Cheetah: Developing a German HPSG grammar from the Tiger treebank

Bart Cramer

Saarland University, Germany

July 2009, DELPH-IN Annual Meeting, Barcelona

Outlook

- Treebank-based PCFG parsers have high coverage, and large statistical models can be built from them.
  (Charniak and Johnson, 2005)
- Hand-written deep grammars include more linguistic information, but are also more susceptible to robustness issues.
  (Flickinger, 2000; Butt et al., 2002)
- Recent efforts to combine these approaches have been very successful, but hinge heavily on specific properties of English.
  (Hockenmaier and Steedman, 2002; Miyao, Ninomiya, and Tsujii, 2004; Cahill et al., 2004)
- So: let's try it on German.

Deep grammar extraction

- Deep grammar extraction is the process of deriving a deep grammar from annotated material.
- Key in this process is the conversion from a source treebank (e.g. the Penn Treebank) to a target formalisms (e.g. HPSG, CCG, LFG), before the derivation process can take place.
- The amount of information in the source treebank (loosely) determines the depth of the resulting grammar.

German language

Some characteristics that make German a harder language to parse:

- Richer morphology (case system, noun compounding)
- More long-distance dependencies
- A more free word order:

a. Der Präsident hat gestern das Buch gelesen.
The.NOM President.NOM has yesterday the.ACC book.ACC read.PERF.
'The president read the book yesterday'

b. Gestern hat der Präsident das Buch gelesen.

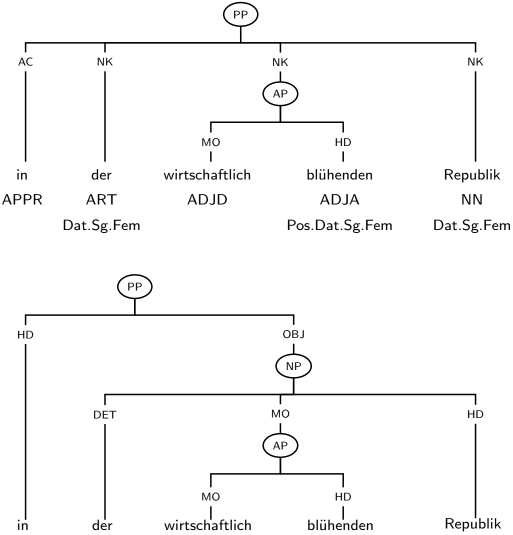
c. Das Buch hat der Präsident gestern gelesen.

Tiger treebank

(Brants et al., 2002)

- The Tiger treebank is a dependency treebank, consisting of just over 50K sentences of newspaper text (17.6 w/s).
- It allows crossing branches (33% of sentences), mostly for object/modifier fronting and extraposed relative clauses or comparatives.
- The annotation includes morphological properties and syntactic functions.

Tiger treebank



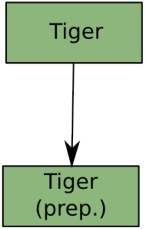
Differences

- Compared to previous deep grammar extraction methods, this method is deeper:
  - The core grammar is more elaborate, with detailed analyses for the German word order, coordinations, direct speech, etc.
  - The lexicon derivation algorithm will pick up a larger number of linguistic facts, with a higher degree of abstraction.
  - It contains a core lexicon for the most frequent and/or semantically marked lemmas.
- The source treebank (Tiger) is not converted to HPSG. Instead, the lexicon is read off directly.

