

## 1

---

# Two Strategies for Text Parsing

JOAKIM NIVRE

In a previous paper (Nivre, 2005) I have discussed two different notions of parsing that appear in the literature on natural language processing. The first, which I call *grammar parsing*, is the well-defined parsing problem for formal grammars, familiar from both computer science and computational linguistics; the second, which I call *text parsing*, is the more open-ended problem of parsing unrestricted text in natural language, which I define as follows:

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

The main conclusion in Nivre (2005) is that grammar parsing and text parsing are in many ways radically different and therefore require different methods. In this paper, I will concentrate on text parsing and compare two different methodological strategies, which I call the *grammar-driven* and the *data-driven* approach (cf. Carroll, 2000). To some extent, these approaches can be seen as complementary, and many existing systems combine elements of both. Nevertheless, from an analytical perspective it may be instructive to contrast the different ways in which they tackle the problems that arise in parsing unrestricted natural language text.

## 1.1 Grammar-Driven Text Parsing

In the grammar-driven approach to text parsing, a formal grammar  $G$  is used to define the language  $L(G)$  that can be parsed and the class of analyses to be returned for each string in the language. Given my

definition of text parsing, it is clear that the grammar-driven approach is based on a crucial assumption, namely that the formal language  $L(G)$  is a reasonable approximation of the language  $L$  that we want to process. In practice, we know that most if not all of the formal grammars that have been developed for natural languages to date fail to meet this assumption, and many of the research directions in natural language parsing during the last two decades can be seen as motivated by the desire to overcome these problems.

One of the hardest problems for the grammar-driven approach has traditionally been to achieve *robustness*, where robustness can be defined as the capacity of a system to analyze any input sentence. The shortcomings of grammar-driven systems in this respect can be traced back to the fact that some input sentences  $x_i$  in a text  $T$  are not in the language  $L(G)$  defined by the formal grammar  $G$ .

Theoretically speaking, it is possible to distinguish two different cases where  $x_i \notin L(G)$ . In the first case,  $x_i$  is a perfectly well-formed sentence of the language  $L$  and should therefore also be in  $L(G)$  but is not. This is sometimes referred to as the problem of *coverage*, since it should be eliminated by increasing the coverage of the grammar. In the second case,  $x_i$  is considered not to be part of  $L$ , and should therefore not be in  $L(G)$  either, but can nevertheless be analyzed syntactically. This can then be called the problem of robustness proper. However, even though there are many clear-cut examples of both kinds, there are also many cases where it is difficult to decide whether a sentence that is not in  $L(G)$  is in  $L$ , at least without making appeal to a prescriptive grammar for the natural language  $L$ .

As pointed out by Samuelsson and Wirén (2000), there are essentially two methods that have been proposed to overcome the robustness problem for grammar-driven systems. The first is to relax the grammatical constraints of  $G$  in such a way that a sentence outside  $L(G)$  can be assigned a complete analysis. The second is to maintain the constraints of  $G$  but to recover as much structure as possible from well-formed fragments of the sentence. This leads to the notion of *partial parsing*, which has been explored within a number of different frameworks such as deterministic parsing (Hindle, 1989), finite state parsing (Koskeniemi, 1990, 1997), and Constraint Grammar parsing (Karlsson, 1990, Karlsson et al., 1995).

Another major problem for grammar-driven text parsing is the problem of *disambiguation*, which is caused by the fact that the number of analyses assigned to a string  $x_i$  by the grammar  $G$  can be very large, while text parsing requires that a small number of analyses (preferably a single one) are selected as appropriate in the context of the text  $T$ .

Again, we can make a theoretical distinction between two reasons that the grammar parser outputs more than one analysis for a given string. On the one hand, we have cases of true ambiguity, i.e. where  $x_i$  admits of more than one syntactic analysis in the language  $L$ , even though only one of them is appropriate in the textual context, and where the grammar  $G$  captures this by assigning several analyses to  $x_i$ . On the other hand, it may be the case that the grammar  $G$  contains rules that license analyses for  $x_i$  that are never encountered in  $L$ . The latter problem is sometimes called the *leakage* problem, in allusion to Sapir's famous statement that '[a]ll grammars leak' (Sapir, 1921, 39). Although one might argue that it is only the former problem that relates to disambiguation proper, it is again very difficult in practice to draw a sharp distinction between problems of leakage and disambiguation.

Early work related to the ambiguity problem used specialized grammars for different domains of text, which can drastically reduce the number of analyses assigned to a given string, compared to broad-coverage grammars. Another approach is to use deterministic processing and try to ensure that, as far as possible, a correct decision is made at each nondeterministic choice point corresponding to an ambiguity (Hindle, 1989). Disambiguation can also be facilitated by the choice of parsing methodology. For example, the eliminative parsing strategy used e.g. in Constraint Grammar, where parsing consists in successively eliminating candidate analyses, integrates disambiguation into parsing.

