Joint Training for Open-domain Extraction on the Web:

Exploiting Overlap when Supervision is Limited

Rahul Gupta*
Google Research

Sunita Sarawagi
IIT Bombay

* Work done at IIT Bombay

Query-driven Extraction on the Web

User →

Gran Torino	Walt Kowalski	2008
Dirty Harry	Harry Callahan	1971

Collective Extraction

19. Sudden Impact (1983) .... Harry Callahan
20. Honkytonk Man (1982) .... Red Stovall
21. Firefox (1982) .... Mitchell Gant
22. Any Which Way You Can (1980) .... Philo Bedd
23. Bronco Billy (1980) .... Bronco Billy
24. Escape from Alcatraz (1979) .... Frank Morris

The Dead Pool (198
Heartbreak Ridge (
Pale Rider (1985)
Tightrope (1984)
City Heat (1984)
Sudden Impact (1983)

Joe Kidd (1972)	Joe Kidd
Dirty Harry (1971)	Inspector Harry Callahan
Play Misty for Me (1971)	Dave
The Beguiled (1971)	John McBurney
Kelly's Heroes (1970)	Kelly

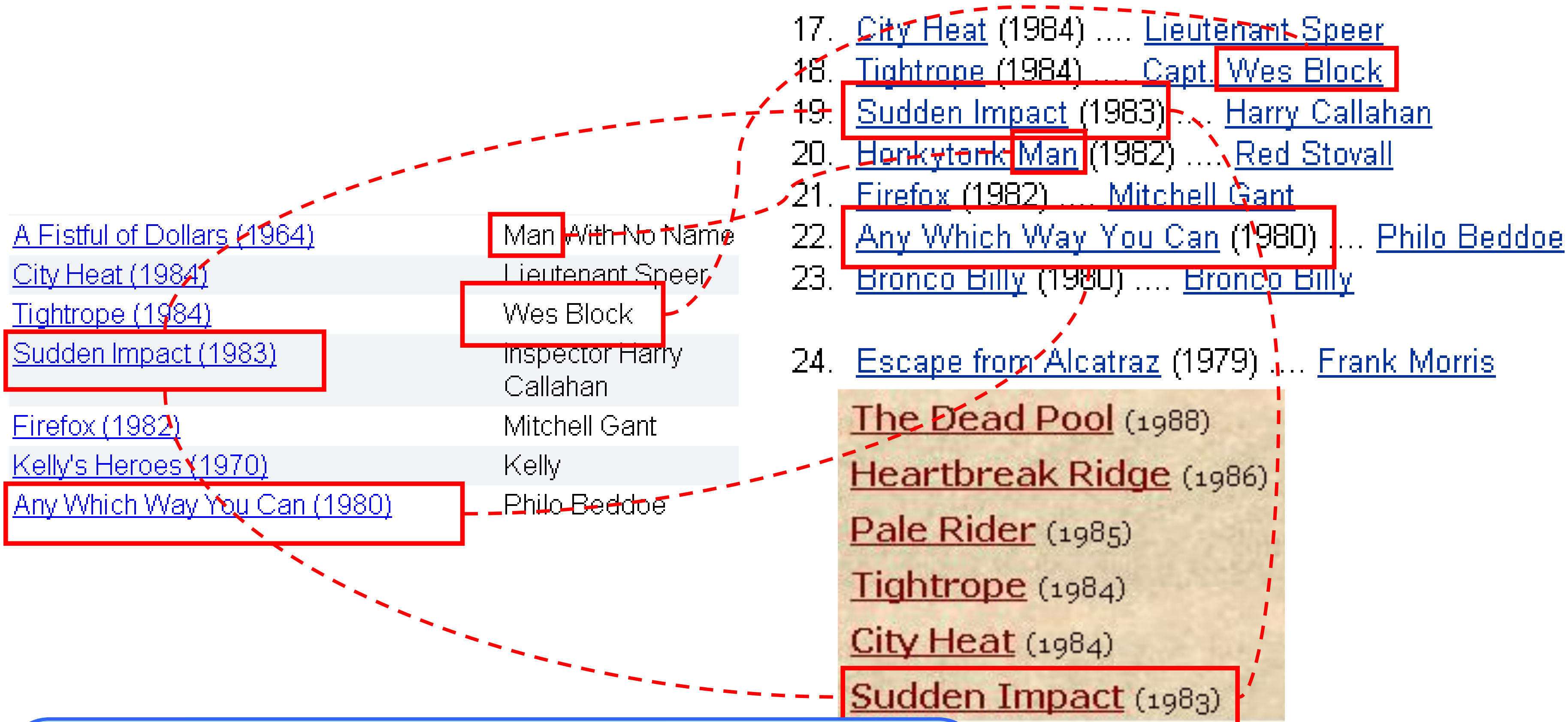
Firefox	Mitchell Gant	1982
…	…	…
…	…	…

City Heat	-	1984
…	-	…
…	-	…

Joe Kidd	Joe Kidd	1972
…	…	…
…	…	…

Merge & de-duplicate, Rank, Display to the user
(World Wide Tables, Gupta & Sarawagi VLDB '09)

Flavors of Content Overlap



A shared segment can be

- Arbitrarily long
- Across arbitrary number of sources
- Potentially a false-positive!

