# 11

# Two Notions of Parsing

JOAKIM NIVRE

The term *parsing*, derived from Latin *pars orationis* (parts of speech), was originally used to denote the grammatical explication of sentences, as practiced in elementary schools. The term was later borrowed into computer science and linguistics, where it has acquired a specialized sense in connection with the theory of formal languages and grammars. However, in practical applications of natural language processing, the term is also used to denote the syntactic analysis of sentences in text, without reference to any particular formal grammar, a sense which is in many ways quite close to the original grammar school sense.

In other words, there are at least two distinct notions of parsing that can be found in the current literature on natural language processing, notions that are not always clearly distinguished. I will call the two notions *grammar parsing* and *text parsing*, respectively. Although I am certainly not the first to notice this ambiguity, I feel that it may not have been given the attention that it deserves. While it is true that there are intimate connections between the two notions, they are nevertheless independent notions with quite different properties in some respects. In this paper I will try to pinpoint these differences through a comparative discussion of the two notions of parsing. This is motivated primarily by an interest in the problem of text parsing and a desire to understand how it is related to the more well-defined problem of grammar parsing. In a following companion paper I will go on to discuss different strategies for solving the text parsing problem, which may or may not involve

*A Finnish Computer Linguist: Kimmo Koskenniemi*
*Festschrift on the 60th birthday*.
Editors: Arppe, Carlson,
Heinämäki, Lindén,
Miestamo, Piitulainen, Tupakka,
Westerlund, Yli-Jyrä,
jne... (lisättävä puuttuvat).

| | | | | | |
|---|---|---|---|---|---|
| S | → | NP VP PU | JJ | → | Economic |
| VP | → | VP PP | JJ | → | little |
| VP | → | VBD NP | JJ | → | financial |
| NP | → | NP PP | NN | → | news |
| NP | → | JJ NN | NN | → | effect |
| NP | → | JJ NNS | NNS | → | markets |
| PP | → | IN NP | VB | → | had |
| PU | → | . | IN | → | on |

FIGURE 1 Context-free grammar for a fragment of English

grammar parsing as a crucial component.

## 11.1 Grammar Parsing

The notion of grammar parsing is intimately connected to the notion of a formal grammar $G$ defining a formal language $L(G)$ over some (terminal) alphabet $\Sigma$. The *parsing problem* can then be defined as follows:

> Given a grammar $G$ and an input string $x \in \Sigma^*$, derive some or all of the analyses assigned to $x$ by $G$.

The analysis of formal grammars and their parsing problems goes back to the pioneering work of Noam Chomsky and others in the 1950's and continues to be a very active area of research. The most widely used formal grammar, both in computer science and in computational linguistics, is the *context-free grammar* (CFG) of Chomsky (1956). Figure 1 shows a context-free grammar defining a fragment of English including the sentence analyzed in Figure 2, which is taken from the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993).

Over the years, a variety of different formal grammars have been introduced, many of which are more expressive than the CFG model and motivated by the desire to provide a more adequate analysis of natural language syntax. This development started with the transformational grammars of Chomsky (1957, 1965) and has continued with syntactic theories like Lexical-Functional Grammar (Kaplan and Bresnan, 1982) and Head-Driven Phrase Structure Grammar (Pollard and Sag, 1994). In recent years, there has been a special interest in so-called mildly context-sensitive grammars, exemplified by Tree-Adjoining Grammars (Joshi, 1985) and Combinatory-Categorial Grammar (Steedman, 2000), which appear to strike a good balance between linguistic adequacy and computational complexity. However, there has also been considerable interest in grammars that are less expressive but more efficient, notably frameworks based on finite-state techniques (cf. Koskenniemi,

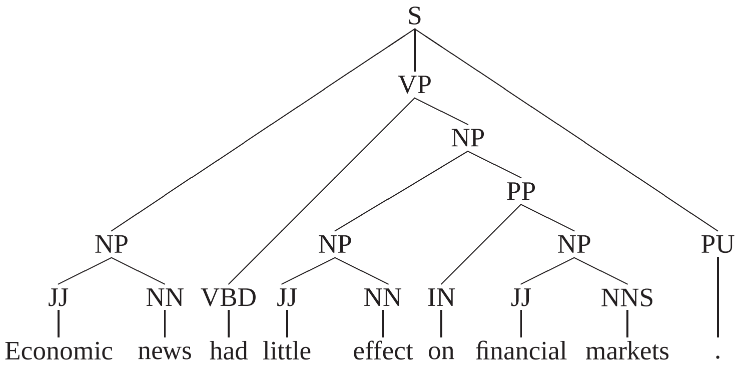

FIGURE 2 Constituent structure for English sentence

1997).

