Proceedings of VEXTAL'99 - Venezia, San Servolo, V.I.U. - 22-24 November 1999

Adaptive Parsing and Lexical Learning

R. Basili and M.T. Pazienza and M.Vindigni
University of Rome Tor Vergata,
Department of Computer Science, Systems and Production,
00133 Roma (Italy),
{basili,pazienza,vindigni}@info.uniroma2.it

Abstract

Most of NLP based systems crucially depend on parsing as one of the early stages of analysis. When computational efficiency is a key issue, the availability of a specific and extensive lexical knowledge becomes crucial to limit the intrinsic combinatorial complexity of syntactic processing.
We present here an adaptive corpus-driven approach to build a lexical resource based on a feedback interaction between a robust parser and a concept learner. We show that this methodology is able to improve the parser performances through an extensive experimental evaluation against "gold standard" data.

1 Introduction

Although the key role of lexical information in NLP tasks is widely understood, it has become clear that trying to cover the full range of *opacity* respect to different tasks (multilingual MT, IR, IE, etc.) often leads to lexicons swamped with unneeded information: this has a dramatic impact on the overall performances of text processing systems. A more successful approach is to face the problem in a bottom-up fashion, taking early into account the application domain, the target task and/or the amount of the available background knowledge.

The role of applications and sublanguages has been widely discussed (e.g. (Church and Mercer, 1993), (Basili et al., 1996)). Since specific domains are idiosyncratic with respect to common sense linguistic knowledge (so that general purpose resources are often inadequate to provide specific explanations), lexical phenomena in sublanguages are relatively regular, and corpora can be effectively used to induce background knowledge.

Looking early at the task leads to redefine notions expressed more widely in most linguistic theories under a computational perspective. In particular, NLP parsing of typical sentences in corpora always faces the problem of limiting the search within the huge hypothesis space. Grammatical constraints posed only on syntactic categories are in fact too weak to deal with the intrinsic ambiguity of natural languages. To represent the effective contributions that lexical items bring on sentence well-formedness it has been introduced ((Chomsky, 1965), (Pollard, 1994), (XTAG, 1995)) the concept of subcategorisation frame: a subcategorisation frame is a specification of the number and types of elements that lexical items need in order to completely express their meaning (the related events in case of verbs). Such a kind of knowledge could be seen as a control strategy for parsing systems, able to provide systematic principles for guiding the search for grammatical structure. His crucial role in parsing can be appreciated by looking at the Penntree bank (Santorini et al., 1993) sentence

below:

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

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*). However, two of the three PPs belong to the subcategorisation information of a verb: *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 to the parser (by the subcategorisation frames *SUBJ-contribute-OBJ-PP(to)* and *SUBJ-value-OBJ-PP(at)*) pushes for early decision, thus not only disambiguating but simply avoiding the proliferation of the ambiguities shown above.

For the purpose of parsing, a fine distinction between true arguments (i.e. constituents necessary to fully express the verb meaning) and adjuncts (i.e. constituents due only to contextual information) is almost useless, becoming irrelevant when the behavior of adjuncts is, in a target language, regular enough. They provide in fact the same "control" information. Thus, the notion of subcategorisation frame used here is simpler than its traditional definition: it includes only the involved syntactic relations (e.g. subject, direct/indirect object, the prepositions handling PP costituents, as well as clausal conjuctions). This assumption is justified by the fact that such information can be automatically acquired from a corpus and it is expressive enough to effectively guide parsing. Although a good number of methodologies for automatic acquisition have been proposed ((Brent, 1992; Briscoe and Carrol, 1997)), usually their evaluation is carried out by direct comparison with hand coded material and the impact on the parsing accuracy has never been extensively estimated. Brent (Brent, 1992), in particular, proposes a statistical method reaching recall (about 70%) and precision (about 90%) in the task of mapping verbs into subcategorization schemata (e.g. cl, THATclause, ...). Although the method learns and is tested on the PT material, no evaluation of improvements of the parsing accuracy is carried out.

We describe in this paper an architecture for integrating a learning system with a robust parser, as a way of (1) extracting relevant lexical information and (2) exploiting it for improving parsing accuracy. The result is a grammatical processor with adaptive capabilities, able to improve itself with respect to a specific target domain and the related sublanguage phenomena. Section 2 describes the adaptive architecture. Extensive experiments over "gold standard" data are discussed in Section 3.

