated to preliminary stages of lexical acquisition processes (e.g. (Basili *et al.*, 1992)). On the other hand, robust parsers arose as tools able to deal with free occurrence texts, (Carroll & Briscoe, 1996), and thus they are usually more ambitious tools producing a more general set of grammatical information: from, possibly ambiguous, dependency graphs (e.g. (Grinberg *et al.*, 1996) to disambiguated parse trees, (Srinivas, 1997)). Robust parsers can be those based on rich logical formalisms (e.g. morphosyntactic constraints as in (Aït-Mokhtar & Chanod, 1997), or lexicalized super-tags (Srinivas, 1997)) or numerical approaches (e.g. statistical estimation as in data oriented parsing (Bod, 1993)).

Even if the principles inspiring shallow parsing and robust parsing techniques differ, the two approaches have several commonalities. Both relay on partial parsing methodology and multi-stage approaches (e.g. the finite automaton cascade (Abney, 1996; Ait-Mokhtar & Chanod, 1997)).

The good trade-off between expressiveness and efficiency in shallow and robust parsers is a basic property to strengthen their portability throughout changing operational environments, with respect to sub-languages and NLP tasks. The high level of re-usability lies in the fact that the grammatical recognition for a shallow and robust parser is under-specified: the variety of target phenomena is rather small and the underlying resources (e.g. grammars) are not fully specialized. This property is also a weakness. In fact, the range of phenomena that can be treated in a reliable way is not very large with respect to the potential application needs. The case of event recognition, in Information Extraction (IE) (MUC-6, 1995; Pazienza, 1997), is just an example.

Portability is very high in shallow parsers as they usually do not employ lexical knowledge in parsing. The domain independence of the overall process increases as no lexicon nor sub-language specific information is necessary. However this may result in a reduction of accuracy (i.e. low precision). Look for example, at the Penn Tree-bank (Marcus *et al.*, 1993) sentence #1692:

(wsj_1692) As part of the agreement, Mr. Gaubert contributed real estate valued at \$ 25 million to the assets of Independent American.

The prepositional phrases in this sentence are intrinsically ambiguous: *at \$ 25 million, to the assets* and *of Independent American* can refer to verbs (e.g. *contributed, valued*) and nouns (i.e. *real estate, millions* and *assets*). This produces a proliferation of alternative readings that highly affects precision. Lexical information here may play a crucial role. Two of the three PPs are in fact argumental (i.e. belong to a verb argument structure): *at \$ 25 million* is in fact subcategorized by the verb *valued* and *to the assets* is clearly the "recipient/destination" argument of verb *contribute*. Making this information available drastically reduces the ambiguity in a shallow parsing framework. Aim of the current work is, on the one hand, to investigate the possibility of improving the accuracy of a shallow parser by making it sensitive to verb subcategorization lexical information, on the other, to propose and evaluate a bootstrapping architecture that permits to maintain the low costs in design and maintenance of grammars and the portability throughout the changing of operational environments, typical characteristics of shallow parsing processors. The bootstrapping architecture consists of a shallow parser sensitive to lexical information and a verb subcategorization lexicon learner. The potentials of the technology are investigated through a large scale evaluation (cf. sec. 4).

The parsing processor we are here investigating and integrating in the architecture is *Chaos* (*Chunk analysis oriented system*), an English and Italian robust parser based on stratification and lexicalization. It is based on the largely shared principle that verbs play the role of determining the semantics of a sentence, and, thus, of projecting most of its grammatical structures. The lexicalised grammar rules employed in *Chaos* are, thus, the subcategorization frames for verbs. The advantage of this parser is that, when possible, it exploits the available subcategorization lexicon, but, it reduces to a shallow parser otherwise. Its description of *Chaos* is given in Sec. 2. In the overall architecture, the parser is coupled with a learning module *RGL*, the Rome Galois Lattice that derives the subcategorization lexicon. The adaptivity of the architecture to different sublanguages, is discussed in sec. 3.2 where the bootstrapping architecture is also described.

2. A not-so-shallow parser: *Chaos*

In order to improve the accuracy of a pure shallow parsing approach a given level of lexical information should be made available, especially with respect to phenomena whose resolution is crucial in application scenarios like IE. Here we present *Chaos*, a robust and shallow syntactic analyzer sensible to subcategorization lexical information. The underlying principles are discussed in sec. 3, while its architecture is presented in sec. 3.1. The bootstrapping architecture is finally presented in sec. 3.2.

