Syntax (2/2)
(POS tagging, chunking, parsing)

Ing. Roberto Tedesco, PhD
roberto.tedesco@polimi.it

arcslab
adaptable, relational and cognitive software environments

NLP – AA 16-17
Prof. L. Sbattella

POS tagging

- Morphological analysis considers one word at a time
- Often, this is not enough to disambiguate part-of-speech
  - E.g. *Talk*: verb or name?
- The context is *needed*
- POS taggers do just that

Methodologies for POS tagging

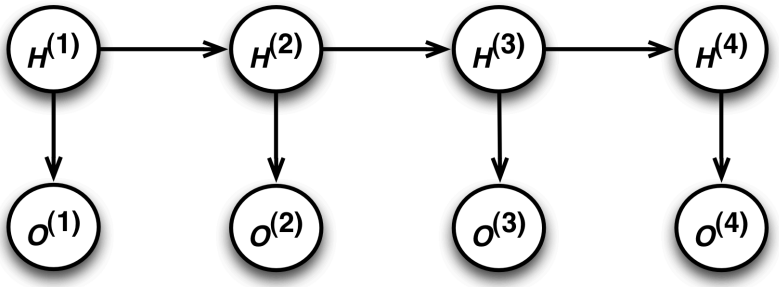
- A *language model* is needed
- Several approaches
  - Stochastic and non stochastic
- Two popular stochastic models:
  - HMM-based
  - Entropy Maximization-based

HMM POS tagging

- Works a sequence at a time
- Hidden variable: tag set
  - E,g, stochastic var. $T=\{t_a, t_b, \ldots\}$: tag set (hidden state)
- Observable variable: the set of the word forms (lexicon)
  - E.g. stochastic var. $W=\{w_a, w_b, w_c, \ldots\}$: word set (output)
- Given a sequence of words $<w^{(i)}>$, Viterbi calculates the most probable sequence of tags $<t^{(i)}>$

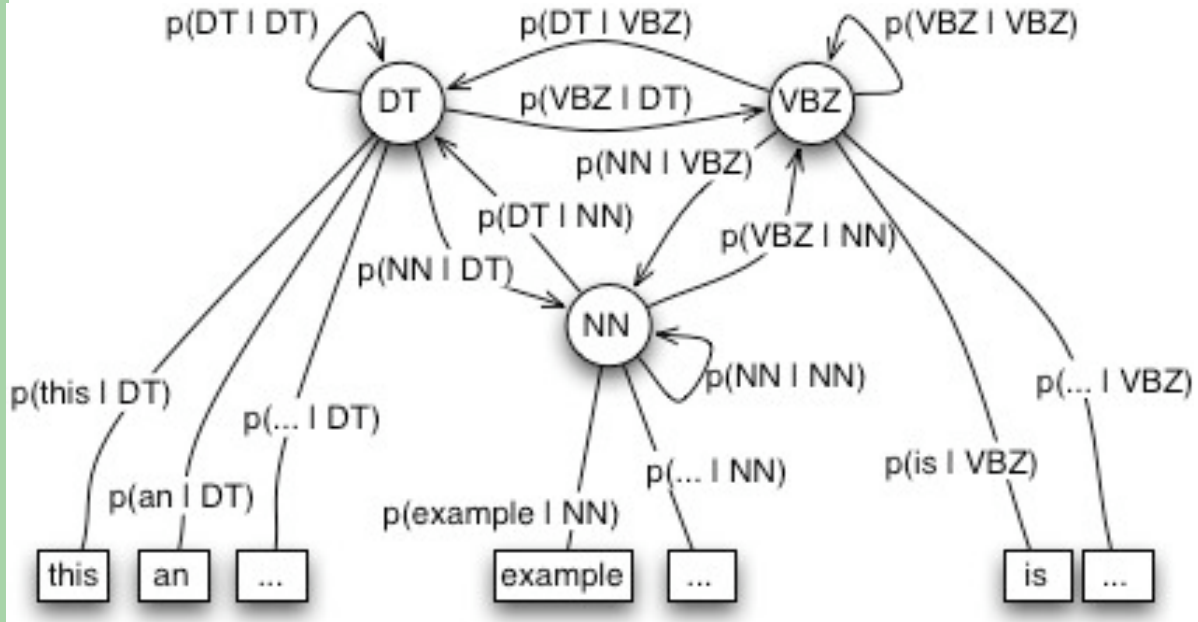
Hidden Markov Models (HMM)

