research = re- + *kikro-

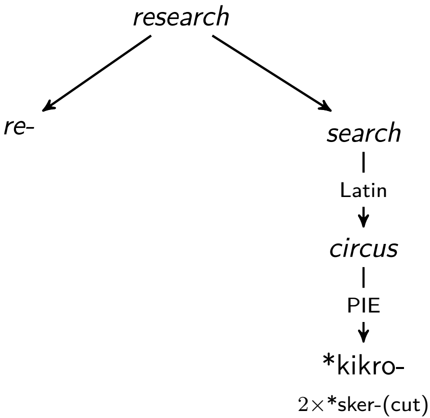
Revisit the Development of Probabilistic Symbol-Refined Grammars

Weiwei Sun

Wangxuan Institute of Computer Technology
Peking University

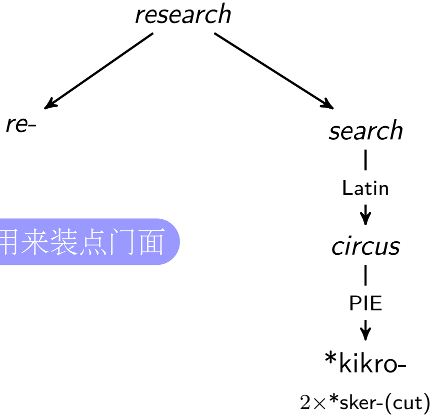
@StudentWorkshop.NLPCC2019

The etymological meaning of *research*



from http://www.etymonline.com/

The etymological meaning of *research*



知识可以用来装点门面

from http://www.etymonline.com/

This talk

A historical introduction to one type of parsing model

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? Simple PCFG Again

This talk

A historical introduction to one type of parsing model

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? Simple PCFG Again

some thoughts

Outline

Let's start from *Friends*

Let's start from *Friends*

Well, ya know how I always wanted to go out with Chip Matthews in high school? Well, tonight I actually went out with Chip Matthews in high school.

go out with Chip Matthews in high school.

- *go out in high school.*
- *Chip Matthews in high school.*

Let's start from *Friends*

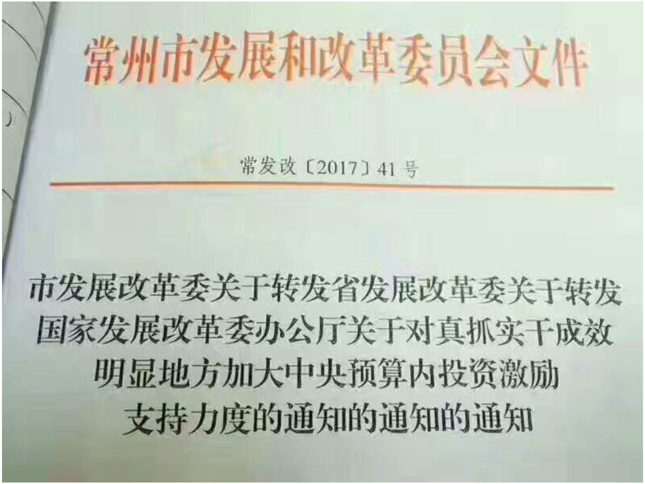
Well, ya know how I always wanted to go out with Chip Matthews in high school? Well, tonight I actually went out with Chip Matthews in high school.

go out with Chip Matthews in high school.

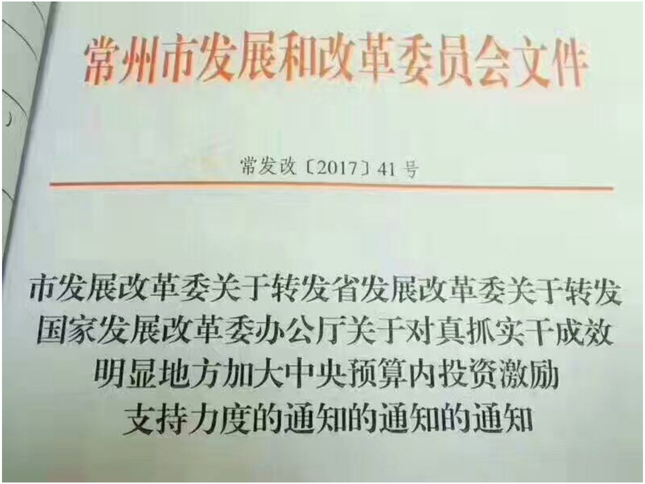
- *go out in high school.*
- *Chip Matthews in high school.*

`With syntax, we can distinguish between them.`

NL sentences could be really long.



NL sentences could be really long.



With syntax, we can understand very long sentences.