3. The principles of *Chaos*

The parsing method presented here is based on the basic assumption that verbs determine the semantics of a sentence and its surface realization is strictly dependent on this fact. Verbs characterize the set of syntactic restrictions over the grammatical representation of the target sentence. They are widely recognized as the heads of sentences (Pollard & Sag, 1994). Furthermore, several NLP tasks are based on the relations that verbs establish with the other words.

Let us examine, in the example sentence (*wsj_1692*) of the previous section, how a stratified and lexicalized approach results in an increase of the parsing accuracy. A first phase, *chunking* (Abney, 1996), could be designed to pack segments whose structures are independent from any verb grammatical projection. Simple noun phrases (e.g. *Mr. Gaubert, real estate*) and modifiers (e.g. *to the assets, at \$ 25 million*) are examples of these structures. Looking at grammatical relations among chunks in the example sentence, only those established by the verb *contribute* with its subject (i.e. *Mr. Gaubert*), and with its (adjacent) object *real estate* can be detected in unambiguous way. Without any other lexical information the legal and unambiguous relations are very few.

Let us suppose that the parser dispose of *subcategorization* information (i.e. it knows argument structures fully describing the verb syntactic behavior). *To contribute* could be associated with a direct object but also a recipient (or beneficiary) argument, resulting in a frame like *contribute-NP-PP(to)*¹. *To value* is associated with an object (i.e. the evaluated entity) and to the prepositional phrase expressing the "degree/amount" (usually ruled by preposition *at*), i.e. *value-NP-PP(at)*. Most of the ambiguities in the sample sentence disappear since they are resolved on lexical basis. With respect to the example sentence, a strategy that uses a combination of clause boundary recognition and verb argument detection could decide that:

1. *valued* is linked to *at \$ 25 million* so that the maximal lower bound of the corresponding clause is extended to include such a PP (i.e. *(valued at \$ 25 million)_{VP}*);
2. *contributed* is linked to *to the assets* so that, similarly, the maximal lower bound of its clause is *(Mr. Gaubert contributed real estate valued at \$ 25 million to the assets)_{VP}*.

Note that, links derived on lexical basis (i.e. according to subcategorization information) have important effects on the remaining ambiguities: other potential attachment sites of the argumental PPs like *at \$ 25 million* and *to the assets* are discarded. Moreover, persistent ambiguity is reduced. The *(of Independent American)_{PP}* structure is no longer allowed to attach to nouns like *real estate* or *million* as illegal bracket crossing of the clause related to *contribute* would be generated: as a result the only allowed attachments are with the verb *contribute* itself or with the noun *assets*.

¹Note that the subject is missing from the subcategorization dictionary as it is mandatory in syntax, although it can be omitted.

The Chaos parser, briefly summarized in Section 3.1, is designed as a stratified recognizer (i.e. applying a cascade of processing steps), based on (verb) lexical information according to the strategy suggested in the above example. As performances of a lexicalized parser strictly depend on the quality of the available lexical information a crucial problem is how to make this information available avoiding the huge costs required by manual compilation of a lexicon. This is why our experimental set-up foresees the use of an automatic acquired lexicon in a unique structured architecture.

In this perspective, *Chaos* is able to run without access to verb subcategorization lexical information (*blind parsing*), thus supporting a bootstrapping approach:

1. a corpus is firstly processed without a subcategorization lexicon;
2. detected grammatical information is used as a basis to induce subcategorization information;
3. the lexicalized parser is finally applied to the target corpus.

The bootstrapping architecture is presented in Section 3.2.

3.1. The Chaos architecture

The overall architecture has been designed to exploit an available verb subcategorization lexicon. The resulting parser should inherit both the computational efficiency of a shallow parser and the accurate syntactic information typically produced by a lexicalized approach. The design choice to give priority to the verb argumental connections induces a stratification of the parsing processor. The first stage has the role of packing the ambiguities that are not under the control of the verb projections, i.e. the *cores* of nominal phrases, prepositional phrases, adjectival phrases, and verbal phrases. This level of stratification introduces an intermediate level between words and sentences, the level of *chunks*. The recognition of these bigger units has a computational cost comparable to the one of a finite state automaton. In the overall architecture (Fig. 1), this stage is embodied by the *Chunker* module. Tokenized and morphologically annotated sentences (*am sentences* in figure) are given as input to the *Chunker*.