2 An adaptive parsing framework

Adaptive parsing is intended as a syntactic process able to self-organize with respect to the target sublanguage, given an extensive sample of it (i.e. the corpus). It is realized in an incremental way, where a first scan of the corpus is accomplished in order to trigger an adaptation phase. Induction of relevant phenomena (subcategorization information, in our case) is carried out by a learning subsystem: results are directly input to a new

("informed") parsing phase. The extracted (lexical) information is expected to improve the accuracy of the overall syntactic processing.

In our framework, adaptivity is pursued by integrating a robust parser (CHAOS, (Basili et al., 1998a)) and a learning system (RGL, (Basili et al., 1997)). The parser is first fed with no lexical information, thus providing more ambiguous data ("blind" phase). The detected verbal dependencies are used as input for the learning system and the acquired subcategorization information used as a source lexical information for the second "informed" parsing stage. Fig. 1 shows the overall architecture.

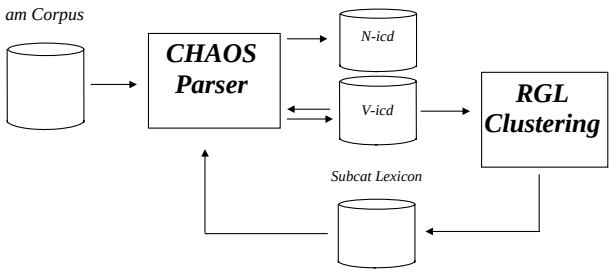

Figure 1: The Subcat Frames learning system

In Fig. 1 the source corpus (i.e. morphologically annotated sentences, *am Corpus*) is first parsed. Nominal and verbal dependencies (i.e. $N-icd$s and $V-icd$s, respectively) are detected and activate the learning phase (*RGL Clustering*). The interaction between parsing and learning produces a lexicon that is aimed to optimize parsing. The main two components of the adaptive architecture are described in the next two subsections.

2.1 Chaos: A Robust Parser for IE

CHAOS (Basili et al., 1998a) is a robust parser based on a stratified approach. The overall architecture is sketched in Fig.2. A chain of processing steps is adopted to incrementally improve the quality of the analysis and tackle the problem of combinatorial search.

The first phase is *chunking* (Abney, 1996) that aims to pack segments whose structures are independent from verb grammatical projections. Simple noun phrases (e.g.*Mr. Gaubert*, *real estate*) as well as modifiers (e.g. *to the assets*, *at $ 25 million*) are examples of these structures: unambiguous syntactic constituents (called *chunks*) are built according to a set of regular expressions, resulting in an intermediate level of interpretation between words and sentences.

The VSG (Verb shallow recognizer) and NSG (Noun shallow recognizer) processing modules take care of detecting inter-chunk verbal ($V-icd$s), and nominal ($N-icd$s) dependencies, respectively. The CBR (Clause Boundary Recognizer) module is responsible of guessing clause boundaries and determine hierarchical relationships among them. Details can be found in (Basili et al., 1998a). The interaction (looping) between VSG and

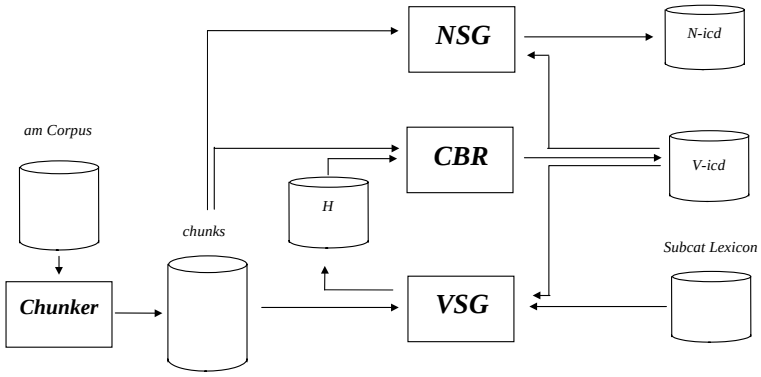

Figure 2: The Architecture of the CHAOS Parser

CBR models the interleaving process of (1) detecting verbal dependencies and (2) determining the boundaries between different sentence clauses.