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

0110101000001001111001100110011111110011101111001100100100001

Structuring a Sentence

01010101010101010101010101010101010101010101010101010101010101010101010

011010100000100111100110011001111110011101111001100100100001

$$\sqrt{2} - 1$$

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

0110101000001001111001100110011111110011101111001100100100001

$$\sqrt{2} - 1$$

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

0110101000001001111001100110011111110011101111001100100100001

$$\sqrt{2} - 1$$

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

0110101000001001111001100110011111110011101111001100100100001

$$\sqrt{2} - 1$$

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知

Structuring a Sentence

0101010101010101010101010101010101010101010101010101010101010

0110101000001001111001100110011111110011101111001100100100001

$$\sqrt{2} - 1$$

市发展改革委关于转发省发展改革委关于转发国家发展改革委办公厅关于对真抓实干成效明显地方加大中央预算内投资激励支持力度的通知的通知的通知

Phrase structure

The basic idea

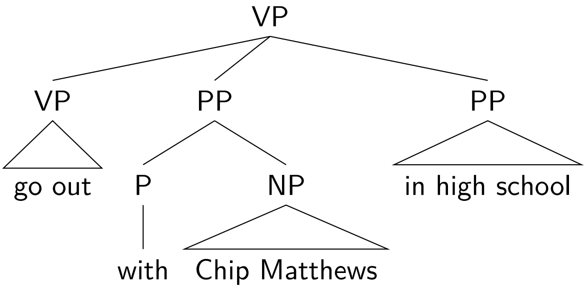
Phrase structure organizes words into nested constituents, which can be represented as **a tree**.

Phrase structure

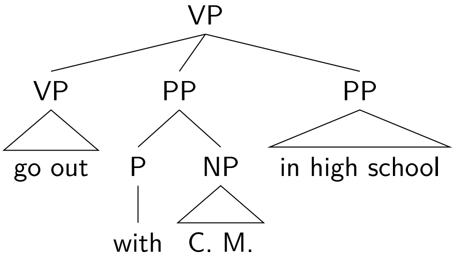
The basic idea

Phrase structure organizes words into nested constituents, which can be represented as **a tree**.

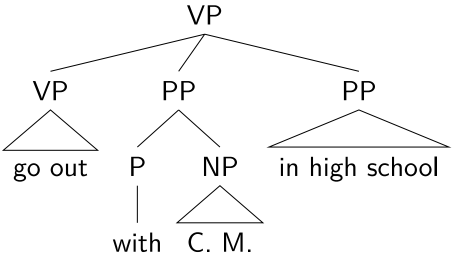
Example

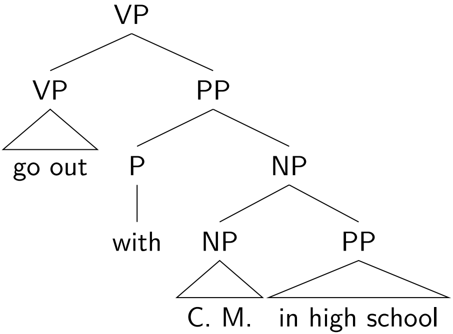


Different structures; different meaning

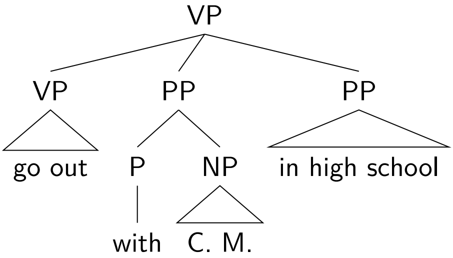


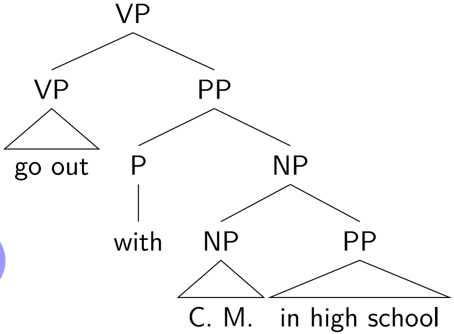
Different structures; different meaning





Different structures; different meaning

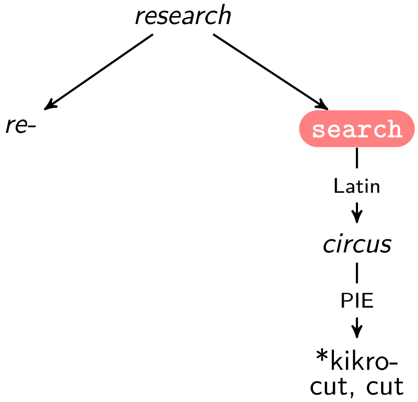




Parsing
To get trees automatically

Outline

search



Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0
Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S $\rightarrow$ NP VP	1.0	NP $\rightarrow$ AdjP NP	0.5
VP $\rightarrow$ V AdvP	1.0	NP $\rightarrow$ N	0.5
AdvP $\rightarrow$ Adv	1.0	AdjP $\rightarrow$ Adj	1.0

Adj $\rightarrow$ colorless	0.5	Adj $\rightarrow$ green	0.5
N $\rightarrow$ ideas	1.0	V $\rightarrow$ sleep	1.0
Adv $\rightarrow$ furiously	1.0		

4. S

We can derive the structure of a string.