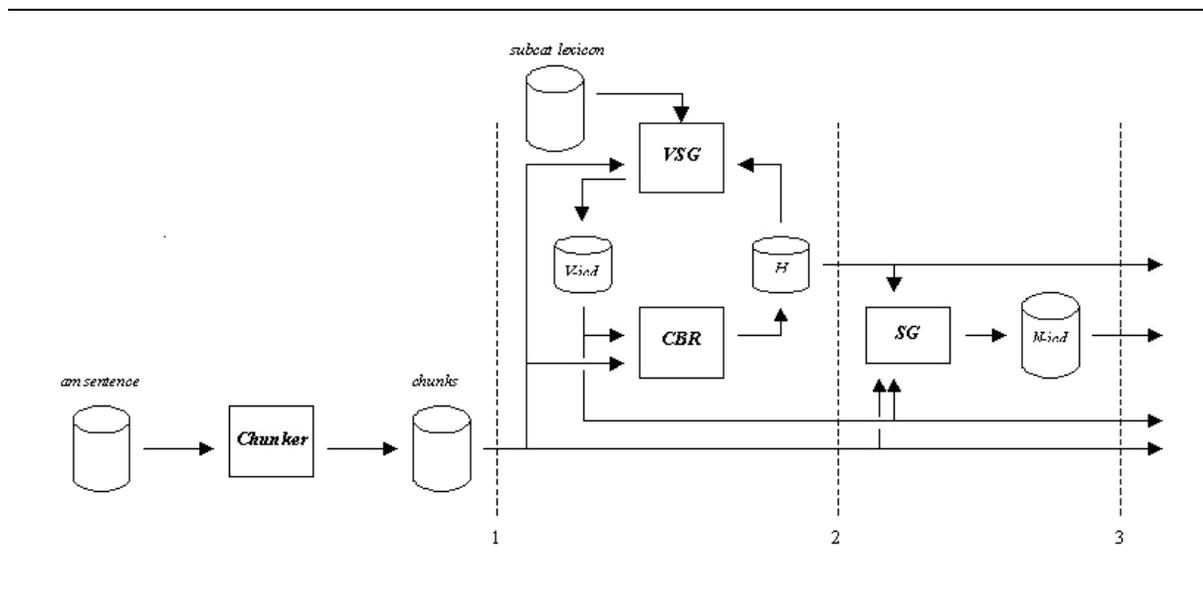


Figure 1: The Chaos syntactic processor architecture

The second stage uses the verb subcategorization lexicon in order to detect the verb arguments in the sentence. The adopted strategy investigates the arguments of verbs exploiting the approximation of clause boundaries.

In the system (Fig. 1), chunks are used as input to the Clause Boundary Recognition (*CBR*) aiming to recognize clauses and structure them in a hierarchy (*H*). The recognition of clauses is integrated with a special purpose parser (Verb Shallow Recognizer, *VSG*) aiming to detect relations between a verb and members of its subcategorization pattern (i.e. its arguments). The interaction between the *CBR* and *VSG* provides a combined recognition of the clause hierarchy and the set of argumental dependencies of verbs, namely *Verbal inter-chunk dependencies* (*V-icds*). The interleaving between verb argument and clause boundary detection makes these last constantly upgraded, so that bracket crossing is used as an incremental constraint on the later steps. A right-to-left analysis is carried out in this phase.

Finally, the third step of analysis is the Shallow recognizer (*SG*) triggered by *Chunks*, the clause hierarchy *H* and the known (i.e. detected) argumental relations (*V-icd*, verb inter-chunk dependencies).

The final representation of the sentence is a graph whose nodes are words and whose edges are inter-chunks dependencies (*iwds*²). The graph gathers the set of alternative planar graphs (Grinberg *et al.*, 1996) representing the grammatical information of the sentence. *Plausibility*, as a degree of confidence, is associated to each *iwd* (Basili *et al.*, 1992). Unambiguous links are associated with the plausibility of 1. Lower plausibility will score ambiguous dependencies (e.g. persistently ambiguous PPs, like *of Independent American*)_{PP} structure in the above example).

