

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>



The European Journal for the Informatics Professional  
<http://www.upgrade-cepis.org>

Vol. VIII, issue No. 1, February 2007

#### Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática (<http://www.ati.es/novatica/>), journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by Novática

UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETWORK), the network of CEPIS member societies' publications, that currently includes the following ones:

- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novática**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

#### Editorial Team

Chief Editor: Llorenç Pagés-Casas, Spain, [pages@ati.es](mailto:pages@ati.es)

Associate Editors:

François Louis Nicolet, Switzerland, [nicolet@acm.org](mailto:nicolet@acm.org)

Roberto Carniel, Italy, [rcarniel@dgf.uniud.it](mailto:rcarniel@dgf.uniud.it)

Zakaria Maamar, Arab Emirates, [Zakaria.Maamar@zu.ac.ae](mailto:Zakaria.Maamar@zu.ac.ae)

Soraya Kouadri Mostéfaoui, Switzerland,

[soraya.kouadrimostefaoui@gmail.com](mailto:soraya.kouadrimostefaoui@gmail.com)

Rafael Fernández Calvo, Spain, [rfcvalvo@ati.es](mailto:rfcvalvo@ati.es)

#### Editorial Board

Prof. Wolfried Stucky, CEPIS Former President

Prof. Nello Scarabottolo, CEPIS Vice President

Fernando Píera Gómez and

Llorenç Pagés-Casas, ATI (Spain)

François Louis Nicolet, SI (Switzerland)

Roberto Carniel, ALSI - Tecnoteca (Italy)

#### UPENET Advisory Board

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)

Brian Runciman (ITNOW, United Kingdom)

Franco Filippazzi (Mondo Digitale, Italy)

Llorenç Pagés-Casas (Novática, Spain)

Veith Risak (OCG Journal, Austria)

Panicos Masouras (Pliroforiki, Cyprus)

Andrzej Marciniak (Pro Dialog, Poland)

Rafael Fernández Calvo (Coordination)

**English Language Editors:** Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by Concha Arias Pérez

"Gaia gateway" / © ATI 2007

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Rames

Editorial correspondence: [Llorenç Pagés-Casas <pages@ati.es>](mailto:Llorenç.Pagés-Casas@ati.es)

Advertising correspondence: [novatica@ati.es](mailto:novatica@ati.es)

UPGRADE Newslist available at

<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

#### Copyright

© Novática 2007 (for the monograph)

© CEPIS 2007 (for the sections UPENET and CEPIS News)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (April 2007)

**"Information Technologies  
for Visually Impaired People"**

(The full schedule of UPGRADE  
is available at our website)

### Monograph: Next Generation Web Search

(published jointly with Novática\*)

Guest Editors: *Ricardo Baeza-Yates, José-María Gómez-Hidalgo, and Paolo Boldi*

- 2 Presentation. The Future of Web Search — *Ricardo Baeza-Yates, Paolo Boldi, and José-María Gómez-Hidalgo*
- 5 Efficient Sparse Linear System Solution of the PageRank Problem — *Gianna M. Del Corso, Antonio Gullì, and Francesco Romani*
- 12 Learning to Analyze Natural Language Texts — *Giuseppe Attardi*
- 19 SNAKET: A Personalized Search-result Clustering Engine — *Paolo Ferragina and Antonio Gullì*
- 27 The Multimodal Nature of the Web: New Trends in Information Access — *Luis-Alfonso Ureña-López, Manuel-Carlos Díaz-Galiano, Arturo Montejo-Raez, and M<sup>a</sup> Teresa Martín-Valdivia*
- 33 Adversarial Information Retrieval in the Web — *Ricardo Baeza-Yates, Paolo Boldi, and José-María Gómez-Hidalgo*
- 41 GERINDO: Managing and Retrieving Information in Large Document Collections — *Nivio Ziviani, Alberto H. F. Laender, Edleno Silva de Moura, Altigran Soares da Silva, Carlos A. Heuser, and Wagner Meira Jr.*
- 49 Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web — *Iadh Ounis, Christina Lioma, Craig Macdonald, and Vassilis Plachouras*
- 57 Yahoo! Research Barcelona: Web Retrieval and Mining — *The Yahoo! Research Team*

