TEXT, SPEECH
AND LANGUAGE
TECHNOLOGY

INDUCTIVE DEPENDENCY PARSING

By Joakim Nivre

Springer

Inductive Dependency Parsing

by

Joakim Nivre
Växjö University, Sweden

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 1-4020-4888-2 (HB)
ISBN-13 978-1-4020-4888-3 (HB)
ISBN-10 1-4020-4889-0 (e-book)
ISBN-13 978-1-4020-4889-0 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

Printed in the Netherlands

Preface

This book is based on work carried out over a period of roughly three years with the support of a number of people and organizations that deserve my heartfelt gratitude. In the first place, I want to thank my PhD students Johan Hall and Jens Nilsson, who have been involved in the project from the start. I also want to thank all the people who are part of the research group in computer science at Växjö University, for providing a stimulating environment to work in, and the Swedish Research Council for a grant that supported part of the work reported in this book (Vetenskapsrådet, 621-2002-4207).

Among the many people who have contributed, directly or indirectly, to the ideas and results presented in the book, I specifically want to mention Eckhard Bick, Sabine Buchholz, John Carroll, Atanas Chanev, Yuchang Cheng, Walter Daelemans, Ralph Debusmann, Denys Duchier, Gülsen Eryiğit, Jason Eisner, Kilian Foth, Kadri Hacioglu, Jan Hajič, Erhard Hinrichs, Tomáš Holan, Viggo Kann, Matthias Trautner Kromann, Geert-Jan Kruijff, Sandra Kübler, Marco Kuhlmann, Haitao Liu, Welf Löwe, Svetoslav Marinov, Erwin Marsi, Yuji Matsumoto, Ryan McDonald, Pierre Nugues, Tomasz Obrębski, Guy de Pauw, Aarne Ranta, Mario Scholz, Noah Smith, Antal van den Bosch, Hiroyasu Yamada, Anssi Yli-Jyrä, and Daniel Zeman. I also want to thank my editor, Jolanda Voogd, for practical assistance, and the series editors, Nancy Ide and Jean Véronis, for promoting the publication of the book in the first place.

I owe a special debt to John Carroll, Walter Daelemans and Welf Löwe, who scrutinized the first draft of the manuscript and suggested innumerable improvements, and to Viggo Kann, who spotted an error in one of the proofs. All remaining inadequacies are entirely my own responsibility.

Finally, I want to express my love and gratitude to my wife Elisabeth and my son Fredrik for making my life such a wonderful experience.

Växjö, February 2006 *Joakim Nivre*

Contents

1 **Introduction** 1
1.1 Inductive Dependency Parsing 1
1.2 The Need for Robust Disambiguation 5
1.3 Outline of the Book 6

2 **Natural Language Parsing** 9
2.1 Syntactic Representations 10
2.2 Two Notions of Parsing 12
2.2.1 Grammar Parsing 13
2.2.2 Text Parsing 16
2.2.3 Competence and Performance 19
2.3 Methods for Text Parsing 20
2.3.1 Grammar-Driven Text Parsing 20
2.3.2 Data-Driven Text Parsing 27
2.3.3 Converging Approaches 37
2.3.4 Inductive Dependency Parsing 40
2.4 Evaluation Criteria 41
2.4.1 Robustness 41
2.4.2 Disambiguation 41
2.4.3 Accuracy 42
2.4.4 Efficiency 43

3 **Dependency Parsing** 45
3.1 Dependency Grammar 46
3.1.1 The Notion of Dependency 47
3.1.2 Varieties of Dependency Grammar 50
3.2 Parsing with Dependency Representations 55
3.2.1 Grammar-Driven Dependency Parsing 56
3.2.2 Data-Driven Dependency Parsing 61
3.2.3 The Case for Dependency Parsing 66

3.3 A Framework for Dependency Parsing 67
3.3.1 Texts, Sentences and Tokens 68
3.3.2 Dependency Graphs 69
3.3.3 Dependency Parsing 72
3.4 Parsing Algorithm 72
3.4.1 Configurations 72
3.4.2 Transitions 74
3.4.3 Deterministic Parsing 76
3.4.4 Algorithm Analysis 79
3.4.5 Evaluation Criteria Revisited 85