- S $\Rightarrow$ NP VP   1.0
  $\Rightarrow$ N VP   0.5

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0
  ⇒ ideas V AdvP 1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0
  ⇒ ideas V AdvP 1.0
  ⇒ ideas sleep AdvP 1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0
  ⇒ ideas V AdvP 1.0
  ⇒ ideas sleep AdvP 1.0
  ⇒ ideas sleep Adv 1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP   1.0
  ⇒ N VP   0.5
  ⇒ ideas VP   1.0
  ⇒ ideas V AdvP   1.0
  ⇒ ideas sleep AdvP   1.0
  ⇒ ideas sleep Adv   1.0
  ⇒ ideas sleep furiously   1.0

Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

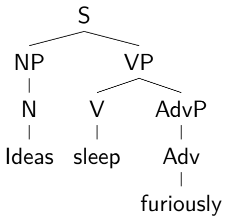
S → NP VP	1.0	NP → AdjP NP	0.5
VP → V AdvP	1.0	NP → N	0.5
AdvP → Adv	1.0	AdjP → Adj	1.0

Adj → colorless	0.5	Adj → green	0.5
N → ideas	1.0	V → sleep	1.0
Adv → furiously	1.0		

4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0
  ⇒ ideas V AdvP 1.0
  ⇒ ideas sleep AdvP 1.0
  ⇒ ideas sleep Adv 1.0
  ⇒ ideas sleep furiously 1.0



Probabilistic Context-Free Grammar

1. $V = \{\text{S}, \text{NP}, \text{VP}, \text{AdjP}, \text{AdvP}\} \cup \{\text{N}, \text{V}, \text{Adj}, \text{Adv}\}$
2. $T = \{$colorless, green, ideas, sleep, furiously$\}$
3. P

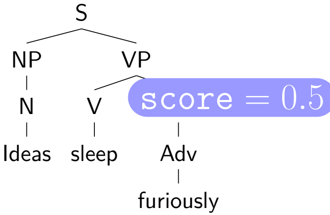
S → NP VP	1.0		NP → AdjP NP	0.5	
VP → V AdvP	1.0		NP → N	0.5	
AdvP → Adv	1.0		AdjP → Adj	1.0	

Adj → colorless	0.5		Adj → green	0.5
N → ideas	1.0		V → sleep	1.0
Adv → furiously	1.0			

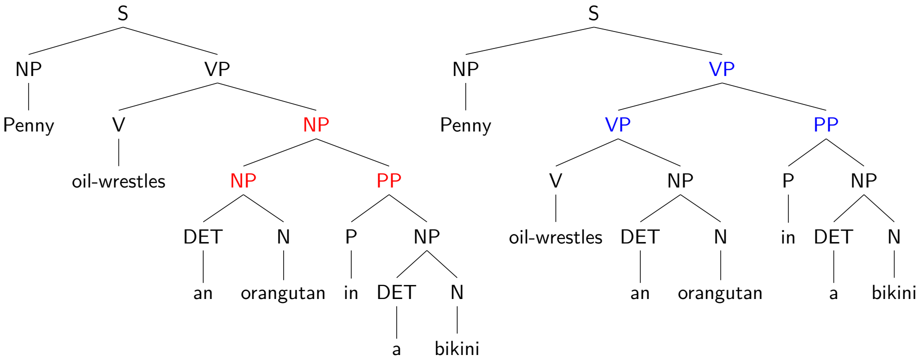
4. S

We can derive the structure of a string.

- S ⇒ NP VP 1.0
  ⇒ N VP 0.5
  ⇒ ideas VP 1.0
  ⇒ ideas V AdvP 1.0
  ⇒ ideas sleep AdvP 1.0
  ⇒ ideas sleep Adv 1.0
  ⇒ ideas sleep furiously 1.0

