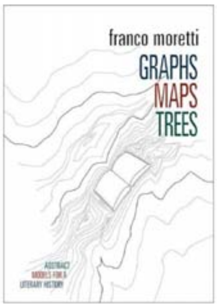
# Natural Language Processing as Philology

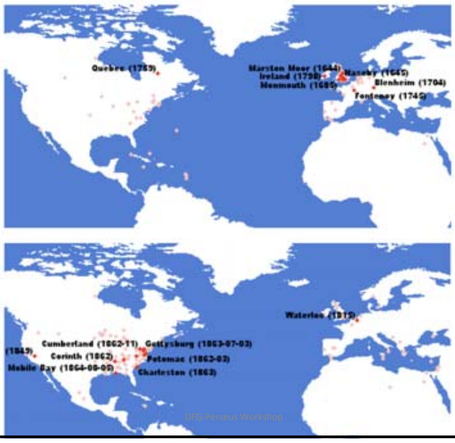
David Smith
Department of Computer Science
UMass Amherst

*Roman Jakobson*

Philology is the art of reading slowly.

*Franco Moretti*

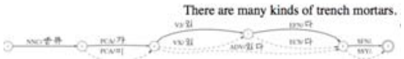


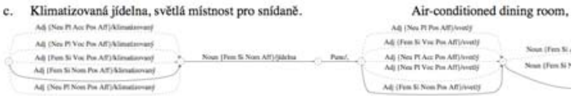


# NLP Highlights

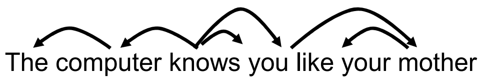
- Efficient algorithms for linguistic inference
  - Joint inference across many layers of language
- Adaptation to new languages and domains
- Inferring structure in large, noisy collections
  - Detecting text reuse and linkage
  - Inferring temporal sequence of events

# Morphological Disambiguation

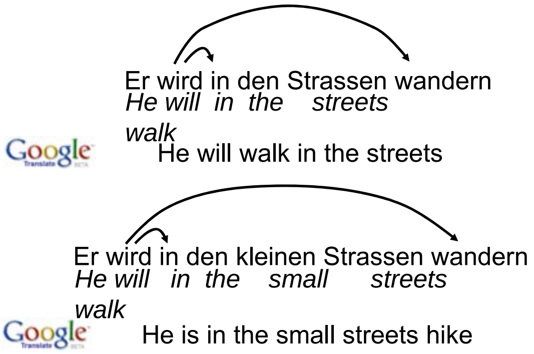



c. Klimatizovaná jídelna, světlá místnost pro snídaně.
Air-conditioned dining room,

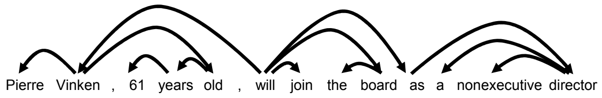
# Bare-Bones Dependency Structure



# Syntax in Translation


Er wird in den Strassen wandern
*He will in the streets walk*
He will walk in the streets
Er wird in den kleinen Strassen wandern
*He will in the small streets walk*
He is in the small streets hike

# Who Did What To Whom?

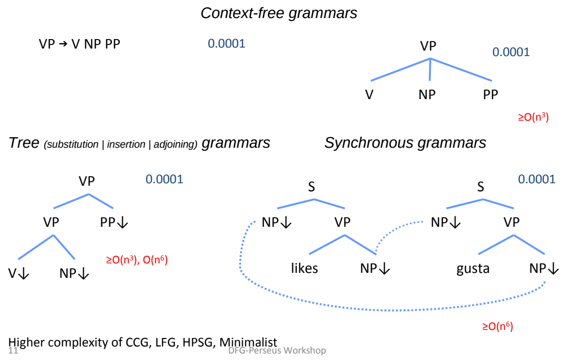

Pierre Vinken , 61 years old , will join the board as a nonexecutive director

PropBank join predicate

| ARG0 | ARG1 | ARG-PRD |
|---|---|---|
| Vinken | board | director |

## SYNTAX AND PARSING

# Grammars and Trees



Higher complexity of CCG, LFG, HPSG, Minimalist

# Dependency Parsing as Graph Inference

Raw sentence

He reckons the current account deficit will narrow to only 1.8 billion in September.

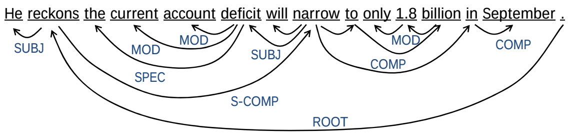
Part-of-speech tagging

POS-tagged sentence

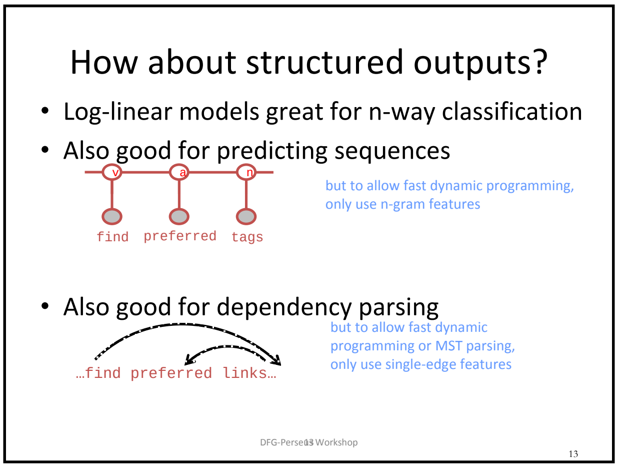
He reckons the current account deficit will narrow to only 1.8 billion in September.
PRP VBZ DT JJ NN NN MD VB TO RB CD CD IN NNP .

