Adaptive parsing for time-constrained tasks

Roberto Basili, Maria Teresa Pazienza, Michele Vindigni and Fabio Massimo Zanzotto

Department of Computer Science, Systems and Production,
University of Rome Tor Vergata,
Via di Tor Vergata 110, 00133 Roma (Italy),
{basili,pazienza,vindigni,zanzotto}@info.uniroma2.it

Abstract

Real Natural Language Processing (NLP) applications often involve cooperation among different processing modules, involving various degrees of linguistic skill. Advanced NL parsers are expected to recognize grammatical phenomena with a throughput suitable to satisfy "time constraints" in real applications. We present a robust and efficient syntactic recognizer, *Chaos* (*Chunk analysis oriented system*), able to capture at least the grammatical information assumed to be crucial for several linguistic and non linguistic inferences as required by an application system. The parser inherits both the computational efficiency of a shallow parser and the accurate syntactic information typically produced by a lexicalized approach. The potentials of the technology are investigated through different corpora. The parsing architecture proposed is open to the integration of domain specific lexical information, thus realizing an explicit level of adaptativity.

1. Introduction

Real Natural Language Processing (NLP) applications often involve cooperation among different layers, involving various degrees of linguistic skill. For instance, in interactive systems speech is potentially a very friendly interface: human-computer interaction involves not only voice recognition, but mainly the ability to reason upon the interaction itself and to build a discourse model: sentence analysis become thus important for triggering the required inference. The range of needed skills could vary along a continuum with respect to the involved background knowledge. This poses serious problems to the scalability and adaptivity of linguistic tools to new sub-domains. Advanced NL parsers are expected to recognize grammatical phenomena with a throughput suitable to satisfy "time constraints" in real applications. *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 and porting. They are usually based on efficient representation and algorithms (e.g. finite state automaton) and are focused on very specific phenomena (e.g. noun phrases parsing) (Appelt *et al.*, 1993), or dedicated to preliminary stages of lexical acquisition processes (e.g. (Basili *et al.*, 1992)) that could be easility customised on the specific task.

On the other hand, *robust parsers* arose as tools able to deal with free occurrence texts, (Carroll & Briscoe, 1996) 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).

Even if the principles inspiring shallow parsing and robust parsing techniques differ, the two approaches have several commonalities. 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.

Several research works suggest that efficient and robust syntactic processing is viable through processes of decomposition of the grammatical knowledge and lexicalization (D.Grinberg, 1996; Carrol & Briscoe, 1996; Abney, 1996). By sharing this basic assumption we have realized a robust and efficient syntactic recognizer, *Chaos* (*Chunk*

analysis oriented system), able to capture at least the grammatical information assumed to be crucial for several linguistic and non linguistic inferences as required by an application system. The approach is based on two major principles: *lexicalization* and *stratification* of the parsing process. In particular, the *stratification* is realized by a cascade of processing steps, as will be hereafter described.

The employed notion of *lexicalization* is mainly based on the use of subcategorization information as a control [1] strategy for the analysis: it is commonly argued that verbs play an important role in determining the semantics of a sentence, and, thus, in projecting most of its grammatical structures. Verb subcategorization frames are employed in *Chaos* as lexicalised grammar rules. The advantage of this parser is that, when possible, it exploits the available subcategorization lexicon, but, it reduces to a shallow parser otherwise. A syntethic description of *Chaos* is given in the next section. The potentials of the technology are investigated through different corpora in sec. 3, and some conclusions will be derived in final section.

2. The Chaos architecture

The overall framework of the syntactic processor to exploit a viable verb subcategorization lexicon (deeply described in (Basili *et al.*, 1998b)) is shown in Fig. 1. The resulting parser should inherit both the computational efficiency of a shallow parser and the accurate syntactic information typically produced by a lexicalized approach. A stratification of the parsing process is naturally induced by the design choice to assign priority to the verb argumental connections.