### UPENET (UPGRADE European NETWORK)

- 59 From **Novática** (ATI, Spain)  
Informatics Profession  
The Maturity of IT Professionalism in Europe — *Sean Brady*
- 68 From **Pro Dialog** (PTI-PIPS, Poland)  
Graphical Interfaces  
Portable Declarative Format for Specifying Graphical User Interfaces — *Zbigniew Fryźlewicz and Rafał Gierusz*
- 75 From **Novática** (ATI, Spain)  
Next-generation Web  
Blogs: On the Cutting Edge of the Next-generation Web — *Antonio Miguel Fumero-Reverón and Fernando Sáez-Vacas*

### CEPIS NEWS

- 83 Harmonise Project: Building up to the Final Report—*François-Philippe Dragnet*
- 84 News & Events: European Funded Projects and News Updates

\* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by Novática, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

# Learning to Analyze Natural Language Texts

Giuseppe Attardi

*Linguistic analysis is rarely used in information retrieval applications like Web search, classification or summarization. Recent advances in statistical and machine learning techniques have spawned developing tools such as parsers or machine translators which are accurate and effective enough for large scale deployment. Future generation Web search engines might perform linguistic analysis of documents to extract semantic relations and to enrich their indexes to provide more sophisticated services than document retrieval. To illustrate these techniques, we outline how to build a dependency parser which learns from examples.*

**Keywords:** Information Access, Information Extraction, Natural Language Processing, Opinion Mining, Parsing, Question Answering.

*"The difference between the intelligence of humans and other mammals is that we have language"*

Jeff Hawkins, "On Intelligence", 2004.

## 1 Introduction

Language is one of the fundamental abilities of the human mind: language not only enables us to communicate but it also shapes our thoughts. A recent study [6] showed that members of a Brazilian tribe, whose language does not define numbers above two, were unable to reliably tell the difference between four objects placed in a row and five in the same configuration.

Language is used to select or to create associations and to communicate them: "Through language one human can invoke memories and create next juxtapositions of mental objects in another human" as Jeff Hawkins says in his fascinating book "On Intelligence" [12].

Studies suggest that in most corporations, around 70% of information is stored in human language text, rather than in structured forms like databases: email, memoranda, reports, operating handbooks, policy and position papers, etc. On the Web an even higher percentage of information is expressed in terms of textual documents, and even rich multimedia documents eventually boil down to carrying language expressions (e.g. song lyrics and movie dialogues).

Since people access information through the Web as well as use computers for communicating with other people (email, messaging, etc.), one would expect that the ability to process language would be an essential part in many computer applications. Nevertheless Natural Language Processing (NLP) tools are still rarely used. Indeed, most applications in the field of Information Retrieval (IR), including document search, classification, summarization, filtering and so on, are based on representing a document as simply a *bag of words* and on performing statistical analysis based

## Author

**Giuseppe Attardi** is professor at the Dipartimento di Informatica of the University of Pisa. He has been visiting professor at MIT, at ICSI in Berkeley, at the Sony Computer Science Laboratory in Paris and the Yahoo! Research Center in Barcelona. Prof. Attardi has been project leader of several European and national projects. He has funded two research spin-offs, DELPHI SpA, specialized in Unix workstations and Ideare SpA, specialized in search engines. He has been involved in the development of the university network and the national research network GARR. Prof. Attardi is the author of innovative software systems, including the MIT Lisp Machine Window System, the garbage collector used by Sun Microsystems in the implementation of the Java language, and IXE, the search engine used by La Repubblica and Tiscali. His current research interests include document analysis, parsing, question answering, and Web computing. <attardi@di.unipi.it>.

on counting the frequencies of occurrence of words in a document. Most attempts to exploit deeper linguistic analysis of documents failed to achieve significant improvements in these tasks.

Despite its effectiveness in certain tasks, the bag of words approach appears limited, as it can only use those pieces of information that are explicitly mentioned in the documents, it fails to exploit synonyms, gets confused by ambiguous terms, and misses important contextual information, for instance the presence of negations.

Why is NLP not used, and some may even say not needed in IR and in tools for information access on the Web? One reason is that typically document retrieval accuracy has been used as a primary measure of information retrieval success. Document retrieval reduces the need for NLP techniques, since discourse factors can be ignored, and using many query words provides word-sense disambiguation. Another reason is the lack of robustness of many NLP techniques, which are typically not as robust as word indexing in handling thousands of real word texts.

There are tasks however where a linguistic analysis of texts may seem to be essential.