In CHAOS verbal subcategorization frames act as constraints input to the clause recognizer and guide the determination of maximal and minimal clause boundaries[1]. Verbal dependencies derived on lexical basis (i.e. according to subcategorization information) have important effects on the remaining ambiguities: other potential attachment sites of subcategorised PPs are discarded, thus reducing persistent ambiguity.

In the last phase (i.e. *NSG*), the syntactic recognition is extended to a set of non argumental dependencies between chunks (e.g. post nominal prepositional phrases) by a technique already proposed in the SSA parser (Basili et al., 1992). In this phase inter-chunk dependencies *within* the determined clauses (i.e. infra-clausal dependencies) are extracted using a robust discontinuous grammar. Note that in this phase clausal boundaries effectively constraint the scope of ambiguous PPs.

For example, in the (*wsj_1692*) sentence lexical information on the verb contribute (i.e. *SUBJ-contribute-OBJ-PP(to)*) supports the detection of the V-PP *contribute-to the asset*. After this choice is made, 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: the only allowed attachments are with the verb *contribute* itself or with the noun *assets*.

The result of the parsing in CHAOS is a graph whose nodes are *chunks* and links are (potentially ambiguous) *icd*s.

[1] If no true lexicalised information is available, CHAOS makes a default guess about potential arguments, only looking for *Subject*, *Object* as well as *that* clauses. This heuristics is irrespective of the specific verb but is activated whenever default arguments are matched in the incoming sentence

2.2 RGL: A Conceptual clustering system for learning verb subcategorization frames

The RGL system (Basili et al., 1997) is a conceptual clustering tool based on conceptual lattices (Carpineto and Romano, 1993). The clustering technique derives classes as conjunctive concepts according to a boolean feature value language. Each derived concept is a couple *(S,F)* where S is a subset of instances, called *extent*, and F is the set of features of the cluster, called *intent*. The whole lattice is both complete and correct, i.e. for every subset of the source examples there is exactly one node whose intent is the conjunction of the shared features. The set of couples can be organized according to the standard set inclusion on the extents S and the corresponding partial order produces a lattice structure.

The functional architecture of RGL system is presented in fig. 3.

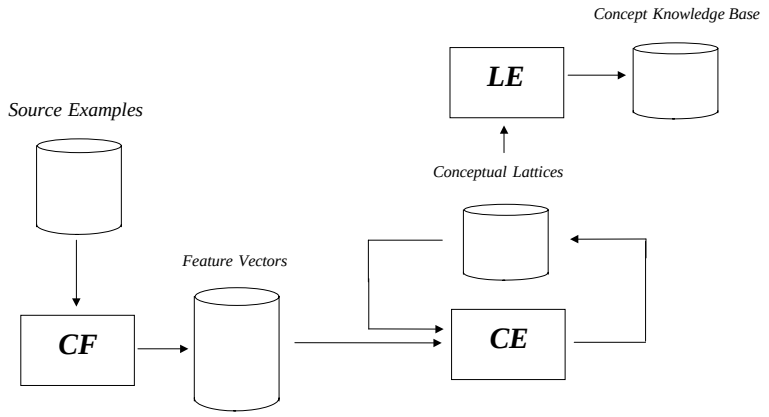

Figure 3: The RGL Conceptual Clustering system

To learn verb subcategorization frames, syntactic graphs are collected from the corpus by the robust parser and given as input to RGL. RGL is expected to clusters the verbal phrases into classes each of one suggests a subcategorisation frame. Syntactic graphs are processed by the Context Formatter (*CF*) module, that rewrites them in linear attribute-value vectors[2]: adopted features are grammatical dependencies expressing potential arguments (e.g. Subj, Objects, PPs). PP are denoted by the related prepositions.

For each verb, the Clustering Engine (*CE*) incrementally builds a lattice, whose nodes are sets of contexts expressing the same grammatical structure. Note that extents are the source clauses, so that contextual information for later processing/validation is provided. The logical decription of nodes (i.e. their intent) expresses the grammatical patterns found in similar examples (e.g. *SUBJ-verb-OBJ* or *SUBJ-verb-OBJ-PP(to)*), that suggest subcategorisation frames.

In order to select suitable frames, nodes are weighted by the Lattice Evaluator (*LE*) according to a task-oriented measure (i.e. *selectivity*, see (Basili et al., 1997) for details). Selectivity takes into account statistical

[2]Each clause is described by one vector

and linguistic relevance. The best representatives (i.e. nodes whose values are over an empirical threshold) are the only subcategorization patterns being projected in the Lexicon.