Solving the parsing problem for a specific type of grammar requires a parsing algorithm, i.e. an algorithm that computes analyses for a string $x$ relative to a grammar $G$. Throughout the years a number of different parsing algorithms for different classes of grammars have been proposed and analyzed. For context-free grammars, some of the more well-known algorithms are the Cocke-Kasami-Younger (CKY) algorithm (Kasami, 1965, Younger, 1967), Earley's algorithm (Earley, 1970), and the left corner algorithm (Rosenkrantz and Lewis, 1970). These algorithms all make use of tabulation to store partial results, which potentially allows exponential reductions of the search space and thereby provides a way of coping with ambiguity. This type of method, which constitutes a form of *dynamic programming* (Cormen et al., 1990), can also be generalized to more expressive grammar formalisms.

Traditional parsing algorithms can be described as *constructive* in the sense that they analyze sentences by constructing syntactic representations in accordance with the rules of the grammar. An alternative to this is to use an *eliminative* parsing strategy, with treats the grammar as a set of constraints and views parsing as a constraint satisfaction problem. In this approach, which is found in different forms in frameworks such as Constraint Grammar (Karlsson, 1990, Karlsson et al., 1995), Parallel Constraint Grammar (Koskenniemi, 1990, 1997), and Constraint Dependency Grammar (Maruyama, 1990), sentences are analyzed by successively eliminating representations that violate constraints until only valid representations remain.

I will make no attempt to review the vast literature on grammar parsing here but will concentrate on some general observations concerning the properties of the parsing problem and the methods used to solve it. First of all, it

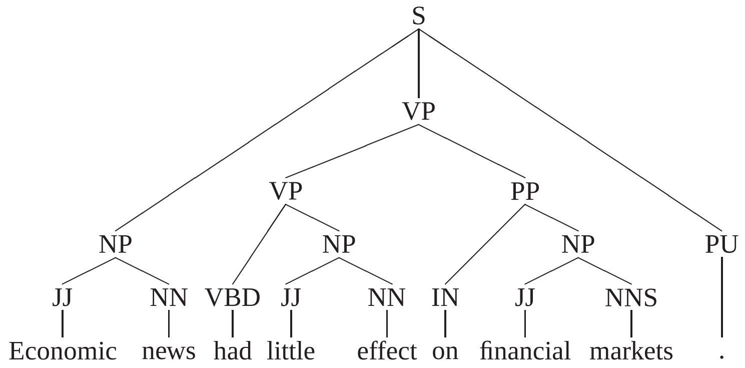

FIGURE 3 Alternative constituent structure for English sentence

is worth noting that the parsing problem for a class of grammars is a well-defined *abstract problem* in the sense of algorithm theory (Cormen et al., 1990), i.e. a relation between a set $I$ of inputs, which in this case are pairs consisting of a grammar $G$ and a string $x$, and a set $O$ of outputs, which are syntactic representations of strings in $L(G)$. A parsing algorithm provides a solution to this problem by computing the mapping from arbitrary inputs to outputs.

Secondly, the parsing problem for formal grammars is intimately tied to the corresponding *recognition problem*, i.e., the problem of deciding whether the string $x$ is in $L(G)$. It is only strings in $L(G)$ that receive an analysis in the parsing process, and most parsing algorithms in fact solve the recognition problem simultaneously.

Thirdly, we note that the analyses to be assigned to a particular input string $x$ are completely defined by the grammar $G$ itself. For example, if $G$ is a context-free grammar, we may be interested in the number of distinct parse trees that result from derivations of $x$ from the start symbol $S$ of $G$. In principle, this means that the correctness of a parsing algorithm can be established without considering any particular input strings, since the set of all input-output pairs are given implicitly by the grammar $G$ itself.

