Incremental Coarse-to-Fine Parsing

Marten van Schijndel
Department of Linguistics
The Ohio State University

April 20, 2012

Incremental Motivation

Understanding ... One Step at a Time

- Cognitive motivations
  - Operates on incomplete information (Cloze testing)
- Engineering motivations
  - Can make use of information about recent content/structure (coreference, pragmatics)
  - Unsegmented input
  - $\mathcal{O}(n)$ Streaming task

Coarse-to-Fine Motivation

What is it?

Coarse-to-Fine Motivation

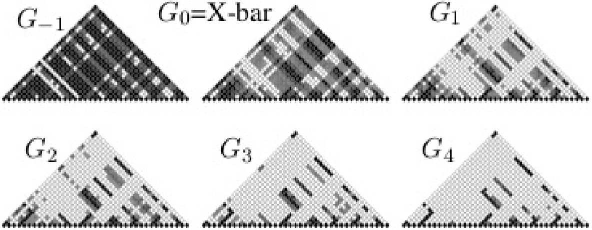
What is it?

- A way of improving parse speed/accuracy through pruning the search space.

Coarse-to-Fine Motivation

What is it?

- A way of improving parse speed/accuracy through pruning the search space.
- It has massively sped up parsers in the recent past
  - [Petrov and Klein, 2007] 50x



CTF Theory

How does it work?

CTF Theory

How does it work?

Parse in phases [Charniak et al., 2006]

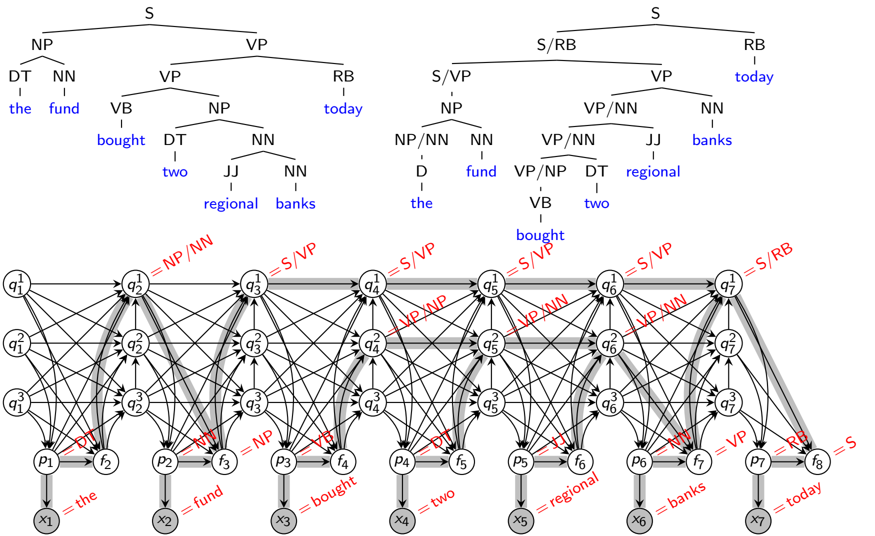
$$
\begin{array}{llll}
0 & 1 & 2 & 3 \\
P & \left\{ \begin{array}{ll}
HP & \left\{ \begin{array}{ll}
S\_ & \left\{ \begin{array}{l} S \\ VP \\ SBAR \\ \ldots \end{array} \right. \\
N\_ & \left\{ \begin{array}{l} NP \\ NAC \\ NX \\ \ldots \end{array} \right.
\end{array} \right. \\
MP & \left\{ \begin{array}{ll}
A\_ & \left\{ \begin{array}{l} ADJP \\ QP \\ CONJP \\ \ldots \end{array} \right. \\
P\_ & \left\{ \begin{array}{l} PP \\ PRT \\ RRC \\ WHADJP \\ \ldots \end{array} \right.
\end{array} \right.
\end{array} \right. & &
\end{array}
$$

CTF History

Some ways of implementing Coarse-to-Fine:

- Do it by hand [III and Kaplan, 1993, Charniak et al., 2006] or machine [Petrov and Klein, 2007]
- Single or Multi-layered
- If we assume the Berkeley Parser paradigm:
  - Trainer derives split-merge grammar files
  - Initialization phase creates a predictive chain back to coarse grammar

Sequence Model Parsing



Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  (S (ADVP happily) (NP-SUBJ John)...)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  (S^g_10 –> ADVP^g_21 NP^g_4 1.462527E-18) WOW!

Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  (S (ADVP happily) (NP-SUBJ John)...)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  (S^g_10 -> ADVP^g_21 NP^g_4 1.462527*E*-18) WOW!

Profit:

- Accuracy

Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  (S (ADVP happily) (NP-SUBJ John)...)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  (S^g_10 -> ADVP^g_21 NP^g_4 1.462527*E*-18) WOW!

Profit:

- Accuracy

Cost:

- Training time
- Increased size of grammar

Sequence Model Training

Sequence Model Conversion

- Input: Sleek newly obtained PCFG
  (S^g_10 –> ADVP^g_21 NP^g_4 1.462527*E*-18)
- Generate virtual trees to give probabilities of component productions
- Output: Phase-, depth-specific grammar
  (B 2 S^g_10 ADVP^g_21 –> NP^g_4 2.348767*E*-20)

Sequence Model Training

Sequence Model Conversion

- Input: Sleek newly obtained PCFG
  (S^g_10 –> ADVP^g_21 NP^g_4 1.462527E-18)
- Generate virtual trees to give probabilities of component productions
- Output: Phase-, depth-specific grammar
  (B 2 S^g_10 ADVP^g_21 –> NP^g_4 2.348767E-20)

Profit:

- Information about upcoming categories
- Information about embedding depth

Sequence Model Training

Sequence Model Conversion

- Input: Sleek newly obtained PCFG
  (S^g_10 –> ADVP^g_21 NP^g_4 1.462527E-18)
- Generate virtual trees to give probabilities of component productions
- Output: Phase-, depth-specific grammar
  (B 2 S^g_10 ADVP^g_21 –> NP^g_4 2.348767E-20)

Profit:

- Information about upcoming categories
- Information about embedding depth

Cost:

- Training time
- Increased size of grammar

Mix it all up

How does this work?

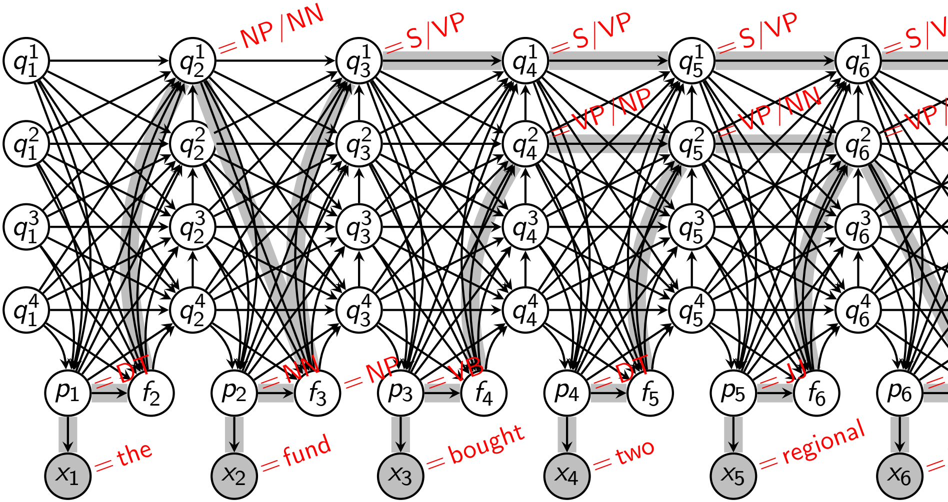
- Approximate Inference

Variable Descriptions

- q_t^d represents an element of working memory/incomplete constituent
- These are decomposed into a_t^d and b_t^d
- x_t is the observation at time t
- p_t is the preterminal that expands into that observation
- f_t is the final state obtained by integrating a new observation into the parse (expansion state)

