

Acquiring a wide-coverage lexicalized grammar from Penn Treebank

Yusuke Miyao
University of Tokyo

Motivation

- Difficulty of deep parsing of real-world text
- Bottleneck: difficulty in the development of wide-coverage grammars
 - Scaling up grammars to process real-world text requires more than a decade
- Necessity of a methodology for efficient grammar development

Topic of this talk

- Corpus-oriented development of an HPSG grammar
 - The principal aim of grammar development is **treebank construction**
 - Penn Treebank is converted into an HPSG treebank
 - A lexicon is extracted from the HPSG treebank
 - CCG [Hockenmaier et al. 2002], HPSG [Miyao et al. 2004]
- Parsing models (by the other speakers)
 - Parsing algorithms
 - Disambiguation models

Background: HPSG

- HPSG is a syntactic theory to explain generic regularities that underlie phrase structures, lexicons, and semantics [Pollard & Sag 1994]
- Two components of HPSG:
 - **Lexical entries** represent word-specific constraints
 - **Grammar rules** express generic grammatical regularities

Background: HPSG parsing

- Lexical entries determine syntactic/semantic constraints of words

Lexical entries

[HEAD *noun*
SUBJ <>
COMPS <>]

John

[HEAD *verb*
SUBJ <HEAD *noun*>
COMPS <HEAD *noun*>]

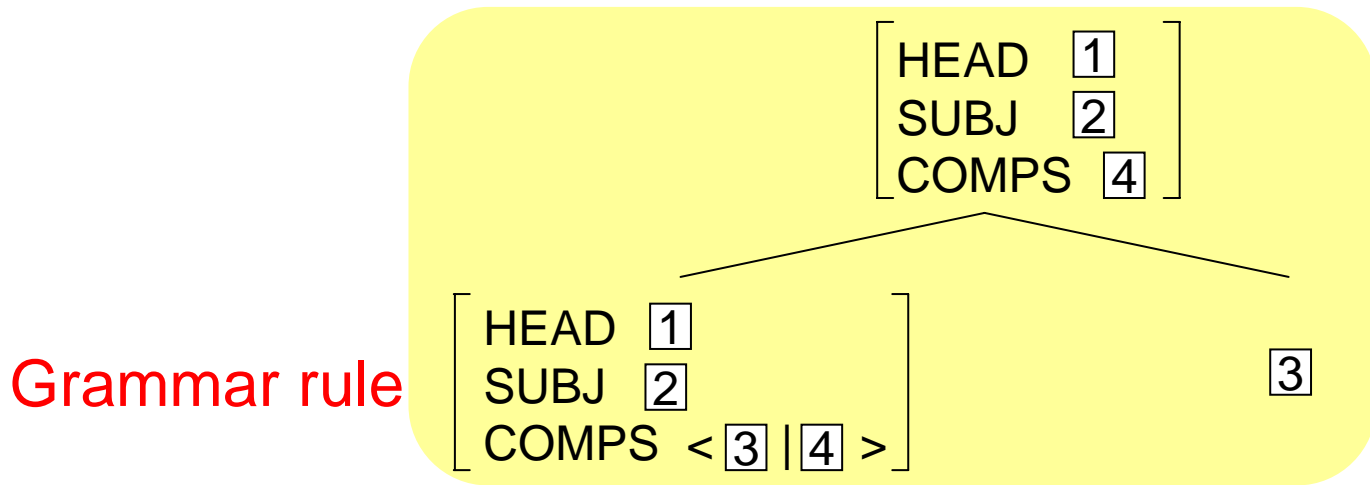
saw

[HEAD *noun*
SUBJ <>
COMPS <>]

Mary

Background: HPSG parsing

- Grammar rules determine generic constraints of grammar (not limited to construction rules)



[HEAD *noun*
SUBJ <>
COMPS <>]

John

[HEAD *verb*
SUBJ <HEAD *noun*>
COMPS <HEAD *noun*>]

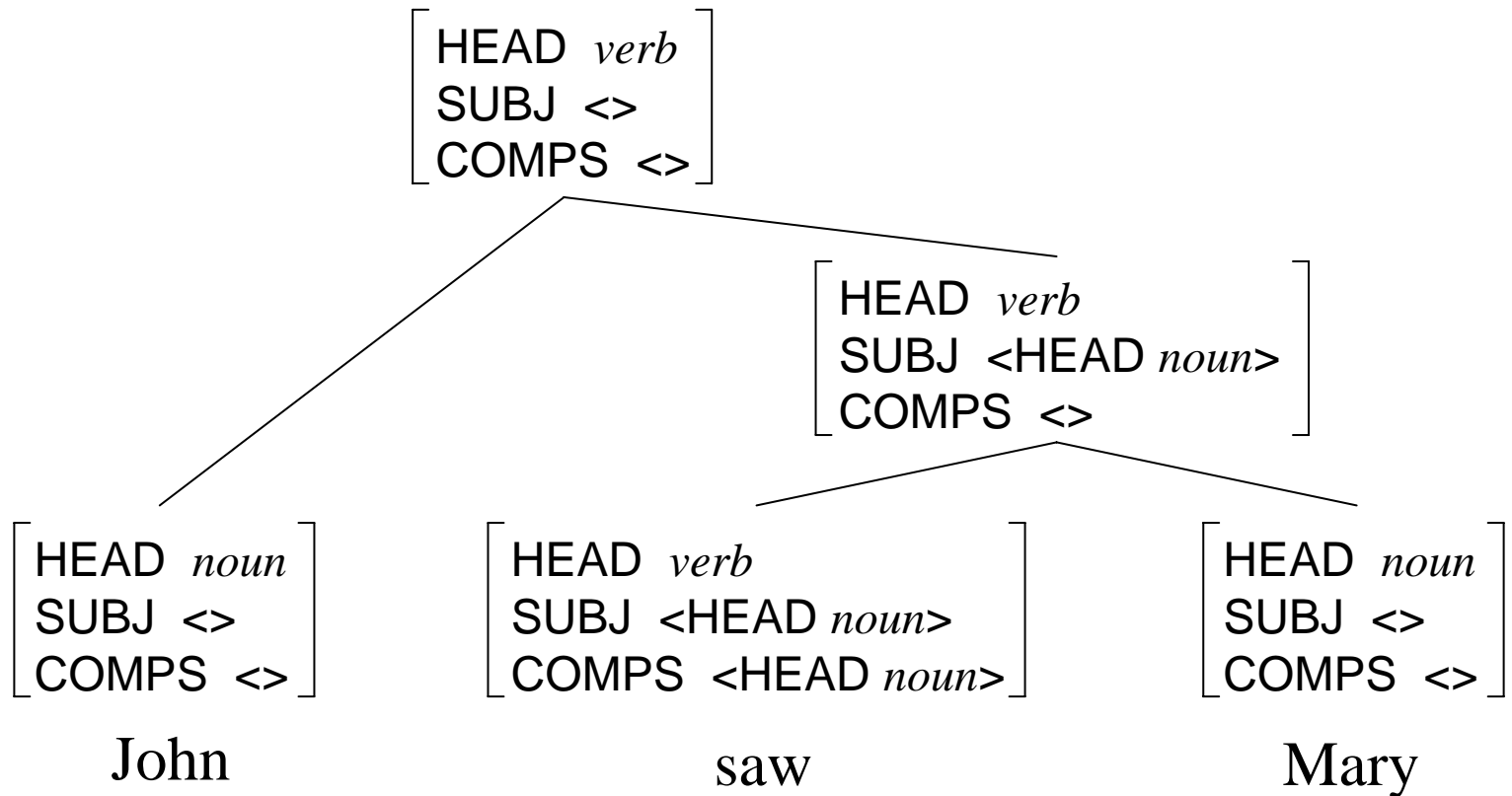
saw

[HEAD *noun*
SUBJ <>
COMPS <>]

