

Luca Dini
Grammatiche Computazionali e Ingegneria del Linguaggio

(Atti del convegno sul trattamento automatico del linguaggio)

1 Introduzione

Nel presente articolo ci occuperemo di grammatiche computazionali, ovvero di moduli in grado di fornire analisi strutturali di enunciati prodotti in forma orale o scritta. Tali moduli costituiscono, in molti casi, il cuore di sistemi complessi di trattamento automatico del linguaggio. Nel seguito faremo uso della definizione di *grammatica computazionale* come

qualunque sistema formalizzato che rappresenti esplicitamente la struttura sintattica dell'italiano.

L'enfasi attribuita ad alcuni dei termini della definizione si spiega sulla base delle seguenti considerazioni:

- Una grammatica computazionale deve essere un *sistema*, nel senso che tutti i componenti e le generalizzazioni che la compongono devono essere in grado di interagire propriamente tra di loro, ovvero essere in grado di produrre tutte e solo le strutture previste da una data teoria o richieste da una data applicazione. Da un punto di vista operativo ciò si traduce con l'attivazione dei concetti di *integrabilità* e *modularità* dell'implementazione.
- Una grammatica computazionale deve essere formalizzata, ovvero deve esistere una procedura in grado di accettare in ingresso le descrizioni in essa contenute e un enunciato in linguaggio naturale, e restituire in uscita la struttura assegnata dalla grammatica a tale enunciato. Da un punto di vista operativo ciò si traduce con l'attivazione del concetto di *correttezza*.
- Le descrizioni della grammatica devono essere *esplicite* ovvero devono rappresentare in maniera chiara ed intuitiva le proprietà strutturali dell'oggetto che descrivono. Non è pensabile che una grammatica possa essere rappresentata semplicemente da funzioni scritte in un qualunque linguaggio di programmazione, ancorché in grado di fornire rappresentazioni strutturali corrette. Da un punto di vista operativo ciò si traduce con l'attivazione del concetto di *dichiaratività*.

I tipi di grammatiche su cui ci soffermeremo in questo lavoro hanno come obiettivo principale l'assegnazione di strutture *sintattiche*. Il motivo per cui l'analisi semantica non del linguaggio non verrà presa in esame, se non marginalmente, è dovuto al fatto che in molti casi l'assegnazione di una struttura semantica viene concepita come una procedura operante sull'*output* della grammatica, e non risulta pertanto interessante dal nostro punto di vista.¹ Inoltre, in molti sistemi, il tipo di rappresentazione semantica fornita è troppo fortemente dipendente dal tipo di applicazione per cui l'analisi è stata concepita per essere trattato in questo contesto.

2 Tipologia delle grammatiche in relazione alla profondità di analisi

Sino al principio degli anni 90, sotto l'influenza degli studi di matrice chomskiana relativi alle grammatiche di competenza², le grammatiche computazionali erano concepite essenzialmente come sistemi in grado di produrre rappresentazioni strutturate di *interi enunciati*. Sotto la spinta delle competizioni tra sistemi di estrazione dell'informazione promossi da entità governative americane (soprattutto la Message Understanding Conference), e sulla base di certi risultati teorici nell'ambito della psicolinguistica e della scienza cognitiva (si veda soprattutto [3] e [4]), negli anni '90 si sono affiancate alle grammatiche tradizionali grammatiche in grado di produrre rappresentazioni poco strutturate di *frammenti* di enunciati. Nel seguito, chiameremo il tipo più tradizionale di grammatica *grammatica di competenza* e assegneremo alle grammatiche in grado di analizzare frammenti di enunciati l'etichetta *grammatiche superficiali*.

¹ Va detto che in taluni casi l'assegnazione della struttura semantica procede di pari passo con l'assegnazione della struttura sintattica (si vedano ad esempio le grammatiche costruite secondo l'impostazione teorica HPSG ([1], [2]), per cui i meccanismi di composizione semantica fanno parte integrante della conoscenza grammaticale.