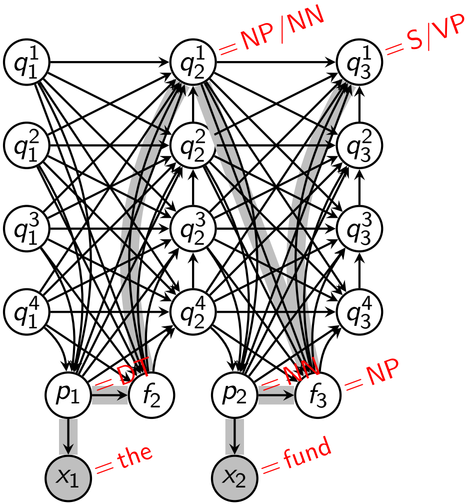
Mix it all up

[q_t^d]



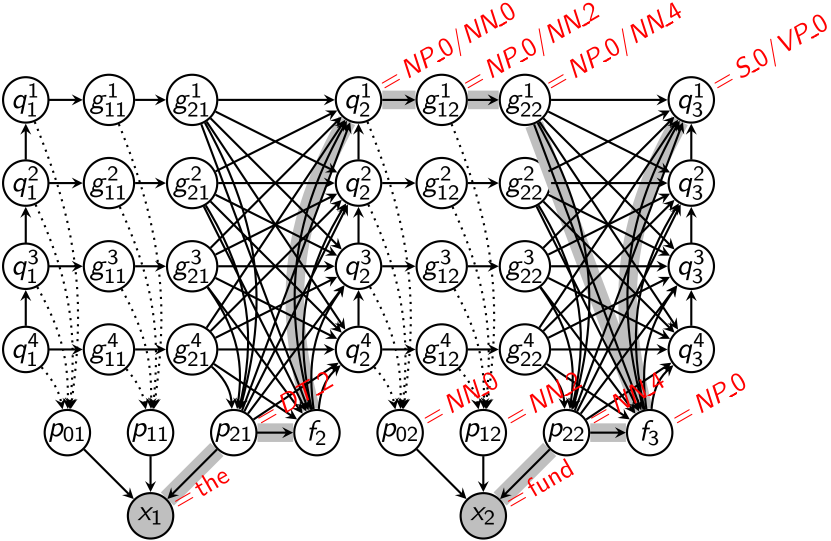
Mix it all up

$[\, q_t^d \,]$



Mix it all up

$[\, q^d_{gt} \,]$



How does it work?

Theory/Equation time

Most likely sequence

$$\hat{q}^{1..D}_{1..T} \stackrel{\text{def}}{=} \underset{q^{1..D}_{1..T}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\theta_Q}(q^{1..D}_t \mid q^{1..D}_{t-1}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{P,d'}}(p_t \mid b^{d'}_t) \cdot \mathsf{P}_{\theta_X}(x_t \mid p_t) \quad (1)$$

where d' is the lowest non-empty q^d_t

How does it work?