## 1.1 Information Extraction

Information extraction consists of automatically extracting structured information from unstructured machine-readable documents. Information extraction involves tasks that use NLP tools such as: Named Entity Recognition: recognition of entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions; Coreference: identification of noun phrases that refer to the same object; Terminology extraction: finding the relevant terms for a given corpus. Extracting relations among entities typically involves using a parser to recover the syntactic relations in the text.

Relation extraction can be used to populate the Semantic Web, i.e. to automatically construct from Web pages the semantic relations and the ontologies which form the core of the Semantic Web, complementing the method of annotating documents using a knowledge representation language like RDF (*Resource Description Framework*) or OWL (*Web Ontology Language*).

## 1.2 Question Answering

A more challenging task for NLP is Question Answering (QA): to give the user a short precise answer to a question expressed in natural language, perhaps supported by evidence. One would expect that NL analysis of the question is required and consequently documents should be processed with NLP techniques to extract the answer and specially to provide the required evidence.

However, it is possible to build a quite effective QA system without relying on NLP techniques at all. The redundancy of a huge repository like the Web can let us get away without NLP. The intuition is that a user's question is often syntactically quite close to sentences that contain the answer. For instance, consider the question: "Where is the Louvre Museum located?". Quite likely within the dozen of billions of document in the Web, at least one will contain "The Louvre Museum is located in Paris". The underlined words occur identical both in the question and in the answer, except in a slightly different order. AskMSR [4] is a QA system based on this idea, which works as follows. First it analyzes the question to obtain a set of question rewrites. For example, for *where* questions, it moves 'is' to all possible locations, obtaining:

"is the Louvre Museum located"  
"the is Louvre Museum located"  
"the Louvre is Museum located"  
"the Louvre Museum is located"  
"the Louvre Museum located is"

The next step consists in submitting all the rewrites to a Web search engine, retrieving the top *n* answers, and then analyzing the matching sentences. Often it is just enough to examine the search engine's "snippets", not the full text of the actual document, to find the correct answer.

Of course some of the variants in step 2 are syntactically incorrect or meaningless, but that is not important, since search is fast enough to afford to perform redundant queries and moreover only a small number of documents

will contain non-grammatical phrases. AskMSR ranked among the top best systems at the annual TREC (*Text Retrieval Conference*) QA task.

Nevertheless, the best ranking systems at most recent TREC QA do indeed exploit linguistic sentence analysis. A set of candidate sentence answers is first retrieved performing a search on the document collection. Questions and candidate answer sentences are parsed to obtain a grammatical parse tree. Parse trees are compared to determine whether they resemble each other enough to conclude that the candidate does indeed contain the answer [1] or to extract a logical representation of the facts in the candidates and try to prove that these facts entail an answer to the question [16].

QA systems also use other linguistic tools to perform Part of Speech tagging, Named Entity recognition, Semantic Role labelling, Coreference resolution as well as linguistic resources like WordNet or gazetteers. This tends to make QA quite a complex task which requires several minutes per question, as opposed to a fraction of a second for a typical Web search.

In order to make QA and other applications which employ NLP techniques more practical, NLP tools need to be made more robust and efficient.

The traditional approach to NL processing involved the use of grammatical formalisms, embedded in either rule-based or logic-based systems. Certain authors have stipulated the existence of a universal grammar, shared by all languages, as an innate capability of the human mind.

More recently, statistical methods, coupled with the ability to process huge document collections allowed significant improvements to be achieved in NLP. The development of statistical parsers is one of the biggest breakthroughs in natural language processing in recent years. Typical statistical parsers still rely on a formal grammar of the language, which is often hard to develop, given the intricacies and subtleties of human languages. Dependency parsers are an alternative approach that avoids the use of a grammar.

## 1.3 Machine Translation

Similarly traditional Machine Translation (MT) paradigms require either extensive transfer-rule writing by linguists or very large parallel training corpora. The former can take person-decades to develop and acquiring parallel text for Statistical MT proves to be a daunting task even for major language pairs. Context-Based Machine Translation<sup>TM</sup> [8] is a new paradigm for corpus based translation that requires neither rules nor parallel corpora.

Both in parsing and machine translation new methods of language analysis are emerging which avoid having to rely on articulated grammar formalisms and instead exploit machine learning techniques applied to large collections of documents. These are fascinating developments since they seem to be getting closer to the psycholinguistic evidence that children learn language without ever being exposed to the notions of grammar.