The abstract nature of the grammar parsing problem is reflected in the evaluation criteria that are usually applied to parsing methods in this context. For example, a parsing algorithm is said to be *consistent* if, for any grammar $G$ and input string $x$, it only derives analyses for $x$ that are licensed by $G$; it is said to be *complete* if, for any $G$ and $x$, it derives *all* analyses for $x$ that are licensed by $G$. For example, the grammar in Figure 1 is ambiguous and assigns to our example sentence not only the analysis in Figure 2 but also the analysis in Figure 3. Thus, a complete parsing algorithm must compute both these analyses, while a consistent algorithm must not compute any other analysis. However, both consistency and completeness can be proven without considering any particular grammar $G$ or input string $x$, given the formal definition of the class of grammars and the relevant notions of derivation and representation.

The same goes for considerations of efficiency, where proofs of complexity, either for particular parsing algorithms or for classes of grammars, provide the most relevant tools for evaluation. For a context-free grammar $G$, parsing can be performed in $O(n^3)$ time, where $n$ is the length of the input string $x$, using a dynamic programming algorithm. For mildly context-sensitive grammars, parsing complexity is still polynomial — typically $O(n^6)$ — while for more expressive formalisms running time becomes exponential in the worst case. By contrast, systems based on finite-state techniques normally support parsing in $O(n)$ time. Research on the complexity of linguistically motivated classes of grammars was pioneered by Barton et al. (1987) and has been followed by a wide range of subsequent studies.

Although complexity results often need to be supplemented by practical running time experiments, as shown for example by Carroll (1994), the role of empirical evaluation remains limited in grammar parsing, especially as far as correctness is concerned. This follows from the fact that grammar parsing is an abstract and mathematically well-defined problem, which can be studied using formal methods only.

## 11.2 Text Parsing

Text parsing[1] is concerned with the syntactic analysis of (more or less) unrestricted text. This notion of parsing therefore applies to concrete manifestations of a language $L$, where we cannot necessarily assume that $L$ is a formal language. In particular, we are of course interested in the case where $L$ is a natural language, or possibly a restricted subset of a natural language. I assume that a *text* in a language $L$ is a sequence $T = (x_1, \ldots, x_n)$ of sentences (strings) $x_i$, and I define the *text parsing* problem as follows:

> Given a text $T = (x_1, \ldots, x_n)$ in language $L$, derive the correct analysis for every sentence $x_i \in T$.

The term *sentence* should be understood in the sense of *text sentence* rather than *system sentence* (Lyons, 1977), i.e., it refers to a segment of text with-

[1] The term *text* in *text parsing* is not meant to exclude spoken language, but rather to emphasize the relation to naturally occurring language use. Although I will have nothing to say about the parsing of spoken utterances in this paper, I want the notion of text parsing to encompass both written texts and spoken dialogues. An alternative term would be *discourse parsing*, but it seems that this would give rise to misleading associations of a different kind.

out any specific assumptions about syntactic completeness or other structural properties. What constitutes a sentence in this sense may differ from one language to the other and may not always be completely clear-cut. In the context of this paper I will simply disregard this problem, although it is well-known that the problem of sentence segmentation in text processing is far from trivial (Palmer, 2000).

To exemplify the notion of text parsing, let us return again to the example sentence from Figure 2. In its original context, which is a text taken from the Wall Street Journal and included in the Penn Treebank, this sentence has an interpretation that corresponds to the analysis in Figure 2 — rather than the alternative analysis in Figure 3. Therefore, the former analysis is the one and only correct analysis in the context of text parsing.

Let us now return to the observations made with respect to grammar parsing in the previous section and see in what respects text parsing is different. First of all, it is not clear that text parsing is a well-defined abstract problem in the same sense as grammar parsing, especially not when we consider texts in a natural language. It is true that text parsing has the structure of a mapping problem, but in the absence of a formal definition for the language $L$, there is no precise delimitation of the input set. Moreover, even if we can agree on the formal properties of output representations, there is no formal grammar defining the correct mapping from inputs to outputs. For example, the syntactic representation in Figure 2 is clearly of the kind that can be defined by a context-free grammar. But according to my conception of the text parsing problem, there is no specific instance of this formal grammar that defines the mapping from input strings to specific representations.