Theory/Equation time

Right-Corner: Single expansion, Single reduction

E-R+, E-R-, E+R+, E+R-

θ_Q

$$
\begin{aligned}
&\mathrm{P}_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D}\ p_{t-1}) \\
&\quad \stackrel{\mathrm{def}}{=} \mathrm{P}_{\theta_F}(\text{'0'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathrm{P}_{\theta_{A,d'}}(\text{'–'} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \cdot [\![a_t^{d'-1} = a_{t-1}^{d'-1}]\!] \cdot \mathrm{P}_{\theta_{B,d'-1}}(b_t^{d'-1} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \\
&\qquad \cdot [\![q_t^{1..d'-2} = q_{t-1}^{1..d'-2}]\!] \cdot [\![q_t^{d'..D} = \text{'–'}]\!] \\
&\quad + \mathrm{P}_{\theta_F}(\text{'0'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathrm{P}_{\theta_{A,d'}}(a_t^{d'} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \cdot \mathrm{P}_{\theta_{B,d'}}(b_t^{d'} \mid a_t^{d'}\ a_{t-1}^{d'+1}) \\
&\qquad \cdot [\![q_t^{1..d'-1} = q_{t-1}^{1..d'-1}]\!] \cdot [\![q_t^{d'+1..D} = \text{'–'}]\!] \\
&\quad + \mathrm{P}_{\theta_F}(\text{'1'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathrm{P}_{\theta_{A,d'}}(\text{'–'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot [\![a_t^{d'} = a_{t-1}^{d'}]\!] \cdot \mathrm{P}_{\theta_{B,d'}}(b_t^{d'} \mid b_{t-1}^{d'}\ p_{t-1}) \\
&\qquad \cdot [\![q_t^{1..d'-1} = q_{t-1}^{1..d'-1}]\!] \cdot [\![q_t^{d'+1..D} = \text{'–'}]\!] \\
&\quad + \mathrm{P}_{\theta_F}(\text{'1'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathrm{P}_{\theta_{A,d'}}(a_t^{d'+1} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathrm{P}_{\theta_{B,d'}}(b_t^{d'+1} \mid a_t^{d'+1}\ p_{t-1}) \\
&\qquad \cdot [\![q_t^{1..d'} = q_{t-1}^{1..d'}]\!] \cdot [\![q_t^{d'+2..D} = \text{'–'}]\!]
\end{aligned} \tag{2}
$$

How does it work?

Theory/Equation time

$\theta_{F,d,g}$

$$\mathsf{P}_{\theta_{F,d,g}}(f_G \mid b_G, p_G) \stackrel{\text{def}}{=} \begin{cases} \mathsf{P}_{\theta_{F,d}}(f_G \mid b_G, p_G) & \text{if } g = 0 \\ 1 & \text{else} \end{cases} \tag{3}$$

$\theta_{A,d,g}$

$$\mathsf{P}_{\theta_{A,d,g}}(a_g \mid b_G, f_G, \pi(a_g)) \stackrel{\text{def}}{=} \frac{max_{a_G \mid a_g \prec a_G} \mathsf{P}_{\theta_{A,d,g}}(a_G \mid b_G, f_G, \pi(a_G))}{max_{a'_G \mid \pi(a_g) \prec a'_G} \mathsf{P}_{\theta_{A,d,g}}(a'_G \mid b_G, f_G, \pi(a'_G))} \tag{4}$$

$$= \frac{max_{a_G \mid a_g \prec a_G} \mathsf{P}_{\theta_{A,d}}(a_G \mid b_G, f_G)}{max_{a'_G \mid \pi(a_g) \prec a'_G} \mathsf{P}_{\theta_{A,d}}(a'_G \mid b_G, f_G)} \tag{5}$$

How does it work?

Theory/Equation time

$\theta_{B,d,g}$

a) Active Transition

$$\mathsf{P}_{\theta_{B,d,g}}(b_g \mid a_g, f_G, \pi(b_g)) \stackrel{\text{def}}{=} \frac{max_{b_G, a_G \mid b_g \prec b_G, a_g \prec a_G} \mathsf{P}_{\theta_{B,d,g}}(b_G \mid a_G, f_G, \pi(b_G))}{max_{b'_G, a_G \mid \pi(b_g) \prec b'_G, a_g \prec a_G} \mathsf{P}_{\theta_{B,d,g}}(b'_G \mid a_G, f_G, \pi(b'_G))} \tag{6}$$