<$H^{(1)}$=DT, $H^{(2)}$=VBZ, $H^{(3)}$=DT, $H^{(4)}$=NN>



<$O^{(1)}$=this, $O^{(2)}$=is, $O^{(3)}$=an, $O^{(4)}$=example>

Unrolled view

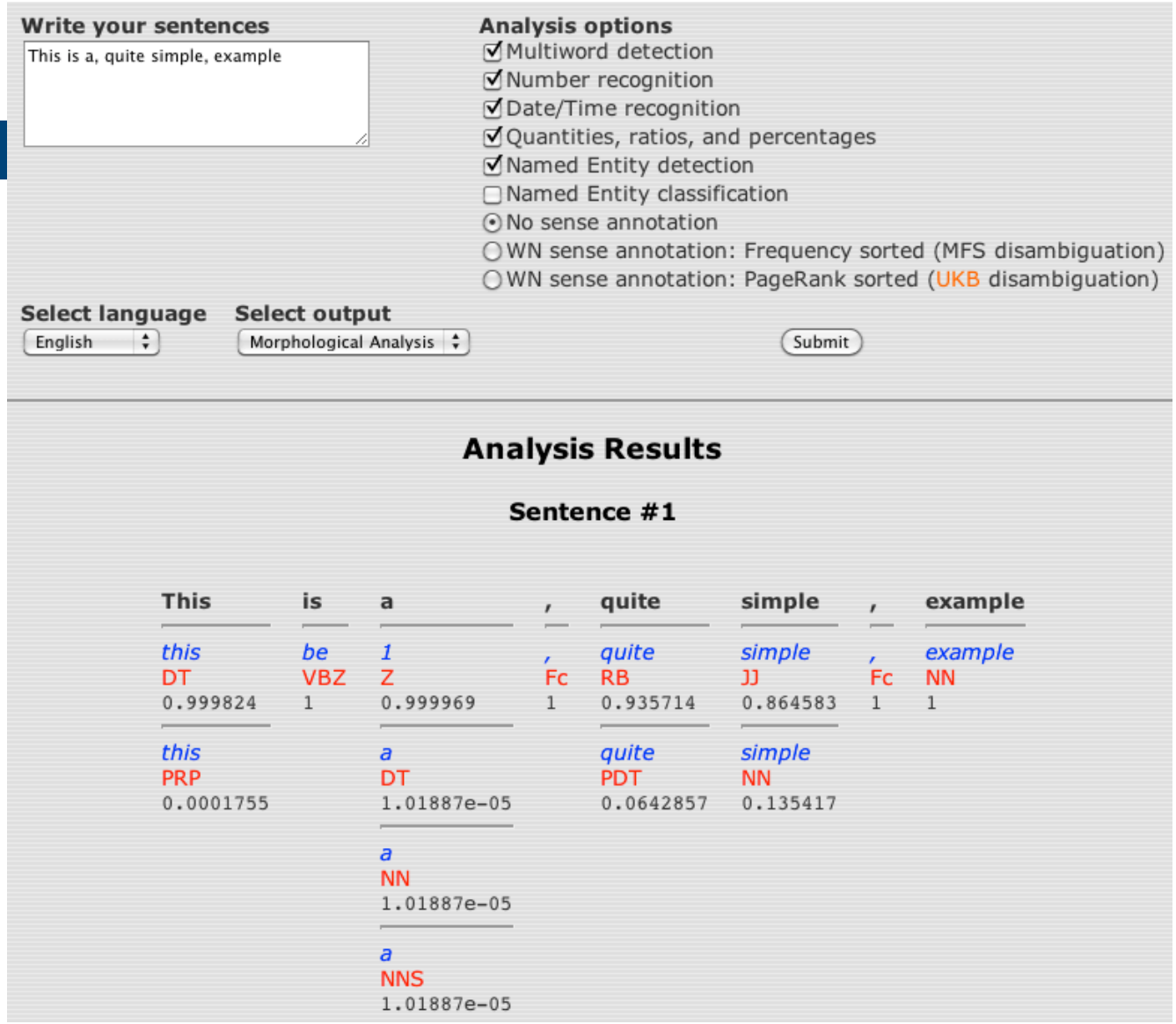


Graph view

- $H^{(t)}$={DT, VBZ, NN, … }
- $O^{(t)}$={…, an, …, example, …, is, …, this, …}

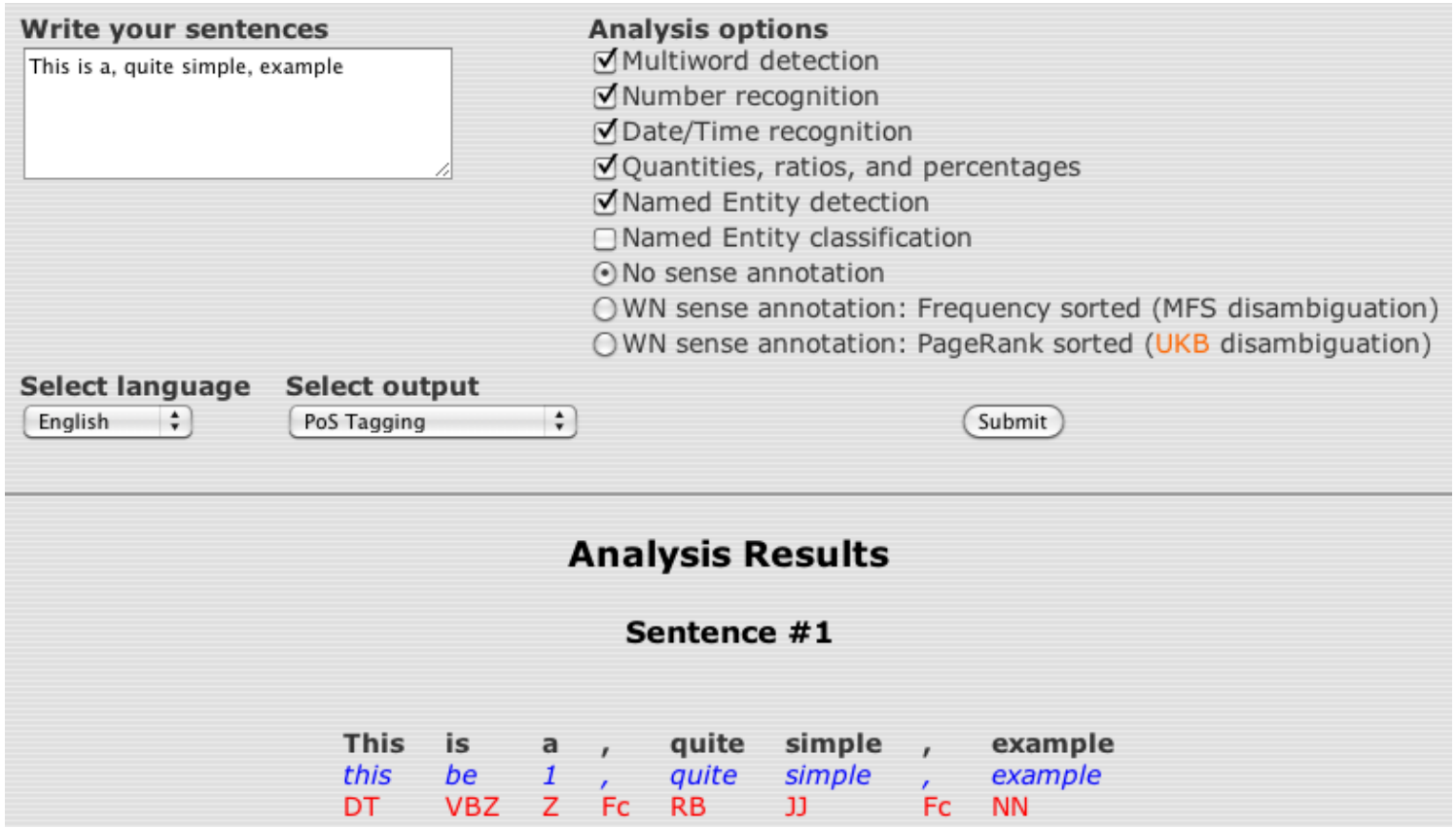
FreeLing

- As a morphologic analyzer



FreeLing

- POS tagging
- HMM



Stanford POS tagger

- Stanford POS Tagger
- Entropy Maximization
  - Uses a CMM, basically a simplified CRF
- Java based

- demo

Chunking (aka shallow parsing)

- Identifying and classifying the flat, non-overlapping segments of a sentence
  - This set typically includes noun phrases, verb phrases, adjective phrases, and prepositional phrases
  - $[_{NP}$ The morning flight] $[_{PP}$ from] $[_{NP}$ Denver] $[_{VP}$ has arrived.]
- Leverages POS tagging
- Two approaches:
  - Finite-state rules able to catch phrase segments (FST)
  - Machine learning. We present this approach

Tags (Penn treebank corpus)