One way of looking at the problem is instead to say that it is an empirical *approximation problem*, where we try to approximate the correct mapping given increasingly large but finite samples of the mapping relation. Needless to say, this is a view that fits very well with a data-driven approach to text parsing, but the main point right now is simply that, unlike grammar parsing, the problem of text parsing lacks a precise characterization in formal terms.

Secondly, text parsing lacks the connection between parsing and recognition that we observed for grammar parsing. This is a direct consequence of the fact that the input language is not formally defined, which means that recognition is not a well-defined problem. Therefore, we can no longer *require* that an input string be part of the language to be analyzed. In most cases, we instead have to *assume* that any text sentence is a valid input string. And if we want to be able to reject some input strings as ill-formed, then we cannot refer to a formal language definition but must appeal to some other criterion.[2]

[2] For certain practical applications, such as grammar checking, it is obviously both relevant and necessary to reject certain strings by appeal to a prescriptive grammar, but it can be prob-

Thirdly, while there is no reference to a grammar in the definition of text parsing, there is reference to a sequence of sentences providing a textual context for each sentence to be analyzed. This is based on the assumption that text parsing deals with language use, and that the analysis assigned to a sentence is sensitive to the context in which it occurs. In particular, I assume that each text sentence has a single correct analysis, even if the string of words realizing the sentence may be found with other interpretations in other contexts. In other words, text parsing entails disambiguation.

However, the absence of a formal grammar also means that we need some external criterion for deciding what is the correct analysis for a given sentence in context. For natural languages, the obvious criterion to use is human performance, meaning that an analysis is correct if it coincides with the interpretation of competent users of the language in question. This leads to the notion of an *empirical gold standard*, i.e. a reference corpus of texts, where each relevant text segment has been assigned its correct analysis by a human expert. In the case of syntactic parsing, the relevant segments are sentences and the corpus will normally be a *treebank* (Abeillé, 2003, Nivre, 2005). Thus, my reason for saying that the analysis given in Figure 2 is correct is simply that this is the analysis found in the Penn Treebank.

The use of treebank data to establish a gold standard for text parsing is problematic in many ways, having to do both with the representativity of the corpus material and the reliability and validity of the treebank annotation. And even if we can establish a gold standard treebank, it will only provide us with a finite sample of input-output pairs, which means that any generalization to an infinite language will have to rely on statistical inference. This is in marked contrast to the case of grammar parsing, where the consistency and completeness of parsing algorithms, for any grammar and any input, can be established using formal proofs.

The empirical nature of the text parsing problem is reflected also in the evaluation criteria that are applied to parsing methods in this context. Since notions of consistency and completeness are meaningless in the absence of a formal grammar, the central evaluation criterion is instead the empirical notion of *accuracy*, which is standardly operationalized as agreement with gold standard data. However, it is important to remember that, even though it is often difficult to apply formal methods to the text parsing problem itself given its open-ended nature, the parsing methods we develop to deal with this problem can of course be subjected to the same rigorous analysis as algorithms for grammar parsing. Thus, if we are interested in the efficiency of different methods, we may use results about theoretical complexity of algorithms as well as empirical running time experiments. However, for the central notion

lematic in the general case.

of accuracy, there seems to be no alternative but to rely on empirical evaluation methods, at least given the current state of our knowledge.

## 11.3 Competence and Performance

The discussion of grammar parsing and text parsing leads naturally to a consideration of the well-known distinction between *competence* and *performance* in linguistic theory (Chomsky, 1965).[3] It may be tempting to assume that grammar parsing belongs to the realm of competence, while text parsing is concerned with performance. After all, the whole tradition of generative grammar in linguistics is built on the idea of using formal grammars to model linguistic competence, starting with Chomsky (1957, 1965). The idea that natural languages can be modeled as formal languages unites theorists as different as Chomsky and Montague (1970). Within this tradition, it might be natural to view the study of grammar parsing, when applied to natural language, as the study of idealized human sentence processing.

The traditional notion of linguistic competence has recently been called into question, and it has been suggested that many of the properties typically associated with linguistic performance, such as frequency effects and probabilistic category structure, also belong to our linguistic competence (Bod et al., 2003). While the nature of linguistic competence is a hotly debated and controversial issue, it seems unproblematic to assume that text parsing is concerned with performance, at least if we want to use text parsing methods to build systems that can handle naturally occurring texts. This means that a model of linguistic competence is of use to us only if it can be coupled with an appropriate model of performance. So, regardless of whether grammar parsing is a good model of linguistic competence or not, it is still an open question what role it has to play in text parsing (cf. Chanod, 2001).