²Una grammatica di competenza (*competence grammar*) è una forma di organizzazione grammaticale in grado di modellare la competenza del parlante relativa ad un certa lingua. Tali tipi di grammatiche cercano dunque di fornire rappresentazioni adeguate alla teoria di riferimento e sono normalmente improntate alla massima generalità ed indipendenza dal dominio.

2.1 Grammatiche di competenza

Le grammatiche di competenza mirano a produrre rappresentazioni degli enunciati che rispecchino in qualche maniera la competenza del parlante rispetto ad una data lingua. L'individuazione delle caratteristiche che modellano tale competenza è spesso affidata ad una teoria linguistica di riferimento. Inoltre, Dal momento che il parlante è normalmente in grado di comprendere *per intero* un enunciato che gli venga sottoposto, tali grammatiche cercano di raggiungere una rappresentazione strutturale che copra l'intera frase di input, raggruppando tutto il materiale linguistico sotto un'unica etichetta sintattica. Qualora tale operazione fallisca, nessun *output* viene prodotto, a meno che non siano attivi meccanismi di *robustezza* che sono comunque sempre implementati tramite procedure esterne alla grammatica stessa.

Come accennato, le grammatiche di competenza sono *intrinsecamente* dipendenti da una teoria linguistica. Al momento le teorie che maggiormente hanno fornito ispirazione per la creazione di grosse grammatiche di competenza (e che in certi casi hanno tratto spunto da certe implementazioni computazionali) sono FUG, ([5]), GPSG, ([6]), UCG, ([7]), Dependency Grammar, ma, soprattutto HPSG ([1],[2]) e LFG ([8]).

Altra caratteristica tipica delle grammatiche di competenza è rappresentata dalla netta distinzione tra conoscenza grammaticale e conoscenza implementativa. L'idea di base è che la lingua possa essere descritta da un insieme astratto di generalizzazioni del tutto indipendenti dal linguaggio di programmazione che le interpreterà ed eventualmente applicherà all'enunciato da analizzare. Di conseguenza sono stati definiti numerosi *frameworks* dedicati proprio a fornire al codificatore di grammatiche il linguaggio ideale per la codifica delle informazioni grammaticali pertinenti. Tra questi, citiamo, non esaustivamente, LFG ([8]), PATR ([9]), ALE ([10]), STUFF ([11]), ET-6/1 ([12]), CLE ([13]), TDL ([14]) TFS ([15]).

2.2 Grammatiche superficiali

Le grammatiche superficiali sono nate essenzialmente come risposta ad applicativi volti ad estrarre informazione standardizzata (spesso sotto forma di *templates*) da grosse moli di testi. Di fatto, sotto l'etichetta di grammatiche superficiali si tende ad accorpare almeno due distinte classi di moduli di analisi piatta:

1. Sistemi di chunking ([16], [17], [18], [19], [20], [21]), il cui scopo principale è identificare i costituenti non ricorsivi di una frase, raggruppandoli sotto alcune etichette standard.
2. Sistemi di analisi semantica piatta, in cui il raggruppamento delle categorie lessicali è guidato dalle proprietà semantiche di certi costituenti "forti" (teste verbali e nomi predicativi) ([22], [23], [24]). Si vedano anche le descrizioni dei sistemi presentati a MUC4, MUC5 e MUC6).

La maggior parte dei sistemi esistenti si basa sulla combinazione di queste due tecniche: il chunking produce un testo marcato in cui vengono riconosciuti i costituenti principali; il modulo di analisi semantica piatta sfrutta questo premarcamento al fine di identificare le relazioni semantiche esistenti fra i vari costituenti.