Mary

Background: HPSG parsing

- Grammar rule applications produce syntactic/semantic structures of sentences

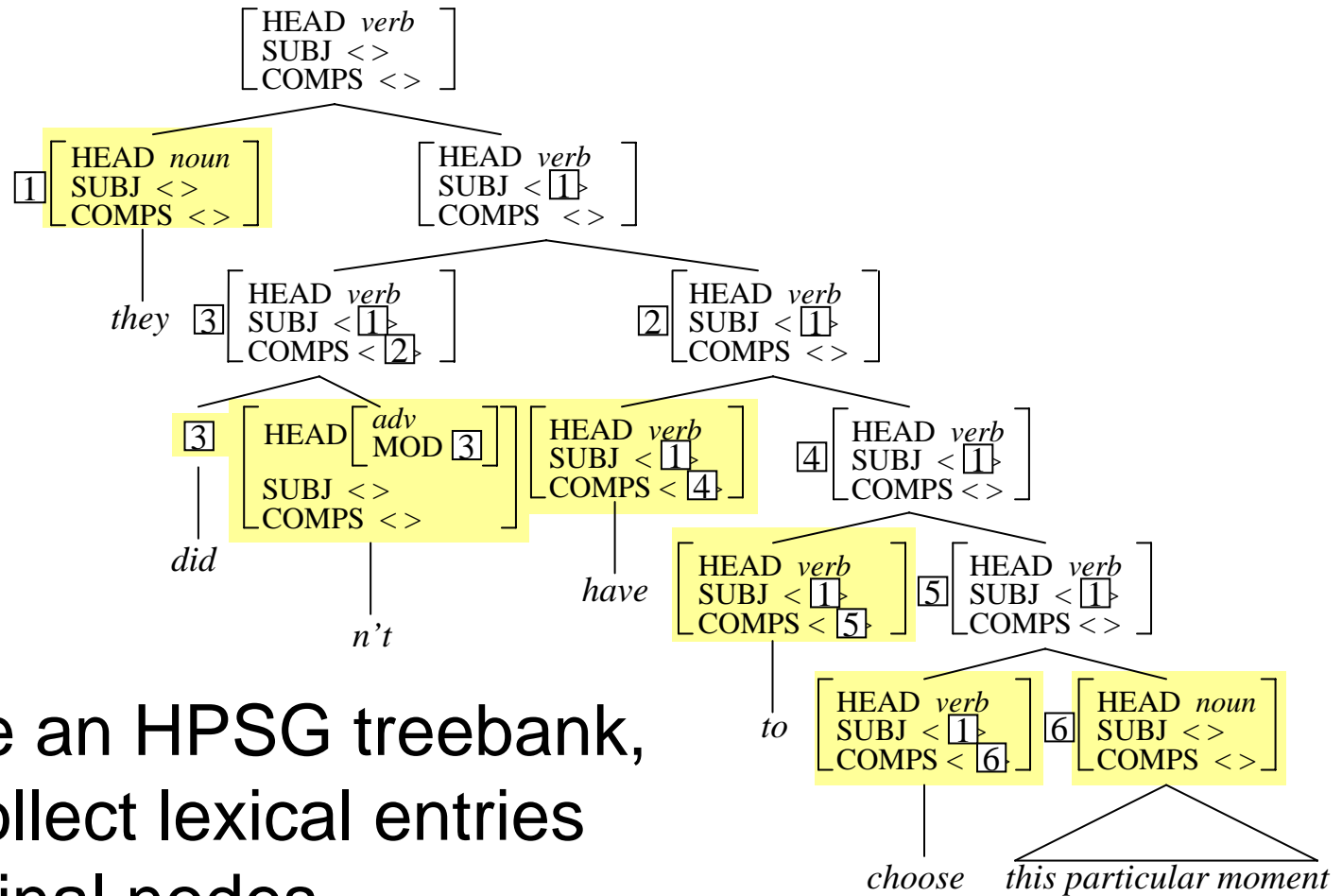


Requirements

- For HPSG parsing, we require:
 - Grammar rules
 - Lexical entries
 - Treebank
 - For statistical modeling
 - For grammar testing

What is the fastest way to the development of these three resources?

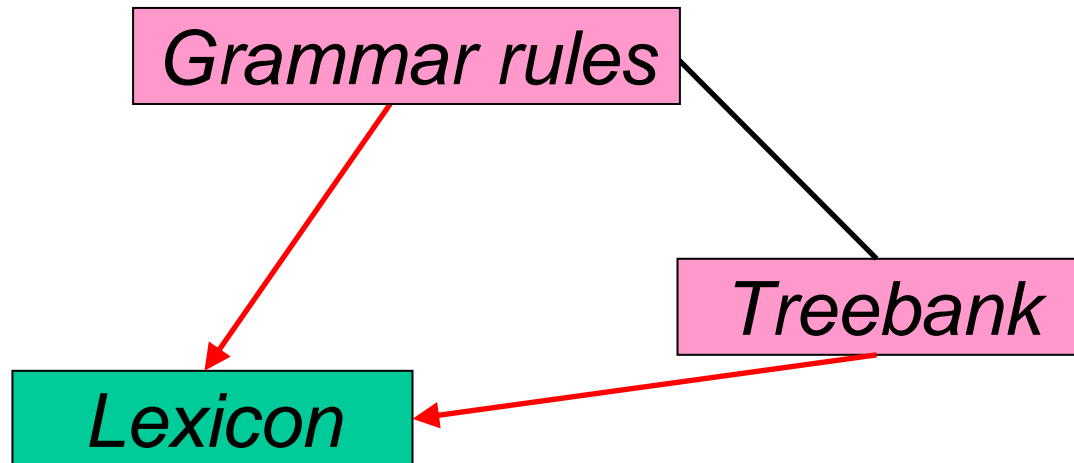
Treebank > Lexicon



- If we have an HPSG treebank, we can collect lexical entries from terminal nodes

