Computational Morphology and Syntax of Natural Languages

Daniel Zeman

October 2, 2020




Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

- Presentations and talks will be in English
  - This year even if all students present understand Czech (the talks will be recorded and made available for those who could not be present)
- Questions welcome in both English and Czech
  - If you do not want to be recorded, keep your question until after the talk
  - Or write me at zeman@ufal.mff.cuni.cz any time
- And I have many examples from Czech ☺

Getting Credits

- 2–3 smaller tasks
  - homework style
  - less flexible deadlines
- Alternatively: one larger project
  - ask me if interested
  - can be combined with your mgr. (or bc.) thesis

An “Unbalanced” Course

- 1/3 linguistics, 2/3 tools
- 1/3 lab work, 2/3 lectures
- 3/4 morphology, 1/4 syntax
- Mostly rule-based
  - almost no machine learning
  - no neural networks

Outline: Morphology

- Morphemic segmentation
  - *un* + *beat* + *able*
- Phonology ("morphonology") and orthography
  - *baby* + *s* = *babies*
- Inflectional vs. derivational morphology
- Morphological analysis: word form → lemma + morphosyntactic features (tag)
- Tagging (context-aware disambiguation)
- Unsupervised affix detection in corpus
- Mining of word forms from corpus

Morphological Analysis

- Input:
  - word form (token)
- Output:
  - set (possibly empty) of analyses
  - an analysis:
    - lemma (base form of the lexeme)
    - tag (morphological, POS)
    - ... part of speech
    - ... features and their values

MA Example

- Language: Czech
- Input: *malými*
- Output (only one selected analysis here):
  - lemma = *malý* “small”
  - tag = `AAFP71A`
    - part of speech = `AA` (adjective / přídavné jméno)
    - gender = `F` (feminine / ženský)
    - number = `P` (plural / množné)
    - case = `7` (instrumental / 7. pád)
    - degree of comparison = `1` (positive / 1. stupeň)
    - polarity = `A` (affirmative / kladné)

MA Example

- Language: 🇬🇧 English
- Input: *flies*
- Output:
  - lemma 1 = *fly-1* (to move in the air)
  - tag 1 = `VBZ` (verb, present tense 3rd person singular)
  - lemma 2 = *fly-2* (an insect)
  - tag 2 = `NNS` (noun, plural)
- Output is not disambiguated with respect to context

MA vs. Tagging

- By tagging we usually mean context-based disambiguation
- Most taggers employ machine learning methods
- Taggers may or may not work on top of MA
  - MA may provide readings not known from training
  - If a tagged corpus is available but MA is not, a tagger can still be trained on the corpus

Morphemic Segmentation

- Morpheme is the smallest unit of language that conveys some meaning
- Morpheme segmentation = finding morpheme boundaries within words
- Typically part of MA:

Morphemic Segmentation

- Morpheme is the smallest unit of language that conveys some meaning
- Morpheme segmentation = finding morpheme boundaries within words
- Typically part of MA:
  - input: *closed*

Morphemic Segmentation

- Morpheme is the smallest unit of language that conveys some meaning
- Morpheme segmentation = finding morpheme boundaries within words
- Typically part of MA:
  - input: *closed*
  - identify the morphemes: *close* + *d*

Morphemic Segmentation

- Morpheme is the smallest unit of language that conveys some meaning
- Morpheme segmentation = finding morpheme boundaries within words
- Typically part of MA:
  - input: *closed*
  - identify the morphemes: *close* + *d*
  - interpret them: *verb (close)* + *past tense*

Morphemic Segmentation

- Morpheme is the smallest unit of language that conveys some meaning
- Morpheme segmentation = finding morpheme boundaries within words
- Typically part of MA:
  - input: *closed*
  - identify the morphemes: *close* + *d*
  - interpret them: *verb (close)* + *past tense*
  - output: *close* + `VBD`

Morphemic Segmentation

- Sometimes it is useful to know the morphemes even if we cannot interpret them
- Data sparseness, e.g., in machine translation:
  - en: *city*

Morphemic Segmentation

- Sometimes it is useful to know the morphemes even if we cannot interpret them
- Data sparseness, e.g., in machine translation:
  - en: *city*
  - cs alignments in parallel corpus: *město* (nom/acc/voc sing, 42×), *města* (gen sing, nom/acc/voc plur, 40×), *městě* (loc sing, 32×), *měst* (gen plur, 9×), *městské* (adj, 7×), *městem* (ins sing, 7×), *městských* (adj, 4×), *městská* (adj, 4×), *městský* (adj, 2×), *městu* (dat sing, 2×), *městech* (loc plur, 2×) … total 11 forms seen

