AX

Lexico-syntactic Information from Corpora
Josefův Důl, 21.1.2003

Ondřej Bojar

Motivation

- Distant goals:
  - Machine translation, grammar checking, text summary, ...
- Steps in processing sentences:
  - Morphological analysis ($\sim$ done)
  - Syntactic analysis (difficult, lexico-syntactic information needed)
  - Deeper analyses (more difficult, out of scope of this presentation)

Information for Verbs

- Verbs "organize" the sentence, let's start with them.
- Verbs "require" specific (types of) complements (modifications).
- *Observed frame* = list of sons in a dependency tree.
- *Surface frame* = list of "typical" complements of (a type of) a verb. (e.g. *protestovat proti_čemu*, not *protestovat kdy*)
- Steps in automatic extraction of surface frames:
  - Obtain observed frames.
  - Filter the complements to obtain surface frames.
  - (Further inference to obtain valency frames.)

Surface $\simeq$ *Subcategorization frame*, *Subcategorization* $\neq$ *Valency frame*

Data Available

- Prague Dependency Treebank (PDT)
  - Syntactic anotation (manual).
  - ∼1.5 million words, ∼98 thousand sentences.
  - Observed frames “free of charge”.
- Czech National Corpus (CNC)
  - Morphological anotation (automatically disambiguated).
  - ∼100 million words, ∼1.8 million sentences.
  - Observed frames are hard to extract

    (Sample input sentence: *Nové předpisy vyžadují mj. též povinnost určit u každého kvantitativního výsledku měření jeho nejistotu.*)

Data for Individual Verbs

		Observations in CNC	PDT	
	Verbs	(min, ∅, max)	Observations	Examples
1	1	11 253 207	1 737	být
2	1	2 175 254	513	mít
3	2	570 234, 851 099,5, 1 131 965	116, 159,5, 203	moci, muset
4	21	140 362, 243 575,3, 522 307	21, 106,3, 524	říci, chtít, jít, dát, uvést
5	53	68 535, 92 773,1, 126 411	2, 45,2, 133	čekat, zůstat, znamenat
6	99	40 248, 50 606,4, 68 004	1, 31,8, 99	představit, věnovat, vyjít
7	164	23 011, 30 707,7, 40 210	1, 18,7, 79	číst, přicházet, končit
8	317	10 970, 15 861,7, 22 982	1, 10,5, 83	dokončit, svědčit, přejít
9	818	3 551, 6 137,0, 10 966	1, 4,5, 29	uznávat, reprezentovat
10	20 800	0, 241,7, 3 548	0, 1,8, 79	ztroskotat, rýsovat

More Syntactically Anotated Data

- Simple scheme:
  - (Get more texts, e.g. from the Internet.)
  - Morphologically anotate and disambiguate the sentences.
  - Employ one of the parsers available for Czech to get the dependency trees.
  - Extract the desired lexico-syntactic information.
- But current parsers are not accurate enough (we shall see).

Better Anotated Data

- Parsers are not accurate enough, so:
  - Morphologically anotate (+disambiguate) the sentences.
  - Choose "simple" sentences, easier to parse.
  - Employ one of the parsers...
- The filtration must be *linguistically motivated*.
  - Keep sentences containing the observed phenomenon.
  - Filter out all the sentences where the phenomenon is "hidden" and/or interferes with other phenomena.
  - (Partial parsing needed for some of the decisions.)
- Goal dependent filtration $\Rightarrow$ a "scripting" language would help creating different filters $\Rightarrow$ system AX.

AX

- Goals:
  - Easy formulation of filters to reject sentences.
  - Easy implementation of partial syntactic analysis of input sentences.
- Overall scheme:
  - Input sentence = list of feature structures.

    1 input word = 1 fs of (ambiguous) morphological information.
  - The script is a pipeline of *filters* and *rewriting rules*.
  - Blocks of rules update the lists of feature structures, generate different "readings" of the sentence.