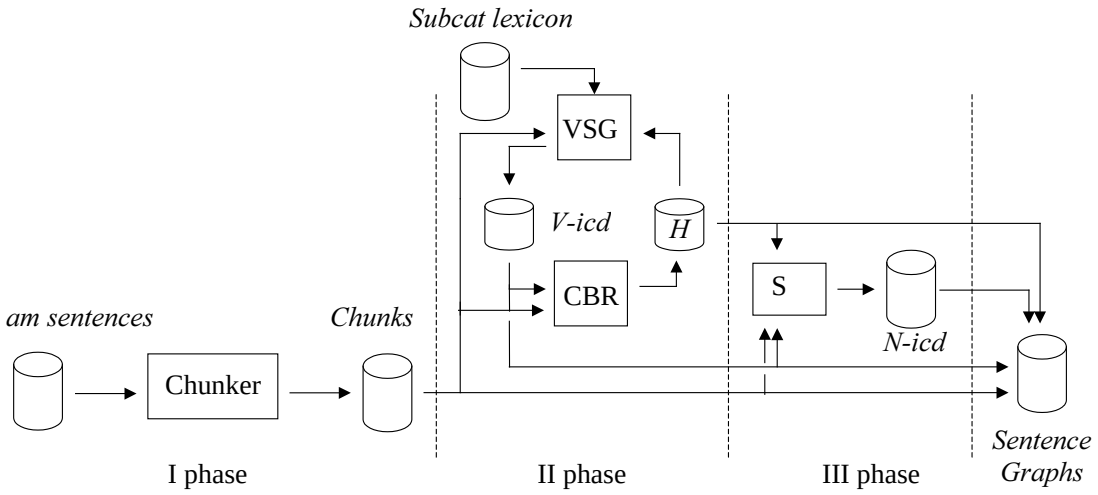

Figure 1: The Chaos syntactic processor architecture

The syntactic analysis starts from tokenized and morphologically annotated sentences (*am sentences* in figure).

2.1. The Chunker

The first processing step 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 and realises an intermediate level between words and sentences, the level of *chunks*.

This module, no more complex than a finite state automaton disburdens later phases of bottom-up parsing. The chunker is based on the notion of *island of non ambiguity* (Basili *et al.*, 1998a) for a grammar. It fully characterizes the nature of those unambiguous fragments of sentences that can in fact appear in a chunk. For instance, consider

[1] As several linguistic theories (e.g. HPSG) and parsing frameworks (e.g. LTAG, SLTAG, lexicalized probabilistic parsing) suggest, lexicon-driven systems ensure the suitable forms of grammatical control for many complex phenomena

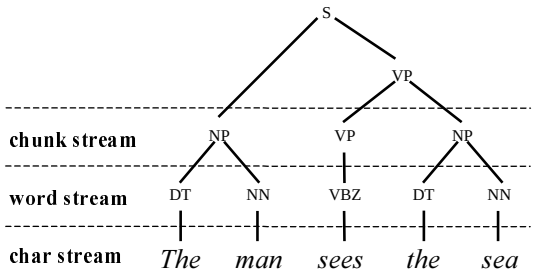

Figure 2: Interpretation levels of a simple derivation tree

the fragment *to find a consistent solution for the analysis* that is ambiguous and generates the two interpretation trees for the sentence:

```
(to find (a consistent solution) (for the analysis))
(to find (a consistent solution (for the analysis)))
```

The innermost equivalent subgraphs:

```
(to find)
(a consistent solution)
(for the analysis)
```

are those islands of non ambiguity claimed to be the focus of chunk analysis. Words belonging to equivalent subgraphs are characterized by specific sequences of morphological classes. The definition of a chunk prototype for each specific sequence characterizes all and only those fragments to be collapsed, as unambiguous. *A sequence of words can thus be considered a chunk if and only if it is an instance of a chunk prototype*. The unambiguous islands can be entirely defined once a general grammar for the underlying language is available. In (Abney, 1996) a formal derivation of chunk prototypes from a grammar is proposed and developped in (F.M.Zanzotto, 1997).

To fulfill the requirement of a shallow parser, inter-chunk dependencies (icd) must be detected by further processing steps. Since verbs play a crucial role in determining syntactic relations between words, and thus chunks: then our strategy will look first for verbal icds (i.e. those including at least a verbal chunk). To select first the more significant among these verbal icds appears to be a critical choice. We propose here to use verb subcategorization frames for such a task. As sentences have more than one verb and verbs define the different sentence clauses, the recognition of argumental icd influences also the identification of clause boundaries. Problems such as coordination and subordination between clauses are solved on the basis of verb arguments recognition.

The recognition of the complete hierarchy of the sentence clauses is refined incrementally along with the discovery of verb argumental icds for the different verbs.