Word dependency parsing

Word dependency parsed sentence


He reckons the current account deficit will narrow to only 1.8 billion in September .
SUBJ
MOD
MOD
SUBJ
MOD
COMP
SPEC
COMP
S-COMP
ROOT

slide adapted from Yuji Matsumoto



## How about structured outputs?

- Log-linear models great for n-way classification
- Also good for predicting sequences

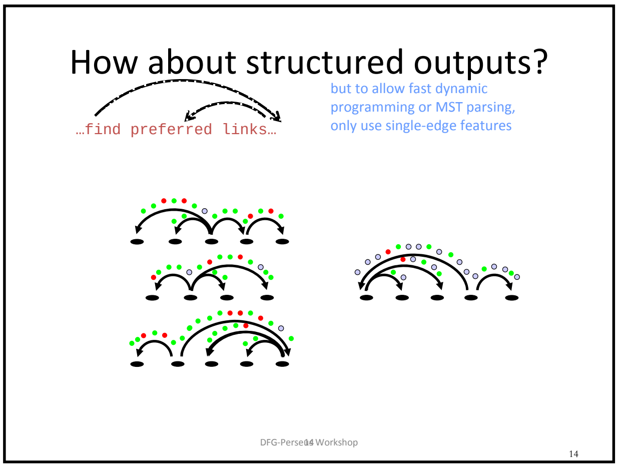
v
a
n
find preferred tags

but to allow fast dynamic programming, only use n-gram features

- Also good for dependency parsing

…find preferred links…

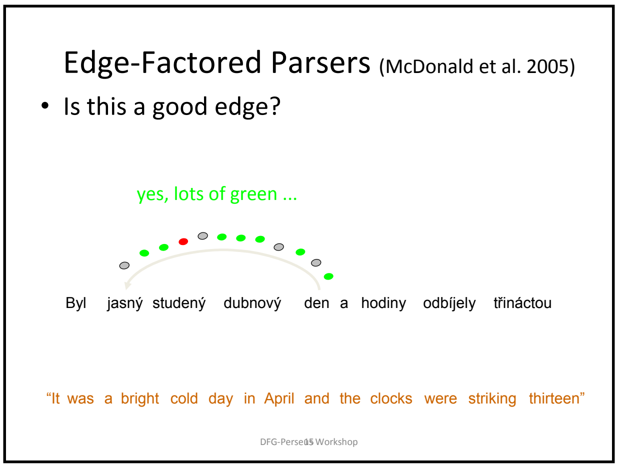
but to allow fast dynamic programming or MST parsing, only use single-edge features



## How about structured outputs?

…find preferred links…

but to allow fast dynamic programming or MST parsing, only use single-edge features



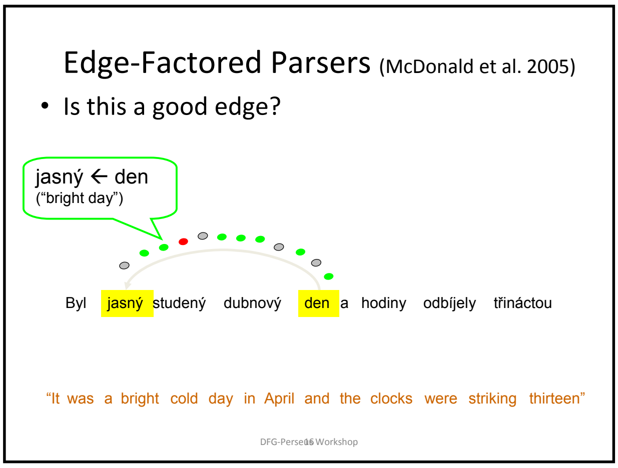
## Edge-Factored Parsers (McDonald et al. 2005)

- Is this a good edge?

yes, lots of green ...

Byl jasný studený dubnový den a hodiny odbíjely třináctou

“It was a bright cold day in April and the clocks were striking thirteen”



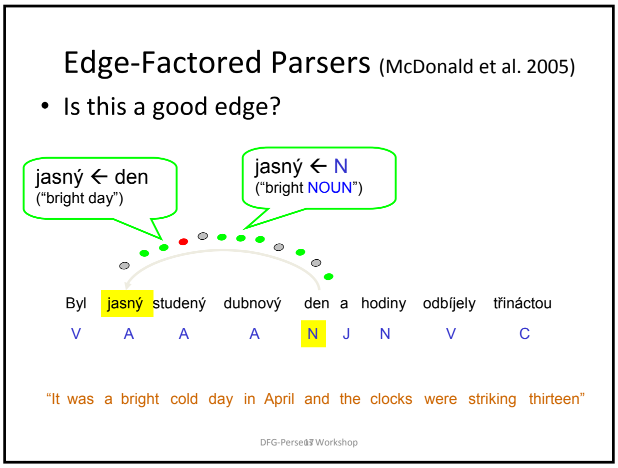
## Edge-Factored Parsers (McDonald et al. 2005)

- Is this a good edge?

jasný ← den
(“bright day”)

Byl jasný studený dubnový den a hodiny odbíjely třináctou

“It was a bright cold day in April and the clocks were striking thirteen”



## Edge-Factored Parsers (McDonald et al. 2005)

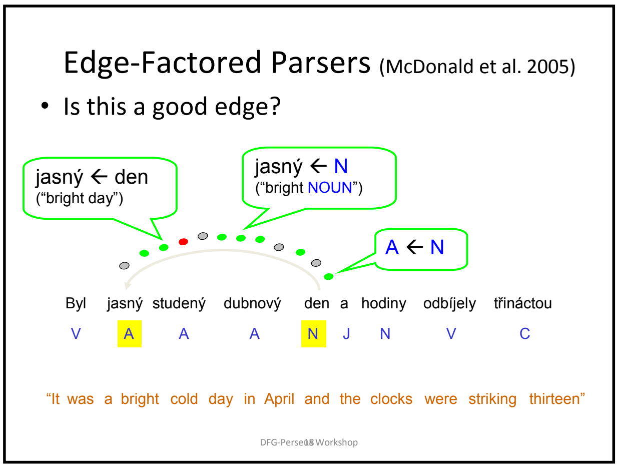
- Is this a good edge?