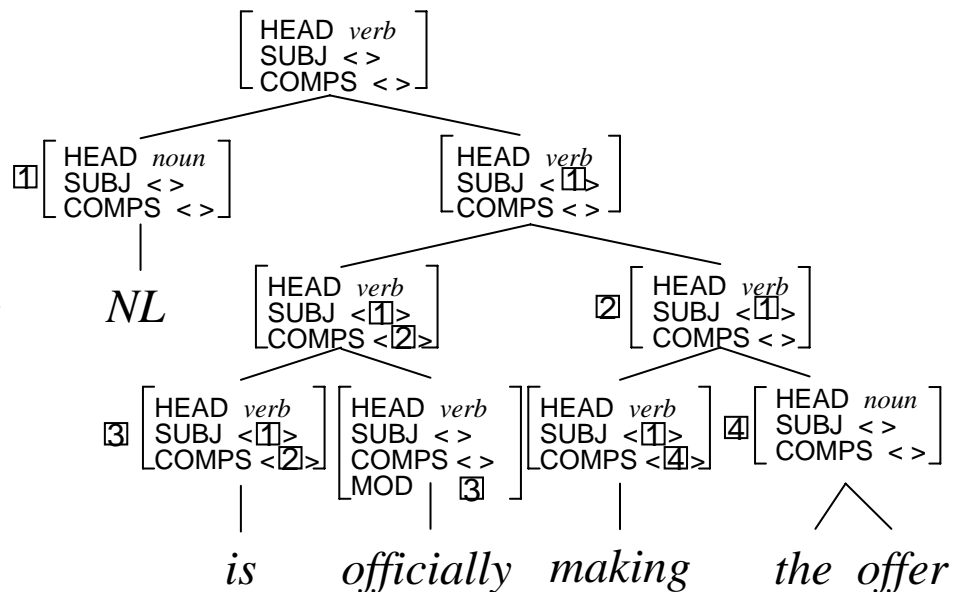
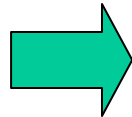
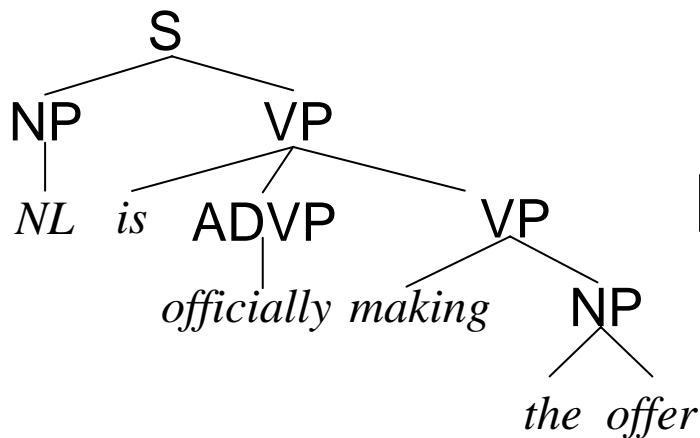
Our approach

1. Develop grammar rules and an HPSG treebank
2. Collect lexical entries from the HPSG treebank

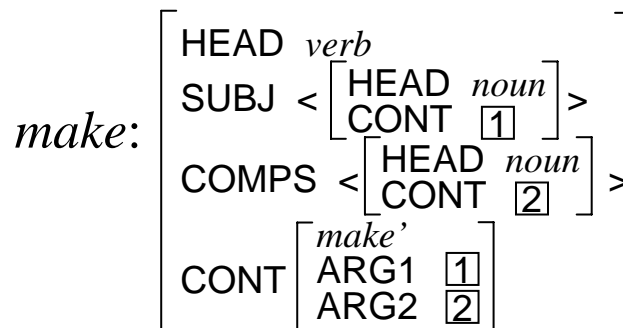
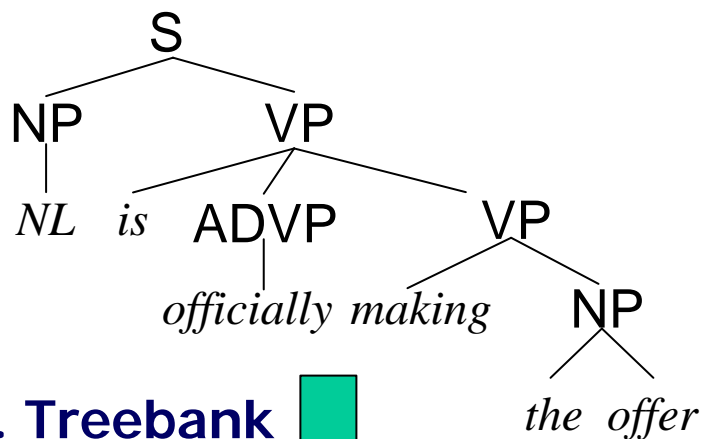


How to make an HPSG treebank?

- Convert Penn Treebank into HPSG-conformant structures
- Grammar development = restructuring a treebank in conformity with HPSG grammar rules



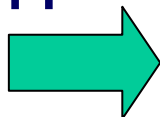
Overview of grammar development



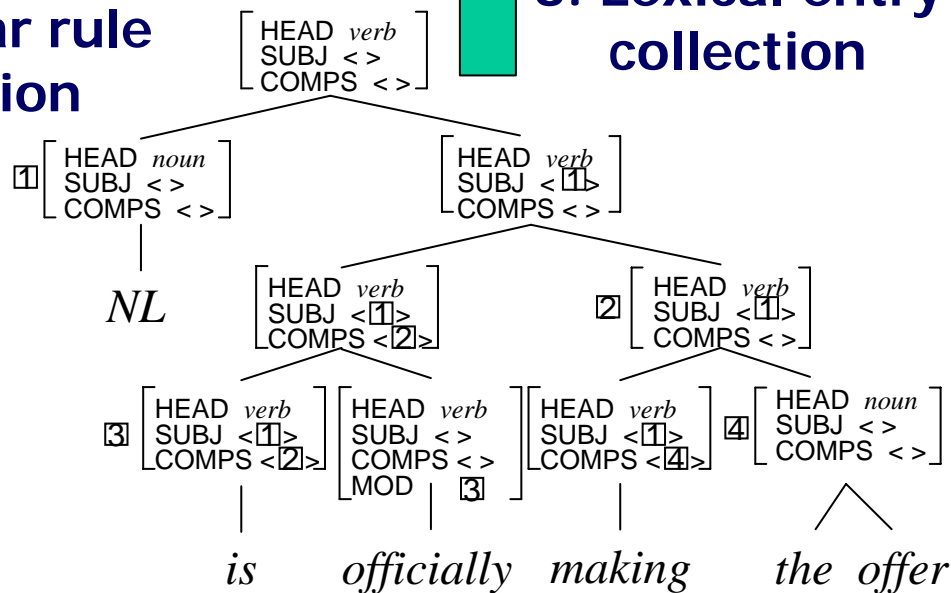
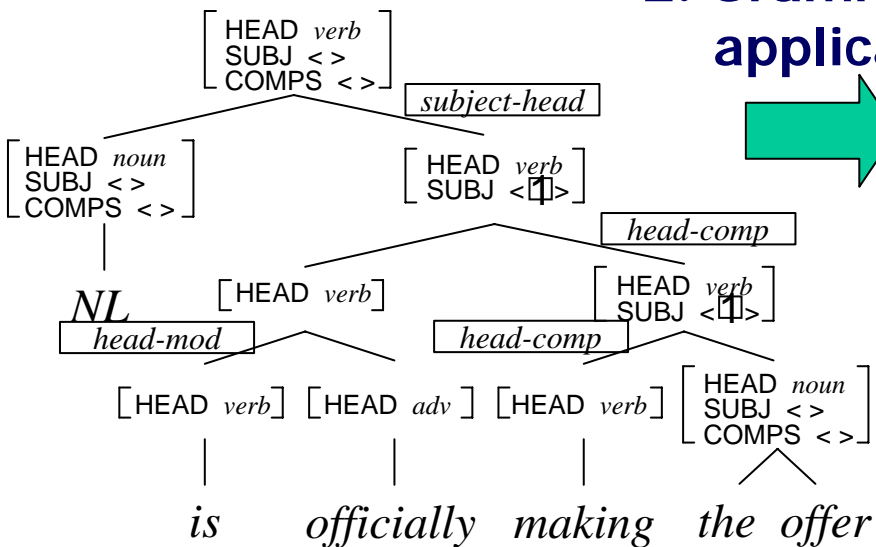
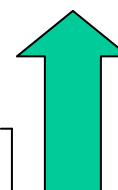
1. Treebank conversion