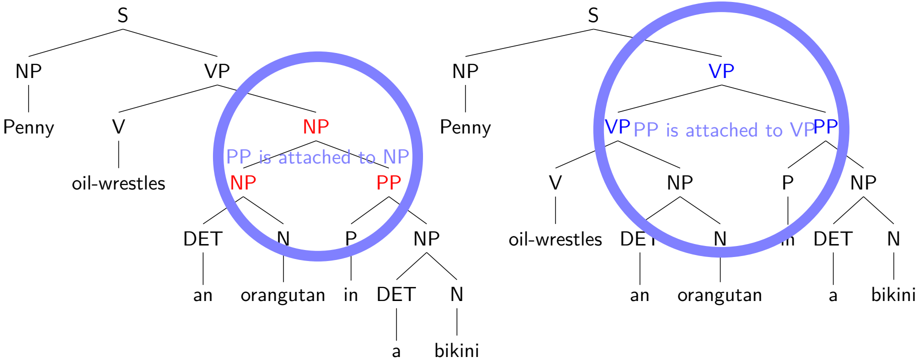


Naive application is not successful



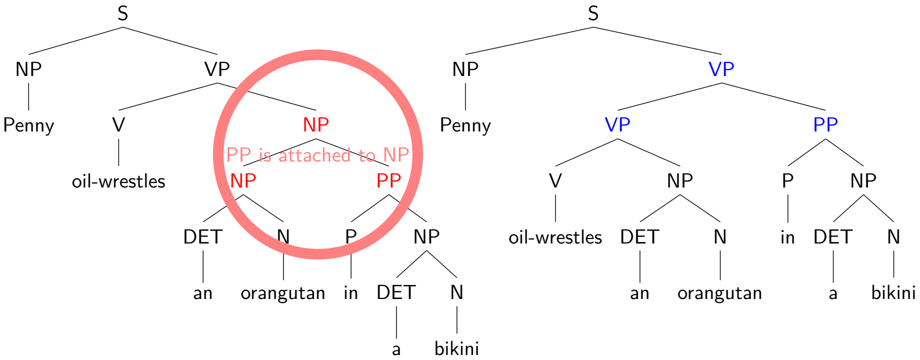
- q(NP→NP PP) $>$ q(VP→VP PP): The first one
- q(NP→NP PP) $<$ q(VP→VP PP): The second one

Naive application is not successful



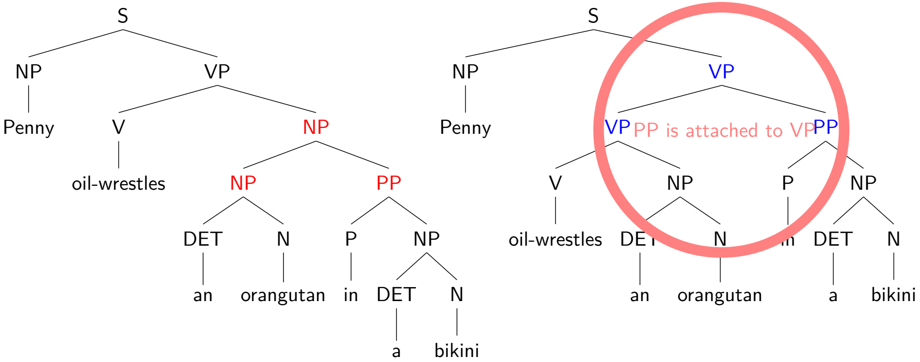
- $q(\text{NP} \rightarrow \text{NP PP}) > q(\text{VP} \rightarrow \text{VP PP})$: The first one
- $q(\text{NP} \rightarrow \text{NP PP}) < q(\text{VP} \rightarrow \text{VP PP})$: The second one

Naive application is not successful



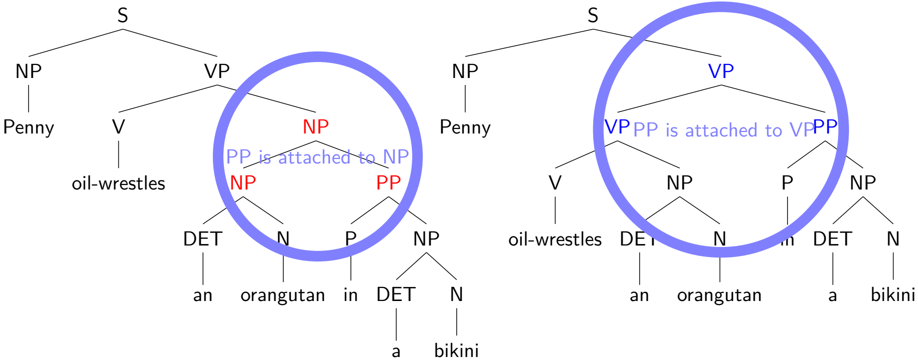
- $q(\text{NP}\rightarrow\text{NP PP}) > q(\text{VP}\rightarrow\text{VP PP})$: The first one
- $q(\text{NP}\rightarrow\text{NP PP}) < q(\text{VP}\rightarrow\text{VP PP})$: The second one

Naive application is not successful



- q(NP→NP PP) $>$ q(VP→VP PP): The first one
- q(NP→NP PP) $<$ q(VP→VP PP): The second one

Naive application is not successful



- q(NP→NP PP) $>$ q(VP→VP PP): The first one
- q(NP→NP PP) $<$ q(VP→VP PP): The second one