3 Procedural Evaluation of Subcategorisation Frame Acquisition

The evaluation of the learning process by RGL has been carried out by measuring the improvement of the parser accuracy. Note that this kind of procedural evaluation is also a test for accessing viability of the grammatical bootstrap. The architecture has been tested over the PennTree Bank (PT) (Santorini et al., 1993), often used as a standard "reference" set. PT syntactic trees have been rewritten in CHAOS planar graphs, projecting, when possible, the coherent information. Sentences whose constituents were not isomorphic to any CHAOS grammatical relations have been simply ignored: this produced a subset of about 40,000 test sentences. The parsing accuracy obtained by the adaptive architecture has been evaluated against the bare parser:

- *no_lex* run: is the parsing via the "blind" system (i.e. CHAOS plus heuristics for verb attachments).
- *lex* run: is the adopted feedback cycle (i.e. CHAOS + learned lexicon) shown in fig. 1.

Results of the two runs are compared against the flattend trees in the test set. After the learning phase 1,993 different verbs have been processed resulting in a set of 4,077 lexicalized patterns of subcategorization.

Recall and Precision (and the associated F-measure as a synthetic index ($\alpha = 0.5$))

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

express the accuracy of the detected dependencies.

As accuracy is expected to depend on the complexity of incoming sentences a further empirical measure has been defined as follows:

$$Sentence\ Complexity = \left\lfloor \frac{\#V_{icds} + \#N_{icds}}{\#Verbs} \right\rfloor$$

where $\#V_{icds}$ and $\#N_{icds}$ are respectively the number of verbal and nominal dependecies (i.e. Subj, Obj, VP-PP vs. NP-PP) in the test tree, while $\#Verbs$ is the number of verbs in the sentence.

Table 1 reports the number of sentences for a subset of the different complexity classes.

Column 3 and 4 are a further expression of the parsing complexity. *VPP Rec* and *VPP Prec* express the accuracy of a disambiguation rule, based on a "blind" heuristics: the *closest* potential head for a PP has been selected by a *Right Attachment* heuristics[3]. The rule has been applied to all the ambiguous verbal dependencies and

[3]This criterion has been often used as a lower bound of PP disambiguation methods (e.g. (Hindle and Rooths, 1993))

Table 1: Number of sentences vs. Complexity Index in the Test set

Complexity Index	*#sentences*	VPP Rec	VPP Prec
0	507	0.36	0.81
1	11,239	0.37	0.79
2	15,893	0.35	0.77
3	7,674	0.38	0.84
4	3,850	0.38	0.89
5	1,585	0.45	0.85
6	1,051	0.21	0.72
7	365	0.60	1.00
8	244	0.42	1.00

precision/recall (i.e. VPP Rec and VPP Prec) have then been measured over the entire test set. As expected, the VPP attachment task is very difficult for more complex sentences: low recall suggests that attaching the PP to the closest verb result in only detecting about 33% to 50% of the correct verbal dependencies. These "average" complexity shows that the test set is a good candidate for evaluating parser performance.

Table 2 reports detection of verbal PP dependencies. The attachment of PP to verbs is a difficult task as the low recall of unambiguous V-PP icds suggests: when no lexicon is employed the parser collect only 58% of the correct ones. The use of the induced lexical information results in a relevant increase of unambiguous dependencies (70% recall); the lower precision is mainly due to errors related to noun modifiers wrongly detected as arguments, because of the conflicts between heading prepositions (e.g. *transferring items in iron*). It is worth noticing that data in row 2 (*no_lex* lex and *all icd*s) express the maximal recall reachable as all the colliding dependencies are retained. As expected, the precision is weak (0.58).

Using the lexicon has a strong impact on the coverage of the method: relying on subcategorization information allows to fix the set of verbal dependencies in the very early phases of the analysis (VSG and CBR). These *choices* represent a first level of disambiguation and enlarge the set of unambigously detected *icd*s (0.58 to 0.70) with a reasonable precision (0.86). Note that this measure is related to the capability of PP-attachment disambiguation but do not cover the entire variety of lexical effects on parsing

Lexicon	***icd*s**	***icd*s Type**	R	P	$F(\alpha = 0.5)$
no_lex	*unambiguous*	V-PP	0.58	0.94	0.72
no_lex	*all*	V-PP	0.82	0.58	0.68
lex	*unambiguous*	V-PP	0.70	0.86	0.77

Table 2: verb arguments

Other effects produced by the lexical information available after learning reflect on the detection of noun PP modifiers. Results are shown in Table 3.