4 Inductive Dependency Parsing 87
4.1 A Framework for Inductive Dependency Parsing 88
4.1.1 Data-Driven Text Parsing 88
4.1.2 Inductive Inference 89
4.1.3 History-Based Models 90
4.1.4 Parsing Methods 92
4.1.5 Learning Methods 94
4.1.6 Oracle Parsing 96
4.2 Features and Models 100
4.2.1 Feature Functions 101
4.2.2 Static Features 105
4.2.3 Dynamic Features 107
4.2.4 Feature Models 108
4.3 Memory-Based Learning 110
4.3.1 Memory-Based Learning and Classification 110
4.3.2 Learning Algorithm Parameters 112
4.3.3 Memory-Based Language Processing 115
4.4 MaltParser 117
4.4.1 Architecture 118
4.4.2 Implementation 120

5 Treebank Parsing 121
5.1 Treebanks and Parsing 122
5.1.1 Treebank Evaluation 123
5.1.2 Treebank Learning 128
5.1.3 Treebanks for Dependency Parsing 129
5.2 Experimental Methodology 132
5.2.1 Treebank Data 132
5.2.2 Models and Algorithms 139
5.2.3 Evaluation 140
5.3 Feature Model Parameters 142
5.3.1 Part-of-Speech Context 143
5.3.2 Dependency Structure 145
5.3.3 Lexicalization 147

5.3.4 Efficiency 149
5.3.5 Learning Curves 155
5.4 Learning Algorithm Parameters 158
5.4.1 Neighbor Space and Distance Metric 159
5.4.2 Weighting Schemes 161
5.5 Final Evaluation 163
5.5.1 Accuracy and Efficiency 163
5.5.2 Related Work 168
5.5.3 Error Analysis 171

6 Conclusion 175
6.1 Main Contributions 175
6.2 Future Directions 179

References 183

Index 209

1

Introduction

The automatic analysis of syntactic structure, or parsing, is a core component in many systems for natural language processing. This monograph explores the framework of *inductive dependency parsing* as an efficient method for syntactic parsing of unrestricted natural language text under the joint requirements of robustness and disambiguation. That is, given as input a natural language text, consisting of a sequence of sentences, we want the parser to assign to every sentence at least one analysis (*robustness*) and at most one analysis (*disambiguation*). Needless to say, we also want the single analysis assigned to a sentence to be correct as often as possible (*accuracy*). Finally, we want the computation for each sentence to take as little time and memory as possible (*efficiency*). Maximizing accuracy and efficiency while maintaining robustness and disambiguation is the problem that we have set ourselves. Finding out whether inductive dependency parsing can provide a solution to this problem is the topic of this book.

1.1 Inductive Dependency Parsing

In the framework of inductive dependency parsing, the syntactic analysis of a sentence amounts to the derivation of a dependency structure, using inductive machine learning to guide the parser at nondeterministic choice points. This methodology combines a number of themes that are prominent in the recent natural language processing literature, although the particular combination of ideas embodied in the resulting framework appears to be original. More precisely, inductive dependency parsing can be regarded as the simultaneous instantiation of two notions that have played a more or less central role in natural language parsing in recent years:

- Dependency-based parsing
- Data-driven parsing

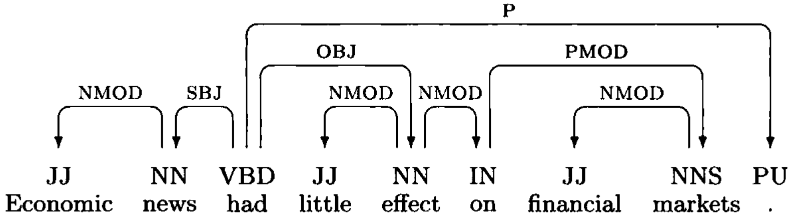
P
OBJ
PMOD
NMOD
SBJ
NMOD
NMOD
NMOD
JJ NN VBD JJ NN IN JJ NNS PU
Economic news had little effect on financial markets .

Fig. 1.1. Dependency structure for English sentence from the Penn Treebank

The fundamental idea in dependency-based parsing is that parsing crucially involves establishing binary relations between words in a sentence. This is illustrated in figure 1.1, which depicts the analysis of a short sentence taken from the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993, 1994). In this example, the syntactic structure is built up by recognizing a subject relation (SBJ) from the finite verb *had* to the noun *news*, a nominal modifier relation (NMOD) from *news* to the adjective *Economic*, an object relation (OBJ) from *had* to the noun *effect*, and so on.

