# A one-pass valency-oriented chunker for German

Adrien Barbaresi

# A one-pass valency-oriented chunker for German

**Adrien Barbaresi**

ICAR Lab
ENS Lyon & University of Lyon
15 parvis René Descartes, 69007 Lyon, France
adrien.barbaresi@ens-lyon.fr

**Abstract**

Non-finite state parsers provide fine-grained information but they are computationally demanding, so that it can be interesting to see how far a shallow parsing approach is able to go. The transducer described here consists in a pattern-based matching operation of POS-tags using regular expressions that takes advantage of the characteristics of German grammar. The process aims at finding linguistically relevant phrases with a good precision, which enables in turn an estimation of the actual valency of a given verb. The chunker reads its input exactly once instead of using cascades, which greatly benefits computational efficiency. This finite-state chunking approach does not return a tree structure, but rather yields various kinds of linguistic information useful to the language researcher: possible applications include simulation of text comprehension on the syntactical level, creation of selective benchmarks and failure analysis.

**Keywords:** chunking, valency, shallow parsing, finite-state automata, syntactic phrases

## 1. Introduction

### 1.1. Finite-state transducers applied to German

The idea to use finite-state automata to approximate a grammar became popular in the early nineties, following the work of Pereira (1990) among others. As Karttunen (2001) reports, after a few decades of work on more powerful grammars due to the "persuasiveness of syntactic structures", computational linguists began working again with finite-state automata. The notion of chunk parsing (Abney, 1991) has been crucial for the evolution of finite-state parsers, as well as the notion of cascaded transducers.

Especially on the application side, the fact that these automata do not yield full parses but rather a series of indications obtained faster was considered to be particularly relevant, so that the authors of the information extractor FASTUS stated that simple mechanisms can achieve a lot more than had previously been thought possible (Hobbs et al., 1997). German is not an exception, as Neumann et al. (1997) show.

The growing interest among the research community towards the parsing using finite-state transducers of unrestricted texts written in German led to the publication of several mature parsers during the last decade. Kermes and Evert (2002) as well as Schiehlen (2003) use several levels of parsing to achieve a better precision, as they most notably enable to resolve ambiguities and to check the parsing structures for correctness. The finite-state approach proved adequate to German, as Hinrichs (2005) mentions: "It turns out that topological fields together with chunked phrases provide a solid basis for a robust analysis of German sentence structure".

The mature work on FST bears useful insights on the organization of German. FST parsers do have problems with certain types of clauses for instance, which is one reason why there were primarily dismissed by the advocates of generative grammar (Müller, 2007). Since Müller's doctoral thesis in 2007, little has been done to try to provide an overview of the state of the art, which may be explained by the efficiency of the parsers.

### 1.2. Practical interest of a valency-oriented tool

Given these abilities, a less powerful approach could prove efficient when it comes to study various syntactic phenomena, by using the strengths of the FST on one hand and exploiting the irregularities in the output from natural language processing tools on the other hand, such as part-of-speech taggers, in order to detect linguistic phenomena. In fact, non-finite state parsers have been found to provide helpful features but they are computationally demanding, and it can be interesting to see how far a finite-state approach is able to go when it comes to deliver fine-grained information.

Practical applications include readability assessment, isolation of difficult parts of a text, creation of selective benchmarks for parsers based on particular syntactical asperities as well as failure analysis. The hints can be used to assess text quality and/or quality of POS-tagger output, as the valency analysis reveals the existence of sentences without verb or the lack of frequent constituents such as head nouns for instance. This proves useful in non-standard text analysis, typically in learner or web corpora. Furthermore, it can also be used in these cases to assess the syntactical difficulty of a given phrase or sentence, which is considered an important criteria in readability assessment (Dell'Orletta et al., 2011).

This approach could also encompass what Biber (1992) calls "information packaging", saying that more detectable features linked to this notion could enable fuller models of discourse complexity. In a similar effort to combine different linguistic levels to get a more precise picture of text difficulty, DellOrletta et al. (2011) deal with "parse tree depth features", like the depth of embedded complement chains and the number of verb dependents. They take over research by Pitler and Nenkova (2008) who also used parser output features to detect syntactic complexity.

Thus, the use of the by-products of such tools to derive information about a text is common among researchers. However, the parsers employed in these studies are computationally complex, which makes analysis of large corpora

dependent on time and resources. To our best knowledge it has not been tried so far to give an approximation for it produced by simplified models designed on purpose.