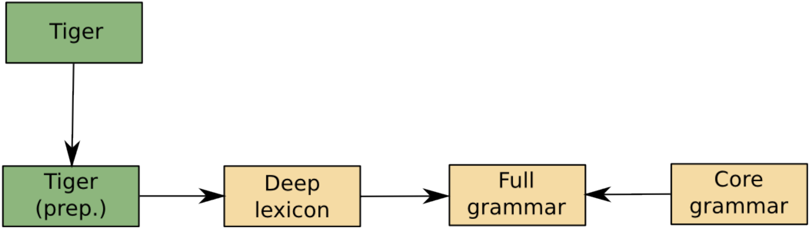
Core grammar

- The core grammar currently supports basic grammatical constructions: Mittelfeld scrambling, relative phrases, and direct speech. It consists of:
    - Rules: 56 rules, almost half of them coordination.
    - Lexicon: around 720 lexical items.
- No content words → no coverage.
- No morphology → each word form receives its own lexical item.

Workflow



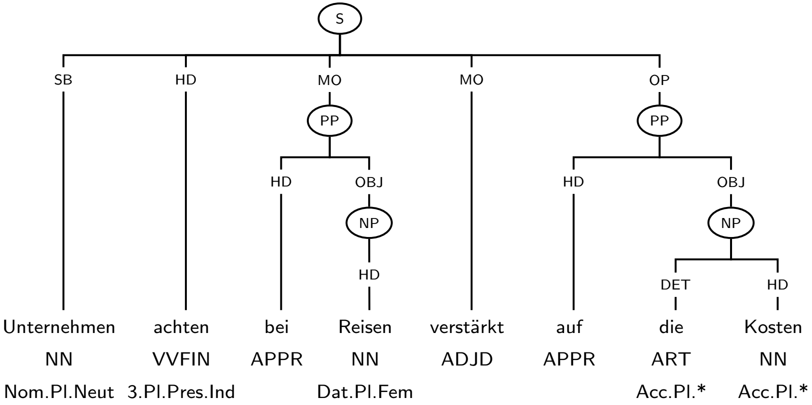
Workflow



Derivation procedure

- Traverse the graph top-down.
- For each node:
    - Identify the node's head (or the deepest verb in the verb cluster);
    - For each complement of this node, add this complement to the head's subcategorisation frame.
    - For each modifier, add this head to the possible MOD values of the modifier's head.
- For each lexical item, a mapping of (lemma, morphology) $\rightarrow$ word form is created.

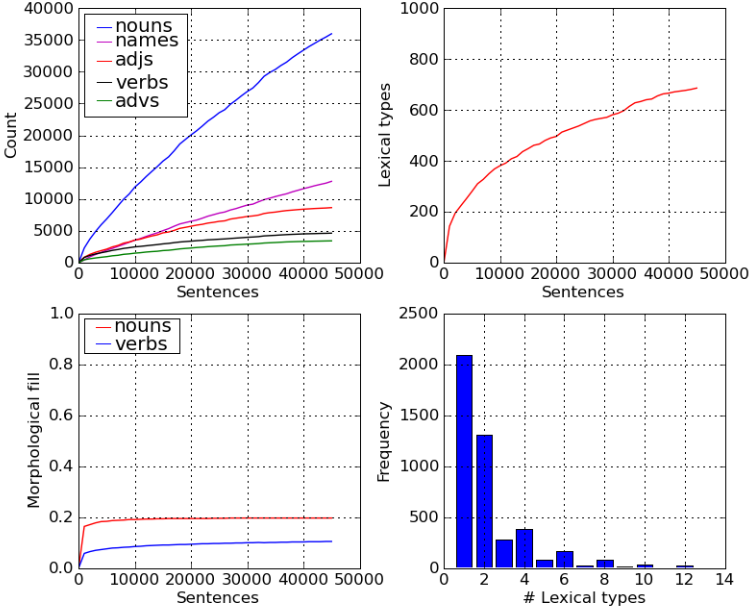
Derivation procedure



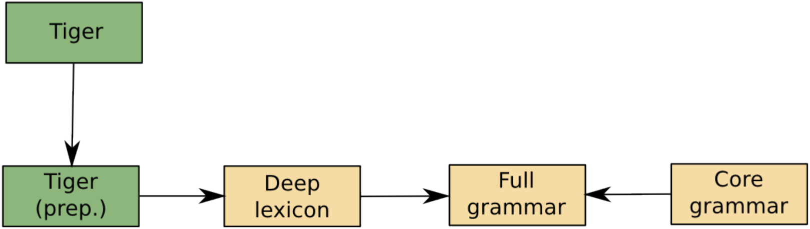
The companies look-after by travels stronger at the costs.