I sistemi basati su grammatiche superficiali mostrano forti caratteristiche di *robustezza*, dal momento che riescono sempre a produrre un output, a prescindere dalla copertura della grammatica, dalla completezza del lessico, e dalla buona formazione dell'input. Inoltre, la semplicità della tecnologia adottata consente di produrre sistemi assai efficienti, in grado di analizzare grossi moli di testi in tempi relativamente ridotti.³

2.3 Alcuni punti problematici

Entrambi i tipi di grammatica sopra descritti presentano problemi in parte tuttora irrisolti. Per quanto riguarda le grammatiche di competenza, possiamo isolare i seguenti punti:

- Non sono in grado di produrre risultati in tempi "accettabili". Come afferma Pereira in [25], sezione 3.6.4, "...however, even polynomial algorithms may not be sufficiently fast for practical applications...."
- Non sono robuste, ovvero è sufficiente una piccola lacuna nel lessico o una lieve malformatezza nella frase da analizzare per inibire la produzione di un qualsiasi risultato

³ Ad esempio IUTA (Italian Unrestricted Text Analyzer, [31]) è in grado di analizzare 68,9 parole al secondo su un Pentium 133. SMES ([27]) analizza testi della lunghezza approssimativa di una pagina in un tempo variabile da 1 a 10 secondi su una Sun UltraSparc.

- La copertura non è sufficientemente ampia e non tiene in debito conto fenomeni irrilevanti dal punto di vista della teoria linguistica ma assolutamente centrali per l'analisi di testi reali.⁴

- Hanno tempi di sviluppo piuttosto lunghi. In [26] si stima che il tempo di produzione per una grammatica ad unificazione dotata di media copertura si aggiri sui quattro anni uomo.

Da tali problemi sono normalmente immuni le grammatiche piatte, che, per contro, presentano limitazione relative all'accuratezza ed alla generalità. Infatti□:

- Le rappresentazioni da esse fornite sono incomplete, dal momento che codificano esclusivamente la parte di testo rilevante per l'applicazione per cui sono state concepite.
- il tasso di successo resta spesso ai limiti dell'accettabilità, dal momento che i sistemi migliori sono in grado di produrre risultati con un'accuratezza intorno al 60% ([27]).
- Sono scarsamente estendibili attraverso domini diversi e la loro utilizzabilità resta spesso limitata a specifiche applicazioni ([28]).
- Non essendo stato definito nessun tipo di linguaggio utente per la codifica di grammatiche piatte, il riutilizzo ed il mantenimento può talvolta essere molto costoso.

Come vedremo nella prossima sezione, la comunità scientifica internazionale sta mettendo a punto tecniche che costituiscono una valida risposta alle limitazioni delle attuali grammatiche, siano esse piatte o di competenza.

3 *L'ingegneria del linguaggio nel campo dello sviluppo grammaticale*

A partire dalla fine degli anni ottanta si è andato delineando un nuovo settore della linguistica computazionale detto *ingegneria del linguaggio* (□'Language Engineering'). Questa la definizione di ingegneria del linguaggio data in MLAP (1993)□:

Language Engineering is concerned with computerised language processing covering several dimensions, such as theoretical and descriptive linguistics, lexicology, computer science and engineering. Its principal aim is to bridge the gap between traditional computational linguistics research and the implementation of practical applications with potential for real world use. Thus Language Engineering is a means for permitting the transition from research and laboratory prototypes to marketable products.

Se lo scopo dell'ingegneria del linguaggio è la migrazione da prototipi di ricerca in prodotti, gli strumenti di cui si avvale sono ricalcati su quelli in uso nel campo dell'ingegneria del software, intesa come "...the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software...". L'utilizzo di procedure desunte dall'ingegneria della qualità ha dato in questi ultimi anni un notevole impulso allo sviluppo di risorse grammaticali. In particolare ha ormai senso parlare di fasi di design, sviluppo, verifica e validazione anche nel campo dello sviluppo di risorse grammaticali. A questo riguardo si registrano in particolare i seguenti progressi□:

- Sono state definite piattaforme di sviluppo (*ALEP* ([28]) e *Xerox LFG Grammar-Writer's Workbench*, per le grammatiche di competenza, *GATE* ([29]) per le grammatiche piatte) in grado di facilitare il lavoro del codificatore di grammatiche fornendogli importanti strumenti di design, sviluppo e *debugging*.
- Sono stati definiti metodi di valutazione quantitativa relativi alla prestazione di grammatiche (si veda il progetto *TSNLP, Test Suites for Natural Language Processing (LRE)*, per le grammatiche di competenza, così come le procedure di valutazione create nel corso della *Message Understanding Conference*).
- Si stanno gradualmente definendo metodologie di documentazione standardizzate, indispensabili per una riduzione dei tempi di sviluppo delle risorse grammaticali e per una più proficua cooperazione tra centri diversi operanti nell'ambito di progetti centrati sulla produzione di conoscenza grammaticale (si veda il massiccio lavoro di documentazione svolto nell'ambito del progetto *LS-GRAM*, ed il modello di documentazione grammaticale di *HyperGram* ([30]) desunto da modelli di documentazione software adottati in *CASE* e *hyperliterate programming*).

A questi progressi ormai consolidati si affiancano settori di ricerca molto promettenti che potrebbero essere in grado, nel giro di pochi anni, di ovviare alle limitazioni descritte in□2.3□.

⁴ Scrive ancora Pereira□in[25], sezione 3.6.4, "...linguistic theories are typically developed to explain puzzling aspects of linguistic competence ... However, actual languages involve a wide range of other phenomena and constructions ... which may not be germane to the issues addressed by a particular linguistic theory or which may offer unresolved challenges to the theory ... Even then, coverage gaps are relatively frequent and difficult to fill, as they involve laborious design of new grammar rules and representations."

In particolare per le grammatiche di competenza si sta tentando di porre rimedio alle deficienze relative a efficienza e robustezza elaborando le seguenti strategie□:

- **Efficienza**□: Sono allo studio tecniche di compilazione delle grammatiche di competenza in più efficienti formati di rappresentazione sintattica. Sembra a questo proposito piuttosto interessante la compilazione da HPSG e LFG in Tree adjoining Grammar (TAG). Inoltre metodi di *Explanation Based Learning* si sono dimostrati assai proficui al fine di guidare le grammatiche di competenza verso un efficiente riconoscimento delle strutture che appaiono più frequentemente in testi reali.
- **Robustezza**□: L'integrazione di grammatiche di competenza con moduli di analisi piatta (sistemi ibridi) contribuirà ad aumentare la robustezza delle grammatiche stesse, dal momento che sarà possibile segmentare la frase in unità più piccole in grado di essere analizzate in maniera indipendente. Inoltre esistono tecniche per cui è possibile trarre vantaggio dal fatto che anche un'analisi fallita produce una un insieme di analisi parziali da cui è comunque possibile trarre utile informazione.

Analoghi progressi si verificano nel campo delle grammatiche superficiali, dove, come abbiamo visto, i punti di debolezza cruciali sono rappresentati dalla loro scarsa riutilizzabilità e generalità.

- **Riutilizzabilità**□: I più recenti sistemi di estrazione dell'informazione sono basati su un linguaggio utente piuttosto trasparente, che permette il facile riutilizzo quantomeno della struttura interpretativa delle regole grammaticali ([31], [27])
- **Generalità**□: A partire dalla terza *Message Understanding Conference*, sono stati fatti grossi sforzi per la creazione di sistemi portabili attraverso domini differenti. Nella maggior parte dei casi tale esigenza ha dato un forte impulso allo sviluppo di tecniche per la creazione automatica di regole di *pattern matching* sintattico, entrate lessicali, e dizionari concettuali. Tali tecnologie sono al momento in grado di fornire risultati piuttosto accurati, e richiedono un minimo impegno di forza lavoro.