In the following sections we will concentrate on pars-

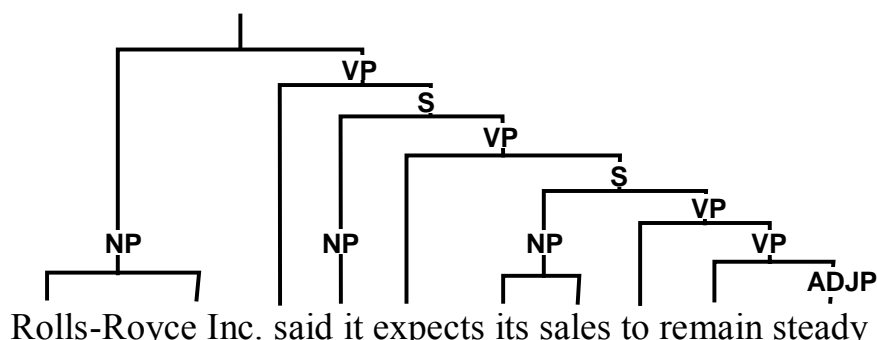


Figure 1: The Parse Tree for the Sentence in the Text.

ing, showing how to build a parser that learns how to parse text without using an explicit grammar formalism and exhibiting a psycholinguistic correct behaviour.

**2 Parsing and Language Analysis**

Constituent parsing consists in analyzing a sentence to produce a parse tree corresponding to a phrase structure grammar for the language. For instance, for the sentence "Rolls-Royce Inc. said it expects its sales to remain steady", the parser would produce the parse tree shown in Figure 1.

Given a grammar for a language, a statistical parser can be trained on a corpus of manually annotated documents to select the most suitable parse tree for a sentence.

Statistical parsers are trained on examples of sentences annotated with the corresponding parse tree. The parser extracts from each example a set of features and learns how to associate those features to the correct tree, i.e. it learns a mapping  $F : S \rightarrow T$  from the set of sentences  $S$  to the set of trees  $T$ .

A linear parsing model consists in:

1. a generator  $GEN : S \rightarrow 2^T$  which generates a set of candidate trees for a sentence
2. a feature extractor  $\Phi : T \rightarrow \mathbb{R}^n$  which gives a vector of the weights of each feature in a tree. A feature corresponds to one dimension in a typically large feature space
3. a vector of weights for the features  $W \in \mathbb{R}^n$ .

When given a new sentence to parse, the parser extracts the features from the sentence and tries to predict which parse tree is the most likely for the sentence.

The process can be summarized in the diagram presented in Figure 2.

The grammar is used to generate all possible parse trees for the sentence, a set of features is extracted for each tree and a statistical model is used to score them, in order to chose the set with the highest score and hence the corresponding tree. The features can be any property of the trees, for instance one feature could be the count of VP nodes in

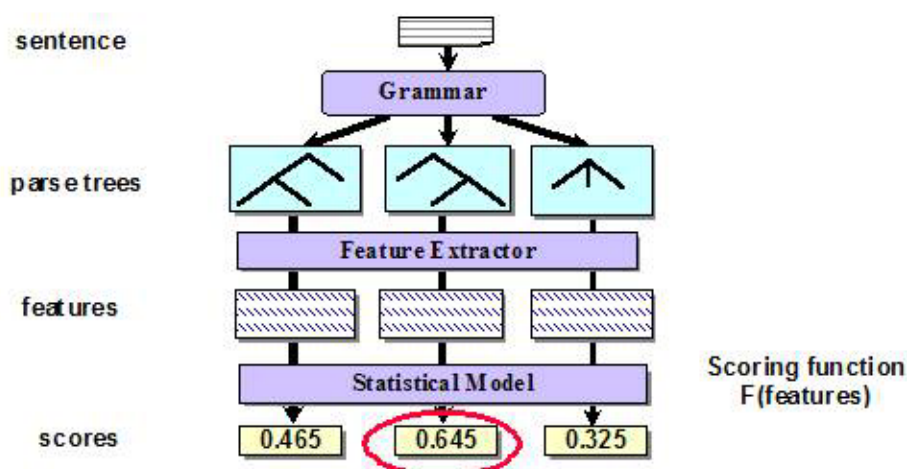


Figure 2: The Process of Predicting the most likely Parse Tree for a Sentence.