2. Grammar rule application

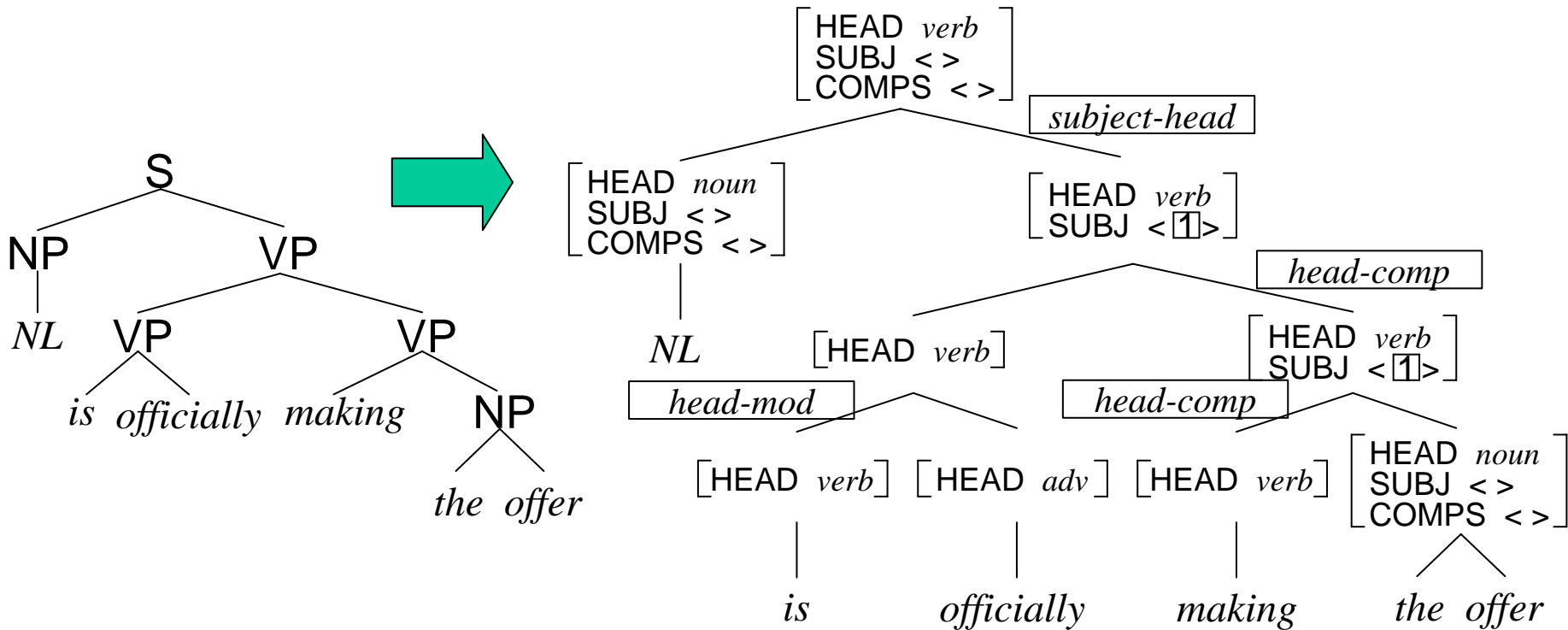


3. Lexical entry collection



1. Treebank conversion

- Modify constituent structures
- Add feature structures



Currently implemented constructions (1/2)

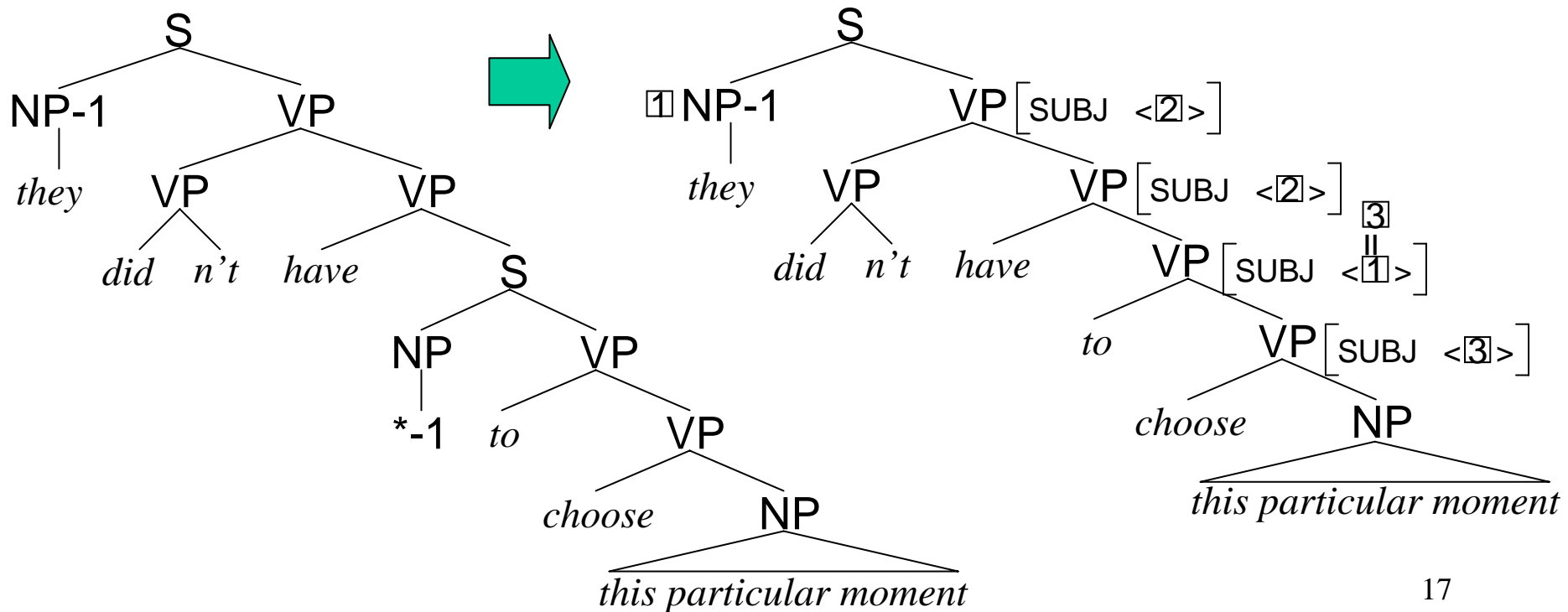
- Most of the constructions discussed in “Syntactic Theory” [Sag et al., 2003]
 - Subcategorization and modification
 - Coordination
 - HEAD features (CASE, INV, VFORM, etc.)
 - Imperative/interrogative
 - Predicative constructions
 - Passives, auxiliary/control verbs
 - Long distance dependencies (WH-movement and topicalization)