The companies watch the travel costs more closely.

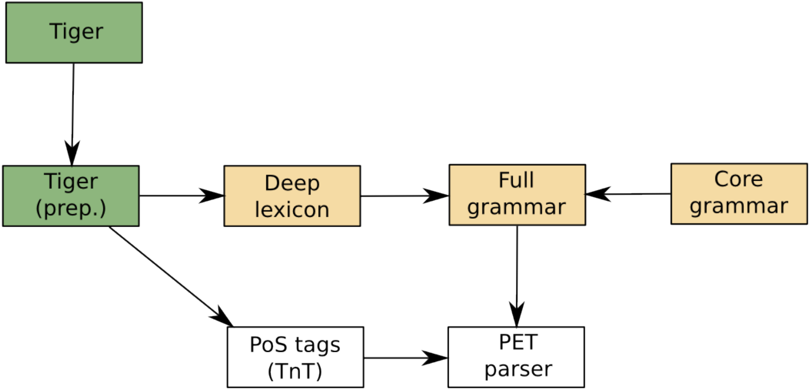
Derivation results



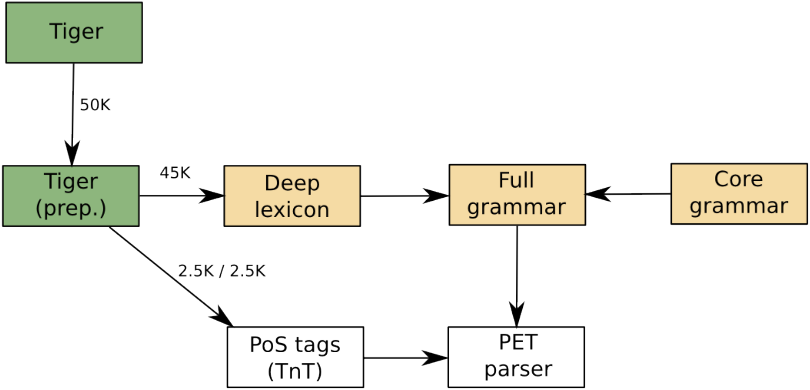
Workflow



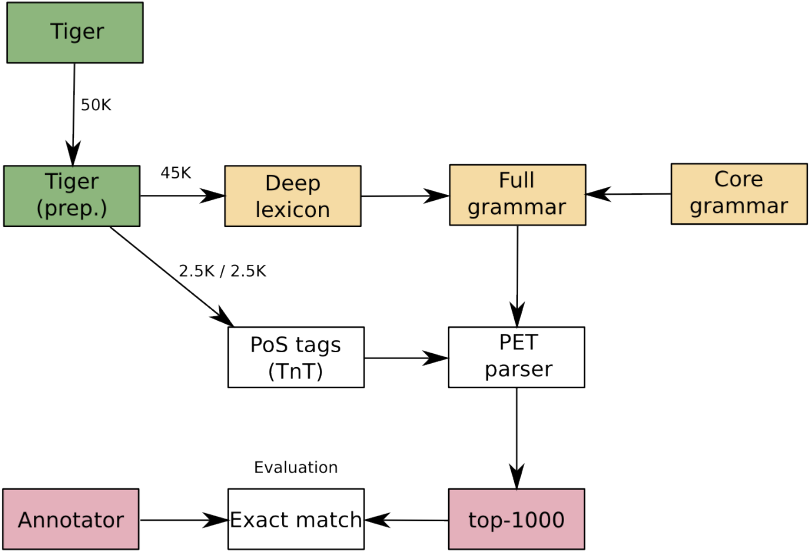
Workflow


Tiger
Tiger (prep.)
Deep lexicon
Full grammar
Core grammar
PoS tags (TnT)
PET parser