3.2. Bootstrapping a robust syntactic processor

In order to approach the problem of manual compilation of the lexicon a bootstrapping architecture is here proposed. It is designed both to lower the costs of manual compilation and to improve the accuracy of the parsing over the target corpus. This architecture has two basic components: *Chaos* and a conceptual clustering module employed for subcategorization frame learning. The learning module *RGL*, the Rome Galois Lattice (Basili *et al.*, 1997), uses of a clustering algorithm based on conceptual lattices, extended with linguistic rules suitable for the specific learning problem.

The strength of the syntactic processor is the ability to run at different levels of lexicalization. This allows to design an adaptive approach to parsing. The grammatical information gathered from the corpus by the parser *Chaos* without subcategorization lexical information is used to feed *RGL*, the learner of verb subcategorization frames. The resultant subcategorization lexicon can be then reused to improve the parsing performances. Note that, in absence of lexical information, the basic heuristics on arguments is that a generic verb has a subject and an object. Note that unambiguous modifiers (e.g. adjacent PPs) are also attached with maximal plausibility.

4. Performance Evaluation

To evaluate the impact of an automatically acquired subcategorization lexicon on the performances of the parser, extensive experiments have been carried out. The aim is both to study the

²As the purpose of this paper is to evaluate the parser with respect to a constituency based tree bank, inter-chunk dependencies will be treated here as inter-word dependencies, where chunks are mapped into words corresponding to their heads

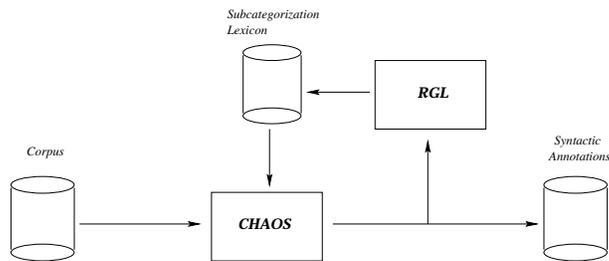


Figure 2: Bootstrapping architecture

limits of the proposed bootstrapping architecture and to demonstrate that the lexical sensitivity of *Chaos* produces an improvement on the parsing accuracy whenever lexical information is made available. In order to measure the accuracy a systematic evaluation method has been defined. The simulation performed allowed to reproduce the operational scenario of a typical NLP application, where a corpus but no lexicon is available in advance. For the evaluation a *syntactically annotated corpus* (SyAC) has been used, as shown in Fig. 3. The grammatical information embodied by the SyAC has been used to acquire the subcategorization lexicon by means of *RGL*. Such architecture supports the evaluation of the upper bound accuracy. In fact, *the syntactic annotation available to the learner RGL is error free, and what is measured is only the effect of the correct lexical information*. In this way errors due to limitations of the corpus blind parsing are missing due to the supervision ensured by the annotations over the source training data.

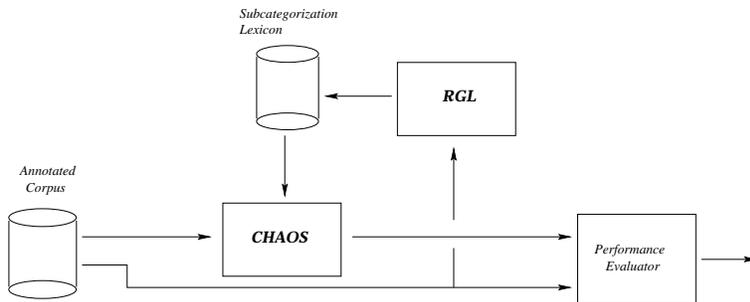


Figure 3: Evaluation architecture: an upper bound

The resulting lexicon is thus the best one (according to the *RGL* inductive capability). Note that in this architecture, any improvement of the parsing accuracy is thus due purely to verb subcategorization lexical information. In the section 4.1, the evaluation scheme is settled, while, in section 4.2, the experiments and their results are discussed.