Currently implemented constructions (2/2)

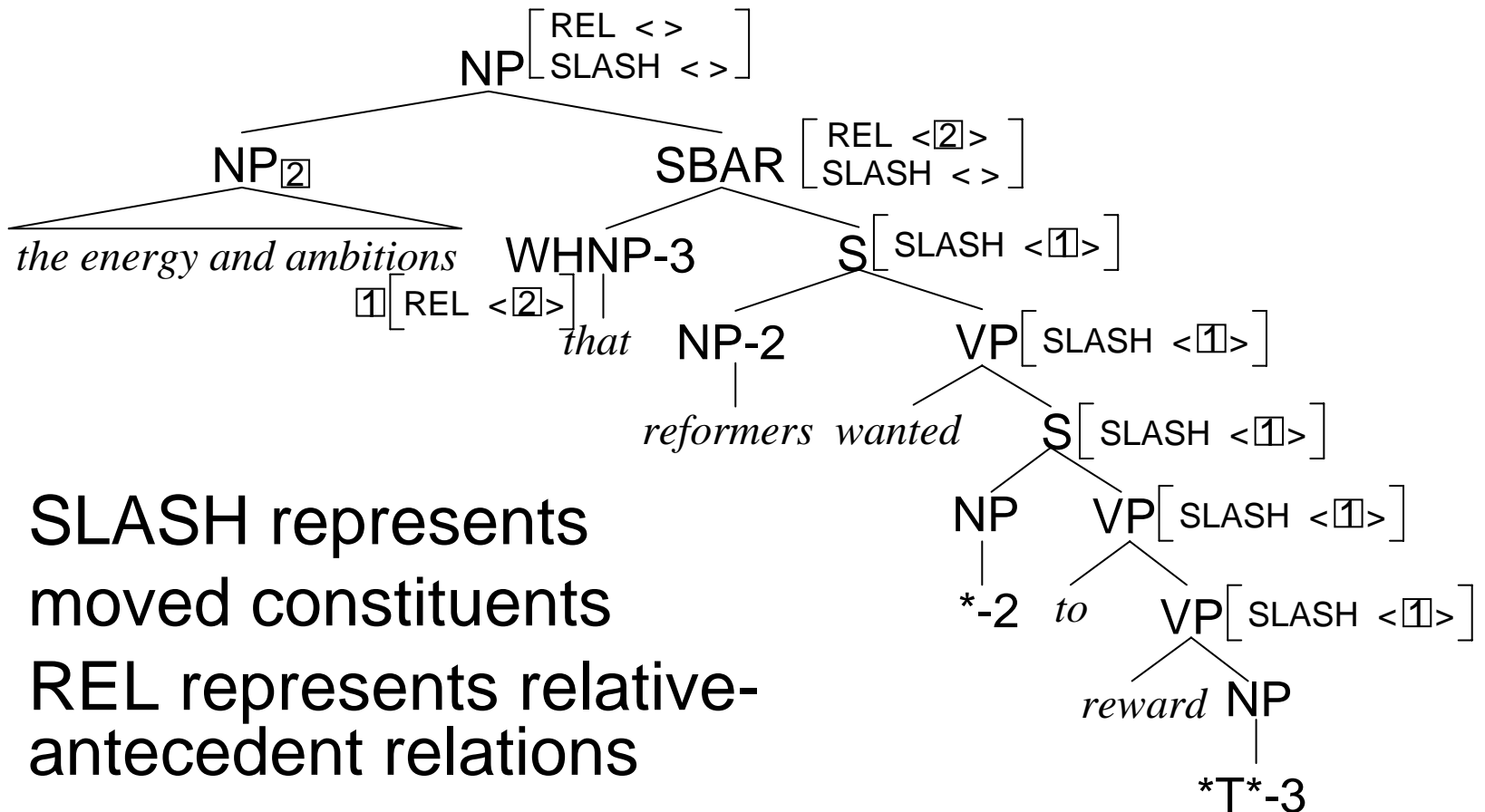
- The grammar is further extended in order to analyze real-world texts
 - Relative clauses (incl. pied-piping and free relative)
 - *tough* constructions
 - Small clauses, “than” clauses, quantifier phrases
 - Participial constructions
 - Inversion (ex. “..., said the president”)
 - Insertion (ex. “The president, he said, will show ...”)
 - Apposition
 - Quotation

Example: auxiliary/control verbs

- Auxiliary/control verbs are annotated as taking unsaturated constituents



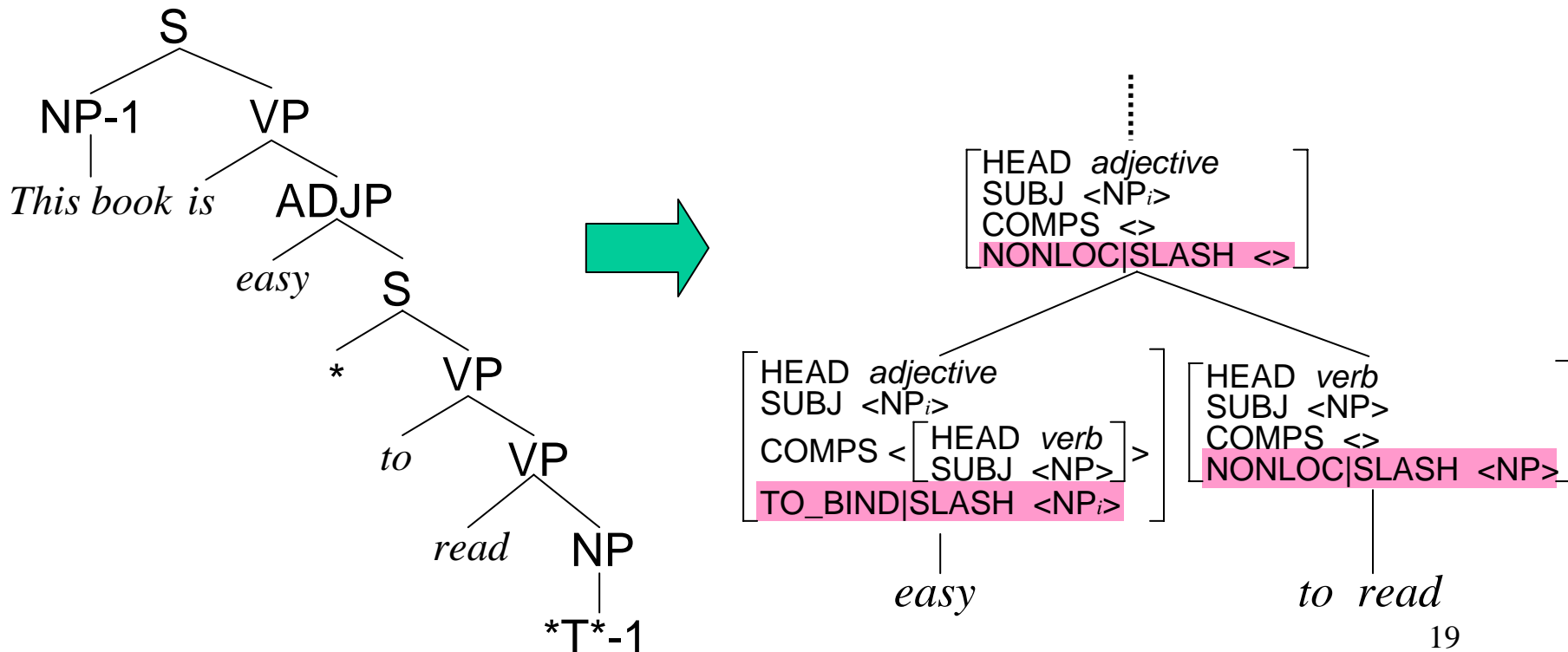
Example: object relative



- SLASH represents moved constituents
- REL represents relative-antecedent relations