However, the most common approach to disambiguation in recent years has been the use of statistical information about the text language  $L$  to rank multiple competing analyses ( $n$ -best parsing) or to select a single preferred analysis. There are several ways in which statistical information can be integrated into the grammar-driven approach, but the most straightforward approach is to use a stochastic extension of a formal grammar, the most well-known example being *probabilistic context-free grammar* (PCFG) (Booth and Thompson, 1973).

The problems of robustness and disambiguation cannot be studied in isolation from the problem of *accuracy*. If robustness and disambiguation have traditionally been considered the stumbling blocks for grammar-driven text parsing, it is often assumed that this approach has an advantage with respect to accuracy, since the grammar  $G$  is meant to guarantee that the analysis assigned to a sentence  $x_i$  in a text  $T$  is linguistically adequate. However, even if we disregard the leakage problem, this argument is weakened by the requirements of robustness and disambiguation. As we have seen above, robustness may require the analysis of strings that are not in the language  $L(G)$  defined by the grammar. And disambiguation normally entails discarding most of

the analyses assigned to a string by the grammar. Other things being equal, these requirements will therefore decrease the likelihood that a given string  $x_i \in X$  is assigned the contextually correct analysis by the parsing system. This means that we need to tackle the joint optimization of robustness, disambiguation and accuracy, even if we can decide to prioritize them differently.

The need for joint optimization also includes the final problem that we will consider, namely *efficiency*, which can be a more or less serious problem for the grammar-driven approach depending on the expressivity and complexity of the formal grammars used. Even if the grammar parsing problem can be solved efficiently in theory, the requirements of robustness and disambiguation can easily compromise efficiency by causing a combinatorial explosion.

## 1.2 Data-Driven Text Parsing

In the data-driven approach to text parsing, a formal grammar is no longer a necessary component of the parsing system. The mapping from input strings to analyses is instead defined by an inductive mechanism that applies to a text  $(S = (x_1, \dots, x_m))$  from the language  $L$  to be analyzed. In general, we can distinguish three essential components in a data-driven text parser:

1. A formal model  $M$  defining possible analyses for sentences in  $L$ .
2. A sample of (possibly annotated) text  $S = (x_1, \dots, x_m)$  from  $L$ .
3. An inductive inference scheme  $I$  defining actual analyses for the sentences of a text  $T = (x_1, \dots, x_n)$  in  $L$ , relative to  $M$  and  $S$ .

The first thing to note is that the formal model  $M$  may in fact be a formal grammar  $G$ , in which case permissible representations will be restricted to strings of the formal language  $L(G)$ . For example, in the standard PCFG model the permissible analyses are defined by a context-free grammar  $G$ . But it can also be a model that provides constraints on representations without defining a string language in the process, such as the robust Data-Oriented Parsing models in Bod (1998), where a permissible analysis is any parse tree that can be composed from subtrees of trees in the text sample, using leftmost node substitution and allowing the insertion of words from the input string  $x$  (even if these do not occur in the training sample).

In the previous section, I observed that grammar-based text parsing rests on the assumption that the text language  $L$  can be approximated by a formal language  $L(G)$  defined by a grammar  $G$ . The data-driven approach is also based on approximation, but this approximation is of an entirely different kind. While the grammar-based approximation in

itself only defines permissible analyses for sentences and has to rely on other mechanisms for textual disambiguation, the data-driven approach tries to approximate the function of textual disambiguation directly. And while the grammar-based approximation is an essentially deductive approach, the data-driven approach is based on inductive inference from a finite sample  $S = (x_1, \dots, x_m)$  to the infinite language  $L$ . Thus, whereas the grammar-driven approach depends on a more or less satisfactory language approximation, the data-driven approach depends on inductive inference from a more or less representative language sample. These different starting points explain why the problems of robustness, disambiguation, accuracy and efficiency may appear quite different in the two extreme approaches. Let us now proceed to an examination of these problems in the context of data-driven text parsing.

Starting with *robustness*, there is no reason that the data-driven approach should be inherently more robust than the grammar-based approach. It all depends on properties of the formal model  $M$  as well as the inference scheme  $I$  used for generalization to unseen sentences. However, it is a contingent fact about most existing data-driven systems for text parsing that these components are defined in such a way that any possible input string  $x$  is assigned at least one analysis, which means that the robustness problem is eliminated. A consequence of the extreme robustness is that these data-driven parsers will analyze strings that are not in the text language  $L$  under any characterization. If we compare this to the grammar-driven language approximation, where the robustness problem arises from the fact that some sentences in  $L$  are not in the language  $L(G)$  defined by the grammar, we can say that the data-driven approach avoids the robustness problem by a kind of superset approximation, i.e. any sentence in  $L$  is a string that can be analyzed by the parser, but not vice versa.