$$= \frac{max_{b_G, a_G \mid b_g \prec b_G, a_g \prec a_G} \mathsf{P}_{\theta_{B,d}}(b_G \mid a_G, f_G)}{max_{b'_G, a_G \mid \pi(b_g) \prec b'_G, a_g \prec a_G} \mathsf{P}_{\theta_{B,d}}(b'_G \mid a_G, f_G)} \tag{7}$$

b) Awaited Transition

$$\mathsf{P}_{\theta_{B,d,g}}(b_g \mid b'_G, f_G, \pi(b_g)) \stackrel{\text{def}}{=} \frac{max_{b_G \mid b_g \prec b_G} \mathsf{P}_{\theta_{B,d,g}}(b_G \mid b'_G, f_G, \pi(b_G))}{max_{b''_G \mid \pi(b_g) \prec b''_G} \mathsf{P}_{\theta_{B,d,g}}(b''_G \mid b'_G, f_G, \pi(b''_G))} \tag{8}$$

$$= \frac{max_{b_G \mid b_g \prec b_G} \mathsf{P}_{\theta_{B,d}}(b_G \mid b'_G, f_G)}{max_{b''_G \mid \pi(b_g) \prec b''_G} \mathsf{P}_{\theta_{B,d}}(b''_G \mid b'_G, f_G)} \tag{9}$$

How does it work?

Theory/Equation time

$\theta_{P,d,g}$

$$P_{\theta_{P,d,g}}(p_g \mid b_g, \pi(p_g)) \stackrel{\text{def}}{=} \frac{max_{p_G, b_G | p_g \prec p_G, b_g \prec b_G} P_{\theta_{P,d,g}}(p_G \mid b_G, \pi(p_G))}{max_{p'_G, b_G | \pi(p_g) \prec p'_G, b_g \prec b_G} P_{\theta_{P,d,g}}(p'_G \mid b_G, \pi(p'_G))} \tag{10}$$

$$= \frac{max_{p_G, b_G | p_g \prec p_G, b_g \prec b_G} P_{\theta_{P,d}}(p_G \mid b_G)}{max_{p'_G, b_G | \pi(p_g) \prec p'_G, b_g \prec b_G} P_{\theta_{P,d}}(p'_G \mid b_G)} \tag{11}$$

$\theta_{X,g}$

$$P_{\theta_{X,g}}(x \mid p_g) \stackrel{\text{def}}{=} \frac{max_{p_G | p_g \prec p_G} P_{\theta_{X,g}}(x \mid p_G)}{max_{p'_G | \pi(p_g) \prec p'_G} P_{\theta_{X,g}}(x \mid p'_G)} \tag{12}$$

$$= \frac{max_{p_G | p_g \prec p_G} P_{\theta_X}(x \mid p_G)}{max_{p'_G | \pi(p_g) \prec p'_G} P_{\theta_X}(x \mid p'_G)} \tag{13}$$

Paydirt

Timing Results

System	CTF-FAWP	FAWP	Diff
5sm-2000	30.3	61.05	0.496
4sm-500	4.16	7.17	0.580
3sm-500	2.11	4.83	0.437
2sm-500	1.64	3.35	0.490
		Ave	0.50

Timing results with varying sm. (sec/sent)

CTF-FAWP Timing Results

System	Time
3sm-2000	8.27
3sm-1000	4.26
3sm-500	2.00
3sm-250	0.87
3sm-100	0.33

Timing results with varying beam-width. (sec/sent)

Paydirt

CTF Accuracy Results

System	Recall	Prec	F
Petrov Klein (Reported, 10-best)	91.2	91.1	91.2
Petrov Klein (5sm, U+B, 1-best)	88.5	88.8	88.7
Petrov Klein (5sm, Binary, 1-best)	88.2	87.9	88.0
FAWP (5sm, b5000)	87.9	87.7	87.8
CTF-FAWP (5sm, b5000)	88.0	87.6	87.8
FAWP (5sm, b2000)	87.7	87.6	87.6
CTF-FAWP (5sm, b2000)	86.2	86.3	86.3

Accuracy of CTF on various incarnations of FAWP.