Tag	Description	Words	Example	%
NP	noun phrase	**DT+RB+JJ+NN + PR**	*the strange bird*	51
PP	prepositional phrase	**TO+IN**	*in between*	19
VP	verb phrase	**RB+MD+VB**	*was looking*	9
ADVP	adverb phrase	**RB**	*also*	6
ADJP	adjective phrase	**CC+RB+JJ**	*warm and cosy*	3
SBAR	subordinating conjunction	**IN**	*whether or not*	3
PRT	particle	**RP**	*up the stairs*	1
INTJ	interjection	**UH**	*hello*	0

CoNLL corpus

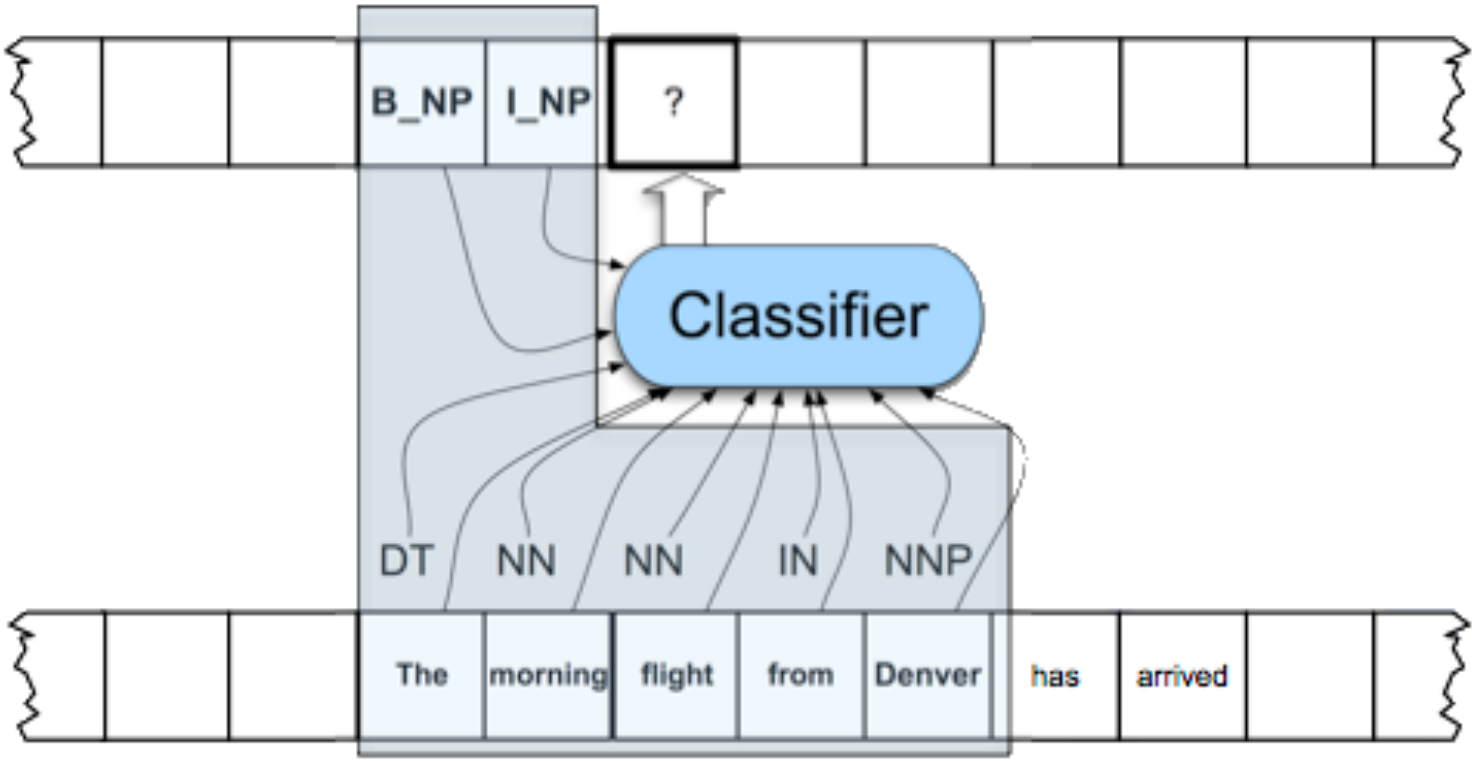
- To train stochastic chunkers
- token, POS, and chunk type
- IBO tagging, for chunk types:

  B_ begin of a chunk
  I_ inside the chunk
  O not part of a chunk

He	PRP	B_NP
reckons	VBZ	B_VP
the	DT	B_NP
current	JJ	I_NP
account	NN	I_NP
deficit	NN	I_NP
will	MD	B_VP
narrow	VB	I_VP
to	TO	B_PP
only	RB	B_NP
#	#	I_NP
1.8	CD	I_NP
billion	CD	I_NP
in	IN	B_PP
September	NNP	B_NP