jasný ← den
(“bright day”)

jasný ← N
(“bright NOUN”)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |

“It was a bright cold day in April and the clocks were striking thirteen”



## Edge-Factored Parsers (McDonald et al. 2005)

- Is this a good edge?

jasný ← den
(“bright day”)

jasný ← N
(“bright NOUN”)

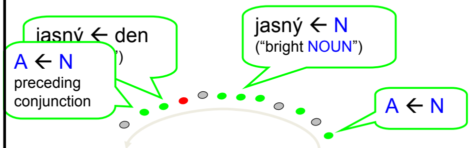
A ← N

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |

“It was a bright cold day in April and the clocks were striking thirteen”

## Edge-Factored Parsers (McDonald et al. 2005)

- Is this a good edge?



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |

"It was a bright cold day in April and the clocks were striking thirteen"

## Edge-Factored Parsers (McDonald et al. 2005)

- How about this competing edge?

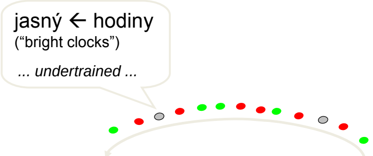
not as good, lots of red ...

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |

"It was a bright cold day in April and the clocks were striking thirteen"

## Edge-Factored Parsers (McDonald et al. 2005)

- How about this competing edge?

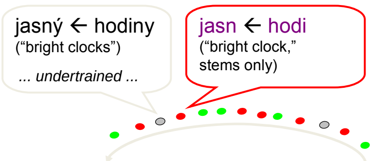


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |

"It was a bright cold day in April and the clocks were striking thirteen"

## Edge-Factored Parsers (McDonald et al. 2005)

- How about this competing edge?

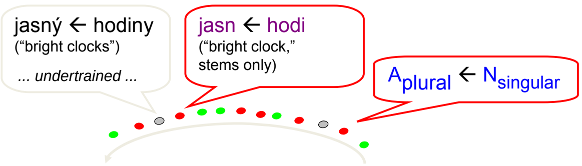


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |
| byl | jasn | stud | dubn | den | a | hodi | odbí | třin |

"It was a bright cold day in April and the clocks were striking thirteen"

## Edge-Factored Parsers (McDonald et al. 2005)

- How about this competing edge?

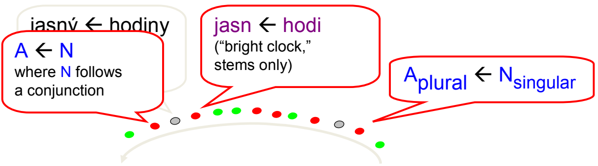


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |
| byl | jasn | stud | dubn | den | a | hodi | odbí | třin |

"It was a bright cold day in April and the clocks were striking thirteen"

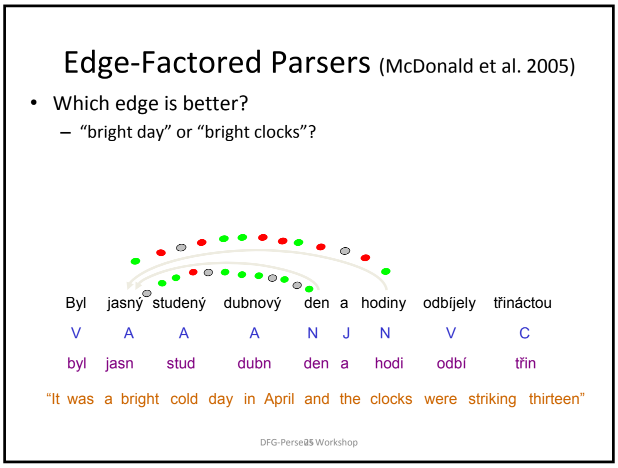
## Edge-Factored Parsers (McDonald et al. 2005)

- How about this competing edge?



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
| V | A | A | A | N | J | N | V | C |
| byl | jasn | stud | dubn | den | a | hodi | odbí | třin |

"It was a bright cold day in April and the clocks were striking thirteen"

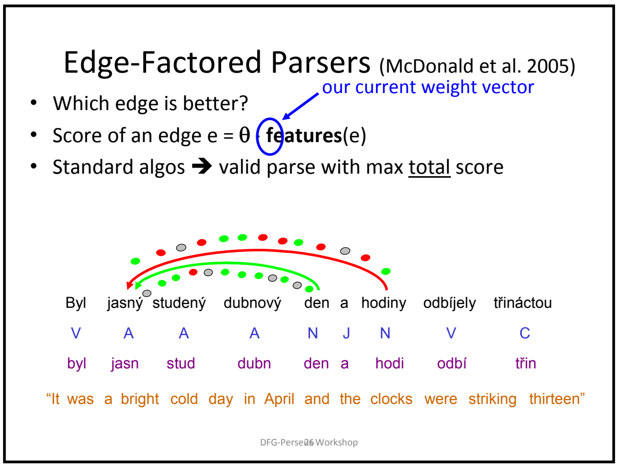


## Edge-Factored Parsers (McDonald et al. 2005)

- Which edge is better?
  - "bright day" or "bright clocks"?

| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
|---|---|---|---|---|---|---|---|---|
| V | A | A | A | N | J | N | V | C |
| byl | jasn | stud | dubn | den | a | hodi | odbí | třin |

"It was a bright cold day in April and the clocks were striking thirteen"