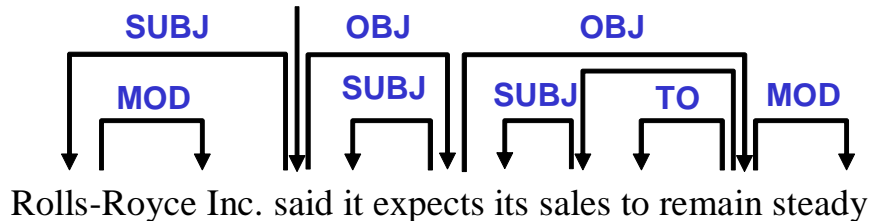


Figure 3: An Example of Dependency Relations.

the tree. The statistical model is trained on a corpus on how to assign weights to each feature, so that the score for a whole tree is just the weighted sum of its features.

Constituent parsing still relies on a grammar for the language, which is often not easy to define, given the intricacies and subtleties of human languages.

As an alternative to constituent parsing, dependency parsers produce dependency trees.

### 3 Dependency Parsing

A dependency tree consists of relations directly between words in the sentence: a *head word* and a *dependent word*. A dependency has an associated type, expressing the grammatical function of the dependency (*Subject, Object, Predi-*

*cate, Determiner, Modifier*). In a *dependency tree*, one word is the head of a sentence, and all other words are either a dependent of that word or else are dependent on some other word which connects to the headword through a sequence of dependencies.

We can see an example of the dependency relations for the example in Figure 3.

A dependency parse tree, while simpler than a constituency tree, still encodes enough semantic information which is useful in further language processing tasks.

Dependency parsing is a simpler task than constituency parsing, since dependency trees do not have extra non-terminal nodes and there is no need for a grammar to generate them.

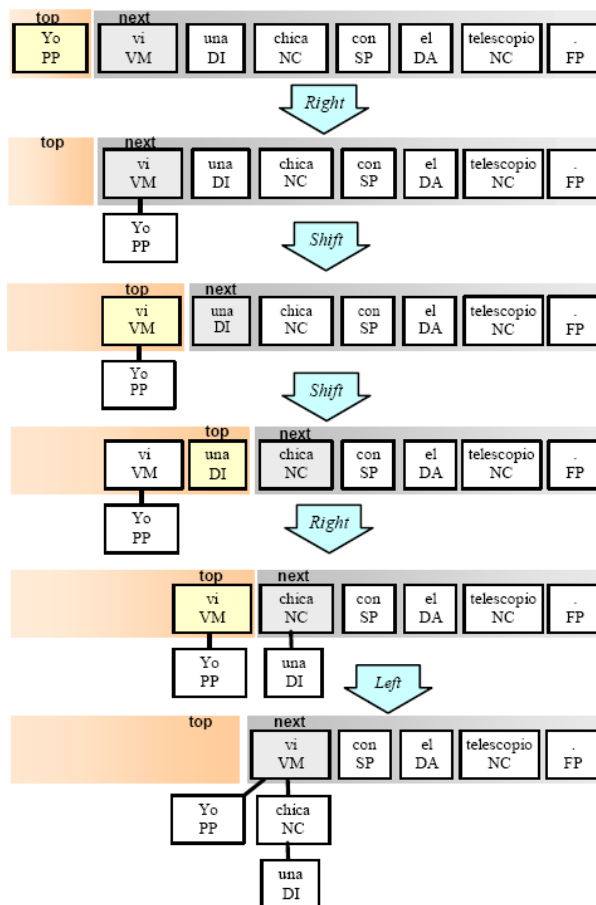


Figure 4: A few Parsing Steps.

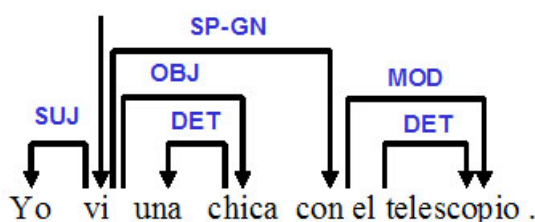


Figure 5: The Parse Tree built from the Parsing Process.

Approaches to dependency parsing either generate such trees by considering all possible spanning trees [15], or build a single tree on the fly by means of shift-reduce parsing actions [18].