Dependency-based methods appear in many guises in the current literature on natural language parsing. On the one hand, we have what we may call dependency parsing in a narrow sense, where the goal of the parsing process is to build a dependency structure, i.e., a graph built from binary dependency relations as in figure 1.1, and where the analysis more or less closely adheres to the theoretical tradition of dependency grammar. Cases in point are Hellwig (1980), Maruyama (1990), Harper and Helzerman (1995), Tapanainen and Järvinen (1997), Menzel and Schröder (1998), and Duchier (1999). On the other hand, we have approaches that can be characterized as dependency-based parsing in a broader sense, where the syntactic analysis may not take the form of a dependency structure, but where the construction of the analysis nevertheless depends on finding syntactic relations between lexical heads. In this category, we find the widely used link grammar parser for English (Sleator and Temperley, 1993), as well as the influential probabilistic parsers of Collins (1997, 1999) and Charniak (2000), but also a variety of other lexicalized parsing models that can be subsumed under the general notion of bilexical grammars (Eisner, 2000). The use of bilexical relations for disambiguation has been a significant theme in research on natural language parsing during the last decade, although the results are not completely unambiguous (Collins, 1999; Gildea, 2001; Klein and Manning, 2003; Bikel, 2004).

The framework we develop in this book falls under the more narrow definition of dependency parsing, at least in the sense that it assumes dependency structures as the only form of syntactic representation. At the same time, we will focus more on formal methods for constructing dependency structures than on details of linguistic analysis, which means that the discussion will remain rather agnostic with respect to different theoretical traditions of dependency grammar. More precisely, we will define the task of dependency parsing relative to a formal framework of dependency graphs, where only minimal assumptions are made concerning the linguistic analysis, but where the notions of robustness, disambiguation, accuracy and efficiency can be given precise definitions.

Although our formal characterization of dependency parsing is compatible with different parsing strategies, we will limit our attention in this study to deterministic methods, which means that we derive a single analysis for each input sentence in a monotonic fashion with no redundancy or backtracking. Historically, deterministic parsing of natural language has been investigated with a view to modeling human sentence processing in a psychologically plausible way, as in the work by Marcus (1980) and Shieber (1983), but it has also been explored as a way of improving the robustness and efficiency of natural language parsing, especially in various approaches to partial parsing using finite-state methods (Ejerhed, 1983; Koskenniemi, 1990; Abney, 1991, 1996).

In the present framework, we want to apply deterministic methods to full parsing, insofar as we want to derive a complete dependency structure for each input sentence. In this way, we hope to combine the gains in efficiency with a deeper analysis of syntactic structure. The parsing algorithm that we use was first presented in Nivre (2003), with a partial analysis of its complexity and robustness properties. It has also been shown that the algorithm favors incremental processing, something that may be desirable both for certain applications, such as language modeling for speech recognition, and for the kind of psycholinguistic modeling that inspired early research on deterministic parsing (Nivre, 2004a). In this book, we will for the first time provide a comprehensive analysis of the parsing algorithm with respect to robustness, disambiguation and complexity.

The second essential component of our methodology is a commitment to data-driven parsing, understood in a broad sense to include all approaches that make essential use of empirical data, in particular treebanks or parsed corpora (Abeillé, 2003b; Nivre, forthcoming), in the development of parsing systems for natural language. Research during the last ten to fifteen years has shown rather conclusively that an empirical approach is necessary in order to achieve accurate disambiguation as well as robustness in parsing unrestricted text, regardless of whether the parser uses a traditional grammar or a more radically data-driven model. In the former case, exemplified by broad-coverage deep parsers such as Riezler et al. (2002) and Toutanova et al. (2002), treebank data are used to tune and optimize the parser, in particular by constructing a statistical model for parse selection. In the latter case, represented by probabilistic parsers such as Collins (1997, 1999) and Charniak (2000), the grammar is replaced by a statistical model, the parameters of which are derived from treebank data using machine learning techniques.