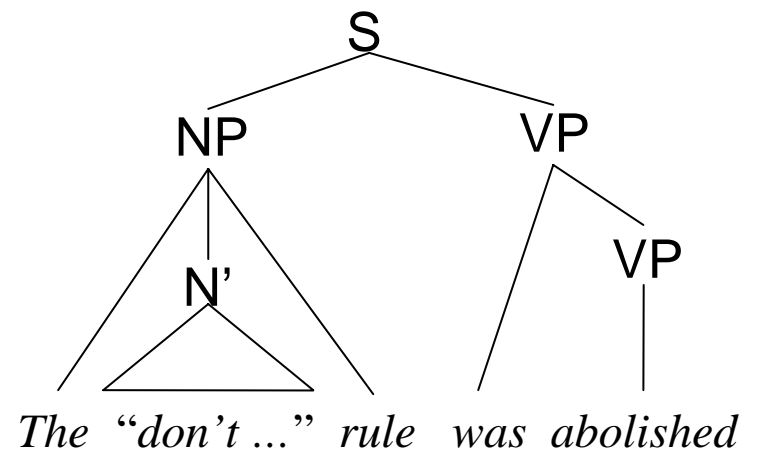
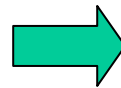
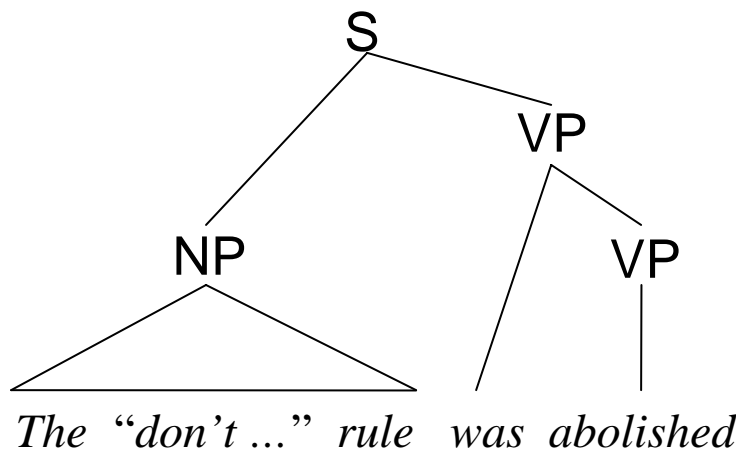
Example: *tough* construction

- “TO_BIND|SLASH” cancels out a trace of the complement



Example: quotation

- “Quotation” can change everything into N’
- Ex. *The “don’t speak out of turn” rule was abolished.*

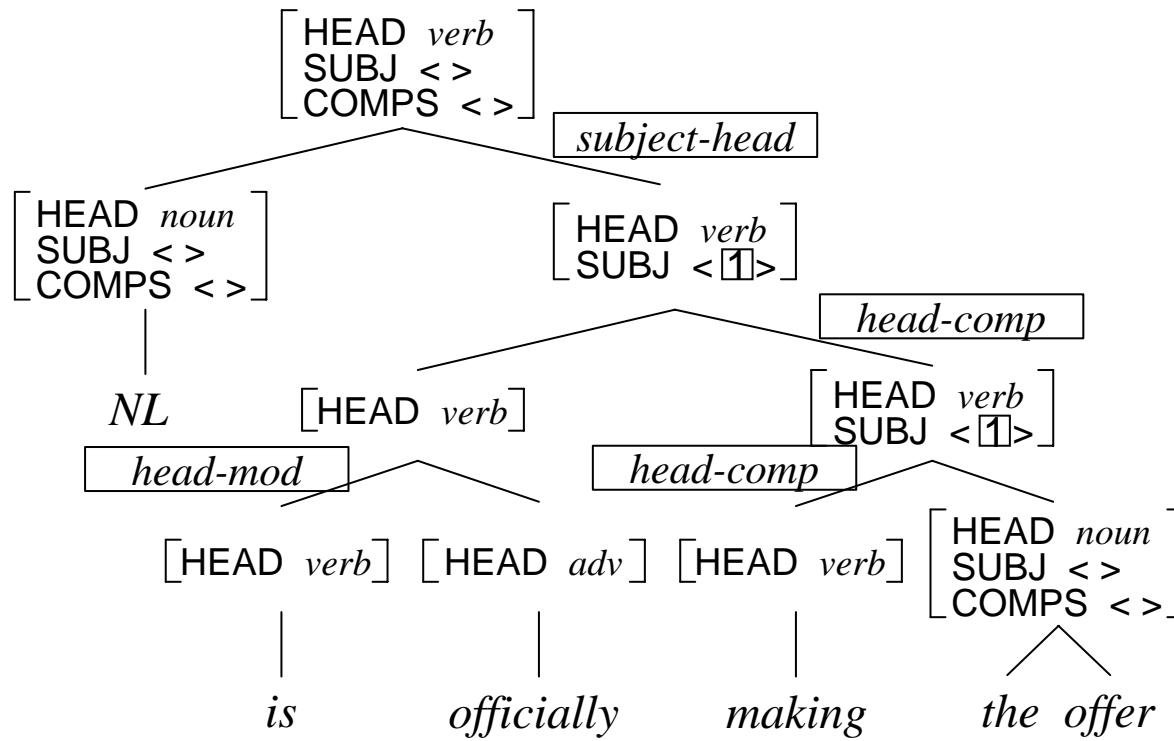


2. Grammar rule application

- Grammar rules are applied to HPSG-style parse trees
 - Grammar rule application fails when a parse tree contains errors/inconsistencies
 - Unspecified feature values are filled
- Resulting parse trees are assured to satisfy constraints of the HPSG theory