F-score $= 72.64$

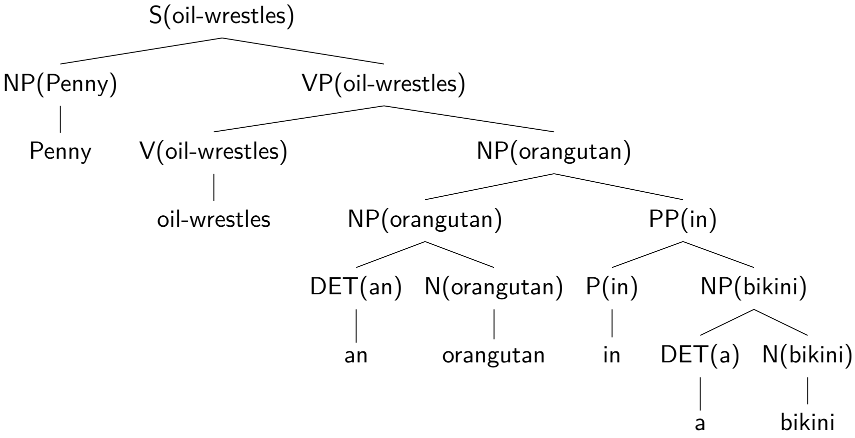
Klein & Manning (2003)

Outline

Lexicalized PCFG

Solution (Collins, 1999)

Add annotations specifying richer information for each rule



Lexicalized CFG

Rewrite rules

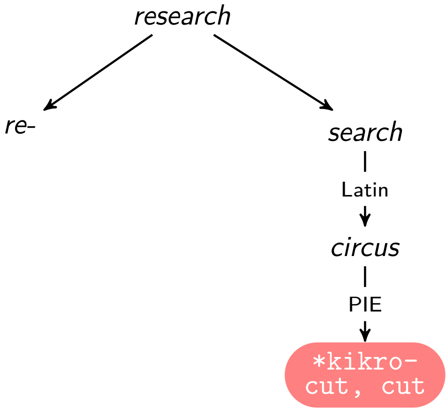
- $X(h) \rightarrow Y(h)\ Z(m)$ for $X, Y, Z \in N$, and $h, w \in T$
- $X(h) \rightarrow Y(m)\ Z(h)$ for $X, Y, Z \in N$, and $h, w \in T$
- $X(h) \rightarrow h$ for $X \in N$, and $h \in T$

S(oil-wrestles) → NP(Penny) VP(oil-wrestles)	1.0
VP(oil-wrestles) → VP(oil-wrestles) PP(in)	0.5
VP(oil-wrestles) → V(oil-wrestles) NP(orangutan)	0.25
...	
V(oil-wrestle) → oil-wrestle	1.0
Det(an) → an	0.4
Det(a) → a	0.6
N(orangutan) → orangutan	0.5
N(bikini) → bikini	0.5
P(in) → in	1.0

Outline

1. Parsing: An (Unnecessary?) Introduction
2. PCFG: Early Failure
3. Lexicalization: A Breakthrough Technique
4. Alternatives? (Simple) PCFG Again

Cutting-edge



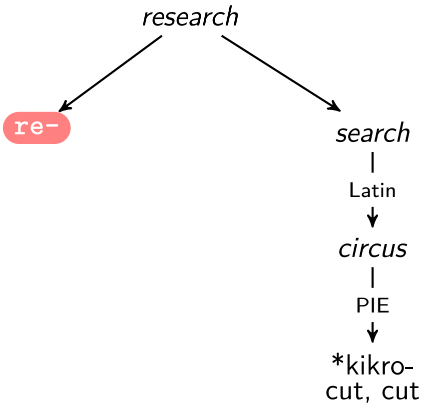
Alternatives?

System name	Short description	Main publications	Software	Results (PARSEVAL)	Comments
Charniak & Johnson's Parser	Lexicalized N-Best PCFG + Discriminative reranking	Johnson and Charniak (2005)	Download	91.4%	also works well on Brown
Self-trained Charniak & Johnson Parser	Above + self-training on ~2 million raw sentences from NANC	McClosky, Charniak, and Johnson (2006)	Download	92.1%	also works well on Brown
Collins' Parser	Lexicalized PCFG	Collins (1999), Bikel (2004)	Dan Bikel's implementation	?	?
Berkeley Parser	Automatically induced PCFG	Petrov et al. (2006), Petrov and Klein (2007)	Berkeley Parser	90.1%	works well also for Chinese and German
Link Grammar	Dependency grammar	Temperley, Sleator, Lafferty, others (1995-2006)	Actively supported project	?	Persian, Arabic, Chinese, German, Russian dictionaries have been developed.