Nivre [17] and Attardi [2] have developed deterministic dependency parsers. In particular the DeSR parser [11] is capable of processing hundreds of sentences per second, and hence is suitable for processing large amounts of text documents, as required, for instance, in information retrieval applications.

#### 4 A Multilanguage Dependency Parser

Instead of learning directly which tree to assign to a sentence, the parser *learns the actions* to use for building the tree. We first illustrate through an example the actions the parser would use to parse a sentence and then we discuss how it can learn how to chose the proper action to perform at the proper time.

The parser constructs dependency trees by scanning input sentences in left-to-right word order. Figure 4 shows the first steps in parsing the sentence "Yo vi una chica con el telescopio". Each token in the sentence is annotated with its part of speech.

The actions (*Shift*, *Right* and *Left*) are applied to two neighbouring tokens, *top* and *next*, which appear respectively as yellow and gray boxes in the figure: *next* is the next input token while *top* is the last of the previously processed tokens that are accumulated on the stack *S*.

A *Right* action creates a dependency relation between two neighbouring words where the *next* token becomes a child of the *top* token.

The first action in the example is a *Right* reduce: its effect is to link "Yo" as a dependent of "vi".

*Right* moves the previous word, if it exists, back to *top*, while it does not change *next*. This allows further *Right* actions to be applied to previously skipped words. Since there are none in this case, a *Shift* is needed. *Shift* does not create any dependencies between the target nodes, but just advances the point of focus to the right.

A further *Shift* action is now performed advancing the input to the right and moving *next* to *top*.

The next step is again a *Right* reduction. Next a *Left* action is performed, which constructs a dependency relation between two neighbouring words where the right node of target nodes becomes a child of the left one, in the opposite direction to the action *Right*. *Left* pops *top* from *S* and pushes it back into *next*.

Note that when either of *Left* or *Right* action is applicable, the dependent child should be a complete subtree to which no further dependent children will be added. For the parser to guarantee this, the parser must be capable of looking at the surrounding context of the target nodes: typically it looks at four nodes forwards, two backwards as well as at the children of *top* and *next*.

The rest of the parsing proceeds with a sequence of *Shift*, *Shift*, *Shift*, *Right*, *Left*, *Left*, *Left*, resulting in the parse tree shown in Figure 5.

The three rules above are sufficient for handling languages with a fairly strict word order, like English. In languages with free word order, like Czech and less frequently also in Romance languages, the dependency trees have arcs that cross, and are called non-projective trees, as in this example presented in Figure 6.

For handling non-projective dependencies additional rules must be added to the parser.

In the literature on parsing, the parsing technique just described is classified as a bottom-up *Shift/Reduce* parser. However, a classical *Shift/Reduce* parser uses a table where to look up which action to apply at each step: for context-free languages this table can be generated from the grammar of the language. Since we are assuming not to have a context-free grammar for natural language, the parser must employ different decision criteria. One solution is to have the parser to learn these criteria from examples.

In the area of Machine Learning several types of algorithms have been developed: a suitable technique for this case is supervised learning, in particular applied to the task of classification. Supervised learning requires a training set,

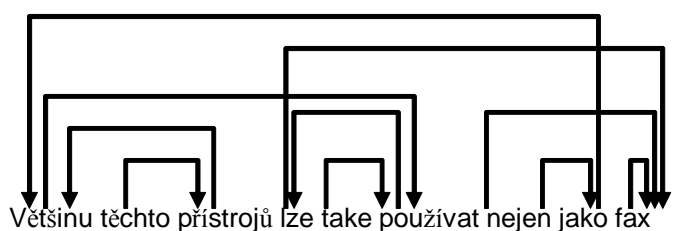


Figure 6: An Example of Dependency Tree for Languages with free Word Order.

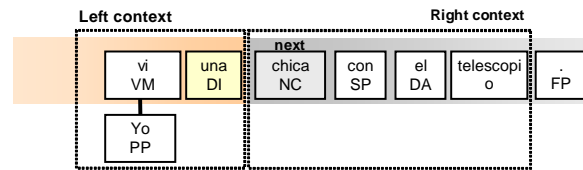


Figure 7: A Parser Context during Learning.

consisting of examples of cases with the associated class to which they belong. A number of classification algorithms exist with good level of accuracy and different performance, including Support Vector Machines (SVM), Maximum Entropy and the Perceptron.