Content Overlap : Another Example

Eric Allman

(web page) Eric Allman is the main author of the sendmail prog
(emails), although certain alternatives have become popular, su
McKusick's partner.

Charles Babbage

Born: Monday, December 26, 1791, in London (England). Died
considered one of the forefathers of computer science for havin
(with the help of Ada Lovelace) the analytical engine, which, alt
(mechanical) computer. See also Babbage's biography on the M

John W. Backus

Born: Wednesday, December 3, 1924, in Philadelphia, Pennsyl
which gave birth to the language FORTRAN (the oldest progran
Calculus, and, of course, assembler). John Backus is the 1977

CS inventors and their inventions

Codd	Relational DB
Cray	Supercomputer

John Atanasoff	Atanasoff–Berry Computer, though it was neithe programmable nor Turing-complete.
Charles Babbage	Designed the Analytical Engine and built a prototype for a less powerful mechanical calcula
John Backus	Invented FORTRAN (*For*mula *Tran*slation), the practical high-level programming language, and formulated the Backus-Naur form that described the formal language syntax.

Extraction Setting and Goal

Setting:

- Low supervision (~3 records)
- Multiple semi/un-structured sources (~20)
- Widely varying/disjoint feature sets across sources
- Significant but arbitrary and noisy content overlap

Goal: Jointly train one extraction model per source so that they agree on the labels of shared segments

Conditional Random Field

Base Model: Linear CRF

Sample sentence: My review of Fermat's last theorem by S. Singh

t	1	2	3	4	5	6	7	8	9
$\mathbf{x}$	My	review	of	Fermat's	last	theorem	by	S.	Singh
$\mathbf{y}$	**Other**	**Other**	**Other**	**Title**	**Title**	**Title**	**other**	**Author**	**Author**

$$y_1 - y_2 - y_3 - y_4 - y_5 - y_6 - y_7 - y_8 - y_9$$

$$\log P_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \sum_t \mathbf{w} \cdot \underbrace{\mathbf{f}(y_t, y_{t-1}, t, \mathbf{x})} - \log Z_{\mathbf{w}}$$

Trained weights

Feature vector at position t

"Log Partition"

(Lafferty et.al. '01)

Possible Alternatives

- Club sources, learn one CRF: Our features are disjoint
- Collective inference: Limited to overlapping content
- Hard label transfer: Co-training, multi-stage learning: prone to error cascades
- Two-source methods: 2-view perceptron/regression: We have multiple sources
- Known joint methods: Compared later

Goal

Input: S data sources, each source i has

Labeled records L_i, Unlabeled records U_i

Set $\mathcal{A} \equiv$ Shared segments across unlabeled records

Goal: Train CRF weights $\mathbf{w_i}$ for each source $i = 1..S$

$$\sum_{(\mathbf{x},\mathbf{y}) \in L_i} \log P(\mathbf{y}|\mathbf{x}, \mathbf{w_i})$$

$$\max_{\{\mathbf{w_1},\ldots,\mathbf{w_S}\}} \sum_{i=1}^{S} \boxed{\text{LogLikelihood}(L_i|\mathbf{w_i})}$$

$$+ \text{AgreementLikelihood}(\mathcal{A}, U_1, \ldots, U_S|\mathbf{w_1}, \ldots, \mathbf{w_S})$$

Goal

$$\max_{\{\mathbf{w_1},\ldots,\mathbf{w_S}\}} \sum_{i=1}^{S} LL(L_i|\mathbf{w_i}) + C \cdot \log \sum_{\mathbf{y}_\mathcal{A}} \underbrace{\prod_{i=1}^{S} \overbrace{p_i^{\mathrm{marg}}(\mathbf{y}_\mathcal{A}|\mathbf{w_i})}^{\text{Marginal prob that } i^{th} \text{ model labels } \mathcal{A} \text{ with } \mathbf{y}_\mathcal{A}}}_{\text{Joint prob that all models label } \mathcal{A} \text{ with } \mathbf{y}_\mathcal{A}}$$

Key Issue: Tractable approximation of the agreement

Re-writing the Agreement Term

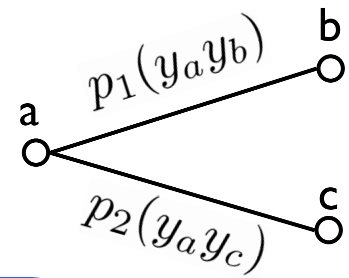
Chain 1

Chain 2

$$\sum_{y_a} p_1^{\text{marg}}(y_a) p_2^{\text{marg}}(y_a)$$

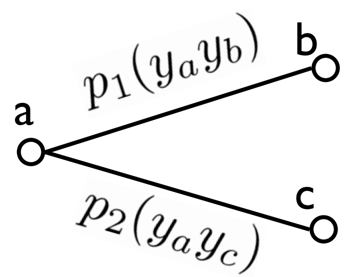
$$= \sum_{y_a, y_b, y_c} p_1(y_a y_b) p_2(y_a y_c)$$

$$= \sum_{y_a, y_b, y_c} \text{Score} \Big($$



$$\Big)$$

$$\approx \text{PartitionFunction}\Big($$



$$\Big)$$

Another Example

Three sentence snippets from different sources:

1987 Matthew “Matt” Groening : Simpsons .

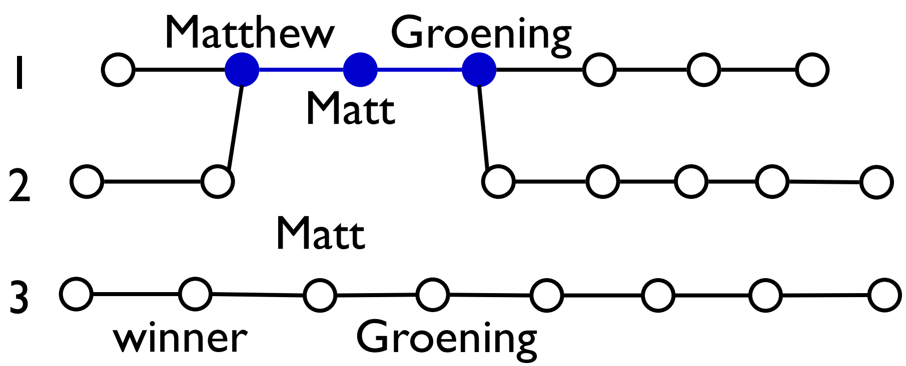
FOX – Matthew “Matt” Groening , The Simpsons , 23rd

Emmy winner Matt Groening , The Simpsons (creator)

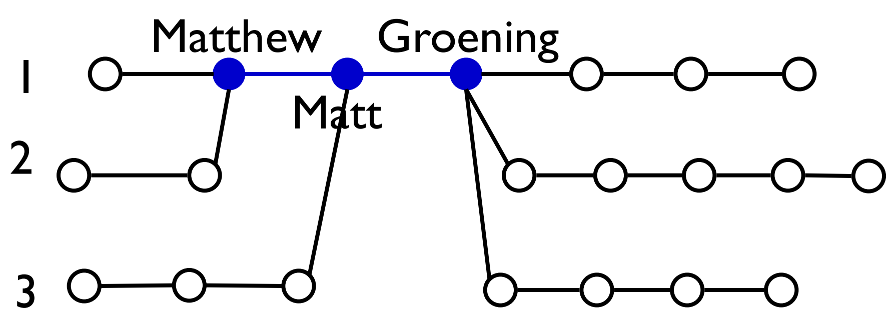
Four shared segments:

- Matthew “Matt” Groening (1,2)
- Matt Groening (1,2,3)
- Matt Groening , The Simpsons (2,3)
- Simpsons (1,2,3)

Collapsing on Shared Segments



Collapse on
"Matthew Matt Groening"

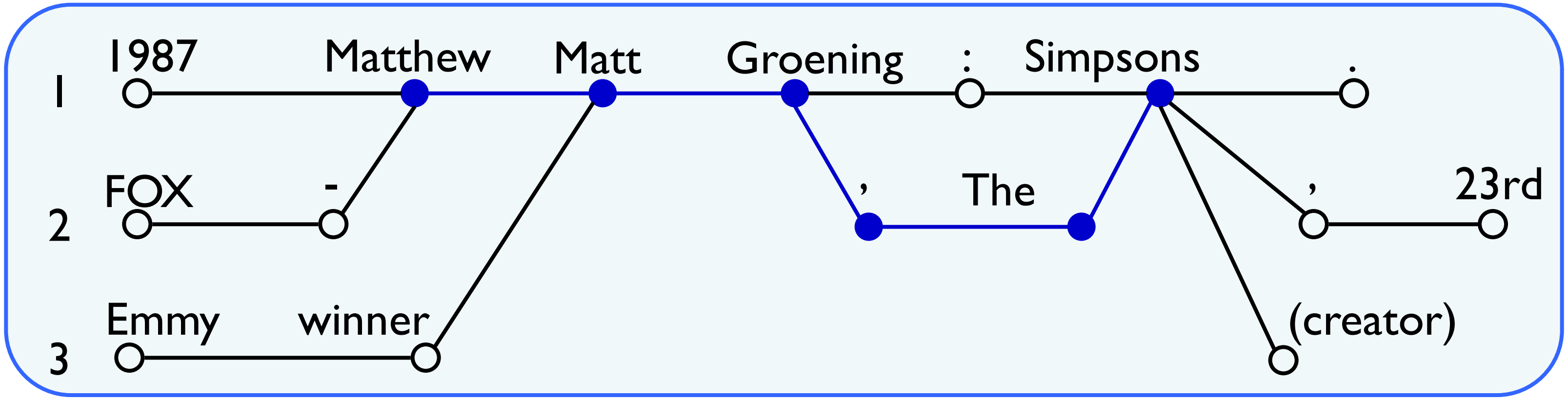


Collapse further on
"Matt Groening"