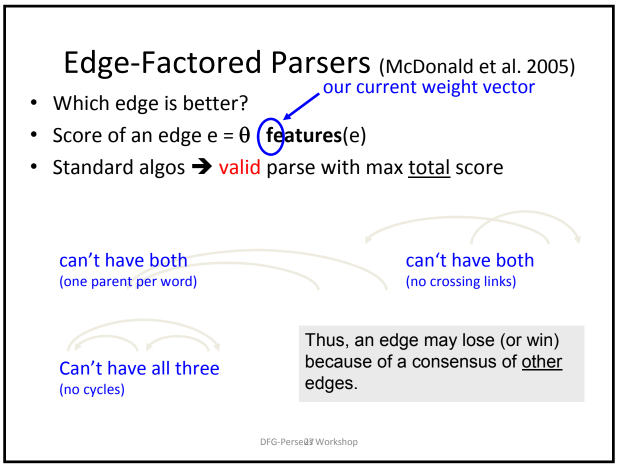
## Edge-Factored Parsers (McDonald et al. 2005)

- Which edge is better?
- Score of an edge e = θ · **features**(e)
- Standard algos ➔ valid parse with max total score

our current weight vector

| Byl | jasný | studený | dubnový | den | a | hodiny | odbíjely | třináctou |
|---|---|---|---|---|---|---|---|---|
| V | A | A | A | N | J | N | V | C |
| byl | jasn | stud | dubn | den | a | hodi | odbí | třin |

"It was a bright cold day in April and the clocks were striking thirteen"



## Edge-Factored Parsers (McDonald et al. 2005)

- Which edge is better?
- Score of an edge e = θ · **features**(e)
- Standard algos ➔ valid parse with max total score

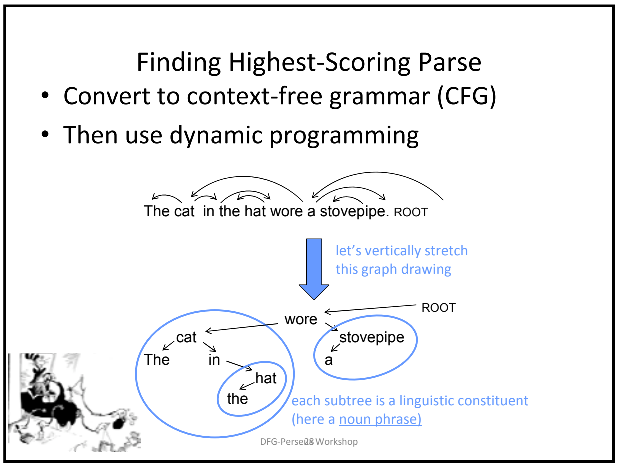
our current weight vector

can't have both
(one parent per word)

can't have both
(no crossing links)

Can't have all three
(no cycles)

Thus, an edge may lose (or win) because of a consensus of other edges.

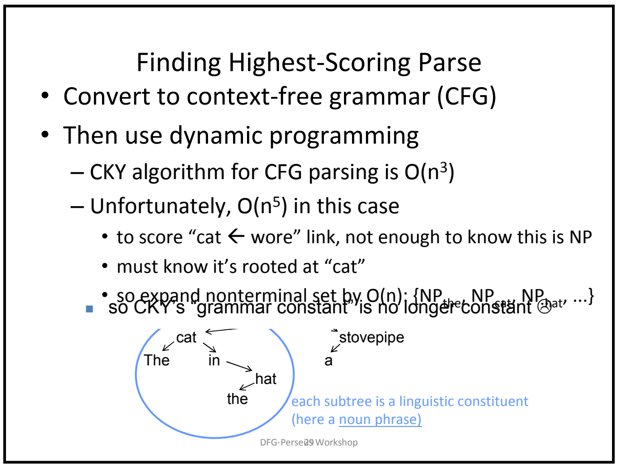


## Finding Highest-Scoring Parse

- Convert to context-free grammar (CFG)
- Then use dynamic programming

The cat in the hat wore a stovepipe. ROOT
let's vertically stretch this graph drawing
ROOT
wore
cat
The
in
hat
the
stovepipe
a

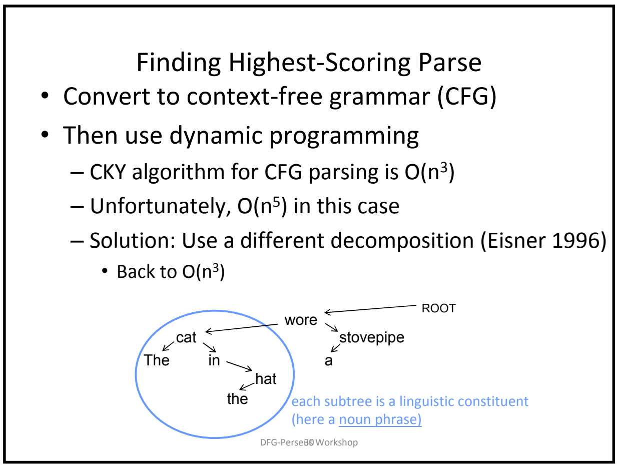
each subtree is a linguistic constituent (here a noun phrase)



## Finding Highest-Scoring Parse

- Convert to context-free grammar (CFG)
- Then use dynamic programming
  - CKY algorithm for CFG parsing is O($n^3$)
  - Unfortunately, O($n^5$) in this case
    - to score "cat ← wore" link, not enough to know this is NP
    - must know it's rooted at "cat"
    - so expand nonterminal set by O(n): {$NP_{the}$, $NP_{cat}$, $NP_{hat}$, ...}
    - so CKY's "grammar constant" is no longer constant ☹