4.1. The evaluation scheme: a re-adapted Parseval

The evaluation of parsing accuracy requires the simulation of an operational scenario. A corpus-oriented evaluation scheme has thus been preferred to a *test-suite*-oriented scheme (Netter *et al.*, 1998; Balkan *et al.*, 1994). Among the proposed evaluation scheme (Carroll & Briscoe, 1998), a paradigm *Parseval*-like (Black *et al.*, 1991) re-adapted in a dependency-based framework has been adopted. Similarly, performance metrics like *Recall*, *Precision* and

f-measure have been defined over a dependency-based representation. The employed corpus and reference syntactic information is the Penn tree bank (PTB) (Marcus *et al.*, 1993). In this evaluation framework, the translation of the PTB constituency-based to the dependency-based annotation scheme, compliant with the evaluation requirements, is a crucial problem. Translation algorithms have been settled in previous works (Lin, 1995; Basili *et al.*, 1998). In the present work the adopted translation algorithm left untranslated about 10% of the *oracle trees* (i.e. reference corpus trees). The resulting evaluation test-set consists of nearly 44,000 sentences.

Under the derived representation *oracle choices* and *system guesses* are represented for a sentence S respectively with:

$$\begin{aligned} G_{oracle}(S) &= (words_{oracle}, iwds_{oracle}) \\ G_{system}(S) &= (words_{system}, iwds_{system}) \end{aligned}$$

where *words* is the ordered set of morphologically analyzed words of S and *iwds* is the set of inter-word dependencies. Note that $G_{system}(S)$ is a family of alternative *planar graphs* representing the sentence S .

Since we assume that $words_{oracle} = words_{system}$, recall, precision, and f-measure over the syntactic phenomena are calculated comparing inter-word dependencies (i.e. *iwds*) assessed by the two sources of information. Our aim is to measure accuracy over different type of grammatical information. Thus, the used instance for recall R and precision P are dependent on the type of syntactic information:

$$\begin{aligned} R &= \frac{card((iwds_{oracle} \cap iwds_{system})|LinkTypes)}{card(iwds_{oracle}|LinkTypes)} \\ P &= \frac{card((iwds_{oracle} \cap iwds_{system})|LinkTypes)}{card(iwds_{system}|LinkTypes)} \end{aligned}$$

LinkTypes are used to project the sets the target syntactic phenomena (e.g. $NP - PP$). A synthetic comparison index is the f-measure $F(\alpha)$:

$$F(\alpha) = \frac{1}{(\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R})}$$

that combines recall and precision with a relative importance factor α . As a consequence, it expresses the global performance improvement achieved by using the lexicon.

4.2. Evaluating the lexicalization of the parser

Experiments aim to demonstrate that lexical information improves the parsing accuracy, and that automatic acquisition of the subcategorization lexicon is viable for the proposed bootstrapping architecture.

Investigated grammatical phenomena are those dependent on the verb subcategorization information. Table 1 focuses on the set of unambiguous dependencies of verbs, and provides evidence on extension of this information when a lexicon is available. Table 2 shows how the information available for verbs reduces the ambiguity over strictly related phenomena, i.e. pp-attachments.

In Table 1 the set of unambiguous verb attachments are characterized by a poorer accuracy. In fact, arguments different from *subjects* and *objects* and recognized unambiguously (i.e. plausibility $pl() = 1$) can be used with a high confidence ($P = 0.94$), but coverage of the phenomena is still far from being satisfactory ($R = 0.58$). The improvement obtained with the use of automatic acquired subcategorization lexicon may be evaluated by comparing the f-measure values. The chosen value for α assigns the same importance to precision and recall measures. While no appreciable improvement exists in the case of *subject* and *object* arguments due to the good performance of the heuristic employed in the shallow version, a significant improvement is obtained for the remaining arguments ($F(\alpha) = 0.72$ vs. $F(\alpha) = 0.77$). Note that, when using the lexicon, the coverage of the phenomena is also largely increased ($R = 0.70$). Note that we are not far from the limit of coverage of the phenomenon typical of the underlying shallow parsing technique. This limit is represented by the value of $R = 0.82$ obtained by the system in absence of subcategorization lexicon considering all the possible links. Under this perspective, the lexicalized version of the parser covers 85% of the phenomena that the blind parser grasps.