## 11.4 Conclusion

The main conclusion that I want to draw from the discussion in this paper is that grammar parsing and text parsing are in many ways radically different problems and therefore require different methods. In particular, grammar parsing is an abstract problem, which can be studied using formal methods and internal evaluation criteria, while text parsing is an empirical problem, where formal methods need to be combined with experimental methods and external evaluation criteria. In a following companion paper I will discuss different methods that have been proposed for text parsing. Some of these methods crucially involve grammar parsing; others do not.

[3] Before Chomsky, similar distinctions had been proposed by Saussure (1916), between *langue* and *parole*, and by Hjelmslev (1943), between *system* and *process*, among others.

## References

Abeillé, Anne, ed. 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers.

Barton, G. Edward, Robert C. Berwick, and Eric Sven Ristad. 1987. *Computational Complexity and Natural Language*. MIT Press.

Bod, Rens, Jennifer Hay, and Stefanie Jannedy, eds. 2003. *Probabilistic Linguistics*. MIT Press.

Carroll, John. 1994. Relating complexity to practical performance in parsing with wide-coverage unification grammars. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 287–294.

Chanod, Jean-Pierre. 2001. Robust parsing and beyond. In J.-C. Junqua and G. van Noord, eds., *Robustness in Language and Speech Technology*, pages 187–204. Kluwer Academic Publishers.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* IT-2:113–124.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton.

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

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1990. *Introduction to Algorithms*. MIT Press.

Earley, J. 1970. An efficient context-free parsing algorithm. *Communications of the ACM* 13:94–102.

Hjelmslev, Louis. 1943. *Omkring sprogteoriens grundlæggelse*. Akademisk forlag.

Joshi, Aravind. 1985. How much context-sensitivity is necessary for characterizing structural descriptions – tree adjoining grammars. In D. Dowty, L. Karttunen, and A. Zwicky, eds., *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives*, pages 206–250. Cambridge University Press.

Kaplan, Ron and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press.

Karlsson, Fred. 1990. Constraint grammar as a framework for parsing running text. In H. Karlgren, ed., *Papers presented to the 13th International Conference on Computational Linguistics (COLING)*, pages 168–173.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, eds. 1995. *Constraint Grammar: A language-independent system for parsing unrestricted text*. Mouton de Gruyter.

Kasami, T. 1965. An efficient recognition and syntax algorithm for context-free languages. Tech. Rep. AF-CRL-65-758, Air Force Cambridge Research Laboratory.

Koskenniemi, Kimmo. 1990. Finite-state parsing and disambiguation. In *Proceedings of the 6th International Workshop on Parsing Technologies (IWPT)*, pages 6–9.

Koskenniemi, Kimmo. 1997. Representations and finite-state components in natural language. In E. Roche and Y. Schabes, eds., *Finite State Language Processing*, pages 99–116. MIT Press.

Lyons, John. 1977. *Semantics*. Cambridge University Press.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19:313–330.

Maruyama, Hiroshi. 1990. Structural disambiguation with constraint propagation. In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*, pages 31–38. Pittsburgh, PA.

Montague, Richard. 1970. Universal grammar. *Theoria* 36:373–398.

Nivre, Joakim. 2005. Treebanks. In *Handbook of Corpus Linguistics*. Walter de Gruyter.

Palmer, David M. 2000. Tokenisation and sentence segmentation. In R. Dale, H. Moisl, and H. Somers, eds., *Handbook of Natural Language Processing*, pages 11–35. Marcel Dekker.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. CSLI Publications.

Rosenkrantz, D. J. and P. M. Lewis. 1970. Deterministic left corner parsing. In *Proceedings of the 11th Symposium on Switching and Automata Theory*, pages 139–152.

Saussure, Ferdinand de. 1916. *Cours de linguistique générale*. Payot.

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press.

Younger, D. H. 1967. Recognition and parsing of context-free languages in time $n^3$. *Information and Control* 10:189–208.