..and so on for the other shared segments

Agreement Term = Log Partition

Final "Fused" Graph: Collapse all shared segments



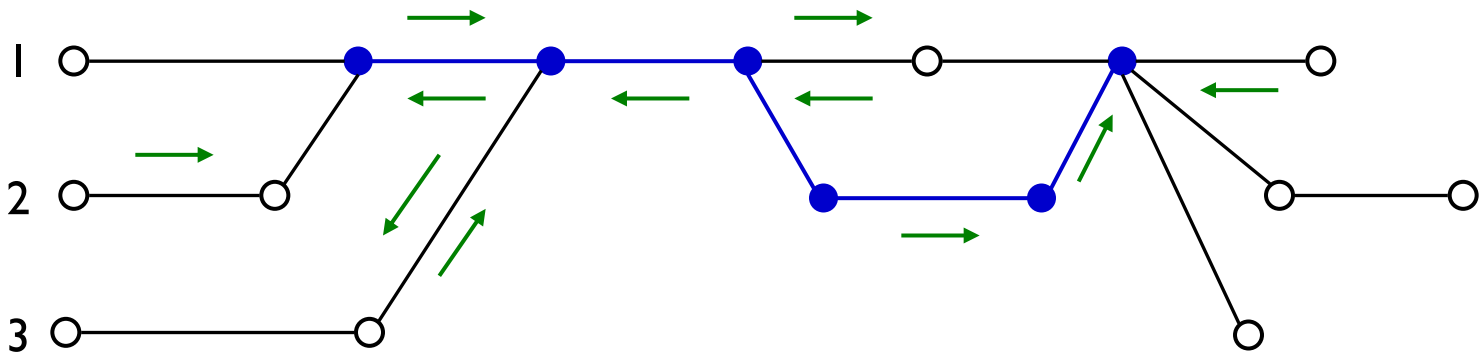
$$\log \sum_{\mathbf{y}_{\mathcal{A}}} \prod_{i=1}^{S} p_i(\mathbf{y}_{\mathcal{A}}|\mathbf{w_i}) = \log Z_{\text{fused}} - \sum_{i=1}^{S} \log Z_i$$

Log Partition of the Fused Graph

Hard if the graph has cycles!

Approximating the Log-Partition

$$\log \sum_{\mathbf{y}_{\mathcal{A}}} \prod_{i=1}^{S} p_i(\mathbf{y}_{\mathcal{A}}|\mathbf{w_i}) = \log Z_{\text{fused}} - \sum_{i=1}^{S} \log Z_i$$



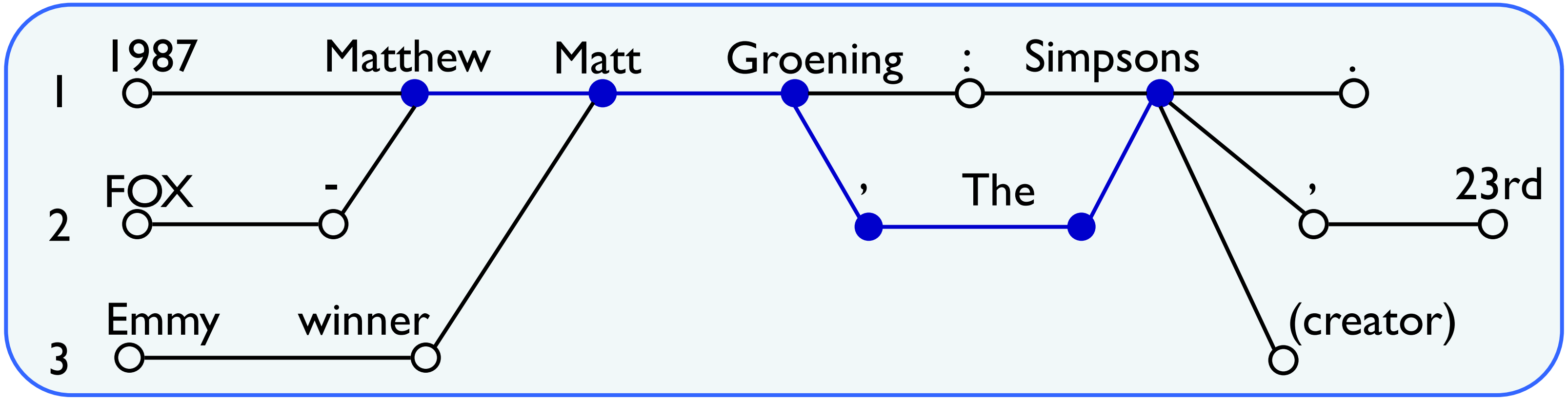
Log Z_{fused} can be approximated by

- Belief propagation (BP) on the fused graph
- Inexpensive variant of BP (Liang et. al. '09)

But…

- BP slow to converge, sometimes inconsistent
- Noisy agreement set => Wrong fused graph!