Morphemic Segmentation

- Sometimes it is useful to know the morphemes even if we cannot interpret them
- Data sparseness, e.g., in machine translation:
  - en: *city*
  - cs alignments in parallel corpus: *město* (nom/acc/voc sing, 42×), *města* (gen sing, nom/acc/voc plur, 40×), *městě* (loc sing, 32×), *měst* (gen plur, 9×), *městské* (adj, 7×), *městem* (ins sing, 7×), *městských* (adj, 4×), *městská* (adj, 4×), *městský* (adj, 2×), *městu* (dat sing, 2×), *městech* (loc plur, 2×) … total 11 forms seen
  - missing cs: *městům* (dat plur), *městy* (ins plur), *městského*, *městskému*, *městském*, *městským*, *městští*, *městskými*, *městskou* (adj remaining forms) … total 9 forms missing

Morphemic Segmentation

- Sometimes it is useful to know the morphemes even if we cannot interpret them
- Data sparseness, e.g., in machine translation

- Stemming = stripping all morphemes but the stem
  - IN: *The British players were unbeatable.*
  - OUT: *The Brit play were beat .*

- Lemmatization = replacing all words with their lemmas (as with tagging, disambiguation may be assumed)
  - IN: *The British players were unbeatable.*
  - OUT: *the British player be (un)beatable .*

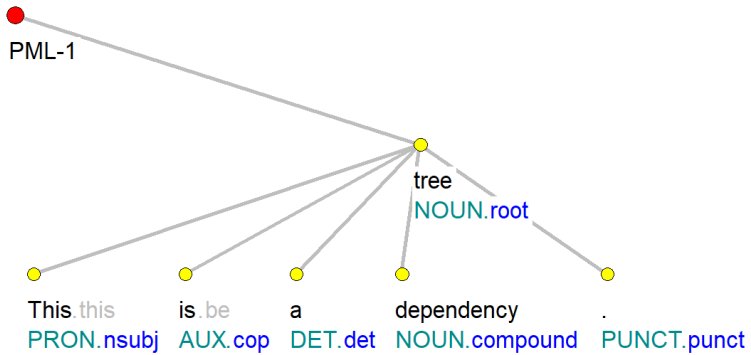
Inflection vs. Derivation

- Derivational morphology:
  - New lemma!
  - Often (but not always) new part of speech
- Inflectional morphology:
  - Set of forms of one lemma (lexeme)
  - The set is called paradigm
- The borderline is sometimes quite fuzzy

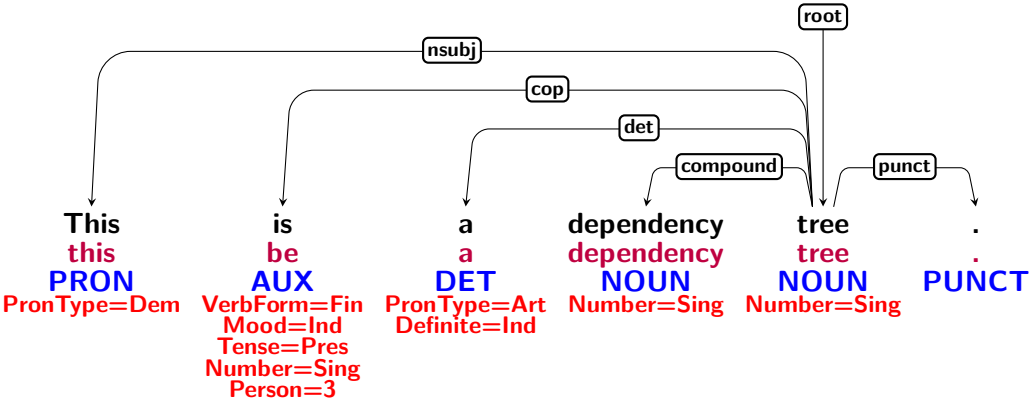
Outline: Syntax

- Constituency vs. dependency
- Context-free grammars
- Transition network grammars
- Shallow parsing (chunking)
- Chart parsers
- Dependency parsers
  - Transition-based
  - Graph-based
- Clause boundaries