Machine learning based chunking

- The chunker slides a context window over the sentence classifying words as it proceeds
- At this point the classifier is attempting to label *flights*

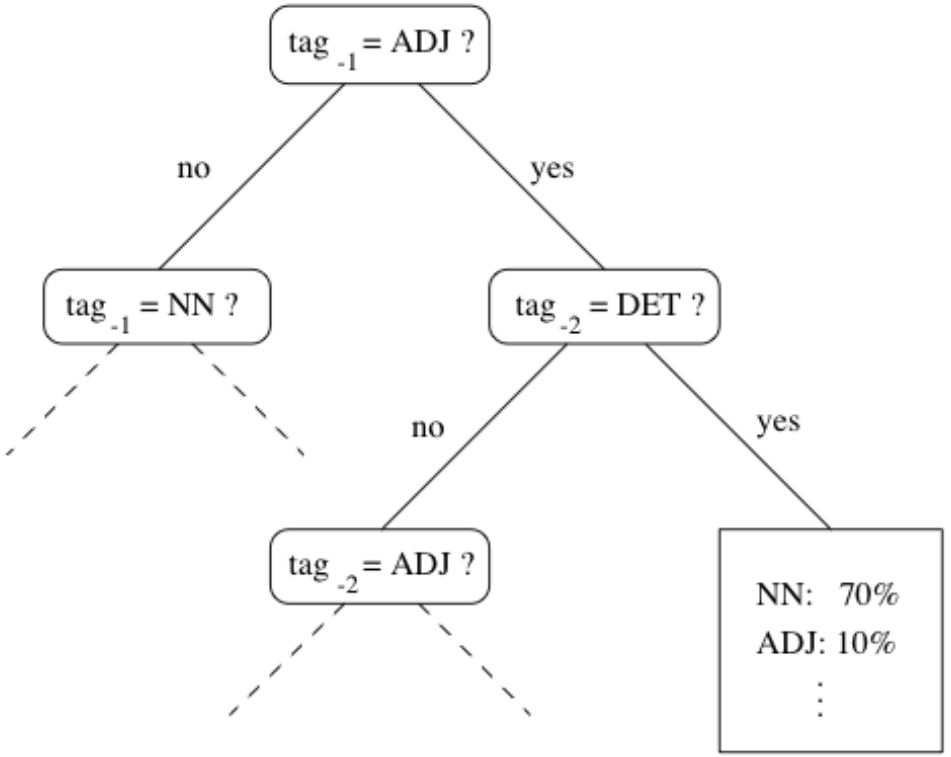


TreeTagger

- POS tagging
- Uses a 2nd order HMM; estimates transition probability by means of a model (not an n-gram)
- Uses a binary decision tree
  - Built from a training corpus of trigrams with POS's

demo

$$p(T^{(t)} \mid T^{(t-1)}, T^{(t-2)})$$



Full parsing

- Classical approach
  - Language model: CFG
- Stochastic approach
  - Language model: PCFG or L-PCFG
  - Often advanced stochastic models are used (multi-step parsing)
- Alternative approaches
  - Language model: Dependency Grammars
  - Language model: Feature-Based Grammars

CFG

- Context-Free Grammar

```
S -> Det N
S -> N
Det -> the | a
N -> dog | cat
```

CFG

- “I saw John with a dog with my cookie”
- top-down, bottom-up, and Earley algorithms
- Five trees found
  - All the trees are compatible with the CFG
  - No way to select the “right one”

PCFG

- Probabilistic CFG

```
S -> Det N [0.8]
S -> N [0.2]
Det -> the [0.6] | a [0.4]
N -> dog [0.5] | cat [0.5]
```

- PCFG, structure and probability, can be learned from a corpus (a treebank)