Alternate Approximation Method

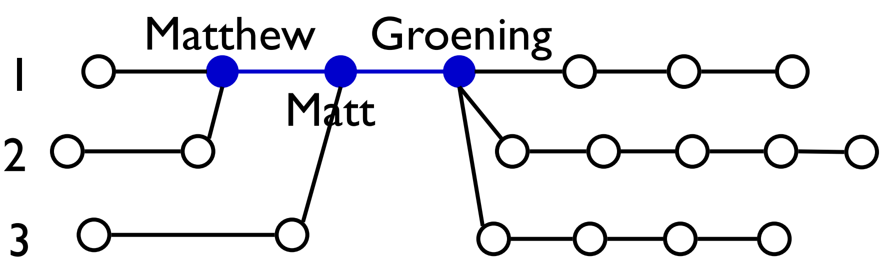


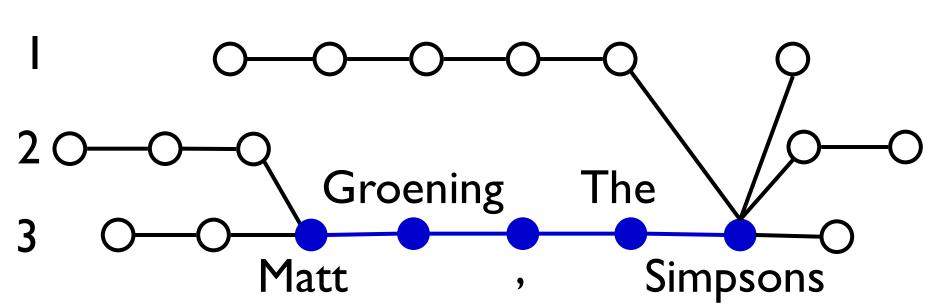
- Collapse on all segments => Intractable cyclic graph
- Collapse on few segments => Maybe get a tractable tree?

Approximation via Partitioning

Partition A into disjoint sets of shared segments $A_1,\ldots,A_k$

$$\log Z_{\text{fused}}(\mathcal{A}) \approx \sum_{i=1}^{k} \log Z_{\text{fused}}(\mathcal{A}_i)$$

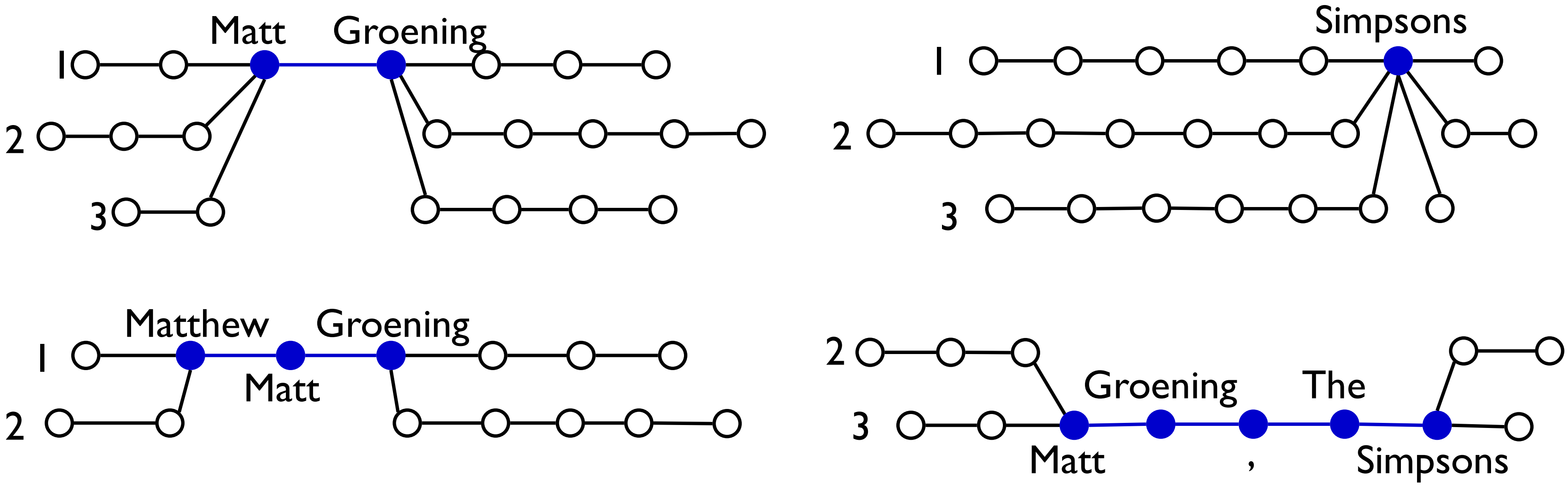




A_1 = Matt Groening,
Matthew Matt Groening

A_2 = Simpsons,
Matt Groening , The Simpsons

Per-segment Partitioning



Each fused graph = a shared segment + its chains = Tree
…But total number of nodes is the highest possible

Partitioning Desiderata

$$\min_{k, \mathcal{A}_1, \ldots, \mathcal{A}_k} \sum_i |\mathrm{FusedGraph}(\mathcal{A}_i)|$$

$$\mathcal{A}_1, \ldots, \mathcal{A}_k \text{ a partition of } \mathcal{A}$$

$$\forall i, \ \mathrm{FusedGraph}(\mathcal{A}_i) \text{ is a tree}$$

- Low runtime: Runtime linear in sizes of fused graphs
- Preserve correlation: Nearby shared segments should go to the same partition

e. g. "Matthew Matt Groening" and "Matt Groening"

Partitioning Desiderata

$$\min_{k, \mathcal{A}_1, \ldots, \mathcal{A}_k} \sum_i |\text{FusedGraph}(\mathcal{A}_i)|$$

$$\mathcal{A}_1, \ldots, \mathcal{A}_k \text{ a partition of } \mathcal{A}$$

$$\forall i, \;\; \text{FusedGraph}(\mathcal{A}_i) \text{ is a tree}$$

- NP-hard in size of agreement set
- Greedy strategy:
  - Grow A_i to maximally reduce objective
  - Tweaks and efficiency measures in paper