The decision problem of the parser needs to be cast into a classification problem. This can be done by starting from a corpus of documents in the given language, annotated with dependency relations. Several such corpora exist for various languages, for instance those that were used in the CoNLL-X Shared task [7]. Corpora for Basque, Castilian Spanish, and Catalan are provided by the CESS-ECE TreeBanks [13].

Each sentence in the corpus can be used as an oracle which tells the correct sequence of actions required to build the sentence tree. In the learning phase the parser is instructed by such an oracle on which action to perform and starts building the tree. At each step the parser collects a set of features which represent the current situation. For example, when the parser reaches the state shown in Figure 7, it will consider a context consisting of two tokens to the left and four tokens to the right. For each token in the context it collects as features the word and its part-of-speech, annotated with their relative position in the context. A feature will consist in a triple (position, type, value), where position is a sequence of integers: -2.1 represents two position backward (-2) and then down to the first (1) child. For the state represented in Figure 7, the features are as follows:

- (-2, lex, vi), (-2, pos, VM), (-2.1, lex, Yo), (-2.1, pos, pp)
- (-1, lex, una), (-1, pos, DI),
- (0, lex, chica), (0, pos, NC),
- (+1, lex, con), (+1, pos, SP),
- (+2, lex, el), (+2, pos, DA),
- (+2, lex, telescopio), (+2, pos, NC).

The action to perform in this state is a *Right*, so a training example is created telling the classifier that this set of features must be classified as belonging to the *Right* class. From a corpus consisting of 15,000 sentences and 500,000 tokens like those for Spanish and Catalan in the CESS-ECE, over 850,000 training examples are generated and 2.5 million sets of features. Maximum Entropy or Perceptron classifiers are capable of handling problems of this size in a few hours on a Pentium PC, while SVM typically take longer. Once the classifier has been trained, the parser can use it to predict which action to perform at each step. Using Maximum Entropy or Perceptron, parsing is much faster than training, analyzing sentences at the rate over 200 per second.

Using an Averaged Perceptron classifier the parser has achieved accuracies ranging from 84% to 92% in parsing the 13 languages of the CoNLL-X Shared Task [7].

Further improvement have been achieved by adding a further processing step which tries to correct the trees produced applying revision rules learned from previous mistakes [3].

The intuitive motivation for the approach is the observation that a dependency parser is mostly right in identifying chunks in the sentence and hence only local corrections are needed to rearrange such chunks, similarly to what people possibly do when they fail to immediately understand a sentence.

### 5 NLP in Information Access

NLP techniques can be used to improve information access in many ways. Information sources like web pages, news and blogs can be analyzed to extract semantic information and semantic relations. This information can be used to create enriched indexes for these documents so that searching may exploit such automatically extracted metadata. For instance Named Entity Recognizers can detect entities named in documents and distinguish the occurrence of common words from those referring to people, locations or corporations, e.g. "apple" vs "Apple Inc.". In financial reports one could identify the analyst name, the target company, the earnings estimate, and the date of the release. Parsing can then be applied to extract the relationships among entities, e.g. who bought what and when. The user can then pose queries in natural language which get parsed in order to identify the focus of the query and the relations being queried. Answers to query can then be synthesized from the analysis of relevant documents. For example a query like: "What is the earnings estimate for Amazon?" might return a combined answer like: "The current consensus estimate for Amazon is \_\_. Specific estimates are: Marcus Teeker: xxx, ...".

Opinion mining is another area where NLP techniques are being exploited. While traditional text classification tries to assign predefined categories to a document, such as spam/no-spam for e-mail, sentiment or opinion mining is a different and challenging task whose goal is the assessment of the writer's attitude toward a subject. Examples include categorization of customer e-mails and reviews by types of claims, modalities, or subjectivities.

For opinion mining, techniques based on extracting dependency relations have proven to be more effective than traditional bag-of-word approaches [14]. Dependency rela-

mation and semantic relations. This information can be used to create enriched indexes for these documents so that searching may exploit such automatically extracted metadata. For instance Named Entity Recognizers can detect entities named in documents and distinguish the occurrence of common words from those referring to people, locations or corporations, e.g. "apple" vs "Apple Inc.". In financial reports one could identify the analyst name, the target company, the earnings estimate, and the date of the release. Parsing can then be applied to extract the relationships among entities, e.g. who bought what and when. The user can then pose queries in natural language which get parsed in order to identify the focus of the query and the relations being queried. Answers to query can then be synthesized from the analysis of relevant documents. For example a query like: "What is the earnings estimate for Amazon?" might return a combined answer like: "*The current consensus estimate for Amazon is \_\_\_. Specific estimates are: Marcus Teeker: xxx, ...*".