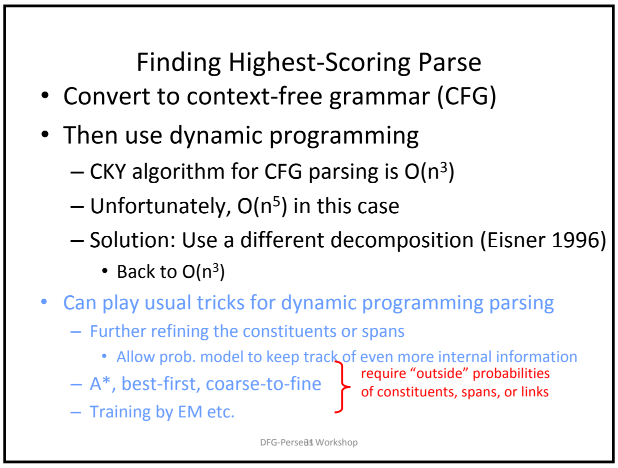
each subtree is a linguistic constituent (here a noun phrase)



## Finding Highest-Scoring Parse

- Convert to context-free grammar (CFG)
- Then use dynamic programming
  - CKY algorithm for CFG parsing is O($n^3$)
  - Unfortunately, O($n^5$) in this case
  - Solution: Use a different decomposition (Eisner 1996)
    - Back to O($n^3$)

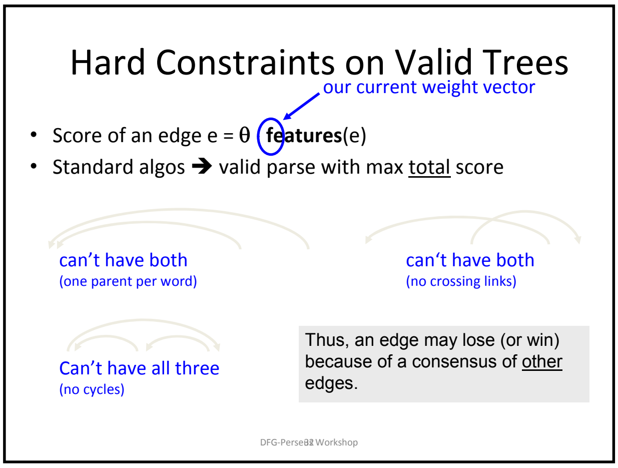
each subtree is a linguistic constituent (here a noun phrase)



## Finding Highest-Scoring Parse

- Convert to context-free grammar (CFG)
- Then use dynamic programming
  - CKY algorithm for CFG parsing is $O(n^3)$
  - Unfortunately, $O(n^5)$ in this case
  - Solution: Use a different decomposition (Eisner 1996)
    - Back to $O(n^3)$
- Can play usual tricks for dynamic programming parsing
  - Further refining the constituents or spans
    - Allow prob. model to keep track of even more internal information
  - A*, best-first, coarse-to-fine
  - Training by EM etc.

require "outside" probabilities
of constituents, spans, or links



## Hard Constraints on Valid Trees

our current weight vector

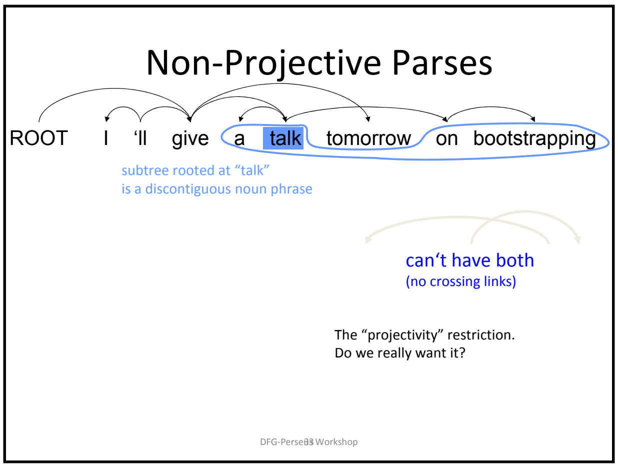
- Score of an edge e = θ · **features**(e)
- Standard algos ➔ valid parse with max <u>total</u> score

can't have both
(one parent per word)

can't have both
(no crossing links)

Can't have all three
(no cycles)

Thus, an edge may lose (or win) because of a consensus of <u>other</u> edges.



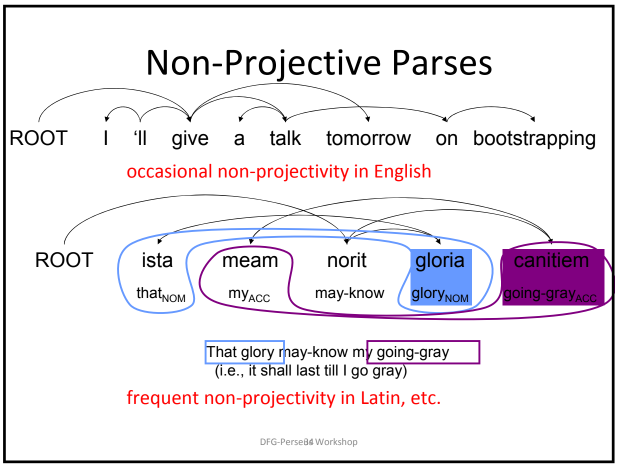
## Non-Projective Parses

ROOT I 'll give a talk tomorrow on bootstrapping

subtree rooted at "talk"
is a discontiguous noun phrase

can't have both
(no crossing links)

The "projectivity" restriction.
Do we really want it?