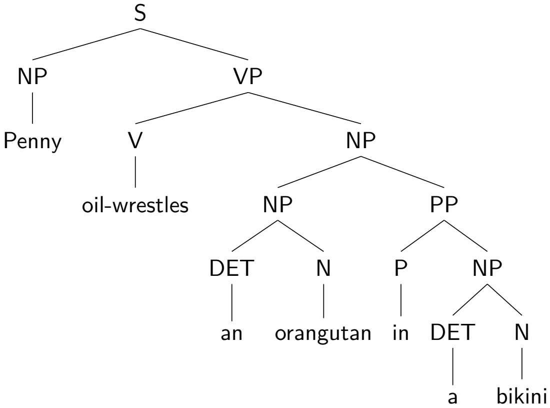
Alternatives?

System name	Short description	Main publications	Software	Results (PARSEVAL)	Comments
Charniak & Johnson's Parser	Lexicalized N-Best PCFG + Discriminative reranking	Johnson and Charniak (2005)	Download	91.4%	also works well on Brown
Self-trained Charniak & Johnson Parser	Above + self-training on ~2 million raw sentences from NANC	McClosky, Charniak, and Johnson (2006)	Download	92.1%	also works well on Brown
Collins' Parser	Lexicalized PCFG	Collins (1999), Bikel (2004)	Dan Bikel's implementation	?	?
Berkeley Parser	Automatically induced PCFG	Petrov et al. (2006), Petrov and Klein (2007)	Berkeley Parser	90.1%	works well also for Chinese and German
Link Grammar	Dependency grammar	Temperley, Sleator, Lafferty, others (1995-2006)	Actively supported project	?	Persian, Arabic, Chinese, German, Russian dictionaries have been developed.

Unlexicalized Parsing


research
re-
search
Latin
circus
PIE
*kikro-
cut, cut

Enriching representations

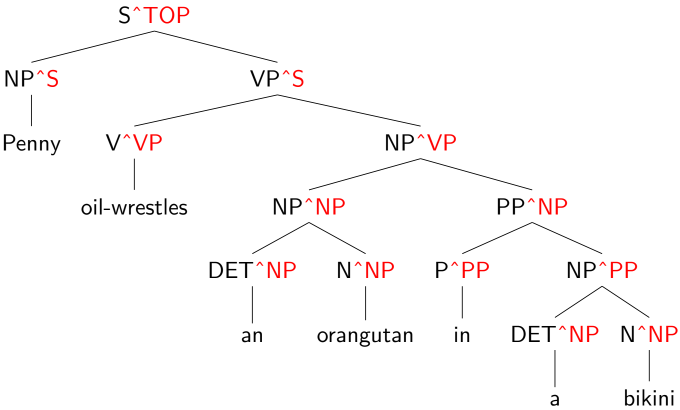


Parent annotation (Johnson, 1998)

Add annotations specifying richer information for each production rule of a given PCFG.

F-score $= 72.64 \rightarrow 76.81$ Klein & Manning (2003)

Enriching representations



Parent annotation (Johnson, 1998)

Add annotations specifying richer information for each production rule of a given PCFG.

F-score $= 72.64 \rightarrow 76.81$ Klein & Manning (2003)

Vertical and horizontal Markovization

Vertical Order		Horizontal Markov Order $h = 0$	$h = 1$	$h \leq 2$	$h = 2$	$h = \infty$
$v = 1$	No annotation	71.27 (854)	72.5 (3119)	73.46 (3863)	72.96 (6207)	72.62 (9657)
$v \leq 2$	Sel. Parents	74.75 (2285)	77.42 (6564)	77.77 (7619)	77.50 (11398)	76.91 (14247)
$v = 2$	All Parents	74.68 (2984)	77.42 (7312)	77.81 (8367)	77.50 (12132)	76.81 (14666)
$v \leq 3$	Sel. GParents	76.50 (4943)	78.59 (12374)	79.07 (13627)	78.97 (19545)	78.54 (20123)
$v = 3$	All GParents	76.74 (7797)	79.18 (15740)	79.74 (16994)	79.07 (22886)	78.72 (22002)

Figure 2: Markovizations: F_1 and grammar size.

Klein & Manning (2003)