Our implementation of a chunk parsing method is part of annotation techniques designed to help qualify texts. More precisely, it is part of criteria which we documented in Barbaresi (2011). These cues consist in a series of approximations to more sophisticated processes that are gathered in order to provide a "reasonable" image of text complexity. They are also a possible input for decision processes in web corpus construction (Schäfer et al., 2013).

## 2. Description

### 2.1. State of the art of this processing step

Several researchers have focused on this particular step, which is most of the time integrated in more complete processing tools. In the FASTUS approach (Hobbs et al., 1997), the basic phrases are such a step, where sentences are segmented into noun groups, verb groups, and particles. Another stage dedicated to complex phrases follows, where complex noun groups and complex verb groups are identified. The authors consider that the identification problems regarding noun phrases (such as the prepositional phrase attachment problem) cannot be solved reliably, but syntactic constructs like noun groups can (i.e. the head noun of a noun phrase together with its determiners and other left modifiers).

Our approach is also comparable to the segmentation part of the Sundance shallow parser (Riloff and Phillips, 2004) as well as to shallow parsing as seen by Voss (2005): the detection of indicators of phrase structure without necessarily constructing that full structure.

### 2.2. Characteristics of valency-oriented phrase chunking

The grouping into possibly relevant chunks enables a valency detection for each verb based on topological fields, which is considered to be a productive approach of German grammar since the seminal work of Reis (1980).

The main difficulty criteria that are addressed by this approach are on the intra-propositional side the syntactic complexity of the groups (and possibly grammatically relevant phrases) and on the propositional side the complementation of the verbs as well as the topological nature of a phrase.

The transducer takes part of speech tags as input and prints as output assumptions about the composition of the phrases and about the position of the verb.

### 2.3. Characteristics of one-pass processing

Our approach aims at robustness. It takes advantage of the STTS tagset (Schiller et al., 1995), and uses the tags as they are produced by the TreeTagger (Schmid, 1994). A more precise version including number, gender and case information provided by the RFTagger (Schmid and Laws, 2008) is possible and is currently under development, nonetheless the newer tagger was significantly slower during our tests and thus it was not used for this study as it defeats the purpose of a one-pass operation in terms of computational efficiency.

The design is similar to parsers like YAC (Kermes and Evert, 2002), except that there is merely one step instead of several ones, as the program is designed to be one an indicator among others. It deals with a linear approach, the transducer takes one tag at a time without having to "look back", which accounts for computational efficiency.

The analysis relies on pattern-based matching of POS-tags using regular expressions (which are themselves finite-state automata). The patterns take into account multiple possible scenarios of tag distribution. At each state, the transducer expects certain types of tags, which allow for a change of state. If the transducer starts but does not find a given tag it comes back to its initial state and ceases to output.

It is tightly dependent on the tagger used as input, forming a sort of ecosystem with the latter that requires to build on a stabilized one, whose decisions in common situations are (at least statistically) known.

Hand-crafted rules have already been considered a noteworthy alternative to machine learning approaches (Müller, 2007). However, because of this fine-tuning, the chunker is limited to German.

### 2.4. Objectives

The purpose is neither to return a tree structure nor to deliver the best results in terms of accuracy (at least not primarily), but rather to yield various kinds of linguistic information useful to the language researcher.

The results are often comparable to text chunks, but the approach is closer to grammatical rules and to the definition of a phrase. The purpose is not to enfold every single particle, i.e. to achieve a good recall, but to find word groups that are linguistically relevant with a good precision.

We share an objective with Voss (2005), which is to approximate a part of syntactic analysis that can be done automatically with few resources and glean as much syntactic information as possible without parsing.

## 3. Implementation

### 3.1. Detection of phrases

The detection of noun phrases and prepositional phrases takes place as shown on Figure 1. Starting from POS-tags, the transducer can go through several states and add tokens to the chunk according to certain transition rules before reaching its final step, which is a common or a proper noun (respectively the tags NN or NE) that is not followed by a word which could be possibly linked to the chunk, like another noun or a tag which leads to the first state.

The detection of prepositional phrases is similar to mentioned scheme, with the main difference being the tags that allow a sequence to begin (APPRART and APPR). The head of the phrase is supposedly on the right of the group. The pattern is greedy: everything that fits under a predefined composition of a phrase counts. While this is a design decision that makes the implementation easier, it does not always perform well.