Lexicon	plaus	Link Type	R	P	$F(\alpha = 0.5)$
<i>no</i>	1	V-Sub	0.75	0.89	0.82
		V-Obj	0.90	0.66	0.76
		V-PP	0.58	0.94	0.72
<i>no</i>	<i>any</i>	V-PP	0.82	0.58	0.68
<i>yes</i>	1	V-Sub	0.76	0.89	0.82
		V-Obj	0.90	0.69	0.78
		V-PP	0.70	0.86	0.77

Table 1: verb arguments

Table 2 shows the effect of lexical information about verbs on the NP-PP attachments. Note that any decision made according to the verbal lexicon reflects, because of the planarity constraints, on the attachments of PPs to nouns. The Table shows an increase of the precision with a corresponding small loss in term of coverage. The global effect is described by an improvement of the f-measure: $F(\alpha) = 0.73$ without lexicon vs. $F(\alpha) = 0.78$ with lexicon.

Lexicon	plaus	Link Type	R	P	$F(\alpha = 0.5)$
<i>no</i>	<i>any</i>	NP-PP	0.85	0.65	0.73
<i>yes</i>	<i>any</i>	NP-PP	0.82	0.75	0.78

Table 2: noun phrases-prepositional phrases attachment

The global performance value for the NP-PP attachment problem is analysed more finely in Fig. 4. The effect of the lexicon is studied with respect to the complexity of the target sentence. An approximate estimation of the complexity of a sentence may be modeled as follows:

$$\text{Sentence Complexity} = \frac{\#LVs + \#LNs}{\#Clauses}$$

where $\#LVs$ and $\#LNs$ are the number of verbal and nominal links (i.e. VP-PP and NP-PP) defined by the oracle, while $\#Clauses$ is the number of clauses in the sentence.

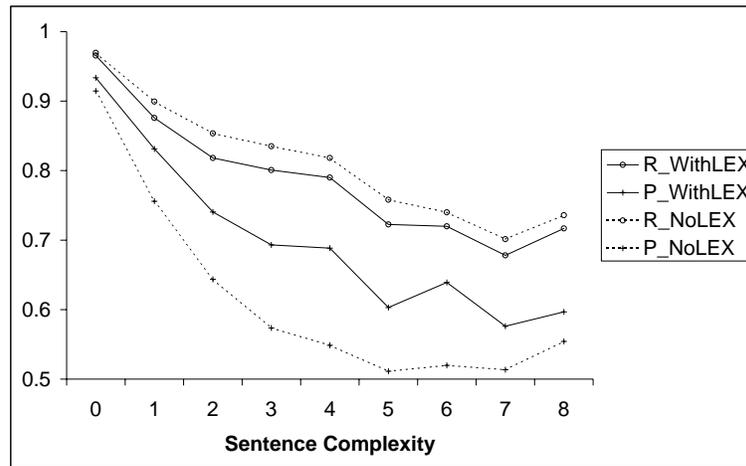


Figure 4: pp-attachment performances vs. sentence complexity

As the Fig. 4 suggests the trend related to the recall plot (and its values) are very similar in both cases (with or without lexicon), while the effect of the lexicon has a stronger effect over more complex sentences (i.e. 3 – 7). A trashing effect emerges for sentences whose level of complexity reaches 8. Due to the very small number of sentences whose complexity was higher than 8, they have been removed from the plot.

5. Conclusions

A structured architecture for an incremental and adaptive approach to parsing has been proposed. A study on the improvements in performance of a shallow lexicalized parser has been extensively carried out. By allowing the parser (*Chaos*) to access corpus-related subcategorization information, the number and the quality of recognized grammatical phenomena has significantly improved. Specific measures of recall and precision have been defined and different phenomena have been studied. Subjects and Objects are easier to be correctly recognized as they generally occur more closely to the verb (at least in the available data sets). On the contrary, PPs show less regularity in the sentence and their ambiguity is very high. The poorer accuracy of the PP recognition is improved by the lexicalization supported by *Chaos*. PP attachments are recognized with higher values of the f-measure when a subcategorization lexicon is used. This is a specific property of the *Chaos* parsing approach. The proposed adaptive architecture makes use of a learning algorithm that makes available the subcategorization information, deriving it directly from the target corpus. The adopted learning method, although tested here in a supervised fashion, is in principle unsupervised. The positive effects on the parsing accuracy can thus be reproduced in a real operational scenario, as no manual compilation of the lexicon is necessary. The viability of the overall approach has been proofed over extensive data sets. Further improvements may be obtained by more complex learning algorithms and better parsing strategies for specific sentence structures (e.g. wh-clauses, relative clauses).