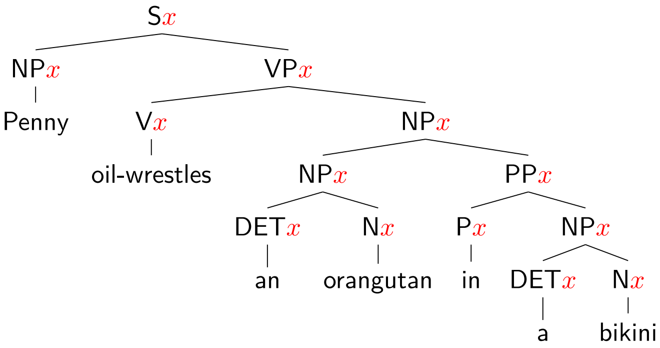
Further annotation enrichment

	Cumulative			Indiv.
Annotation	Size	F_1	Δ F_1	Δ F_1
Baseline ($v \leq 2$, $h \leq 2$)	7619	77.77	–	–
UNARY-INTERNAL	8065	78.32	0.55	0.55
UNARY-DT	8066	78.48	0.71	0.17
UNARY-RB	8069	78.86	1.09	0.43
TAG-PA	8520	80.62	2.85	2.52
SPLIT-IN	8541	81.19	3.42	2.12
SPLIT-AUX	9034	81.66	3.89	0.57
SPLIT-CC	9190	81.69	3.92	0.12
SPLIT-%	9255	81.81	4.04	0.15
TMP-NP	9594	82.25	4.48	1.07
GAPPED-S	9741	82.28	4.51	0.17
POSS-NP	9820	83.06	5.29	0.28
SPLIT-VP	10499	85.72	7.95	1.36
BASE-NP	11660	86.04	8.27	0.73
DOMINATES-V	14097	86.91	9.14	1.42
RIGHT-REC-NP	15276	87.04	9.27	1.94

Klein & Manning (2003)

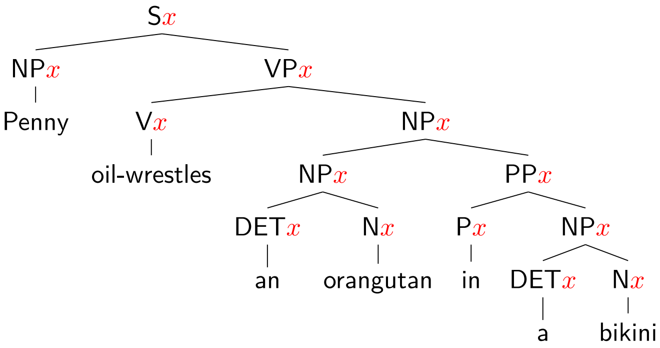
Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)



Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)

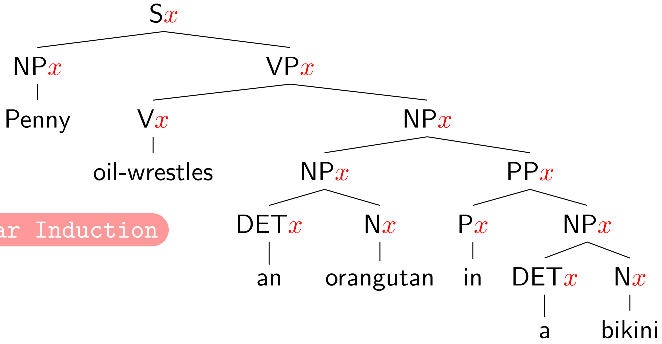


Cool idea; many many challenges

- How to induce x?
- How to find a best parse?

Latent annotations

Additional information as latent variables (Matsuzaki et al., 2005)

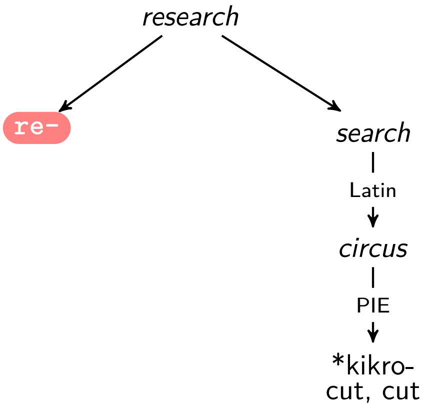


Grammar Induction

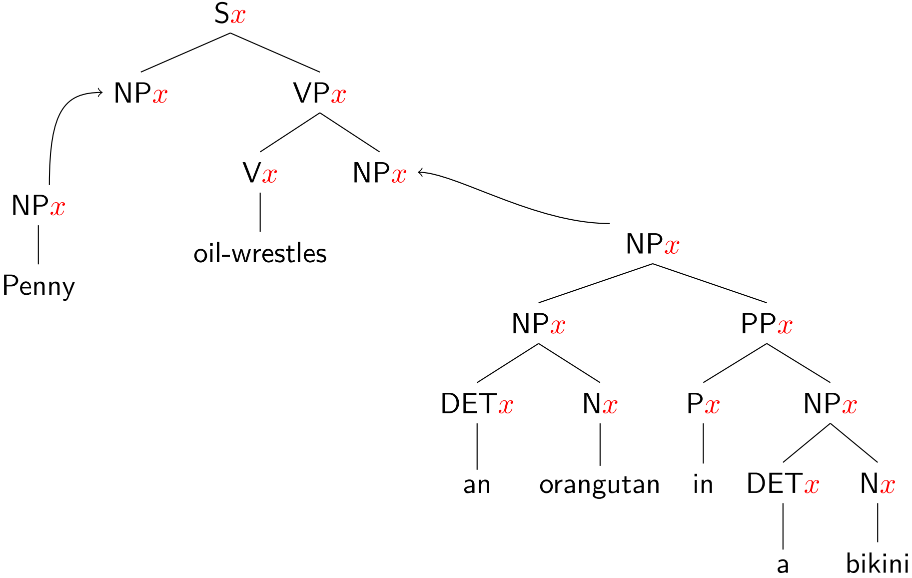
Cool idea; many many challenges

- How to induce x?
- How to find a best parse?