Opinion mining is another area where NLP techniques are being exploited. While traditional text classification tries to assign predefined categories to a document, such as spam/no-spam for e-mail, sentiment or opinion mining is a different and challenging task whose goal is the assessment of the writer's attitude toward a subject. Examples include categorization of customer e-mails and reviews by types of claims, modalities, or subjectivities.

For opinion mining, techniques based on extracting dependency relations have proven to be more effective than traditional bag-of-words approaches [14]. Dependency relations allow the distinguishing statements of opposite polarity, e.g. "I liked the movie", and "I didn't like it" Dependency relations map more easily into semantic relations useful to represent the semantic content of a sentence.

The ability to identify and group documents by intent may lead to new tools for knowledge discovery, for instance for generating a research survey that collects relevant opinions on a subject, for determining prevalent judgments about products or technologies, for analyzing reviews, for gathering motivations and arguments from court decision making or lawmaking debates, for analyzing linkages in medical abstracts to discover drug interactions.

## 6 Conclusions

Statistical methods based on machine learning, coupled with the ability to process huge document corpora are producing significant advances in Natural Language Processing. The new approaches are characterized by learning without the need of explicit grammar formalisms and appeal to psycholinguistic evidence of how humans perform language tasks.

The ability to analyze texts by means of NLP tools is leading to more advanced solutions for information access.

## References

[1] G. Attardi, et al. PiQASso: Pisa Question Answering System. In Proceedings of the Tenth Text REtrieval

Conference, 2001 (TREC 2001).  
 [2] G. Attardi. 2006. Experiments with a Multilanguage non-projective dependency parser. In Proc. of the Tenth CoNLL, 2006.  
 [3] G. Attardi, M. Ciaramita. Tree Revision Learning for Dependency Parsing. In Proc. of HLT/AAACL, 2007.  
 [4] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics, 21(4):pp 543–565, 1995.  
 [5] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. Data-Intensive Question Answering. In Proceedings of the Tenth Text REtrieval Conference (TREC 2001), 393–400.  
 [6] P. Gordon. Numerical cognition without words: Evidence from Amazonia. Science 306 (5695): pp 496–9, 2004.  
 [7] S Buchholz, et al. CoNLL-X Shared Task on Multilingual Dependency Parsing. In Proc. of the Tenth CoNLL, 2006.  
 [8] J. Carbonell, et al. Context-Based Machine Translation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, 19–28, 2006.  
 [9] E. Charniak. A maximum-entropy-inspired parser. In Proc. NAACL, 2000.  
 [10] M. Collins, T Koo. Discriminative Reranking for Natural Language Parsing. Proc. 17th International Conf. on Machine Learning, pp 175–182, 2000.  
 [11] DeSR parser, <<http://medialab.di.unipi.it/Project/QA/Parser>>.  
 [12] J. Hawkins. On Intelligence. Times Books, 2004. ISBN: 0805074562.  
 [13] M. A. Martí, M. Taulé, L. Márquez, M. Bertran. CESS-ECE: A Multilingual and Multilevel Annotated Corpus, 2007. In <<http://www.lsi.upc.edu/~mbertran/cess-ece/publications>>.  
 [14] S. Matsumoto, H. Takamura, M. Okumura. Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In: T.B. Ho, D. Cheung & H. Li (eds), Proceeding of PAKDD'05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. LNCS, vol. 3518, 2005.  
 [15] R. McDonald, et al. Non-projective Dependency Parsing using Spanning Tree Algorithms. In Proc. of HLT-EMNLP, 2005.  
 [16] D. Moldovan, et al. A Temporally-Enhanced PowerAnswer in TREC 2006. In Proc. of TREC 2006.  
 [17] J. Nivre and M. Scholz. Deterministic Dependency Parsing of English Text. In Proc. of COLING 2004, Geneva, Switzerland, pp 64–70, 2004.  
 [18] H. Yamada, Y. Matsumoto. Statistical Dependency Analysis with Support Vector Machines. Proc. of the 8th International Workshop on Parsing Technologies (IWPT), 195–206, 2003.