An even more radical approach is to replace the statistical model by the treebank itself, and to reuse fragments of previously encountered syntactic structures to construct new ones during parsing, as in the framework of Data-Oriented Parsing (DOP) (Bod, 1995, 1998; Bod, Scha and Sima'an, 2003). The DOP model can be seen as an instantiation of the paradigm of memory-based learning, or lazy learning, which is based on the idea that learning is the simple storage of experiences in memory and that new problems are solved by reusing solutions from similar old problems (Daelemans, 1999; Daelemans and Van den Bosch, 2005). Memory-based learning has been successfully applied to a wide variety of problems in natural language processing, such as grapheme-to-phoneme conversion, part-of-speech tagging, prepositional phrase attachment, and chunking (Daelemans et al., 2002). Memory-based approaches to syntactic parsing, in addition to the DOP framework, include Veenstra and Daelemans (2000), Buchholz (2002), De Pauw (2003) and Kübler (2004).

In this book, we will explore a memory-based approach to dependency parsing, using classifiers that predict the next parsing action based on the current state of the parser and a database of previously encountered parser states. Since the state of the parser results from a sequence of previous actions, this can also be seen as a form of history-based parsing (Black et al., 1992; Jelinek et al., 1994; Magerman, 1995), although we prefer the term *inductive dependency parsing* for the general idea of using inductive machine learning to predict the actions of a dependency parser.

An early version of this idea, with a simple probabilistic classifier, was reported in Nivre (2004b). The memory-based version was first presented in Nivre et al. (2004), with an evaluation on Swedish treebank data, and later in Nivre and Scholz (2004), with results from the Wall Street Journal section of the Penn Treebank. This book provides a comprehensive analysis of inductive dependency parsing, including a general characterization of the history-based model and a formal framework for the specification of model parameters. For the deterministic memory-based instantiation of this framework, we also present a detailed discussion of feature selection, and a thorough empirical evaluation of different models using treebank data from both Swedish and English that goes far beyond previously published results.

The framework developed in this book is implemented in a system called MaltParser, which is used in all the experiments reported below. MaltParser can be described as a language-independent parser-generator. When applied to a dependency-based treebank, the system generates a dependency parser for the language represented in the treebank. The memory-based version of this system uses the TiMBL software package (Daelemans et al., 2004) and supports a variety of options with respect to linguistic features as well as learning parameters. A version of MaltParser is freely available for research and educational purposes.[1]

[1] URL: http://www.msi.vxu.se/users/nivre/research/MaltParser.html

1.2 The Need for Robust Disambiguation

The usefulness of parsing in different language technology applications is a point of some controversy. For example, in speech recognition, syntax-based language models have had a hard time improving on the results obtained with probabilistic n-gram models (Rosenfeld, 2000). Similarly, attempts at improving accuracy in information retrieval by incorporating syntactic information have met with very limited success (Tzoukermann et al., 2003). If we move to applications that require some kind of semantic analysis of individual sentences, the role of parsing becomes more evident. For instance, information extraction normally involves at least partial parsing (Cowie and Wilks, 2000), and question answering systems often rely on semantic role labeling, for which full syntactic parsing has been shown to give a substantial improvement over partial parsing (Gildea and Palmer, 2002; Carreras and Màrquez, 2004). In machine translation, parsing has always been a core component of transfer-based systems, but syntax-based models are becoming more prominent also in statistical approaches (Yamada and Knight, 2001; Charniak et al., 2003).

At the end of the day, few researchers would question the relevance of syntactic analysis for the ultimate goal of building computer systems capable of full natural language understanding — however we define this — but there is still no consensus on what form the analysis should take and which methods should be used to derive it. In this book, we will focus on the development of a particular framework for natural language parsing, and even though we will sometimes draw on requirements from applications to motivate certain design choices, we will not be able to demonstrate that these choices actually improve applications, and the potential usefulness of parsing as such will simply have to be taken for granted.

The emphasis on robustness, disambiguation and efficiency in the context of natural language parsing may also need some further motivation. Starting with robustness, we see this as a fundamental requirement in any application of natural language parsing that deals with (more or less) unrestricted text, where the range of permissible inputs cannot be sharply delimited. Even if the likelihood of a correct analysis decreases as the input deviates more and more from our expectations, we want to have a system that degrades gracefully and always delivers some kind of analysis.