re-



Tree Substitution Grammar (Joshi & Schabes, 1997)



Numbers

System name	Short description	Main publications	Software	Resu (PARSE
Charniak & Johnson's Parser	Lexicalized N-Best PCFG + Discriminative reranking	Johnson and Charniak (2005)	Download	91.4%
Self-trained Charniak & Johnson Parser	Above + self-training on ~2 million raw sentences from NANC	McClosky, Charniak, and Johnson (2006)	Download	92.1%
Collins' Parser	Lexicalized PCFG	Collins (1999), Bikel (2004)	Dan Bikel's implementation	?
Berkeley Parser	Automatically induced PCFG	Petrov et al. (2006), Petrov and Klein (2007)	Berkeley Parser	90.1%
Link Grammar	Dependency grammar	Temperley, Sleator, Lafferty, others (1995-2006)	Actively supported project	?

Model	LP	LR	LF
Klein & Manning (2003)	86.9	85.7	86.3
Matsuzaki et al. (2005)	86.1	86.0	
Petrov et al. (2006)	89.8	89.6	
Shindo et al. (2012)			91.1
Shindo et al. (2012)			92.4

Lessons learned

1998 · · · PCFG Models of Linguistic Tree Representations

2003 · · · Accurate Unlexicalized Parsing (ACL best paper)

2005 · · · Probabilistic CFG with Latent Annotations

2006 · · · Learning Accurate, Compact, and Interpretable Tree Annotation

2012 · · · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons learned

- **1998** PCFG Models of Linguistic Tree Representations
- **2003** Accurate Unlexicalized Parsing (ACL best paper)
- **2005** Probabilistic CFG with Latent Annotations
- **2006** Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons learned

- **1998** PCFG Models of Linguistic Tree Representations
- **2003** Accurate Unlexicalized Parsing (ACL best paper)
- **2005** Probabilistic CFG with Latent Annotations
- **2006** Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons learned

- **1998** PCFG Models of Linguistic Tree Representations
- **2003** Accurate Unlexicalized Parsing (ACL best paper)
- **2005** Probabilistic CFG with Latent Annotations
- **2006** Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons learned

- **1998** · PCFG Models of Linguistic Tree Representations
- **2003** · Accurate Unlexicalized Parsing (ACL best paper)
- **2005** · Probabilistic CFG with Latent Annotations
- **2006** · Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons learned

- **1998** · · · PCFG Models of Linguistic Tree Representations
- **2003** · · · Accurate Unlexicalized Parsing (ACL best paper)
- **2005** · · · Probabilistic CFG with Latent Annotations
- **2006** · · · Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** · · · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

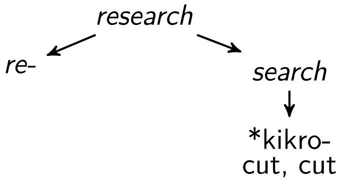
Lessons learned

- **1998** · PCFG Models of Linguistic Tree Representations
- **2003** · Accurate Unlexicalized Parsing (ACL best paper)
- **2005** · Probabilistic CFG with Latent Annotations
- **2006** · Learning Accurate, Compact, and Interpretable Tree Annotation
- **2012** · Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing (ACL best paper)

Lessons that I've learned

Too many; a short list:

- Diversity
- Interpretability
- Revisit old things



Thank You

References I

Collins, M. (1999). *Head-driven statistical models for natural language parsing*. Unpublished doctoral dissertation, University of Pennsylvania.

Johnson, M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics*, *24*(4), 613–632.

Joshi, A. K., & Schabes, Y. (1997). Tree-adjoining grammars. In G. Rozenberg & A. Salomaa (Eds.), *Handbook of formal languages* (Vol. 3, pp. 69–124). Berlin, New York: Springer.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics* (pp. 423–430). Sapporo, Japan: Association for Computational Linguistics.

References II

Matsuzaki, T., Miyao, Y., & Tsujii, J. (2005). Probabilistic cfg with latent annotations. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 75–82). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `http://dx.doi.org/10.3115/1219840.1219850`

Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics* (pp. 433–440). Sydney, Australia: Association for Computational Linguistics.

Shindo, H., Miyao, Y., Fujino, A., & Nagata, M. (2012). Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of the 50th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 440–448). Jeju Island, Korea: Association for Computational Linguistics. Retrieved from `http://www.aclweb.org/anthology/P12-1046`