Run	***icd*s**	***icd*s Type**	R	P	$F(\alpha = 0.5)$
no_lex	*all*	NP-PP	0.85	0.65	0.73
lex	*all*	NP-PP	0.82	0.75	0.78

Table 3: noun phrases-prepositional phrases attachment

Without lexical information the system is able to detect 85% of correct noun PP modifiers: precision tends to be low (i.e. P=0.65) for the inherent overgeneration (i.e. different attachment alternatives are retained). When the lexicon is early used to fix verbal dependencies, the overall scope of PPs is reduced. Corrispondingly, the persistently ambiguous NPP dependencies (i.e. those remaining ambiguous after verb analysis and clause boundary recognition) are detected by the system with a slight loss of recall and a significant improvement of precision (0.65 to 0.75). The above phenomena suggest that lexicon plays not only the role of determining more extensively and with an higher precision verbal dependencies, but, due to the specific nature of CHAOS, it also constraints other forms of ambiguity.

The plot in Figure 4 shows recall and precision with respect to sentence complexity.

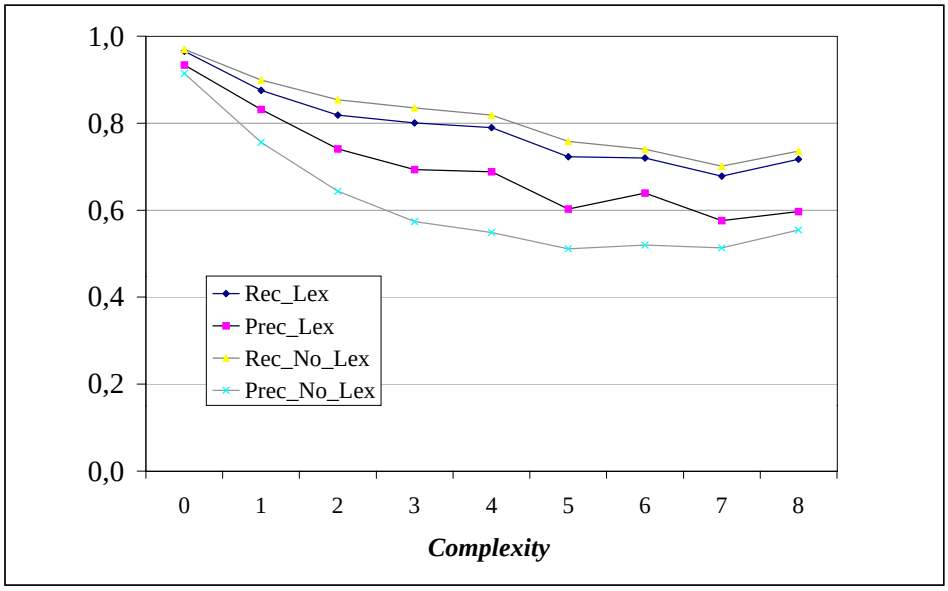

Figure 4: Performance of the NPP attachment vs. sentence Complexity

For highly complex sentences (complexity > 4) the precision of the non lexicalized run is weak.

As previously mentioned, the recall of the "blind" CHAOS system is an upper bound for the system as few decision are taken and all the ambiguities are retained. The behaviour of this "maximal" recall is well approximated by the lexicalized approach. Although decisions are taken very early in the parsing process, by fixing verb arguments and then looking for remaining nominal dependencies, it seems the same correct NPPs are still captured. Only a slight loss is observed (about 2%).

A significant improvement is instead obtained for precision, as expected. Note that for the larger complexity classes the (relative) improvement is higher (e.g. about 25% for complexity 4).

4 Discussion

The described farmework can be assumed as a model for adaptive parsing in specific domains. Effectiveness has been shown on a large scale. Procedural evaluation has been carried out by measuring the impact of the learning technique on parsing accuracy. This has the obvious advantage of producing objective results, also related to the operational scenario typical of an application. Other works in this area rely on human coded lexicons (e.g. (Brent, 1992)) and do not provide such evidencies.