## Non-Projective Parses

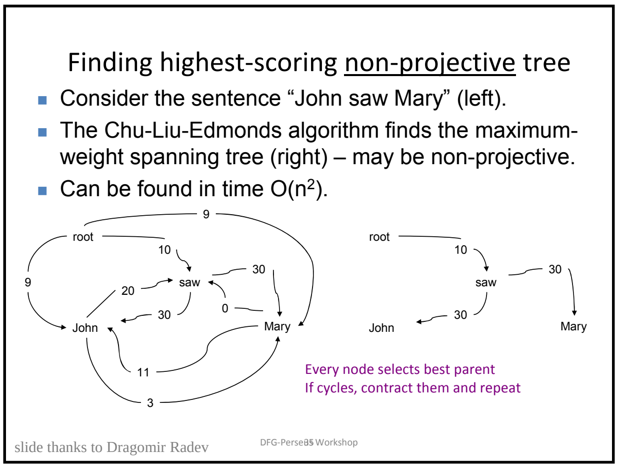
ROOT I 'll give a talk tomorrow on bootstrapping

occasional non-projectivity in English

| ROOT | ista | meam | norit | gloria | canitiem |
|---|---|---|---|---|---|
|  | that$_{NOM}$ | my$_{ACC}$ | may-know | glory$_{NOM}$ | going-gray$_{ACC}$ |

That glory may-know my going-gray
(i.e., it shall last till I go gray)

frequent non-projectivity in Latin, etc.

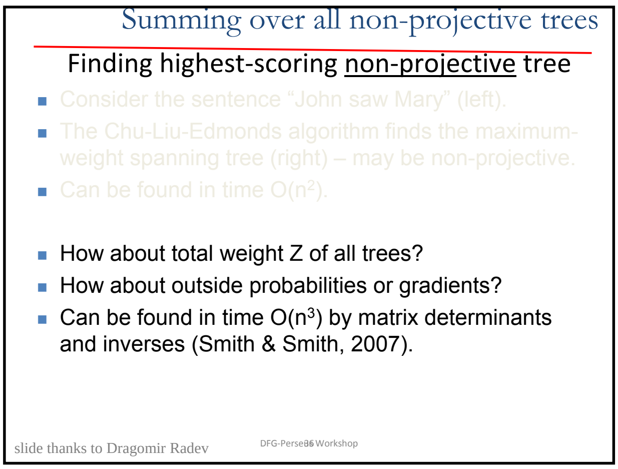


## Finding highest-scoring <u>non-projective</u> tree

- Consider the sentence "John saw Mary" (left).
- The Chu-Liu-Edmonds algorithm finds the maximum-weight spanning tree (right) – may be non-projective.
- Can be found in time $O(n^2)$.

Every node selects best parent
If cycles, contract them and repeat

slide thanks to Dragomir Radev



## Summing over all non-projective trees

## Finding highest-scoring <u>non-projective</u> tree

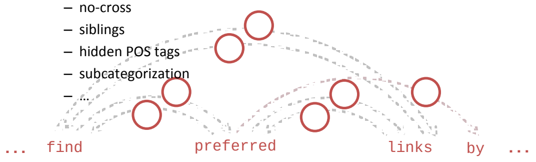
- Consider the sentence "John saw Mary" (left).
- The Chu-Liu-Edmonds algorithm finds the maximum-weight spanning tree (right) – may be non-projective.
- Can be found in time $O(n^2)$.

- How about total weight Z of all trees?
- How about outside probabilities or gradients?
- Can be found in time $O(n^3)$ by matrix determinants and inverses (Smith & Smith, 2007).

slide thanks to Dragomir Radev

# Local factors for parsing

- So what factors shall we multiply to define parse probability?
  - Unary factors to evaluate each link in isolation
  - Global TREE factor to require that the links form a legal tree
    - this is a “hard constraint”: factor is either 0 or 1
  - Second-order effects: factors on 2 variables
    - grandparent
    - no-cross
    - siblings
    - hidden POS tags
    - subcategorization
    - …

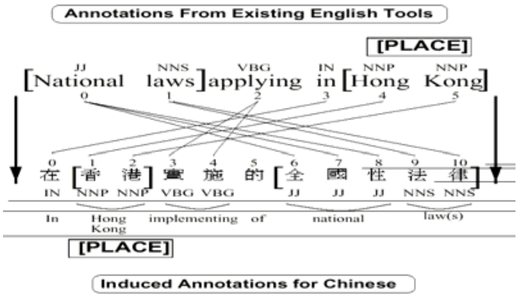


# Future Opportunities

- Efficiently modeling more hidden structure
  - POS tags, link roles, secondary links (DAG-shaped parses)
- Beyond dependencies
  - Constituency parsing, traces, lattice parsing
- Beyond parsing
  - Alignment, translation
  - Bipartite matching and network flow
  - Joint decoding of parsing and other tasks (IE, MT, reasoning ...)
- Modeling sentence processing
  - BP is a *parallel, anytime* process

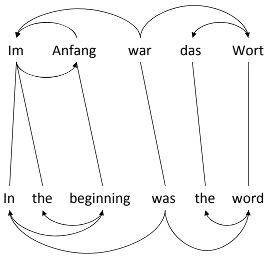
**ADAPTATION**

# Projecting Hidden Structure


Annotations From Existing English Tools
[PLACE]
JJ
NNS
VBG
IN
NNP
NNP
[National laws]applying in [Hong Kong]
0
1
2
3
4
5
0
1
2
3
4
5
6
7
8
9
10
在 [香 港] 實 施 的 [全 國 性 法 律]
IN NNP NNP VBG VBG JJ JJ JJ NNS NNS
In
Hong Kong
implementing
of
national
law(s)
[PLACE]
Induced Annotations for Chinese