And we are done!

$$\max_{\{\mathbf{w_1},\ldots,\mathbf{w_S}\}} \sum_{i=1}^{S} LL(L_i|\mathbf{w_i}) + C \cdot \log \sum_{\mathbf{y}_{\mathcal{A}}} \prod_{i=1}^{S} p_i^{\mathrm{marg}}(\mathbf{y}_{\mathcal{A}}|\mathbf{w_i})$$

Equate to the Log Partition of the Fused Graph

$$\log \sum_{\mathbf{y}_{\mathcal{A}}} \prod_{i=1}^{S} p_i(\mathbf{y}_{\mathcal{A}}|\mathbf{w_i}) = \log Z_{\mathrm{fused}} - \sum_{i=1}^{S} \log Z_i$$

Decompose via Greedy Partitioning into Fused Trees

$$\log Z_{\mathrm{fused}}(\mathcal{A}) \approx \sum_{i=1}^{k} \log Z_{\mathrm{fused}}(\mathcal{A}_i)$$

Experiments: Structured Queries

User →

Gran Torino	Walt Kowalski	2008
Dirty Harry	Harry Callahan	1971

Collective Extraction

19. Sudden Impact (1983) .... Harry Callahan
20. Honkytonk Man (1982) .... Red Stovall
21. Firefox (1982) .... Mitchell Gant
22. Any Which Way You Can (1980) .... Philo Bedd
23. Bronco Billy (1980) .... Bronco Billy

24. Escape from Alcatraz (1979) .... Frank Morris

The Dead Pool (198
Heartbreak Ridge (
Pale Rider (1985)
Tightrope (1984)
City Heat (1984)
Sudden Impact (1983)

Joe Kidd (1972)	Joe Kidd
Dirty Harry (1971)	Inspector Harry Callahan
Play Misty for Me (1971)	Dave
The Beguiled (1971)	John McBurney
Kelly's Heroes (1970)	Kelly

Firefox	Mitchell Gant	1982
…	…	…
…	…	…

City Heat	-	1984
…	-	…
…	-	…

Joe Kidd	Joe Kidd	1972
…	…	…
…	…	…

Merge & de-duplicate, Rank, Display to the user

Experimental Setting

- Extraction on 58 datasets, each representing a relation
  - Oil spills, James Cagney movies, University mottos, Parrots in Trinidad & Tobago, Star Trek novels etc.
  - Each dataset = 2-20 HTML list sources from a 500M crawl
  - Wide range of #columns, #sources, #records, #shared segments, base accuracy, noise
  - Handful (~ 3) labeled records per list source
  - F1 measured using manually annotated ground truth
- Datasets binned by Base model F1 and Average Number of Shared Segments for ease of presentation

Finding the Agreement Set

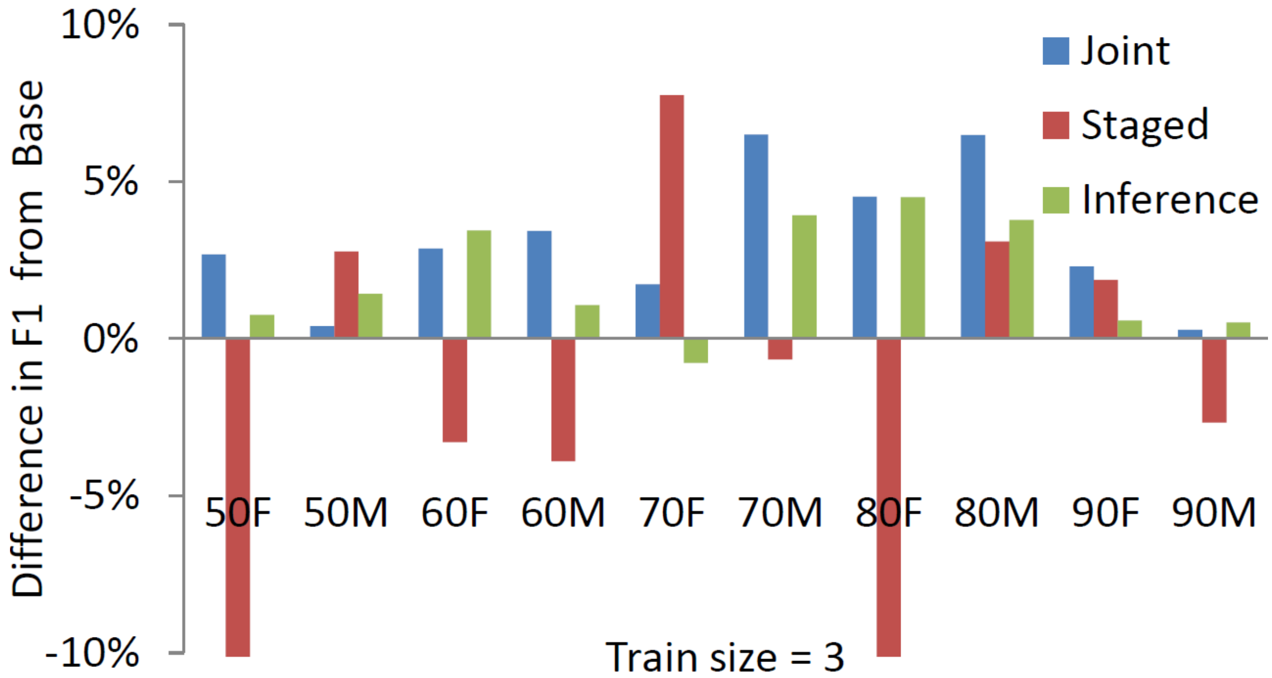
- Traditional: Shared segment = Unigram repetitions
  - Arbitrary, context-oblivious, highly noisy
  - Does not transfer weights of first-order features
- Our strategy:

Shared segment =
Repeating segment in near-duplicate records

Maximally long segment inside a record cluster

Approximate multi-partite matching of sources yields record clusters

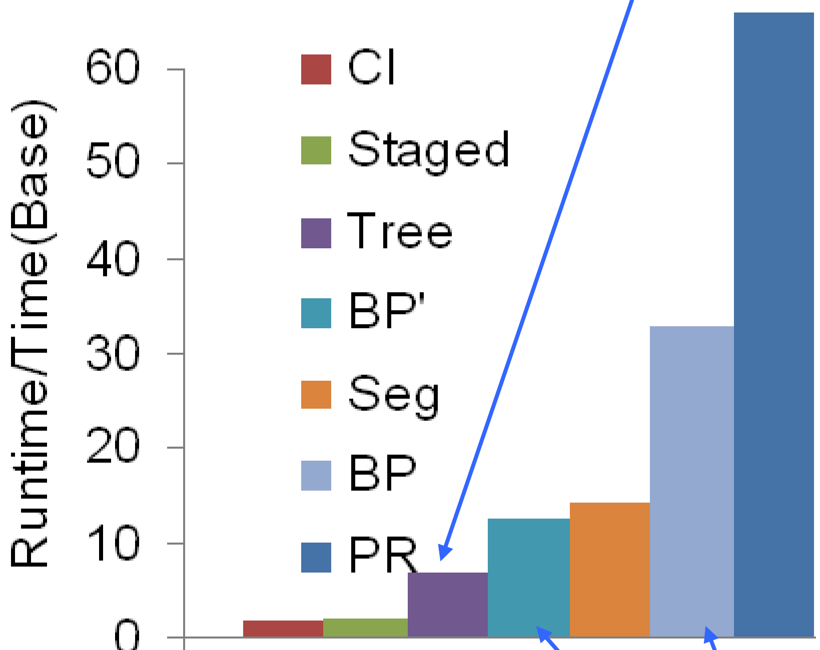
Comparison vs Simpler Methods

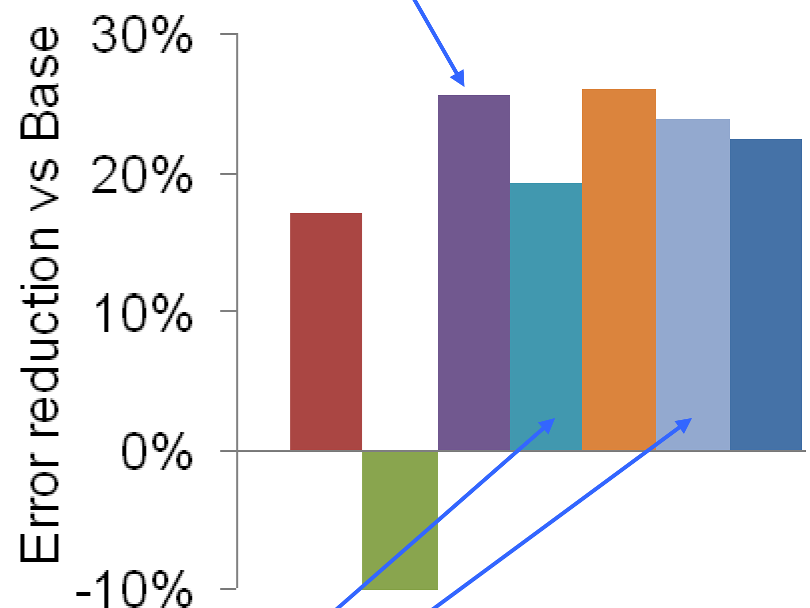


- Label transfer: cascade-prone, 10% drop in some cases
- Collective inference: boosts 83.3% to 86.1%
- Joint training: boosts to 87.5%
  - With 7 training records: boosts 87.4% to 89.2%