4 Grammatiche e Applicazioni

4.1 Grammatiche superficiali

Le tecnologie basate su grammatiche superficiali stanno passando in questo periodo dallo stato di prototipi di ricerca a quello di applicativi. Grazie allo sviluppo di grossi database testuali e di reti di portata mondiale (prima tra tutte Internet, la cui struttura portante resta nonostante il vasto apporto multimediale, di tipo testuale) la necessità di estrarre e selezionare informazione relativa a particolari domini si è andata facendo assolutamente centrale. Tra le applicazioni di maggiore rilevanza si segnalano sistemi per estrarre informazioni contenute in rapporti sulla situazione militare in varie aree del globo (MUC), per classificare i profitti ed i movimenti di denaro delle aziende sulla base di articoli di giornali economici (MUC), per identificare le richieste dell'utente in sistemi di dialogo uomo-macchina (SMES, [27]), per archiviare informazioni relative ad annunci economici (IUTA, [31]). In tutti questi casi l'applicazione prende in esame testi in linguaggio naturale e tenta di produrre *templates* che vengono poi archiviati in un database per una più immediata consultazione. Altre applicazioni che fanno uso di sistemi di analisi piatta comprendono sistemi di classificazione ([32]) e sistemi in grado di produrre riassunti automatici. Nel primo caso si cerca di classificare un certo insieme di documenti, ad esempio la posta in arrivo in una grande azienda, sulla base dei *templates* che da esso possono essere estratti. Nel secondo caso si utilizzano i *templates*□estratti come indicatori dei punti di maggiore rilevanza del testo da riassumere, e si generano enunciati in linguaggio naturale sulla base dei *templates* stessi.

4.2 Grammatiche di competenza

Sulla base dell'attuale stato di sviluppo della tecnologia sembra evidente che le grammatiche di competenza *non* saranno parte integrante della prossima generazione di sistemi di trattamento del linguaggio naturale. Esse sono infatti ancora ad uno stadio prototipale e, come abbiamo visto, sono ancora affette da problemi che ne impediscono l'integrazione in applicativi industriali. Tuttavia è evidente che esiste una classe di applicazioni ([33]) per cui le grammatiche piatte non sono in grado di produrre risultati sufficientemente accurati. E' ad esempio il caso dei sistemi di traduzione automatica, in cui, per la complessità del compito, viene normalmente richiesta un'analisi assai dettagliata del testo in ingresso (si veda il progetto tedesco *Vermobil*). Una situazione analoga si riscontra nel campo della correzione sintattica e dell'insegnamento

automatico delle lingue, dove è indispensabile comprendere con la massima precisione le strutture linguistiche in esame al fine di identificare gli errori dei parlanti e proporre un'eventuale correzione. Infine le grammatiche di competenza verranno a svolgere una funzione decisiva in tutte quelle applicazioni in cui si richiede analisi di testi per cui non è possibile restringere a sufficienza il dominio di applicazione (ad esempio sistemi di interrogazione di database contenenti informazione eterogenee)□: in questi casi i sistemi classici di estrazione di informazione, data la loro natura legata alle aspettative in un certo dominio, difficilmente potranno fornire risultati paragonabili a quelli che le grammatiche di competenza saranno in grado di fornire.

5 Conclusioni

Le poche osservazioni fin qui fatte da un lato evidenziano l'urgente necessità di risorse grammaticali che facilitino un sviluppo equilibrato e consapevole della società dell'informazione, dall'altro sottolineano la facilità con cui tali tecnologie possono essere prodotte alla luce degli ultimi sviluppati raggiunti nel campo dell'ingegneria del linguaggio. È un fatto che alcuni tra i paesi industriali più all'avanguardia (USA, Giappone, Germania) stiano investendo enormi quantità di denaro per lo sviluppo di risorse grammaticali. Tali investimenti rispondono ad una chiara esigenza di salvaguardia dell'identità delle rispettive lingue nazionali, a cui si accompagna una inevitabile spinta verso la salvaguardia dell'identità culturale. Si consideri infatti che, nell'era della comunicazione elettronica, una lingua mal supportata dal punto di vista degli strumenti di analisi sarà destinata a diventare lingua di serie B, fatalmente votata a scomparire dallo scenario della società dell'informazione. Ad esempio, qualora i motori di ricerca non siano in grado di supportare interrogazioni *in e verso* tale lingua, o qualora non esistano strumenti in grado di trattare l'informazione in essa codificata, è estremamente probabile che gli utenti utilizzino lingue maggiormente supportate o rinuncino del tutto a diventare fruitori attivi di certe tecnologie⁵.