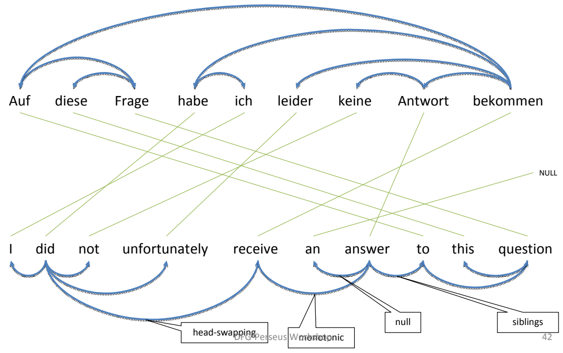
*Yarowsky & Ngai ‘01*

# Projection

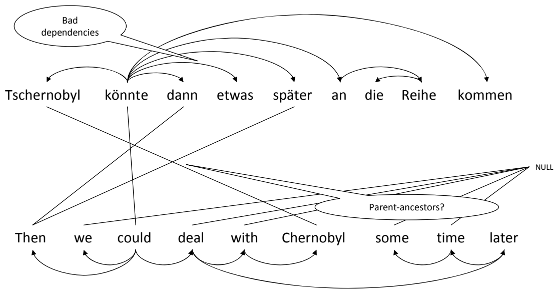


- Train with bitext
- Parse one side
- Align words
- Project dependencies
- Many to one links?
- Invalid trees?
- Hwa et al.: fix-up rules
- Ganchev et al.: trust only some links

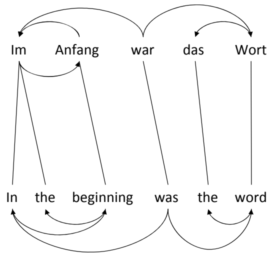
# Divergent Projection



# Free Translation

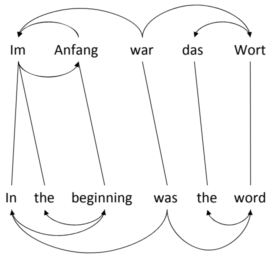


# What's Wrong with Projection?



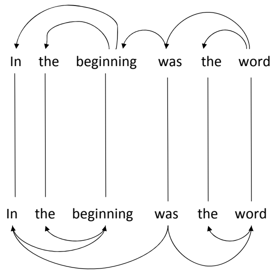
- Hwa et al. Chinese data:
  - 38.1% F1 after projection
  - Only 26.3% with automatic English parses
  - Cf. 35.9% for attach right!
  - 52.4% after fix-up rules
- Only 1-to-1 alignments:
  - 68% precision
  - 11% recall

# Projection



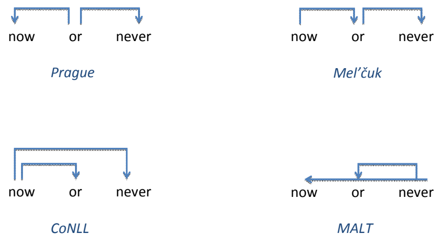
- Different languages
- Similar meaning
- Divergent syntax

# Adaptation

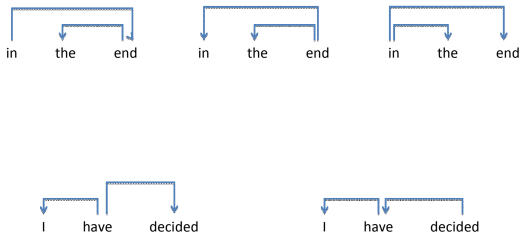


- Same sentence
- Divergent syntax

# A Lack of Coordination



# Prepositions and Auxiliaries



# Adaptation Recipe

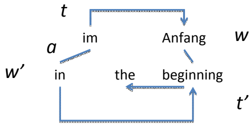
- Acquire (a few) trees in target domain
- Run source-domain parser on training set
- Train parser with features for:
  - Target tree alone
  - Source and target trees together
- Parse test set with:
  - Source-domain parser
  - Target-domain parser

# Why?

- Why not just modify source treebank?
- Source parser could be a black box
  - Or rule based
- Vastly shorter training times with a small target treebank
  - Linguists can quickly explore alternatives
  - Don't need dozens of rules
- Other benefits of stacking
- And sometimes, divergence is very large

**MODEL STRUCTURE**

# What We're Modeling



*This paper*

*Generative* $p(t,a,w \mid t',w')$

*Conditional* $p(t \mid t',a,w,w')$

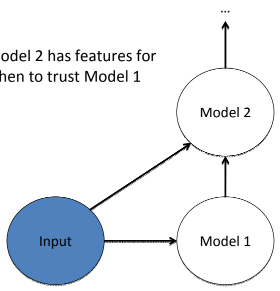
*Ongoing work*

$p(t,t',a \mid w,w')$

$$s(t,t',a,w,w') = \sum_i \theta_i f_i(t,w) + \sum_j \theta_j g_j(t,t',a,w,w')$$

# Stacking

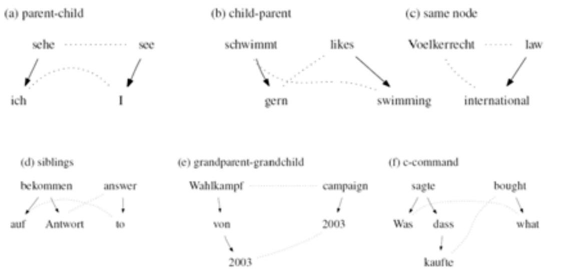
Model 2 has features for when to trust Model 1



# Quasi-Synchronous Grammar

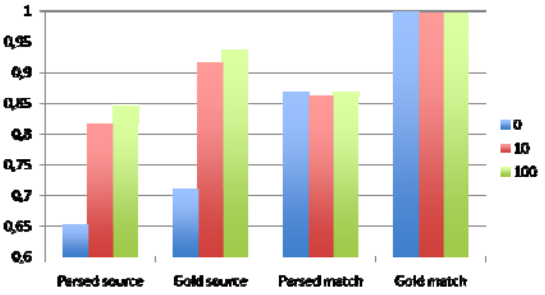
- Generative or conditional monolingual model of target language or tree
- Condition target trees on source structure
- Applications to
  - Alignment (D. Smith & Eisner '06)
  - Question Answering (Wang, N. Smith, Mitamura '07)
  - Paraphrase (Das & N. Smith '09)
  - Translation (Gimpel & N. Smith '09)