Disambiguation is a more controversial requirement, given that part of the information needed to choose between alternative analyses, such as word sense information and extra-sentential context, may be missing at parsing time. This observation leads naturally to the assumption that the parser should simply pass on all analyses that are compatible with the given input and leave the final decision to another component, typically a semantic or pragmatic interpreter. However, the same observation can be made about almost any kind of input analysis, from tokenization and sentence segmentation to semantic and pragmatic analysis. So, unless we adopt a completely holistic integration of all processing levels, there will be decisions at each level that are based on incomplete information. Moreover, the requirement of robustness will often lead to a relaxation of syntactic constraints to the point where the number of analyses compatible with a given input becomes prohibitively large. This means that some degree of pruning is necessary in any case, even though the search space may only be reduced to the n best candidates rather than to a single analysis. Finally, the capacity for disambiguation can be very useful in applications where the parser is not used as part of a processing chain but rather is used to generate features for another kind of analysis. A typical case in point is the use of parse tree information in semantic role labeling referred to earlier. Thus, without wanting to claim that robust disambiguation is the solution to every syntactic analysis problem, we believe that it is useful in many situations, and it will be adopted as a basic requirement for the methods investigated in this book.

Efficiency, finally, is a non-functional requirement for parsers to be usable in practical applications, especially in systems working under hard time constraints, such as speech-based user interfaces, or dealing with large volumes of data, such as information retrieval and extraction systems. In many cases, there is a trade-off between efficiency and accuracy, and although we often give priority to accuracy over efficiency, it is nevertheless a joint optimization problem, since we cannot reduce efficiency to the point where parsers become unusable in practical applications.

The framework developed in this book is the result of a conscious strategy to adopt methods for parsing and disambiguation that are provably robust and efficient, in a sense yet to be made precise, and to work systematically towards higher accuracy while maintaining robustness, disambiguation and (as far as possible) efficiency. Needless to say, this is only one of many conceivable strategies, and it may not be the one that ultimately gives us the highest accuracy, although it should provide us with highly efficient methods with sufficient accuracy for certain applications.

From a scientific point of view, it is also interesting to see how far we can get by adopting an extreme approach and pushing it to its limits. At the very least, this may give a new perspective on results achieved in other frameworks using other strategies. More importantly, however, by concentrating on the systematic study of a few simple ideas and techniques, we may hope to gain a deeper understanding of the way in which they can contribute to improved methods for natural language parsing in general.

1.3 Outline of the Book

In this introductory chapter, we have tried to outline the aims of the study and to motivate the general research directions. The remainder of the book is structured as follows.

Chapter 2
Natural Language Parsing

Chapter 2 discusses the problem of parsing unrestricted natural language text, relating it to other notions of parsing, in particular the one associated with grammars in formal language theory. We compare different strategies for achieving robust disambiguation and define evaluation criteria for the key concepts of robustness, disambiguation, efficiency and accuracy.

Chapter 3
Dependency Parsing

Chapter 3 starts with a review of dependency grammar and its use in syntactic parsing. We then introduce a formal framework for dependency parsing, based on a general definition of labeled dependency graphs with a further characterization of properties such as connectedness, single-headedness, acyclicity, and projectivity. Finally, we present a deterministic parsing algorithm for projective dependency graphs, with proofs of complexity and properties related to robustness and disambiguation.

Chapter 4
Inductive Dependency Parsing

Chapter 4 extends the framework of dependency parsing to incorporate the use of inductive machine learning to guide the parser at nondeterministic choice points. We derive a history-based model of dependency parsing and show how this can be combined with the deterministic parsing algorithm presented in chapter 3 and with discriminative learning methods that induce classifiers from treebank data. We define a formal method for the specification of feature models, we introduce memory-based learning and classification as a method for solving the inductive learning problem defined by the parsing method, and we briefly describe the implemented MaltParser system.

Chapter 5
Treebank Parsing

Chapter 5 contains an empirical evaluation of the parsing methodology with respect to accuracy and efficiency, based on data from Talbanken, a small Swedish treebank, and the Penn Treebank of American English. The chapter starts with a general discussion of treebanks and their use in syntactic parsing, moves on to a description of the evaluation framework and the experimental setup, and concludes with a discussion of the results in relation to previous work on treebank parsing, in particular dependency-based parsing.