In the adaptive architecture the lexicon plays the role of control strategy for a robust parser. The specific nature of the parser allows both the derivation of the early evidence for the conceptual clustering algorithm (i.e. instances of verb behaviour in the source corpus) as well as a direct reuse of the derived information during a renewed (i.e. "informed") parsing process. The resulting parsing architecture, adaptive in a sublanguage, is thus viable and effective. The measured effects of lexicalization on parsing accuracy are a significant increment of the overall precison (relative improvement, 15%) that is even stronger (about 20%) for complex sentences. Evaluation shows that the lexicon has two different effects on parsing:

- it supports a better verb argument detection
- it reduces ambiguities of the remaining sentence fragments, by imposing clause boundaries (and planarity constraints) on nominal PP attachments.

The learning method, in the proposed version, can be applied in a fully automatic way. The only lexical rules used since the beginning are the default heuristics of the parser. No existing lexical resource is assumed. Moreover, empirical estimation is applied only over lattice structures, so that no manually annotated data or external resources are required. It is worth noticing that other works (e.g. (Briscoe and Carrol, 1997)) rely on existing lexicons (e.g. ANLT) for accurate statistical estimation. They cannot be widely applied, especially for languages where such extensive resources are missing.

Note that the current performance represents a lower bound. As a matter of fact, the acquired lexical resource can be used as a basis for human analysis. In particular, the lattice structure built by RGL is suitable for validation and enrichment[4] through browsing the lattice GUI.

Moreover, RGL, as an extractor of verb patterns from corpora, can be suitably adopted as a tuning methodology in Information Extraction (IE). Phenomena specific to the domain are coded in the verb subcategorization lexicon and are ready to be employed as a basic *event recognition* component in IE. A specific architecture for tuning a lexicon in IE is discussed in (Basili et al., 1998b).

Finally, the derived lexical information constitutes a domain-specific lexical knowledge base. Integration with general purpose existing resources (like Wordnet) is possible. The information acquired by RGL can thus be

[4]For example, the compilation of semantic information associated with the arguments, i.e. selectional constraints

exploited to extend an existing lexicon: integration via sense mapping is here necessary and is currently under investigation. Further experimentation in this direction is still needed.

References

S. Abney. 1996. Part-of-speech tagging and partial parsing. In K.Church, S.Young, and G.Bloothooft, editors, *Corpus-based methods in language and speech*. Kluwer academic publishers, Dordrecht.

R. Basili, M.T.Pazienza, P.Velardi. 1992. A shallow syntactic analyser to extract word association from corpora. *Literary and linguistic computing*, 7:114–124.

R. Basili, M.T.Pazienza, P.Velardi. 1996. An Empirical Symbolic Approach to Natural Language Processing. *Artificial Intelligence*, 85.

R. Basili, M.T. Pazienza and M. Vindigni. 1997. Corpus-driven unsupervised learning of verb subcategorization frames. In *Proc. of Conference of the Italian Association for Artificial Intelligence, AI*IA 97, Rome*.

R. Basili, M.T. Pazienza, and Fabio Massimo Zanzotto. 1998a. Efficient parsing for information extraction. In *Proc. of the ECAI98*, Brighton, UK.

R. Basili, M.T. Pazienza, and Fabio Massimo Zanzotto. 1998b. Evaluating a robust parser for italian language. In *Proc. of the THE EVALUATION OF PARSING SYSTEMS Workshop, held jointly with 1st LREC*, Granada, Spain.

M.R. Brent. 1992. *Automatic Acquisition of Subcategorisation Frames from Unrestricted English*. PhD thesis.

T. Briscoe, J. Carrol 1997. Automatic Extraction of Subcategorization from Corpora. In *Proceedings of the 5rd ANLP 97, Washington*.

C. Carpineto, Romano G. 1993. *GALOIS: An order-theoretic approach to conceptual clustering*. Fondazione Ugo Bordoni.

N. Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge.

K. Church, R. Mercer. 1993. Introdution to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics*, 19(1).

D. Hindle and M. Rooths. 1993. Structural ambiguity and lexical relation. *Computational Linguistics*, 19(1).

C.Pollard, I.A.Sag. 1994. *Head-driven Phrase Structured Grammar*. Chicago CSLI, Stanford.

B. Santorini, Marcus M. P., and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313–330.

XTAG Research Group. 1995. A lexicalized tree adjoining grammar for english. Technical report, University of Pennsylvania.