Runtime/Accuracy of All Methods

Greedy-partitioning has the best runtime/accuracy tradeoff





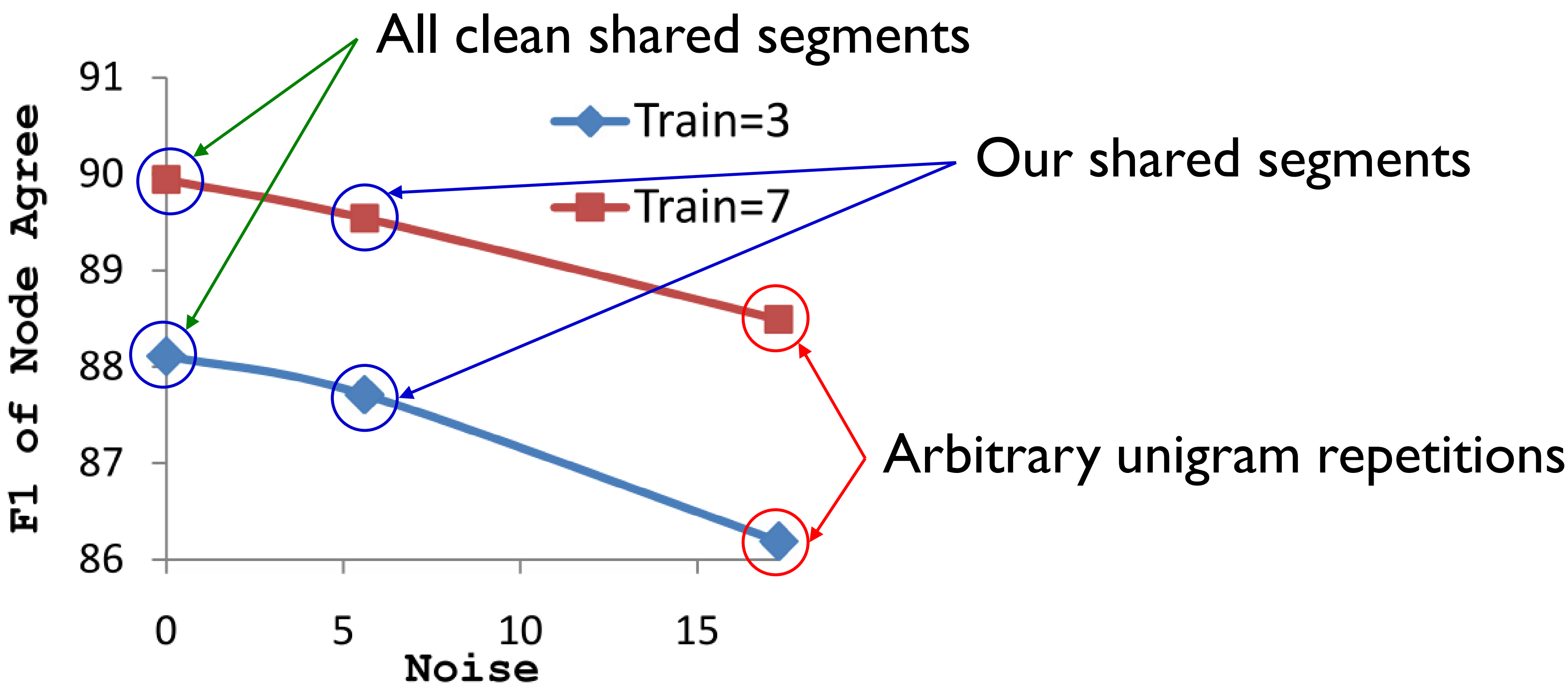
Belief Propagation (BP) quite slow,
Fast variant (BP’) not as accurate

Relative Error Reduction

	50F	50M	60F	60M	70F	70M	80F	80M	90F	90M	All
Absolute F1 Error of Base											
Base	44.8	45.4	33.1	32.7	26.5	23.9	14.4	13.4	5.7	3.9	16.7
Percentage Error Reduction over Base											
CInfer	1.7	3.2	10.4	3.3	-2.9	16.4	31.3	28.2	10.1	13.1	17.0
Tree	6.0	2.3	11.2	9.5	4.4	28.0	38.0	40.6	43.4	13.8	25.5
Seg	6.6	0.6	14.3	9.8	4.5	31.5	38.8	42.7	36.2	9.3	26.8
BP	6.0	2.4	10.6	9.3	3.6	28.7	38.6	42.0	43.3	14.9	26.0
BP'	1.6	2.1	11.8	3.5	-3.1	18.6	34.3	35.0	13.2	-0.5	19.1
PR	2.3	7.9	4.7	10.3	4.1	28.7	30.5	33.3	30.2	9.3	22.4

Red: Increase in error
Green: Best method

Experiments: Noisy Agreement Set



- Our scheme: ~5% token-level noise, small F1 drop
- Arbitrary unigrams: ~15% node noise, significant F1 drop

Related Work

- Agreement-based learning (Liang et.al. ’09)
  - EM-based scheme applied on two sources with clean overlap
- Posterior Regularization (Ganchev et.al. ’08)
  - Different agreement term; used in multi-view
- Two-view perceptron/regression, co - training/boosting/SVMs (Brefeld et.al. ’05, Blum & Mitchell ’98, Collins & Singer ’99, Sindhwani et.al. ’05, Kakade & Foster ’07)
  - Two source and/or hard label transfer
- Multi-task learning (Ando & Zhang ’05)
  - Single source, shared features sought
- Semi-supervised learning (Chapelle et.al. ’06)
  - No training, no support for partially structured overlaps
- Co-regularization, Pooling (Suzuki et.al. ’07)

Summary

- Joint training: Text overlap compensates for supervision
  - Reward agreement of distributions on overlapping text
  - Tractable approximations of the reward
  - Scheme to find low-noise overlapping segments
  - Extensive empirical comparison on many datasets

Best accuracy/speed tradeoff using content overlap
= Decomposing agreement over greedy tree partitions

- Future work
  - Online and parallel collective training

Thanks