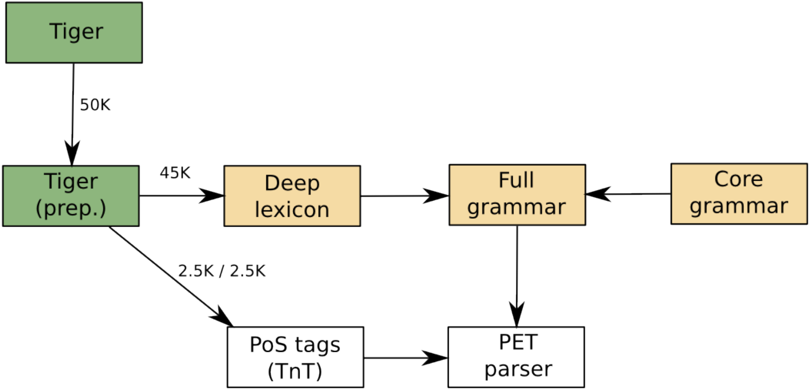
Workflow


Tiger
50K
Tiger (prep.)
45K
Deep lexicon
Full grammar
Core grammar
2.5K / 2.5K
PoS tags (TnT)
PET parser

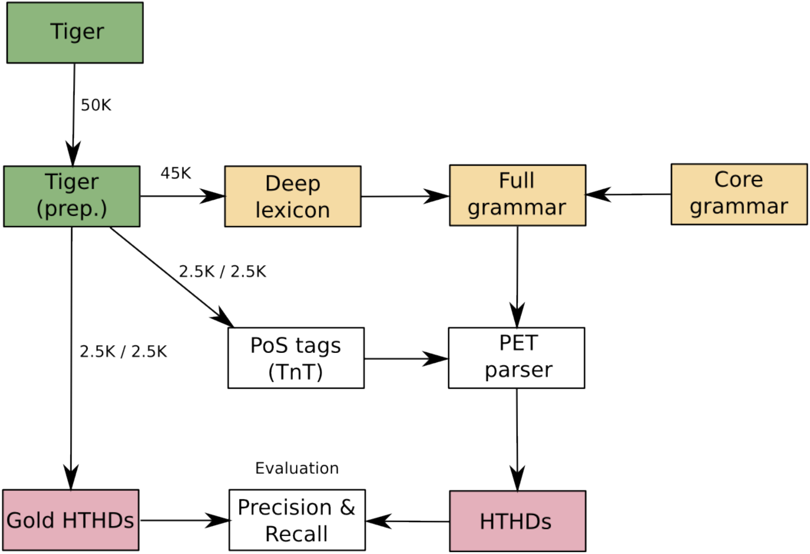
Workflow


Tiger
50K
Tiger (prep.)
45K
Deep lexicon
Full grammar
Core grammar
2.5K / 2.5K
PoS tags (TnT)
PET parser
Evaluation
Annotator
Exact match
top-1000

Workflow


Tiger
50K
Tiger (prep.)
45K
Deep lexicon
Full grammar
Core grammar
2.5K / 2.5K
PoS tags (TnT)
PET parser

Workflow


Tiger
50K
Tiger (prep.)
45K
Deep lexicon
Full grammar
Core grammar
2.5K / 2.5K
PoS tags (TnT)
PET parser
2.5K / 2.5K
Evaluation
Gold HTHDs
Precision & Recall
HTHDs

Results

- 46% of the development set received at least one parse.
- The disambiguation model was trained on 200 hand-annotated sentences.
- In total, the f-score on labeled head-to-head dependencies was 0.303.
- This number was 0.591 for the covered sentences.