The problem of *disambiguation* can in many cases be even more severe in data-driven text parsing than for grammar-driven systems, since the improved robustness resulting from extreme constraint relaxation comes at the expense of massive overgeneration or leakage. However, this is to some extent compensated by the fact that the inductive inference scheme provides a mechanism for disambiguation, either by associating a score with each analysis, intended to reflect some optimality criterion, or by implicitly maximizing this criterion in a deterministic selection. The crucial problem is of course to achieve disambiguation with high *accuracy*, and the development of data-driven text parsing during the last decade has to a very large extent been driven by the desire to improve accuracy, going from the rigid PCFG model to the much richer generative probability models that represent the current

state-of-the-art (Collins, 1997, Bod, 1998). These models estimate the joint probability  $P(x, y)$  of a parse tree  $y$  for a string  $x$  based on a sample of treebank data and performs disambiguation by maximizing this probability for a given string  $x$ .

With respect to the final problem of *efficiency*, the conventional wisdom seems to be that the data-driven approach is superior to the grammar-driven approach, but often at the expense of less adequate output representations (Kaplan et al., 2004). However, in reality we find as much variation among data-driven approaches as among grammar-driven approaches, and the overall picture is in fact very similar. At one end of the scale, we find frameworks where the parsing problem is computationally intractable, such as the original data-oriented parsing models (Bod, 1998). At the other end of the scale, we find highly efficient methods that perform parsing in linear time.

### 1.3 Conclusion

The main conclusion that I want to draw from the discussion in this paper is that the partly conflicting requirements of robustness, disambiguation, accuracy and efficiency give rise to a complex optimization problem, which we can try to solve in different ways but which always requires a joint optimization. The wide variety of different methods for text parsing can to some extent be said to result from different optimization strategies and different goals.

The grammar-driven approach, in its purest form, starts from a system with optimal accuracy, in the sense that only sentences for which the correct analysis can be derived are covered, and gradually seeks to improve the system with respect to robustness and disambiguation. However, this development may compromise efficiency, which therefore has to be optimized together with robustness and disambiguation. By contrast, the data-driven approach, in its most radical form, starts from a system with optimal robustness and disambiguation, in the sense that every sentence gets exactly one analysis, and gradually seeks to improve the system with respect to accuracy. Again, this may lead to problems of efficiency, which has to be optimized together with accuracy.

My characterization of the two approaches is such that many contemporary frameworks for text parsing in fact instantiate both. It is true that we can distinguish approaches that are grammar-driven but not data-driven, such as Constraint Grammar (Karlsson, 1990, Karlsson et al., 1995), and approaches that are data-driven but not grammar-driven, such as Data-Oriented Parsing (Bod, 1998). But we also find frameworks that combine the use of formal grammars with data-driven

methods, such as broad-coverage parsers based on the PCFG model (Black et al., 1993) or on linguistically motivated frameworks such as LFG (Kaplan et al., 2004).

## References

- Black, E., R. Garside, and G. Leech, eds. 1993. *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach*. Rodopi.
- Bod, Rens. 1998. *Beyond Grammar*. CSLI Publications, University of Chicago Press.
- Booth, T. and R. Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers* C-22:442–450.
- Carroll, John. 2000. Statistical parsing. In R. Dale, H. Moisl, and H. Somers, eds., *Handbook of Natural Language Processing*, pages 525–543. Marcel Dekker.
- Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16–23.
- Hindle, D. 1989. Acquiring disambiguation rules from text. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–125.
- Kaplan, Ronald M., Stefan Riezler, Tracy Holloway King, John T. Maxwell III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 97–104.
- Karlssoon, 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.
- Karlssoon, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, eds. 1995. *Constraint Grammar: A language-independent system for parsing unrestricted text*. Mouton de Gruyter.
- 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.
- Nivre, Joakim. 2005. Two notions of parsing. In *Festschrift for Kimmo Koskenniemi*. CSLI Publications.
- Samuelsson, Christer and Mats Wirén. 2000. Parsing techniques. In R. Dale, H. Moisl, and H. Somers, eds., *Handbook of Natural Language Processing*, pages 59–91. Marcel Dekker.
- Sapir, Edward. 1921. *Language: An Introduction to the Study of Speech*. Harcourt Brace.