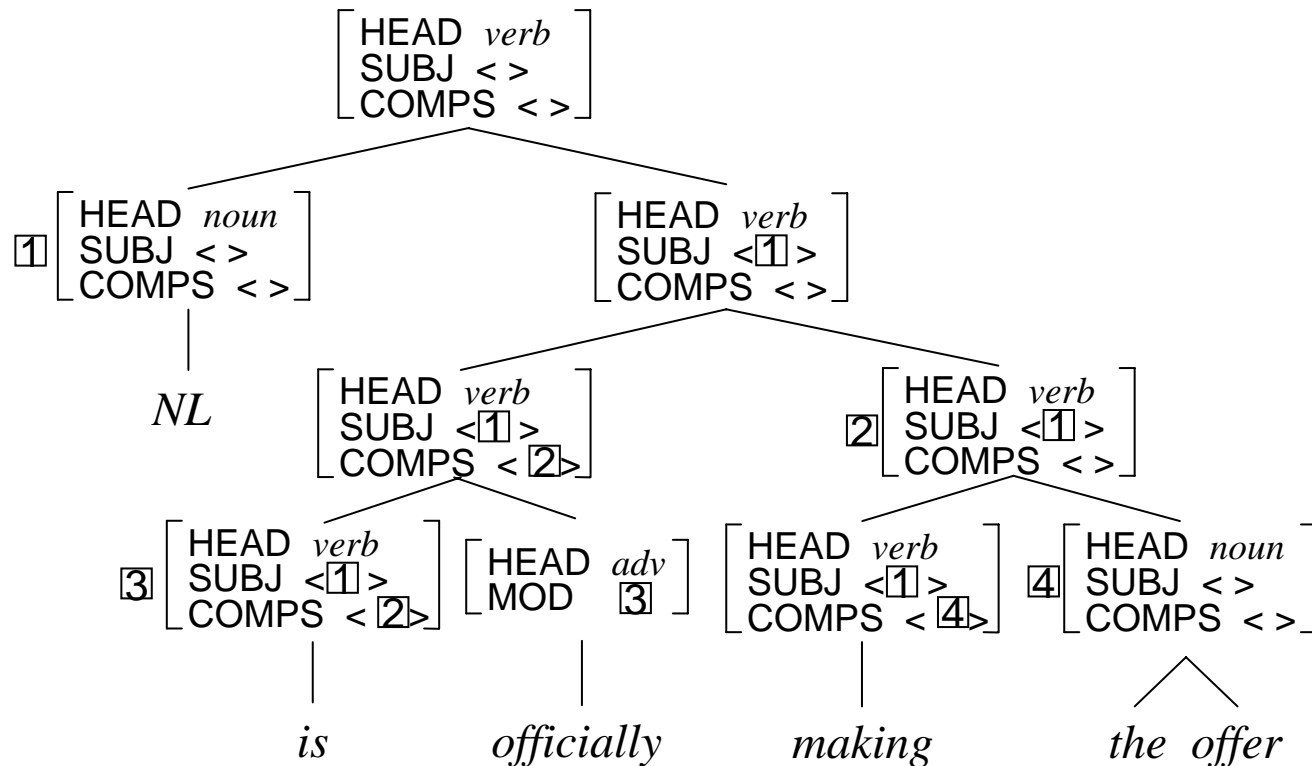
Example

- “*NL is officially making the offer*”



Example

- “*NL is officially making the offer*”

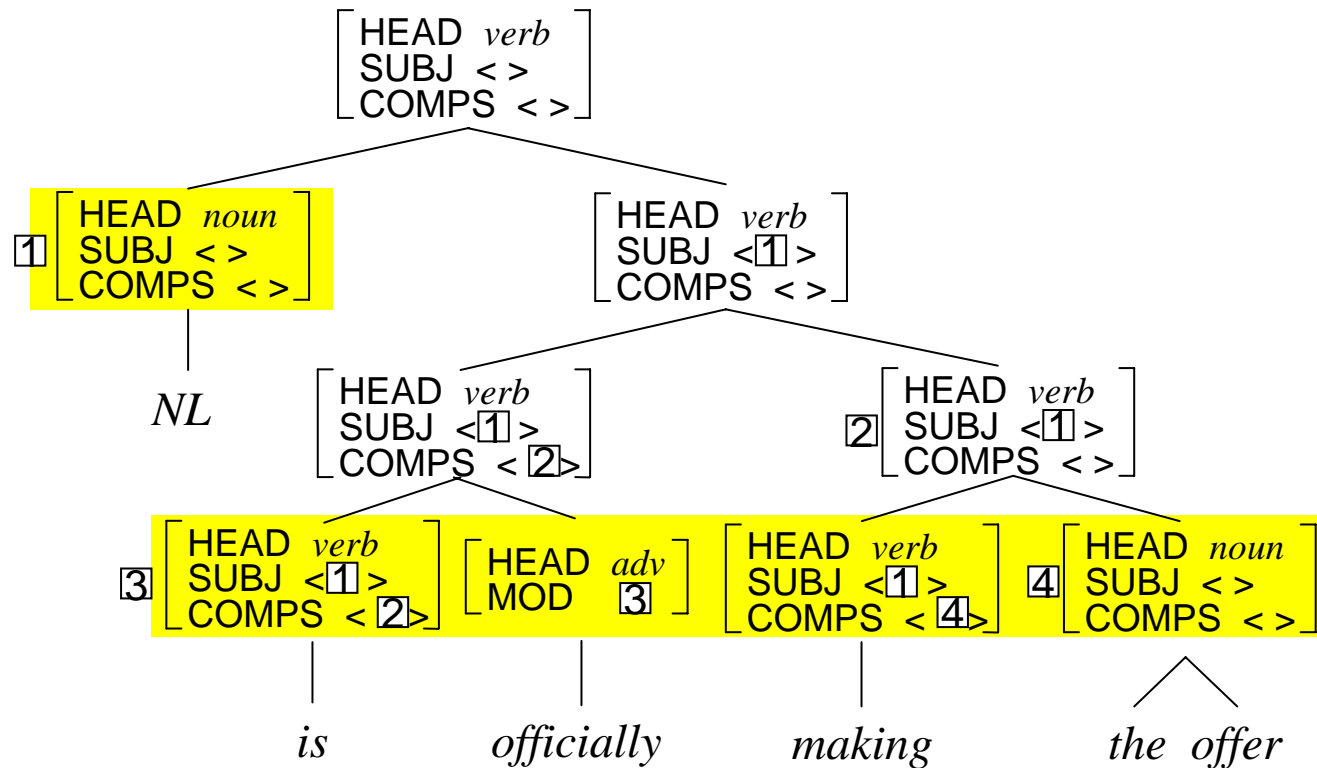


3. Lexical entry collection

- Collect terminal nodes of HPSG parse trees
- Assign predicate argument structures

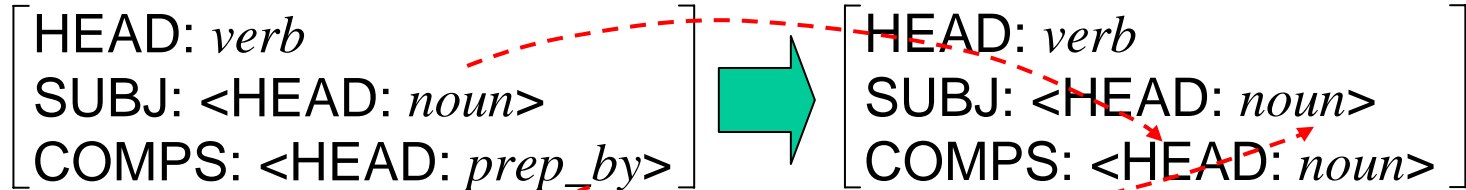
Collecting terminal nodes

- Terminal nodes of HPSG parse trees are instances of lexical entries

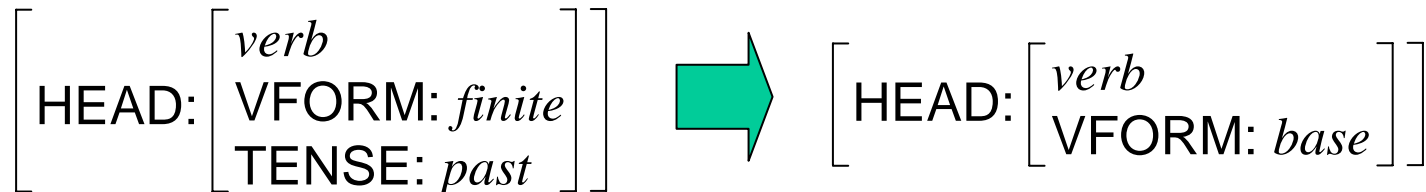


Inverse lexical rules

- *Inverse* lexical rules convert lexical entries of inflected words into lexemes
- Derivational rules: ex. passive rule