Chapter 6
Conclusion

Chapter 6 summarizes the main results of the study and points to promising directions for future research, such as the extension to non-projective dependency structures, which may be needed for languages with more flexible word order; the introduction of mild forms of nondeterminism and stochastic disambiguation; the exploration of alternative learning methods, including an integration of inductive and deductive learning; and the use of more refined evaluation methods.

2

Natural Language Parsing

Research on natural language parsing has over a period of several decades produced a wealth of knowledge concerning different methods for automatic syntactic analysis. Most of the results, however, concern formal grammars and algorithms that are only indirectly related to the more practical problem of analyzing syntactic structure in naturally occurring texts. This has led to a somewhat paradoxical situation where, despite the increase in knowledge about the complexity of problems and algorithms for formal grammars, we know relatively little about the formal properties of text parsing. In fact, it is still not clear that there is a well-defined parsing problem for natural language text that is computable in the strict sense.

In this chapter, we will begin by contrasting the two notions of parsing, the well-defined parsing problem for formal grammars, familiar from both computer science and computational linguistics, and the more open-ended problem of parsing unrestricted text in natural language, which is the focus of the investigations in this book. We will then review different strategies for text parsing, including both grammar-driven and data-driven approaches, and discuss the different kinds of problems that arise with different methods. On the basis of this discussion, we will then define the basic requirements of robustness, disambiguation, accuracy and efficiency, which are central to the investigations of text parsing in this book, and discuss evaluation criteria for each of the requirements.

The primary goal of this chapter is to set the scene for the exploration of inductive dependency parsing in later chapters, by defining the basic problems and evaluation criteria, but in doing so we will also have reason to review some of the more important trends in recent research on natural language parsing. First of all, however, we need to say a few words about the desired output of the parsing process, i.e., about syntactic representations for natural language sentences.

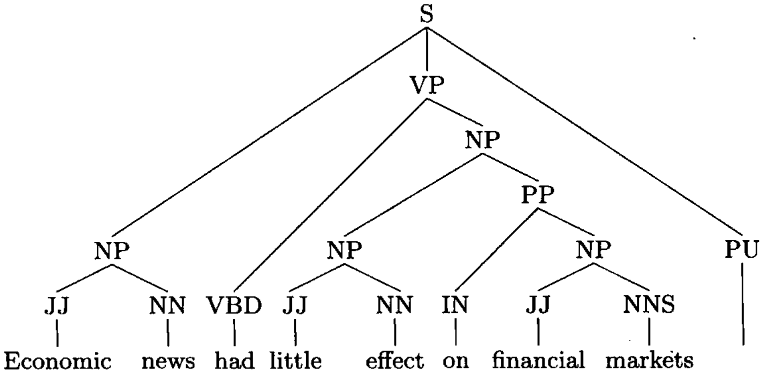

Fig. 2.1. Constituent structure for English sentence from the Penn Treebank

2.1 Syntactic Representations

The type of syntactic representation that has been dominant during the last fifty years, both in theoretical linguistics and in natural language processing, is based on the notion of *constituency*. In this representation, a sentence is recursively decomposed into smaller segments, called *constituents* or *phrases*, which are typically categorized according to their internal structure into *noun phrases*, *verb phrases*, etc. Constituency analysis comes from the structuralist tradition represented by Bloomfield (1933) and was formalized in the 1950s in the model of phrase structure grammar, or context-free grammar (Chomsky, 1956). Figure 2.1 shows a typical constituency representation of an English sentence, taken from the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993, 1994).[1]

A wide range of different theories about natural language syntax are based on constituency representations. In addition to the theoretical tradition of Chomsky (1957, 1965, 1981, 1995), this includes frameworks that are prominent in computational linguistics, such as Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2000), Generalized Phrase Structure Grammar (GPSG) (Gazdar et al., 1985), Tree Adjoining Grammar (TAG) (Joshi, 1985, 1997), and Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987, 1994).

Another type of syntactic representation, which has a long tradition in descriptive linguistics especially in Europe, is instead based on the notion of *dependency*. In this representation, a sentence is analyzed by connecting its words by binary asymmetrical relations, called *dependencies*, which are

[1] The representation is equivalent to the treebank annotation except that the part-of-speech category '.' has been replaced by PU (for *punctuation*) to avoid a name clash with the terminal '.'. This will simplify exposition later on.