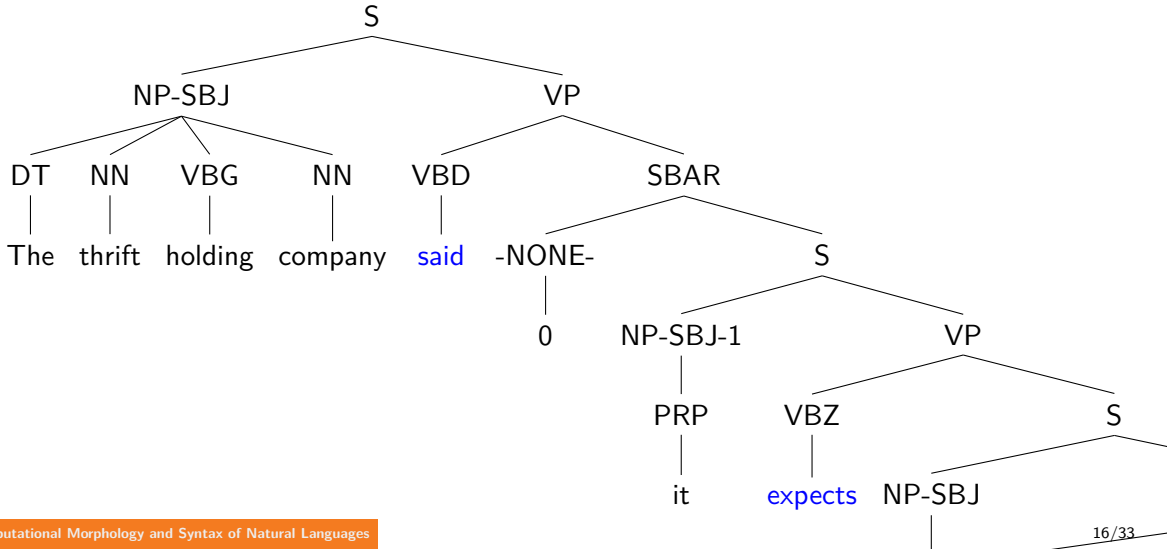
Dependency Tree



Dependency Tree



Phrasal Tree (Penn Treebank)



Applications of Morphology

- First step before broader NLP applications:
  - (Input for syntactic parsing)
  - (Machine translation)
    - Rule-based MT: full-fledged analysis and generation
    - Statistical MT: fighting data sparseness
    - Neural MT: nothing (character embeddings instead)
  - Finding word boundaries (Chinese, Japanese)
  - Dictionaries

Applications of Morphology

- Text-to-speech systems (speech synthesis)
  - Morphology affects pronunciation
    - English *th* is normally pronounced θ or $\eth$
    - However, not in *boathouse (boat + house)*
    - Czech *proudit* =
    - ... *proud* + *it* ("stream" + INF = "flow")
    - ... *pro* + *ud'* + *it* ("through" + "smoke" + INF = "smoke thoroughly")
  - (Speech recognition)
    - Morphology allows for smaller dictionaries

Applications of Morphology

- Word processing
- Typing 🇯🇵 Japanese text
  - Two kana syllabic scripts and kanji (Chinese characters)
  - Typically, people type in kana and system converts to kanji whenever necessary
  - Disambiguation needed!
  - Bound morphemes remain in kana (morphological rules)

Applications of Morphology

- Word processing: find & replace terms
  - Czech: *kniha* “book” → *dílo* “work”
    - *knihy* → *díla*
    - *knize* → *dílu*
    - *knihu* → *dílo*
    - *kniho* → *dílo*
    - *knihou* → *dílem*
    - *knih* → *děl*
    - *knihám* → *dílům*
    - *knihách* → *dílech*
    - *knihami* → *díly*
- Document retrieval
  - Keywords in query are typically base forms
  - The forms in documents are inflected

Morphology-Based Typology

- Isolating languages
  - Chinese: *gǒu bú ài chī qīngcài* = "dog not like eat vegetable"
- Inflectional languages
  - Romance and Slavic languages: Spanish: *pued+es* = *poder* + present indicative, 2nd person, singular
- Agglutinative languages
  - Turkish: *çöplüklerimizdekilerdenmiydi* = *çöp* + *lük* + *ler* + *imiz* + *de* + *ki* + *ler* + *den* + *mi* + *y* + *di* = "was it from those that were in our garbage cans?"
- Polysynthetic languages
  - Eskimo-Aleut languages

Polysynthetic Languages

- Found in Siberia and the Americas
- Intricately compose words of many lexical morphemes that are not easily told apart
  - Typically include both subject- and object-verb agreement
- That is why linguists decided not to separate them orthographically
- Nevertheless, words usually are separated. They are just long
- One long word may cover a whole sentence in other languages
- Chukchi example (Skorik 1961: 102):
  - *Təmeyŋəlevtpəγtərkən.*
  - *Tə-meyŋə-levt-pəγtə-rkən.*
  - `1.SG.SUBJ-great-head-hurt-PRES.1`
  - “I have a fierce headache.”