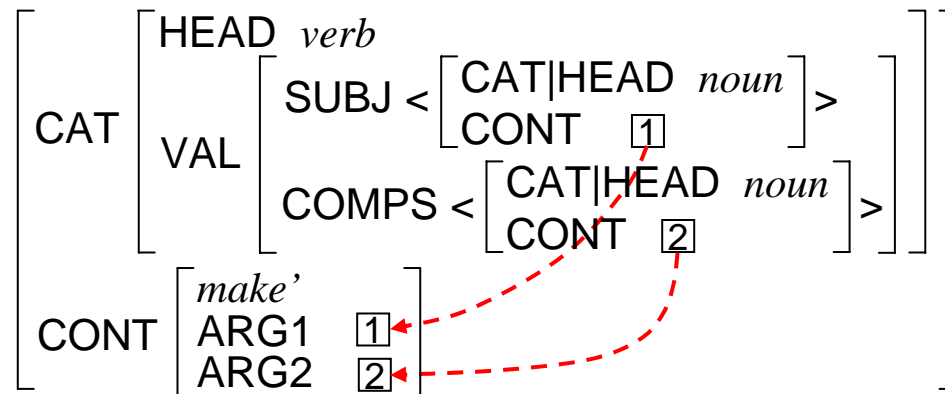
- Inflectional rules: ex. past-tense rule



Assigning predicate argument structures

- Create mappings from syntactic arguments into semantic arguments

Ex. lexical entry for “*make*”



Experiments

- Evaluation of lexical entries extracted from Penn Treebank sections 02-21 (39,832 sentences)
 - Coverage against unseen sentences
 - Relationship between treebank size and coverage
- Test data is an HPSG treebank converted from Penn Treebank section 23

Result of treebank conversion

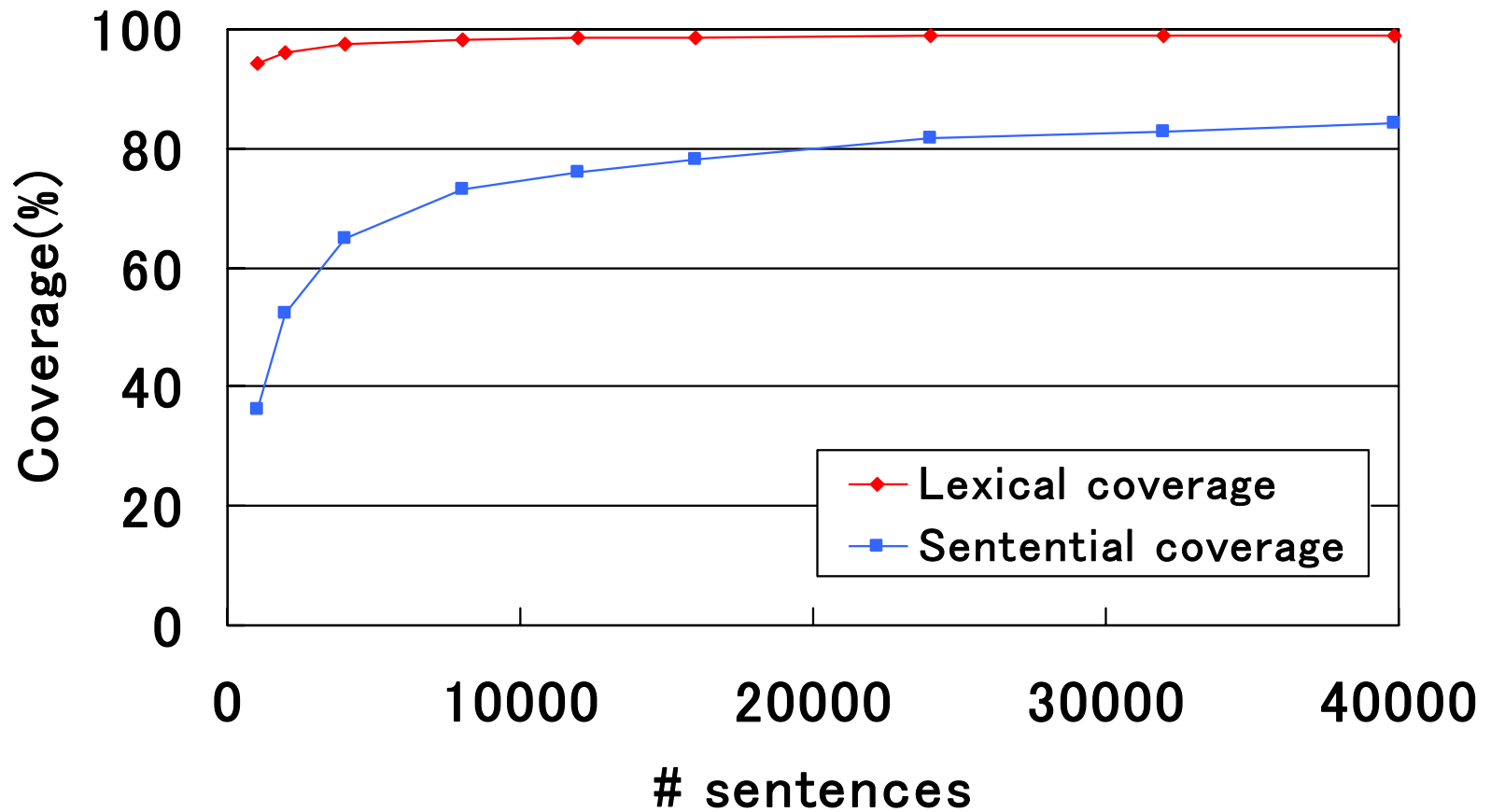
- Treebank conversion and grammar rule application succeeded for 37,589 sentences
- Resulting lexicon:

# words	34,461
# types of lexical entries	1,565
Average # lexical entries/word	1.33

Evaluation of coverage

- Measure: *strong coverage*
 - A word is considered covered when the lexicon has a correct lexical entry for a word
 - A sentence is judged to be covered when all of the words in the sentence are covered
- Lexical coverage: 99.30%
- Sentential coverage: 87.1%

Treebank size vs. coverage



Summary

- Corpus-oriented development of an HPSG grammar is presented
- A wide-coverage HPSG lexicon is obtained
- Future work: improvement of the grammar for supporting following constructions:
 - Expletive pronouns (*it, there*)
 - Idioms (ex. People are *sort of* nervous.)
 - Multiple extraction, parasitic gaps
 - Right-node raising, gapping