2.2. The Lexicalised Parser

The second step uses a 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.

As shown in Fig. 1, chunks are fed to the Clause Boundary Recognition module (*CBR*) that recognizes clauses and structures them in a hierarchy (*H*). The recognition of clauses is strictly coupled with a special purpose parser (Verb Shallow Recognizer, *VSG*) 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.

Further grammatical information may be extracted from the sentence: information concerning non argumental verb modifiers (e.g temporal and spatial expressions) or typical noun modifiers (e.g. prepositional phrases or ad-

jectival specifiers) have not been extracted in the previous phase.

2.3. The Shallow Recognizer

The third step of analysis consists of 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). A special purpose parser is here adopted, following the approach in (R.Basili, 1992; Basili *et al.*, 1994). A discontinuous grammar is applied here to the fragments belonging to the different recognized clauses. Such an infra-clausal analysis allows specific rules being defined to capture binary relations between chunks (e.g. a nominal chunk, type `Nom`, and a prepositional chunk, type `Prep`).

The final representation of the sentence is a graph whose node are words and whose edges are inter-chunks dependencies (*iwds*). 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 in the above example *for the analysis*)$_{PP}$ structure).

The strength of the syntactic processor may be considered as the ability to run at different levels of lexicalization in accordance to the availability of accurate information on verb subcategorization frames [2]. In absence of lexical information, the basic heuristics on arguments assumes that a generic verb has a subject and an object; moreover, unambiguous modifiers (e.g. adjacent PPs) are also attached with maximal plausibility.

3. Performance Evaluation

To be able to use such a parsing framework in time-constrained applications we need to evaluate it and provide coherent scores. The evaluation of parsing results is usually a critical task as most systems are crucially tied to constraints and features directly related with the underlying linguistic theories. Some specific issues have to be highlighted: first, extensive controlled data set are often not available for languages other than English, and our interest is on specialized sublanguages; thus portability and robustness over different knowledge domains are crucial features. The suitability of the reference samples is a critical problem, even more than dimension of test sets; second, system requirements in terms of complexity of the source information (i.e. lexicons and grammars) are also relevant to evaluate portability and robustness. For these reasons we tested our system on samples extracted from corpora related to different domains, with differences in style and grammar.

3.1. Testset Definition

We applied the test over three corpora. A collection of financial news (referred hereafter as $Sole24Ore$), a collection of technical and scientific papers on the environment ($ENEA$) and excerpts of legal documents on italian V.A.T. laws ($Legal$) whose features and processing times are described in Table 1. [3]
Data suggest that chunk analysis provides an effective grouping of words: at least two words over three appear in a non singleton chunk.

Table 1: Features figures of the three corpus

	ENEA	Sole24Ore	Legal
#words	1,149	494	1460
#sentences	56	22	80
average #verbs per sentence	2.14	3.1	2.2
average chunk length	1.53	1.44	1.54

3.2. Adopted Metrics for Evaluation

Traditional $recall$ and $precision$ have been estimated over the set of icds extracted by the system. Manually compiled test sets of icd have been extracted from sample sentences of the three different corpora, and used as

[2] Various different sources have been tested and the corresponding performances have been evaluated in (Basili *et al.*, 1998b). An adaptive architecture that adopt a lexicon of subcategorization patterns automatically acquired from the target corpora, via a learning method based on Galois lattice theory has been presented in (Basili *et al.*, 1999b)

[3] $Legal$, $ENEA$ and $Sole24Ore$ have a size of about 320,000, 350,000 and 1,300,000 words, respectively.

reference set (i.e. $correct\_icd$).

Table 2: Performance figures on the $ENEA$ and $Sole24Ore$ corpus

	ENEA		Sole24Ore	
icd	Recall	Precision	Recall	Precision
Argumental	30.2 %	96.7 %	43.6 %	97.2 %
Unambiguous	58.9 %	88.6 %	63.6 %	88 %
All	75.2 %	72.1 %	69.9 %	72.5 %
Pure SSA	49.9 %	78.8 %	32 %	69.2 %
Processing Speed				
$Chaos$	99.48 w/s		184 w/s	
Pure SSA	105.32 w/s		170 w/s	

Results obtained over the $ENEA$ and $Sole24Ore$ corpus are reported in Table 2. The argumental icd are recognized with high precision although recall is low. However, this specific figure does not distinguish between argumental and other non-argumental verbal icds. A lower recall is related to the higher frequency of non-argumental vs. argumental verb modifiers.