Morphological Devices (Overview)

- Affixes (prefixes and suffixes): concatenative morphology
- Compounding
- Infixation
- Circumfixation
- Root and pattern (templatic) morphology
- Reduplication
- Subsegmental morphology
- Zero morphology
- Subtractive morphology

Affixation

- Most common way of inflection and derivation
- Three morpheme types:
  prefix + radix (stem) + suffix
  - en: *dog* + *s* = *dogs*
    - plural suffix *-s*
  - de: *mach* + *st* = *machst*
    - suffix *-st* denotes present indicative 2nd person singular
  - en: *un* + *beat* + *able*
    - prefix *un-* negates the meaning
    - suffix *-able* converts verb to adjective, expressing applicability of the action of the verb to something

Infixation

- Philippine languages, e.g., Bontoc:
  - *fikas* “strong” → *f+um+ikas* “be strong”
  - *kilad* “red” → *k+um+ilad* “be red”

  - Could be analyzed as prefix to (stem minus initial consonant)

Circumfixation

- Prefix + suffix act together as one morpheme
  - German: *legen* “lay down” → *ge+leg+t* “laid down”
  - Indonesian: *besar* “big” → *ke+besar+an* “greatness”

- Similar but not the same as Czech superlatives
  - *nej+mlad+š+í* “youngest”
  - superlative + stem + comparative + singular nominative

Templatic Morphology

- Semitic languages (Arabic, Hebrew, Amharic…)
-  Arabic:
  - root (usually 3 consonants): *ktb* “write”
  - vowel pattern: *aa* = active, *ui* = passive
  - template: *CVCVC* = first derivational class of verbs (binyan)
  - result: *katab* “write”, *kutib* “be written”

Reduplication

- Copy whole stem or part of it
-  Indonesian plural:
  - *orang* “man” → *orang-orang* “men”
-  Javanese habitual-repetitive:
  - *adus* → *odas+adus* “take a bath”
  - *bali* → *bola+bali* “return”
-  Yidiny (Australian language)
  - *gindalba* “lizard” → *gindal+gindalba* “lizards”

- Reduplication cannot be modeled by finite-state automata!

Subsegmental Morphology

- Irish:
  - *cat* /kat/ = "cat" (singular)
  - *cait* /kat^{j}/ = "cats" (plural)

- The plural morpheme consists just of one phonological feature ("high"), resulting in palatalization

Zero Morphology

- Zero (empty) morpheme, marked sometimes as 0, Ø, λ or ε
- Czech feminine plural case endings for *žena* “woman”:
  - nom: *žen+y* = *ženy*
  - gen: *žen+λ* = *žen*
  - dat: *žen+ám* = *ženám*
  - acc: *žen+y* = *ženy*
  - voc: *žen+y* = *ženy*
  - loc: *žen+ách* = *ženách*
  - ins: *žen+ami* = *ženami*

Subtractive Morphology

- Koasati (Muskogean language):
  - singular verb: *pitaf+fi+n*
  - plural verb: *pit+li+n*
  - singular verb: *lasap+li+n*
  - plural verb: *las+li+n*

- Such examples are rare
- Moreover, one might argue that plural is the base form here

Compounding

- English: maximally two stems written together
- Germanic languages in general favor compounds
- German: *Hotentotenpotentatentantenatentäter*
  - *Hotentot* + *en* + *Potentat* + *en* + *Tante* + *n* + *Atentäter*
  - "Hottentot potentate aunt assassin"
  - "assassin of aunt of potentate of Hottentots"

Further Reading

- James Allen (1995). *Natural Language Understanding.* Benjamin/Cummings, USA
- Richard Sproat (1992). *Morphology and Computation.* MIT Press, USA
- Kenneth R. Beesley, Lauri Karttunen (2003). *Finite State Morphology.* CSLI Publications
- Anna Feldman, Jirka Hana (2009). *A Resource-Light Approach to Morpho-syntactic Tagging.* Rodopi, Netherlands
- Daniel Zeman (2018). *The World of Tokens, Tags and Trees.* ÚFAL, Czechia