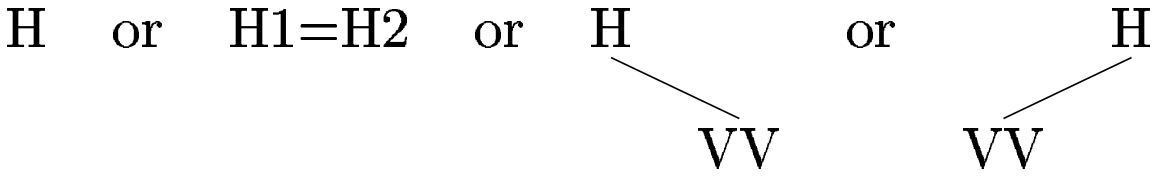
    Not limited to CFG or CSG. Nondeterministic rules allowed.
  - Filters reject inacceptable lists of feature structures.

Sample rules and filters

```
      # Find a sublist of fss matching given RE
      # And replace with output fs or fss or copy \1, \2...
rule "Simple noun phrase:" nounph ---> (adv{0,2} adj)* noun ::
  adj = [cat-adj], noun = [cat-noun], adv = [cat-adv],
  adj.case = noun.case,  # check constrains on rule
  adj.num = noun.num,    # check constrains on rule
  adj.gend = noun.gend,  # check constrains on rule
  nounph = noun,         # fill the output fs
end
filter "Reject readings containing two or more verbs
        or a conjunction:"
  ^ .* [cat-verb] .* [cat-verb] .* $
  |
  ^ .* [cat-conj] .* $
end
```

Filtration to Observe Verb Frames

- Reject complicated punctuation and numbers (colon. . .).
- Combine analytical verb forms (*usnuli jsme*).
- Ignore sentences with more autosemantic verbs.
- Fold simple coordination, reject complicated.
- Accept only sentences with simple structure:



- (Optional) Reject sentences with "suspicious" word order patterns (WOP).

$\Rightarrow$ "very simple sentences, *vss*." ($\sim$15–20 % of sents. in corpora)

Parsers on vss

		Statistical		Hand-made
Correct Dependencies	**Words**	**Collins**	**Zeman**	**Žabokrtský**
All Sentences	126 030	82,51 %	69,15 %	73,8 %
Very simple sentences	20 028	87,70 %	79,40 %	82,3 %
. . . and no suspicious WOP	11 030	87,89 %	79,31 %	83,6 %
Correct Sentences	**Sentences**	**Collins**	**Zeman**	**Žabokrtský**
All Sentences	7 319	30,95 %	15,00 %	18,4 %
Very simple sentences	1 786	47,14 %	29,00 %	31,6 %
. . . and no suspicious WOP	1 113	52,83 %	34,41 %	41,5 %

- $\Rightarrow$ 5–10 % better measured by correct dependencies.
- $\Rightarrow$ 15–20 % better ($\sim$twice) measured by sentences without a mistake.

Parsers on vss (contd.)

		Statistical		Hand-made
Observed Frames Correct	**Verbs**	**Collins**	**Zeman**	**Žabokrtský**
All Sentences	16 329	55,32 %	33,11 %	39,5 %
Very simple sentences	2 472	61,37 %	41,87 %	44,8 %
. . . and no suspicious WOP	1 546	64,68 %	47,02 %	53,8 %
No Sons Missing	**Verbs**	**Collins**	**Zeman**	**Žabokrtský**
All Sentences	16 329	73,73 %	55,86 %	74,7 %
Very simple sentences	2 472	78,52 %	67,76 %	84,1 %
. . . and no suspicious WOP	1 546	81,44 %	71,28 %	84,9 %

- $\Rightarrow$ 10–15 % better measured by correct observed frames.

Summary

- We need syntactic lexicons.
- Treebanks are excellent source, but:
  - expensive to build.
  - small (i.e. many lexemes are rare).
- Use "non-tree" corpora (bigger):
  - + parsers available to obtain syntactic anotation.
  - but select sentences containing an example of the observed phenomenon easier to analyze.
- AX is the system to filter sentences.
  - In fact powerful enough to build partial or full parsers.