The chains of probable tags produced by the tagger as part of its operational design enable pattern analysis, which

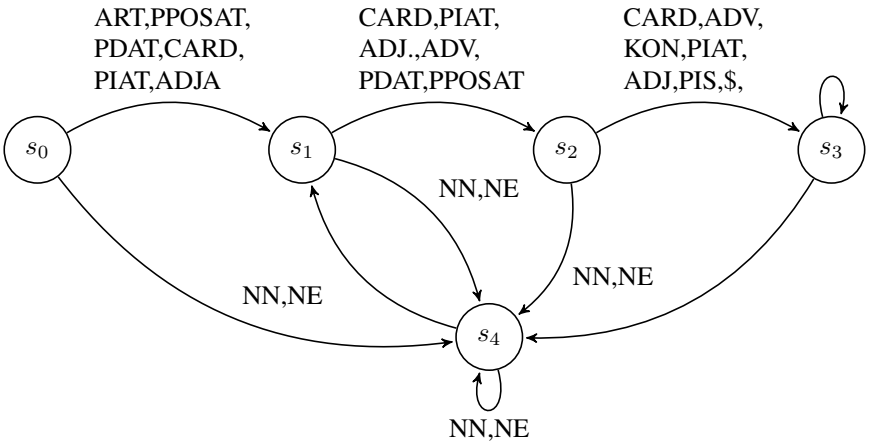


Figure 1: Simplified pattern used for detecting noun phrases on the basis of POS-tags using the STTS tagset (Schiller et al., 1995). The additional APPRART and APPR tags are required to initiate the detection of prepositional phrases.

is based on known syntactical and grammatical rules, simple, well-known patterns, which as such are very likely to give satisfying results.

Thus, the constitution of the surface parser leaves little room for false incorporations, though abusive statements are not prohibited by design. Nonetheless, there is little chance of seeing incoherent output of the parser, since it takes benefit of the analysis by chains done by the tagger. The analysis of the tag probabilities given by the TreeTagger show there are two main cases: either it is quite confident about its output, or it fails at determining a reliable tag, which often affects several tags in a row. When the parser is confronted with such unusual tag chains, it ceases to output.

### 3.2. Actual valency

The purpose is to benefit from the detection mentioned above to give an estimation of the number of arguments that may be syntactically connected to a given verb. In order to do so, there are two operations needed, one on the extra-clausal level and the other one on the intra-clausal one.

First, one has to find the boundaries of the clauses, since the sentence is not a relevant unit. In the case of German, this can often be done by locating the commas, as clauses are very frequently delimited by them, provided that they are not part of enumerations. Then, each head of a chunk found in a given clause increments the actual valency variable.

Due to the greediness of the phrase detection, the value is rather under- than overestimated, which could prove interesting when it comes to comprehension level assessment. In fact, the estimated valency is most of the time between 2 or 5 complements per verb, which confers to a value of 5 or more a decisive character. In fact, this order of magnitude indicates that the sentence is bound to have a complex structure.

### 3.3. Proof of concept and adaptability

The transducer was first implemented to use it as a proof of concept, as a standalone part of a text enrichment workflow. The code for this specific part has been made available online under an open source license[1]. As it is based on a series of conditional statements, IF-ELSIF-ELSE loops following the structure roughly pictured in Figure 1, it can be easily translated to another programming language. Other constraints can also easily be added. All statements can also be used in finite-state formalisms. The main dependency in terms of tools are the tagger and the tagset.

It is conceivable to use a “flat approach” of this issue by using one regular expression containing all the plausible scenarios and applying it directly to a whole text, a whole series of tags to match the candidates. Due to the computational complexity of long strings and multiple OR constraints, which is sometimes deteriorated by automata implementation issues of programming languages (Cox, 2007), this approach was not used for this study. The decomposition of the pattern and the use of a finite-state transducer benefits greatly to the processing speed as well as to the modularity of the analysis.

### 3.4. Example

The output of the finite-state transducer is shown in Figure 2 just under the text (”R” indicates that a ”greedy” right extension pattern was matched, the numbers indicate the state of the finite-state automaton as described in Figure 1), then the valency number guessed by the chunker. Finally, the numbers in bold font show the theoretically expected output.