Lo sviluppo di risorse grammaticali in Italia vive al momento in uno stadio critico. Esiste, è vero, un certo numero di grammatiche, siano esse superficiali o di competenza, che sono state sviluppate da diversi centri nel corso degli anni passati. D'altra parte, è anche vero che tali grammatiche coprono per la maggior parte un insieme comune di fenomeni, risultando difficilmente integrabili. Inoltre soffrono ancora, in misura diversa, dei problemi descritti in 2.3, risultando o di difficile integrabilità in applicazioni reali o di problematica portabilità attraverso domini diversi. Per risolvere tali problemi è necessaria un'azione integrata che da un lato metta insieme tali risolve al fine di monitorare la situazione esistente, dall'altro dia inizio ad un programma integrato in cui tecnologie di ingegneria del linguaggio (vedi sezione 3) possano essere fruttuosamente utilizzate e in cui i vari centri operanti nell'ambito delle risorse grammaticali possano condividere le esperienze accumulate nel corso degli anni al fine di produrre tecnologie effettivamente integrabili in applicativi di tipo industriale.

6 Bibliografia

- [1] Pollard, C. e I. Sag: "Information-Based Syntax and Semantics. Volume I: Fundamentals", CSLI Publications, 1987.
- [2] Pollard, C. e I. Sag□ "Head Driven Phrase Structure Grammar", Chicago Press e CSLI Publications. 1994.
- [3] Abney, S.□: "Parsing by Chunks", In "Principle-Based Parsing□: Computation and Psycholinguistics", Kluwer, 1991.
- [4] Abney, S., "Partial Parsing via Finite-State Cascades", Work in progress, <http://sfs.nphil.uni-tuebingen.de/~abney/>.
- [5] Kay, M.,□: "Parsing in Functional Unification Grammar", in "Natural Language Parsing", D. Dowty, L. Karttunen and A. Zwicky (Eds), Cambridge University Press, 1985.
- [6] Gazdar, G., E. Klein, G.K. Pullum e I. Sag□ "Generalized Phrase Structure Grammar", Basil Blackwell, 1985.
- [7] Zeevat, H., E. Klein e J. Calder□ "Unification Categorical Grammar", in "Categorical Grammar. Unification Grammar and Parsing", Klein e Morrill (Eds), Haddock, 1986.
- [8] Bresnan J.□: "The Mental Representation of Grammatical Relations", Cambridge, Mass., MIT Press, 1982.

⁵ Non è un caso che il tasso di utenti Web in Europa sia straordinariamente alto, oltre che in Gran Bretagna, anche in paesi come Svezia, Norvegia, Islanda, Olanda e Germania, dove cioè lo studio dell'inglese riveste una importanza fondamentale nel ciclo educativo medio.