And Beyond!

Future Work

Where to now?

- Condition on more variables (MaxEnt)
- Weight predictions based on proportion of total beam predictions; More NP predictions make NP a better guess.

Bibliography I

Charniak, E., Johnson, M., Elsner, M., Austerweil, J., Ellis, D., Haxton, I., Hill, C., Shrivaths, R., Moore, J., Pozar, M., and Vu, T. (2006).
Multilevel coarse-to-fine pcfg parsing.
In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 168–175.

III, J. T. M. and Kaplan, R. M. (1993).
The interface between phrasal and functional constraints.
Computational Linguistics, 19(4):571–590.

Bibliography II

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006).
Learning accurate, compact, and interpretable tree annotation.
In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440. Association for Computational Linguistics.

Petrov, S. and Klein, D. (2007).
Improved inference for unlexicalized parsing.
In *Proceedings of NAACL HLT 2007*, pages 404–411, Rochester, New York. Association for Computational Linguistics.

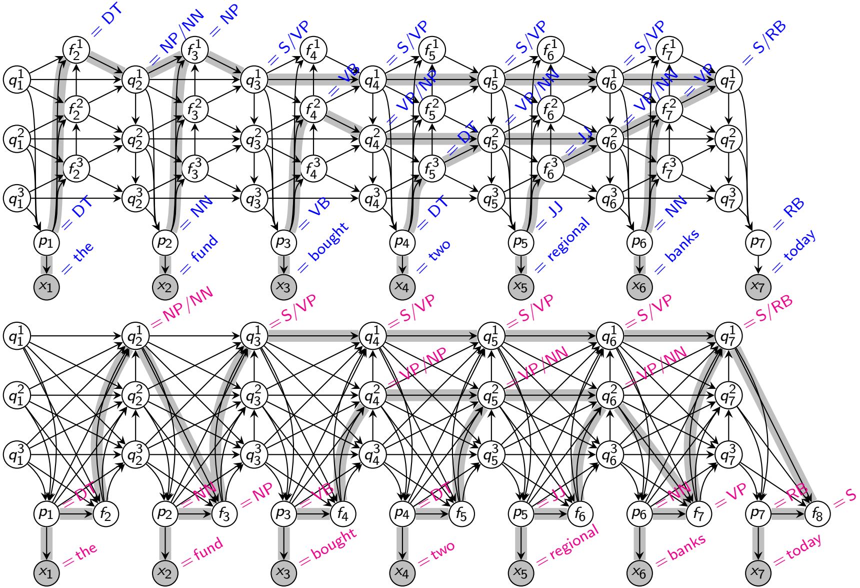
Roark, B. (2001).
Probabilistic top-down parsing and language modeling.
Computational Linguistics, 27(2):249–276.

Bibliography III

Schuler, W. (2009).
Parsing with a bounded stack using a model-based right-corner transform.
In *Proceedings of NAACL/HLT 2009*, NAACL '09, pages 344–352, Boulder, Colorado. Association for Computational Linguistics.

Schuler, W., AbdelRahman, S., Miller, T., and Schwartz, L. (2010).
Broad-coverage incremental parsing using human-like memory constraints.
Computational Linguistics, 36(1):1–30.

The Model


= DT
= NP/NN
= NP
= S/VP
= S/VP
= S/VP
= S/VP
= S/RB
= VB
= VP/NP
= VP/NN
= VP/NN
= VP
= DT
= JJ
= DT
= NN
= VB
= DT
= JJ
= NN
= RB
= the
= fund
= bought
= two
= regional
= banks
= today
= NP/NN
= S/VP
= S/VP
= S/VP
= S/VP
= S/RB
= VP/NP
= VP/NN
= VP/NN
= DT
= NN
= NP
= VB
= DT
= JJ
= NN
= VP
= RB
= S
= the
= fund
= bought
= two
= regional
= banks
= today