## Dependency Relations



+ “none of the above”

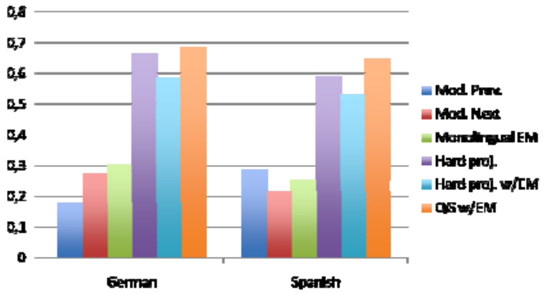
## Adaptation Results



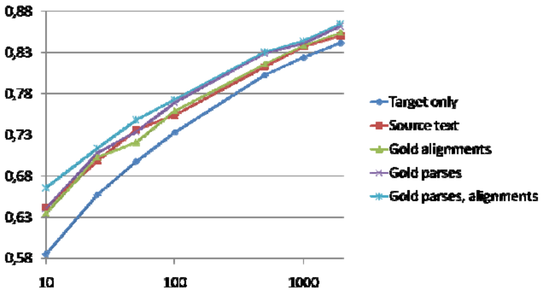
*Different PTB dep. conversions*

*See paper for more results*

## Unsupervised Projection

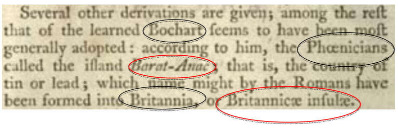


## Supervised Projection



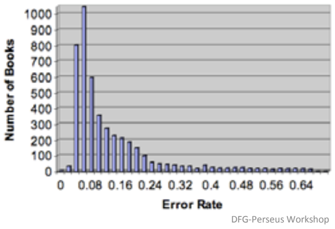
**MINING A MILLION BOOKS**

## Mining Information from Books

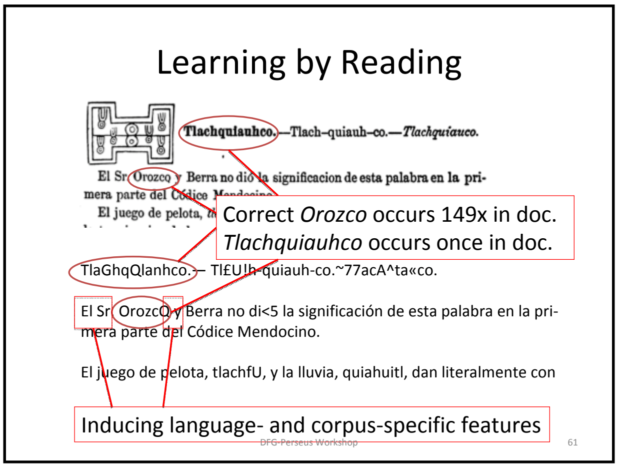


Several other derivations are given; among the reſt that of the learned Bochart ſeems to have been moſt generally adopted: according to him, the Phœnicians called the iſland *Barat-Anac*, that is, the country of tin or lead; which name might by the Romans have been formed into Britannia, or Britannicæ inſulæ.

Bochart
Phoenicians
Barat-Anac
Britannia
Britannicae infulae



- Modern OCR, several errors/page
- Names are worse:
    - In one study, 35% names incorrectly transcribed
- Errors propagate to later steps
- Train language models and name extractors on noisy corpus



## Learning by Reading

Tlachquiauhco.—Tlach–quiauh–co.—*Tlachquiauco.*

El Sr. Orozco y Berra no dió la significacion de esta palabra en la primera parte del Códice Mendocino.

El juego de pelota, *tl*

Correct *Orozco* occurs 149x in doc.
*Tlachquiauhco* occurs once in doc.

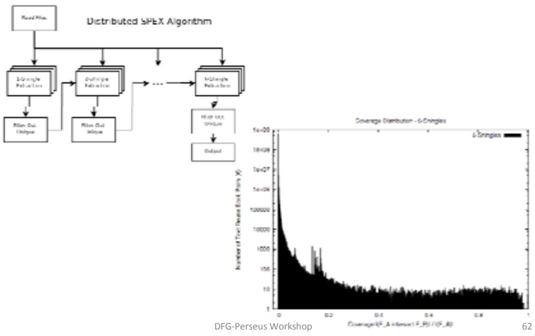
TlaGhqQlanhco.— Tl£U!h-quiauh-co.~77acA^ta«co.

El Sr. OrozcQ y Berra no di<5 la significación de esta palabra en la primera parte del Códice Mendocino.

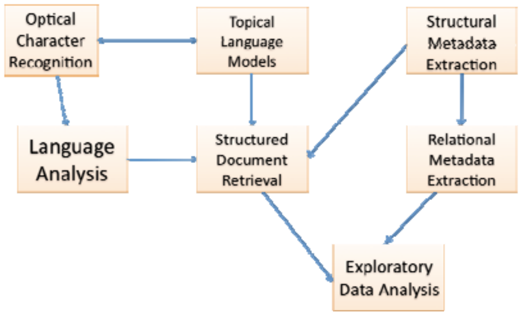
El juego de pelota, tlachfU, y la lluvia, quiahuitl, dan literalmente con

Inducing language- and corpus-specific features

## Reuse & Quotation



## UMass Book Search


Optical Character Recognition
Topical Language Models
Structural Metadata Extraction
Language Analysis
Structured Document Retrieval
Relational Metadata Extraction
Exploratory Data Analysis