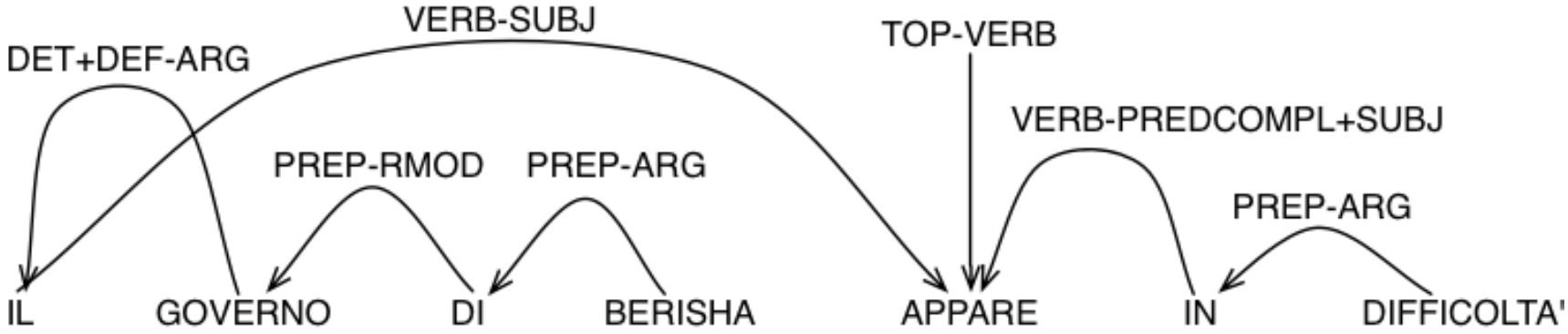
PCFG

- “the boy saw Jack with Bob under the table with a telescope”
- Several trees found
- But now it is possible to rank these trees:
  - The best tree: the most probable tree

Dependency Grammar: TUT

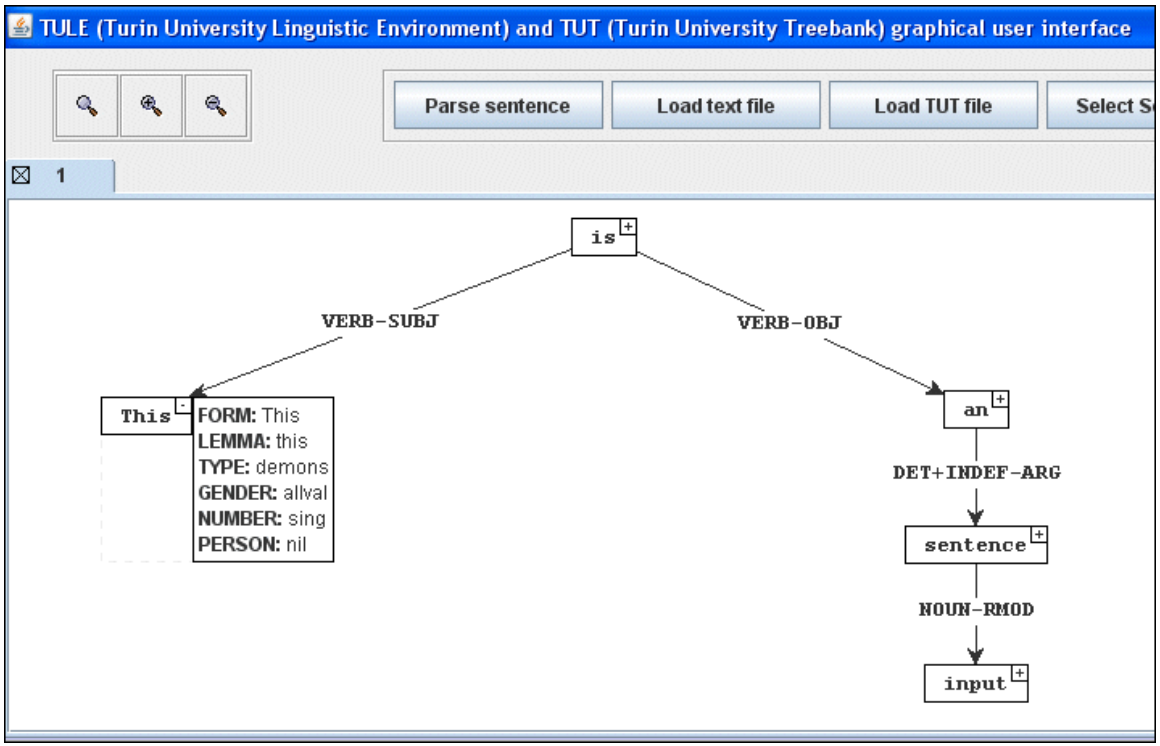
- TUT Treebank contains DG-tagged sentences

1 Il (IL ART DEF M SING) **[5;VERB-SUBJ]**
2 Governo (GOVERNO NOUN COMMON M SING) **[1;DET+DEF-ARG]**
3 di (DI PREP MONO) **[2;PREP-RMOD]**
4 Berisha (BERISHA NOUN PROPER) **[3;PREP-ARG]**
5 appare (APPARIRE VERB MAIN IND PRES INTRANS 3 SING) **[0;TOP-VERB]**
6 in (IN PREP MONO) **[5;VERB-PREDCOMPL+SUBJ]**
7 difficolta' (DIFFICOLTÀ NOUN COMMON F ALLVAL) **[6;PREP-ARG]**
8 . (#\. PUNCT) **[5;END]**