In order to evaluate the benefits of our stratified approach a contrastive analysis with respect to the shallow component (SSA) alone has been applied. The system precision and recall are satisfactory ($> 70\%$) over the different icds types, and, as the argumental icd catching phase is more productive, the precision of the system improves compared to SSA. Moreover, processing speeds, measured in terms of number of words per second for the overall parsing process, is not considerably distant from a pure SSA system performance developped and run on the same platform.

3.3. Evaluating Adaptivity

A specific test has been carried out to estimate differences in using hand-coded and automatically derived lexicons. Two sources have been used for this information:

- a computational lexicon, LIFUV (R.Delmonte, 1992), manually compiled for the 5,000 most frequent Italian verbs
- a lexicon of subcategorization patterns automatically acquired from the target corpora, via a learning method based on Galois lattice theory (R.Basili, 1997).

The results obtained over the $Legal$ corpus are reported in Table 3. Recall and precision over this corpus have been measured against two sources lexical information. The data set from the $Legal$ corpus have been processed in two different experiments. In the first run (see Prec1 and Rec1 in Table 3), the VSG module has been fed with subcategorization information derived from the (hand-coded) LIFUV lexicon. Prec2 and Rec2 are obtained from a run where patterns of subcategorization automatically derived from the corpus (see (R.Basili, 1997)) have been used.

Table 3: Performance on the $Legal$ corpus using two lexicons

Icd	Rec1 (LIFUV)	Prec1 (LIFUV)	Rec2 (Galois)	Prec2 (Galois)
Argumental	28.7 %	88.5 %	29.9 %	89.1 %
Unambiguous	53.6 %	85.8 %	54.4 %	86.3 %
All	67.4 %	71.6 %	68.1 %	72.1 %

The similar results show that using automatically acquired lexical information does not affect the system performances. Efficiency is still very high and does not show relevant changes over the three samples.

3.4. Improving Accuracy

Specific experiments aimed to demonstrate that lexical information improves parsing accuracy, and that automatic acquisition of a subcategorization lexicon from a corpus is viable are presented in (Basili *et al.*, 1999a) [4]. In order to give a qualitative feeling of the impact this information has on the performances, in Fig. 3 is reported a plot of recall and precision figures (with or without the acquired lexicon) for the problem of PP attachment with respect to the complexity [5] of the target sentence.

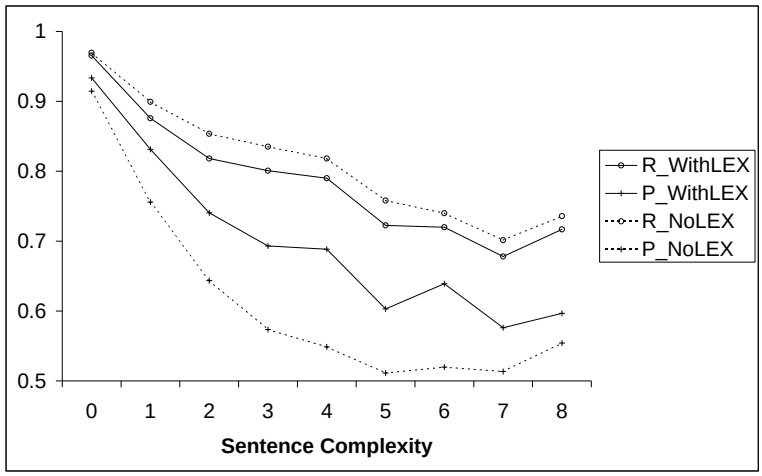

Figure 3: PP-attachment performances vs. sentence complexity

As the Fig. 3 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. $3 - 7$).