Disambiguation models

- It is clear that the previous disambiguation model is insufficient.
- With no immediate training data available, we have to create it:
  - We parsed the training set with Cheetah, and recorded the top-500 parses.
  - For each of the candidate parses for a given sentence, we compute the head-to-head dependency f-score f_i.
  - If the best scoring f-score $f_{max} > \beta$, we used that candidate parse as training material. Otherwise, reject all parses.
- Initial experiments boosted f-scores from 0.60 to 0.65 (roughly). Best results were obtained with low β.

Current state

- Cheetah certainly gained linguistic relevance compared to previous deep grammar extraction approaches. However, coverage on unseen text seems to be prohibitively lower.
  - Use heuristics or hand-written rules to improve on mophological fill of the lexicon.
  - Expand the core grammar to include more common grammatical constructions in German.
  - Apply more advanced robustness techniques.

Current state

- Cheetah certainly gained linguistic relevance compared to previous deep grammar extraction approaches. However, coverage on unseen text seems to be prohibitively lower.
  - Use heuristics or hand-written rules to improve on mophological fill of the lexicon.
  - Expand the core grammar to include more common grammatical constructions in German.
  - Apply more advanced robustness techniques.
- And then?

Scenario 1: Focus on syntax

- So far, the grammar was an attempt to recreate the syntactic dependencies. Because TiGer is a syntactic treebank, making a syntactic grammar was more straightforward.
- The MRS representation is abused: each word receives one relation, qeqs are not used, no modeling of scope, no cross-lingual predicates are used.
- The competition is tough: data-driven dependency parsers (MALT, MST) are probably better in doing this (But also cross-domain? And do we find things DDDPs never find?).
- We might be able to beat the DCU parser for German, though.

Scenario 1: Open-world parsing

- It implements the idea of the open-world assumption, as advocated by Johnson on the last EACL, and will give parses (with coherent output) for almost all possible inputs.
- The idea is to create some robustness rules (possibly supertypes of existing rules). Instead of using a fallback strategy (two-phase strategy), all rules will have the same status (integrated strategy).
- The PET parser has to be changed to make it possible to restrict the complete search space, e.g. using a beam search. The use of robustness rules will then be dispreferred (so often not carried out) by the scoring model.

Scenario 2: Focus on depth/semantics

- The goal here is to have a symbiotic relationship between the hand-written and inferred parta. The emphasis is more on depth.
- Evaluation of this scenario is hard; at least more qualitative. Maybe realisation? A secondary evaluation might be possible by deriving unlabelled dependencies form the parse tree.
  - How deep can such a grammar become?
  - Can we find linguistic generalisations that are hard to find by introspection?
  - Can we test linguistic hypotheses on a larger scale, because of the (hypothesized) better coverage on unseen text?

Scenario 2: Division of labour

- As opposed to relying on hand-writing, we imagine a workflow dependent on the principle of subsidiarity: automate whatever is possible.
- We identify the following areas, running from linguistic/labour-intensive to data-driven, to be developed in this order:
  - Core grammar: hand-written.
  - Modules/parameters: hand-written, but customised.
  - Annotation-driven: derivation is hand-written, but part of the structure is generated automatically.
  - Data-driven: the only human intervention is the bias in the algorithm.
- The academic question to be answered would then be: where are logical boundaries for these regions? To what extent can we push the discovery of phenomena downwards this list?

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

M. Butt, H. Dyvik, T.H. King, H. Masuichi, and C. Rohrer. 2002. The parallel grammar project. In *International Conference On Computational Linguistics*, pages 1–7.

A. Cahill, M. Burke, R. ODonovan, J. Van Genabith, and A. Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Meeting of the ACL*, pages 320–327.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics Morristown, NJ, USA.

D. Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

J. Hockenmaier and M. Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third LREC Conference*, pages 1974–1981.

Y. Miyao, T. Ninomiya, and J. Tsujii. 2004. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proc. IJCNLP*.