TULE

- Dependency grammar
- Client-server
  - Server: LISP-based parser
  - Client: Java-based GUI
- Multilanguage



Lexicalized PCFG

- "workers dumped sacks into a bin"
- PCFG:

  `VP -> VBD NP PP [p=3×10^-5]`
- Lexicalized PCFG:

  1. `VP(dumped) -> VBD(dumped) NP(cats) PP(into) [p=3×10^-11]`
  2. `VP(dumped) -> VBD(dumped) NP(sacks) PP(into) [p=3×10^-10]`
  3. `VP(dumped) -> VBD(dumped) NP(sacks) PP(above) [p=3×10^-12]`

  …

Parser using advanced stochastic models

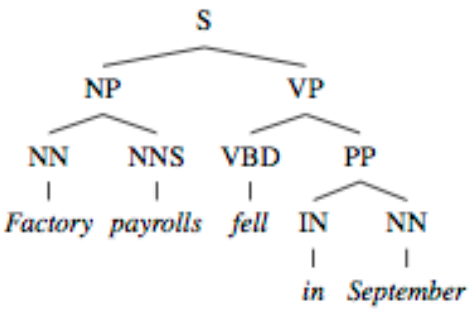
- Factored Lexicalized PCFG (Stanford)
- Lexicalized PCFG + Maximum Entropy reranking (es. Charniak)

Stanford Parser

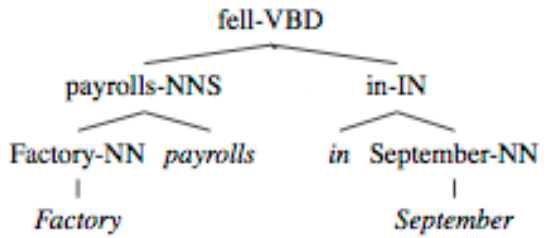
- Lexicalized PCFG are hard to train
  - Data sparsity
- Lexicalized parse tree can be seen as a combination of:
  - Unlexicalized parse tree
  - Dependency tree among words
- These trees can be trained separately
  - Reduce data sparsity

Stanford Parser

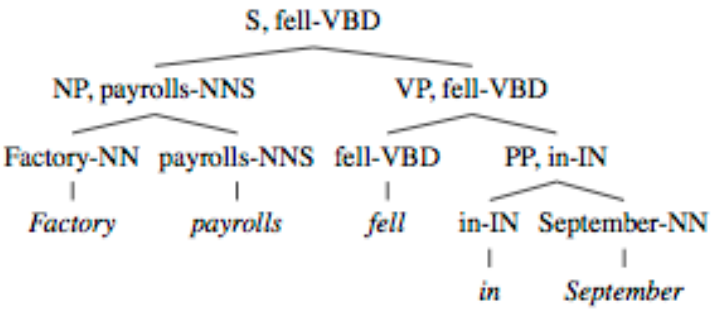
- Starting form a lexicalized Treebank:
  - Reconstruct unlexicalized PCFG
  - Extract headwords and build dependencies among them
- Parsing:
  - P(T): probability of a given unlexicalized parse tree
  - p(D): probability of a given dependency tree (sort of)
- Factored Lexicalized PCFG: p(T,D) = p(T)•p(D)



(a) PCFG Structure



(b) Dependency Structure



(c) Combined Structure

- Demo

Charniak parser

- Based on a Lexicalized PCFG
  - Returns the 50 most probable parse trees
- A Maximum Entropy models reranks the trees and selects the best one
  - 13 feature template, predicating on parse trees
  - A large number of feature instances!

Corpora

- Corpus: a collection of tagged texts
  - Human experts tag texts, by hand, setting the so-called *gold standard*
  - Used to train statistic models
- Well-known corpora, among others:
  - POS tags: Brown Corpus
  - Parsing: Penn Treebank

Brown corpus, an excerpt

The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent/jj primary/nn election/nn produced/vbd ``/`` no/at evidence/nn "/" that/cs any/dti irregularities/nns took/vbd place/nn ./.

Penn Treebank (chunk) an excerpt

“Pierre Vinken , 61 years old , will join the board
As a nonexecutive Director Nov. 29 .”