The enumeration at the beginning is a problem, as it makes the proper identification of the base of the valency complementation a lot more difficult. The chunker fails at it but still manages to count one complementation and not more, for instance because the commas are used as a hint to detect an enumeration. This guess is false from a linguistic point of view, but by design a better precision cannot be achieved in those cases, as the end of the phrase comes unexpectedly late in the processing flow. This first part also illustrates the left-to-right parsing of the syntactic

[1] https://github.com/adbar/valency-oriented-chunker

| Überfüllte | Einzimmerbehausungen | , | moderne | Apartments | oder | Kolonialvillen | im | französischen | Viertel | – | der |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NP0 | NP3 | | NP0 | NP3 | NP3-R | NP3-R | PP0-R | PP1-R | PP3-R | | NP0 |
| | 1 | | | 1 | | | | | | | |
| | | | | | | **1** | | | | | |

| Fotokünstler | Hu | Yang | versucht | mit | seinen | Bildern | , | möglichst | viele | Facetten | seiner | Heimatstadt | einzufangen | . |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NP3 | NP3-R | NP3-R | VP | PP0 | PP1 | PP3 | | | NP0 | NP3 | NP1-R | NP3-R | VP | |
| 2 | | | | | | 3 | | | | 1 | | | | |
| **1** | | | | | | **2** | | | | **1** | | | | |

Figure 2: Example of the chunker output: Sentence text at the first level, the phrases being underlined, chunker output at the second level and valency counter at the third. The gold standard is at the fourth level in bold font. NP, PP and VP are phrase types, the numbers are states described in Figure 1. The letter R implies that an extension on the right as been detected.

components, which could be overridden by a second pass. In this case, it is clear that one trades accuracy against this kind of robustness by adopting the one-pass approach.

The sequence starting with a dash shows a further problem, because in this case the counter should be reset. The noun phrases are identified properly but the valency-complementation values are false.

The last part of the sentence is tagged properly, it shows the ability of the chunker to avoid issues link to the extensions on the right of the noun phrases (proper name and genitive adjuncts), to reset the counter at the beginning of a subordinate clause and to deal with discourse markers.

## 4. Evaluation

### 4.1. Large-scale analysis

Several grammatical particles are not taken into account, such as illocutionary and modal particles, adverbial portions of phrasal verbs and connectors.

So, in order to evaluate the performance of the chunker, one can compute the ratio between the amount of tags that are concerned by the analysis and the amount of tokens for which there is no output. There are no evaluation metrics for the actual valency detection so far, since it is still an experimental feature which relies heavily on other processes.

The corpus used for evaluation consists of 2,416 recent online articles of the German version of the *Geo* magazine[2], comprising a total of 838,790 tokens. There are 469,655 non-verbal tokens for which there is an output and 234,120 verbal tokens (not only verbs but also modifiers like conjunctions or verbal particles) about which the transducer made a statement. Without the punctuation marks (representing 92,680 tokens according to the tags produced by the TreeTagger), that lets about 6 % of the tokens that are possibly words without possible connections.

As already mentioned, the efficiency of the chunker regarding the particles it takes into account is interesting: 547,686 non-verbal tokens in total had a chance to be analyzed, which means that about 86 % of these tokens where considered to be part of a grammatically relevant chunk. If about 14 % of the tags were not incorporated, that means this information could be used to detect difficulties.

[2]http://www.geo.de

The cases for which there is no output are particularly interesting when it comes to text comprehension: if a grammatical structure is not recognized, then it may be a rare form or an error of the tagger. Both could be linked, and both are relevant as a source of processing difficulty by humans or by machines. It can also mean that the structure is particularly long and/or complex, which is also relevant.

This information can also be used in order to isolate difficult parts of a text to compare the existing finite-state parsers, from which it is known that center embedding in noun phrases for instance is a problem (Müller, 2007), as well as recursion issues.

### 4.2. Evaluation in detail

In order to give more precise insights on the performance of the chunker, its output has been evaluated on three different samples of 1,000 tokens in a row extracted from the corpus. The samples comprised a total of 180 sentences spread across eight different articles. The chunker found 831 valency complementations. 95 structures were falsely counted as valency elements, of which the noun phrase was correctly parsed but not numbered properly in 44 cases. 87 relevant structures were missed. Thus, the efficiency in terms of recall is slightly below 90 % on the test samples. The numeration accuracy is around 87 % and the F-measure for the values below is .890.

| Output | Errors | Missed | Precision | Recall |
|---|---|---|---|---|
| 831 | 95 | 87 | .886 | .894 |

### 4.3. Possible improvements

The close evaluation made clear that there are two kinds of problems: those related to linguistics and those related to language processing issues.

On one hand, the impact in terms of valency of reflexive pronouns could be more adequately addressed. Trickier problems arise when loosely defined word categories come into focus, such as discourse markers, whose importance cannot be automatically verified using substitution tests. The task consisting of defining annotation guidelines based on acknowledged word categories in the field of linguistics is a challenge by itself.