Références

ABNEY S. (1996). Part-of-speech tagging and partial parsing. In G. K.CHURCH, S.YOUNG, Ed., *Corpus-based methods in language and speech*. Dordrecht: Kluwer academic publishers.

- AÏT-MOKHTAR S. & CHANOD J.-P. (1997). Incremental finite-state parsing. In *Proceedings of ANLP97*, Washington.
- APPELT D., HOBBS J., BEAR J., ISRAEL D. & TYSON M. (1993). Fastus: a finite-state processor for information extraction from real-world text. In *13th International Joint Conference on Artificial Intelligence*.
- BALKAN L., NETTER K., ARNOLD D. & MEIJER S. (1994). Tsnlp — test suites for natural language processing. In *Proceedings of Language Engineering Convention*, p. 17–22, Edinburgh: ELSNET.
- BASILI R., PAZIENZA M. T. & VELARDI P. (1992). A shallow syntactic analyser to extract word association from corpora. *Literary and linguistic computing*, **7**, 114–124.
- BASILI R., PAZIENZA M. T. & VINDIGNI M. (1997). Corpus-driven unsupervised learning of verb subcategorization frames. Number 1321 in LNAI, Heidelberg, Germany: Springer-Verlag.
- BASILI R., PAZIENZA M. T. & ZANZOTTO F. M. (1998). Evaluating a robust parser for italian language. In *Proc. of the THE EVALUATION OF PARSING SYSTEMS Workshop, held jointly with 1st LREC*, Granada, Spain.
- BLACK E., ABNEY S., FLICKENGER D., GDANIEC C., GRISHMAN R., HARRISON P., HINDLE D., INGRIA R., JELINEK F., KLAVANS J., LIBERMAN M., MARCUS M., ROUKOS S., SANTORINI B. & STRZALKOWSKI T. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proc. of the Speech and Natural Language Workshop*, p. 306–311, Pacific Grove, CA.
- BOD R. (1993). Using an annotated corpus as a stochastic grammar. In *Proceedings of the European Chapter of ACL'93, Utrecht*.
- J. CARROLL & T. BRISCOE, Eds. (1996). *Proceedings of the WORKSHOP ON ROBUST PARSING, held jointly with ESSLI96*, Prague, Czech Republic.
- CARROLL J. & BRISCOE T. (1998). A survey of parser evaluation methods. In *Proc. of FIRST INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION*, Granada, Spain.
- CHOMSKY N. (1957). *Aspect of Syntax Theory*. Cambridge, Massachusetts: MIT Press.
- M. DARLYMPLE, R. M. KAPLAN, J. T. MAXWELL III & A. ZEANEN, Eds. (1995). *Formal Issues in Lexical-Functional Grammar*. US: CSLI Publications.
- GRINBERG D., LAFFERTY J. & SLEATOR D. (1996). A robust parsing algorithm for link grammar. In *4th International workshop on parsing technologies*, Prague.
- LIN D. (1995). A dependency-based method for evaluating broad-coverage parsers. In *Proc. of the 14th IJCAI*, p. 1420–1425, Montreal, Canada.
- MARCUS M. P., SANTORINI B. & MARCINKIEWICZ M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, **19**, 313–330.
- MUC-6 (1995). Proceedings of the sixth message understanding conference(muc-6). In *Columbia, MD: Morgan Kaufmann*.
- NETTER K., AMSTRONG S., KISS T., KLEIN J., LEHMANN S., MILWARD D., REGNIER-PROST S., SCHALER R. & WEGST T. (1998). Diet - diagnostic and evaluation tools for natural language processing. In *Proc. of FIRST INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION*, Granada, Spain.
- PAZIENZA M. T. (1997). *Information Extraction. A Multidisciplinary Approach to an Emerging Information Technology*. Number 1299 in LNAI. Heidelberg, Germany: Springer-Verlag.
- POLLARD C. & SAG I. (1994). *Head-driven Phrase Structured Grammar*. Stanford: Chicago CSLI.
- SRINIVAS B. (1997). *Complexity of Lexical Description and its relevance for partial parsing*. PhD thesis, University of Pennsylvania, Philadelphia.