[Pierre/NNP Vinken/NNP]
,/,
[61/CD years/NNS]
old/JJ ,/, will/MD join/VB
[the/DT board/NN]
as/IN
[a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD]
./.

CoNLL corpus

- To train stochastic chunkers
- POS and chunk types

He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP
to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP

Penn Treebank (tree), an excerpt

“Pierre Vinken ,
61 years old ,
will join the board
As a nonexecutive
Director Nov. 29 .”

```
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken) )
    (, ,)
    (ADJP
      (NP (CD 61) (NNS years) )
      (JJ old) )
    (, ,) )
  (VP (MD will)
    (VP (VB join)
      (NP (DT the) (NN board) )
      (PP-CLR (IN as)
        (NP (DT a) (JJ nonexecutive) (NN director) ))
      (NP-TMP (NNP Nov.) (CD 29) )))
  (. .) )
```

How corpora are built

- Bootstrap
  1. Tag by hand a subset of the corpus
  2. Train a model
  3. Use the model to tag a larger subset of the corpus
  4. Revise and fix taggings
  5. Go to 2
- Kappa measure: agreement among human taggers
  - Human taggers do not fully agree, usually
  - We need a measure of such agreement

Kappa measure

- Compute agreement as:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

$P(A)$ is the observed agreement among the raters

$P(E)$ is the expected agreement (i.e., $P(E)$ is the probability that raters agree by chance)

- The values of the agreement are constrained to [−1, 1]

1="perfect agreement"
0="agreement is equal to chance"
-1="perfect disagreement"

Kappa measure

- Two raters or multi-raters
- Two ways to compute *P*(*E*):
  - Fixed-marginal studies
    - Raters know a priori the quantity of cases that should be distributed into each category (e.g., a rater is free to assign cases to categories as long as there will be a certain, predetermined amount of cases in each category in the end)
    - E.g.: Fleiss' Kappa, Siegel & Castellan's Kappa (multi-raters); Scott's Pi (two raters)
  - Free-marginal
    - Raters do not know a priori the quantities of cases that should be distributed into each category (e.g., raters are free to assign cases to categories with no limits on how many cases must go into each category)
    - E.g. Randolph's K_{free} (multi-raters); Cohen's K (two raters)

$$K_{free} \stackrel{\text{def}}{=} \frac{\frac{1}{N \cdot n \cdot (n-1)} \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{k} n_{i,j}^2 - N \cdot n \right) - \frac{1}{k}}{1 - \frac{1}{k}}$$

N: number of cases to rate; n: number of raters; k: number of categories

$n_{i,j}$: number of raters who assigned the i-th case to the j-th category (the so-called *agreement table)*

NLP Environments

- GATE
- UIMA
- Stanford Core
- OpenNLP
- NLTK

REFERENCES

POS Tagging

- TreeTagger:
  - http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
- Stanford
  - http://nlp.stanford.edu/software/index.shtml
- FreeLing (and parser, morpho analyzer, …)
  - http://nlp.lsi.upc.edu/freeling/

Shallow parsers

- CHAOS
  - http://ai-nlp.info.uniroma2.it/external/chaosproject/
- TreeTagger:
  - http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
- Illinois
  - http://cogcomp.cs.illinois.edu/page/software_view/13

Full parsers

- Stanford parser
  - http://nlp.stanford.edu/software/lex-parser.shtml
- Charniak parser
  - http://www.cs.brown.edu/~ec/
- NLTK (actually, contains a lot of tools…)
  - http://www.nltk.org
- TULE
  - http://www.tule.di.unito.it/

Corpora/1

- Linguistic Data Consortium
  - http://www.ldc.upenn.edu
- European Language Resources Association (ELRA)
  - http://www.icp.grenet.fr/ELRA/
- Int. Computer Archive of Modern English (ICAME)
  - http://nora.hd.uib.no/icame.html
- Oxford Text Archive (OTA)
  - http://ota.ahds.ac.uk/
- Child Language Data Exchange System (CHILDES)
  - http://childes.psy.cmu.edu/

Corpora/2

- NLTK_lite (directory **`corpora`**)
  - Small samples of: Penn Treebank, Brown, ecc.
- Penn Treebank:
  - http://www.cis.upenn.edu/~treebank/
- Brown Corpus:
  - http://en.wikipedia.org/wiki/Brown_Corpus
- American National Corpus:
  - http://americannationalcorpus.org/
- British National Corpus:
  - http://www.natcorp.ox.ac.uk/
- Corpus e Lessico di Frequenza dell'Italiano Scritto:
  - http://alphalinguistica.sns.it/CoLFIS/CoLFIS_Presentazione.htm