On the other hand, a substantial part of the errors deal with tokenization and tagging artifacts such as falsely annotated URL components or punctuation issues. In fact, it

is crucial in this one-pass approach to defined a range of possible clause boundaries, from quotes to commas and to indirect speech markers, as it could improve precision.

## 5. Conclusion

A one-pass chunking and valency detection transducer has been presented. It is mainly linear and uses a bottom-up linguistic model implemented using finite-state automata, which allows by design for a fast processing speed. As such, it satisfies the constraints to work with large corpora.

Although design decisions can account for missing or false results in some cases, evaluation shows that this trade-off seems to be justifiable. There was an existing output for 86 % of the tokens in our corpus, and the valency counter's guesses are correct in 87 % of the cases. The first figure reveals that the chunker is quite permissive, whereas the latter shows that its accuracy is acceptable. A possible application of this tool is precisely what both metrics do not show: what it could not integrate or analyze successfully, as this enables to focus on complex phrases or sentences as well as on irregularities in a corpus.

Future work includes three main topics of interest: first an error analysis concerning on one hand the integration of certain grammatical particles and on the other hand non-standard text-genres and tokenization artifacts. Second the integration of more precise morphosyntactic information which could enable a fine-grained analysis of the right extensions of the noun phrase, like genitive forms following nouns. The third topic of interest deals with metrics for actual valency detection, as the number of verbal dependents could be a highly relevant factor.

## 6. References


Steven P. Abney. 1991. Parsing by chunks. *Principle-based parsing*, 44:257–278.

Adrien Barbaresi. 2011. Approximation de la complexité perçue, méthode d'analyse. In *Actes TALN'2011/RECITAL*, volume 2, pages 229–234, Montpellier, France.

Douglas Biber. 1992. On the complexity of discourse complexity: A multidimensional analysis. *Discourse Processes*, 15(2):133–163.

Russ Cox. 2007. Regular Expression Matching Can Be Simple And Fast (but is slow in Java, Perl, PHP, Python, Ruby, ...). http://swtch.com/ rsc/regexp/regexp1.html.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2011. READ–IT: Assessing Readability of Italian Texts with a View to Text Simplification. In *Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies*, pages 73–83, Edinburgh. Association for Computational Linguistics.

Erhard W. Hinrichs. 2005. Finite-State Parsing of German. In Antti Arppe and et al., editors, *Inquiries into Words, Constraints and Contexts*, pages 35–44. CSLI Publications, Stanford.

Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. *Finite-State Language Processing*, pages 383–406.

Lauri Karttunen. 2001. Applications of Finite-State Transducers in Natural Language Processing. In S. Yu and A. Paun, editors, *CIAA 2000, LNCS 2088*, pages 34–46. Springer, Heidelberg.

Hannah Kermes and Stefan Evert. 2002. YAC – A Recursive Chunker for Unrestricted German Text. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, volume 5, pages 1805–1812.

Frank Henrik Müller. 2007. *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. Ph.D. thesis, University of Tübingen.

Gnter Neumann, Rolf Backofen, Judith Baur, Markus Becker, and Christian Braun. 1997. An Information Extraction Core System for Real World German Text Processing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 209–216.

Fernando Pereira. 1990. Finite-state approximations of grammars. In *Proceedings of the Annual Meeting of the ACL*, pages 20–25.

Emily Pitler and Ani Nenkova. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195.

Marga Reis. 1980. On justifying topological frames: 'Positional field' and the order of nonverbal constituents in German. *DRLAV*, 22(23):59–85.

Ellen Riloff and William Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical report, School of Computing, University of Utah.

Roland Schäfer, Adrien Barbaresi, and Felix Bildhauer. 2013. The Good, the Bad, and the Hazy: Design Decisions in Web Corpus Construction. In Stefan Evert, Egon Stemle, and Paul Rayson, editors, *Proceedings of the 8th Web as Corpus Workshop*, pages 7–15.

Michael Schiehlen. 2003. A Cascaded Finite-State Parser for German. In *Proceedings of the 10th conference of the EACL*, volume 2, pages 163–166.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1995. Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Draft, Universities of Stuttgart and Tübingen.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 777–784.

Helmut Schmid. 1994. Probabilistic Part-Of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, volume 12.

Matthew J. Voss. 2005. Determining syntactic complexity using very shallow parsing. Master's thesis, CASPR Research Report, Artificial Intelligence Center, University of Georgia.