- [9] *Shieber, S. H., Uszkoreit, F.C Pereira, J.J. Robinson e M. Tyson* □ “The Formalism and Implementation of PATR-II”, in “Research on Interactive Acquisition and Use of Knowledge”, SRI International, 1983
- [10] *Carpenter, B.*: “ALE- The Attribute Logic Engine”, Technical Report, Carnegie Mellow University, 1992.
- [11]. *Bouma .G, Koenig, E., and Uszkoreit, H.*: “A flexible graph-unification formalism and its application to natural-language processing”. IBM Journal of Research and Development, 1988.
- [12] *Alshawi, H., Arnold, D. J., Backofen, R., Carter, D. M., Lindop J., Netter, K., Pulman S. G., e Tsuji, J.* □ “Rule formalism and virtual machine design study”. Technical Report ET6/1, CEC, 1991.
- [13] *Alshawi H.*, “The Core Language Engine”. MIT Press, Cambridge, Massachusetts, 1992.
- [14] *Krieger, H., U., and Schaefer, U.* □: “TDL---a type description language of HPSG”. Technical report, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Saarbrücken, Germany, 1994
- [15] *Emele M. e R. Zajac* □ “Typed Unification Grammars”, In “Proceedings of the 13th International Conference on Computational Linguistics”, Helsinki, 1990.
- [16] *Appelt, D., Hobbs, J., Bear, J., Israel, D., Tyson, M.*: “FASTUS: A Finite State Processor for Information Extraction from Real World Text”, in: “Proceedings of IJCAI” 1993.
- [17] *Lehnert, W., Cardie C., Fisher, D., Riloff, E., Williams, R.* □ “University of Massachussets: Description of the CIRCUS System as Used for MUC-3”. Proceedings of the 3rd MUC, San Diego, CA, 1991, pp. 223-234.
- [18] *Koskenniemi K.* □: “Finite State Parsing and disambiguation”, In “COLING-90”, 1990.
- [19] *Kameyama, M., Arima, I.* □ “A minimalist approach to information extraction from spoken dialogues”. In “Proc. of ISSD-93”, Tokyo, Japan, 1993.
- [20] *Gross, M.* □: “The Use of Finite Automata in the Lexical Representation of Natural Language”, in: M. Gross, D. Perrin (eds.): “Electronic Dictionaries and Automata in Computational Linguistics”, Springer-Verlag Berlin 1989.
- [21] *Roche, E.* □: “Text Disambiguation by Finite State Automata, an Algorithm and Experiments on Corpora”, In: “Proceedings of Coling-92” 1992.
- [22] *Cardie, C.* □: “Domain-Specific Knowledge Acquisition for Conceptual Sentence Analysis, PhD Dissertation, Amherst, 1994
- [23] *Goodman, C.* □ “A Casa-Based, Inductive Architecture for Natural Language Processing”, In “Working Notes of the Machine Learning of Natural Language and Ontology”, Stanford University, 1991.
- [24] *Hobbs, J.* □: “The Generic Information Extraction System”, MUC 4, Virginia, 1992.
- [25] *Varile G., Zampolli, A.* □: “Survey of the State of the Art in Human Language Technology”, 1995
- [26] *Erbach G., Uszkoreit H.* □: “Grammar Engineering □: Problems and prospects - report of the Saarbrücken workshop on grammar engineering”. Technical report, University of Saarlandes, Saarbrücken 1990.
- [27] *Neumann, G., Backofen, R., Baur, J., Becker, M., Braun, C.* □: “An Information Extraction Core System for Real World German Text Processing”, in “Proceedings of ANLP 1997”.
- [28] *Cunningham, H., Freeman, M., Black, W.J.* □: “Software reuse, object-oriented frameworks and natural language processing”, in “Proceedings of the conference on New Methods in Natural Language Processing”, Manchester, 1994.
- [28] *Simpkins, N.* □: “ALEP-0 Version 2.2-Prototype Virtual Machine, User Guide”. CEC DG-XIII, Luxemburg, 1992.
- [29] *Cunningham, H., Wilks, Y., Gaizauskas, R.J.* □ “New Methods, Current Trends and Software Infrastructure for NLP”.
- [30] *Mazzini G., Dini, L.* □ “Hypergram” □: Proceedings of the 3rd AUG Workshop, IAI, Saarbrücken, 1997.
- [31] *Bolioli A., Dini L., Goy A., Mazzini G.* □: “IUTA □: Hybrid Technologies for a Portable Information Extraction System for Italian” □: CELI Working Papers □:1-97, 1997.
- [32] *Riloff, E., M.* □ “Information Extraction as a basis for portable text Classification Systems”, PhD Dissertation, 1994.
- [33] *Wilks Y.* □: “Developments in MT Research in the US”, ASLIB Proceedings, Vol. 46#4, April 1994.