4. Conclusions

A novel parsing architecture has been here presented. CHAOS is the system implementing the corresponding stratification and lexicalization principles. Main outcome of the sentence analysis in CHAOS are: (i) a set of unambiguous word chunks; (ii) the hierarchy of clauses recognized in the source sentence; (iii) a set of inter-chunk dependencies (icd) describing major grammatical relations between the recognized constituents. Contrastive analysis demonstrate significant improvements with respect to a simpler parsing technique (SSA (Basili *et al.*, 1994)) over large portions of real corpora in Italian. Evaluation of recall and precision metrics against extensive test data show a good coverage of the target phenomena (basically event structures including PP attachments). Moreover, the extracted information is richer (e.g. clause hierarchy is also built) with respect to other shallow approaches. The processing speed shows that the system can effectively be integrated in a real complex and time contrained NLP system. This makes CHAOS a promising approach to the required linguistic processing of a speech recognition front-end. The variety of the detected information can be effectively used to build sentence and discourse models in dialogue-based systems. Its coverage and speed in fact are crucial advantages with respect to other (numerical or knowledge based) approaches.

A further positive feature of CHAOS, as a linguistic processor in a speech-driven human computer interface (HCI), is the portability to specific domains. In restricted knowledge areas (like those required in HCIs) specific lexical information (i.e. the verb subcategorization lexicon) plays a crucial role. In this paper experiments with automatically acquired lexicons of subcategorization demonstrate an objective increase of the overall performance

[4] In this case, the employed corpus and reference syntactic information was the Penn Tree bank (PTB) (Marcus *et al.*, 1993), often adopted as a golden standard for evaluating parsing accuracy (see *Parseval*-like (Black *et al.*, 1991))

[5] An approximate estimation of the complexity of a sentence may be modeled as follows:

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

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.

(see section 3.4). In other words the parsing architecture proposed in CHAOS is *open* to the integration of domain specific lexical information, thus realizing an explicit level of adaptativity. In order to fully assess the feasibility of the integration with a speech recognizer, several architectural and implementation issues are still open and experiments over attested (i.e. gold standard) test data are needed. This will be part of future work in this direction.

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 Artifical Intelligence*.

BASILI R., MARZIALI A. & PAZIENZA M. T. (1994). Modelling syntactic uncertainty in lexical acquisition from texts. *journal of Quantitative Linguistics*, **1**.

BASILI R., PAZIENZA M. & VINDIGNI M. (1999a). Adaptive parsing and lexical learning. In *VEXTAL International Conference*, Venice,ITaly.

BASILI R., PAZIENZA M. & ZANZOTTO F. (1998a). In *Proceedings of the Workshop Adapting Lexical and Corpus Resources to Sublanguages and Applications, LREC First International Conference on Language Resources and Evaluation*, Granada, Spain.

BASILI R., PAZIENZA M. & ZANZOTTO F. (1999b). Lexicalizing a shallow parser. In *Actes de la 6 Conference annuelle sur le Traitment Automatique des Languages Naturelles*, Cargese, Corse.

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. & ZANZOTTO F. M. (1998b). Efficient parsing for information extraction. In *Proc. of the ECAI98*, Brighton, UK.

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.

CARROL J. & BRISCOE T. (1996). Robust parsing - a brief overview. In *Workshop on robust parsing ESSLLI*, Prague.

J. CARROLL & T. BRISCOE, Eds. (1996). *Proceedings of the WORKSHOP ON ROBUST PARSING, held jointly with ESSLLI96*, Prague, Czech Republic.

D.GRINBERG, J.LAFFERTY D. (1996). A robust parsing algorithm for link grammar. In *4th International workshop on parsing tecnologies*, Prague.

F.M.ZANZOTTO (1997). *Una Metodologia Stratificata per la Analisi del Linguaggio Naturale: il sistema CHAOS.* PhD thesis, Engineering Fac., Univ. of Rome "Tor Vergata".

GRINBERG D., LAFFERTY J. & SLEATOR D. (1996). A robust parsing algorithm for link grammar. In *4th International workshop on parsing tecnologies*, Prague.

MARCUS M. P., SANTORINI B. & MARCINKIEWICZ M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, **19**, 313–330.

R.BASILI, M.T. PAZIENZA M. (1997). Corpus-driven unsupervised learning of verb subcategorization frames. Number 1321 in LNAI, Heidelberg, Germany: Springer-Verlag.

R.BASILI, M.T.PAZIENZA P. (1992). A shallow syntactic analyser to extract word association from corpora. *Literary and linguistic computing*, **7**, 114–124.

R.DELMONTE (1992). *Linguistic and Referential Processes in Text Analaysis by computers*. Venezia: UNIPRESS.

SRINIVAS B. (1997). *Complexity of Lexical Description and its relevance for partial parsing.* PhD thesis, University of Pennsylvania, Philadelphia.