Querying Probabilistic Information Extraction

Daisy Zhe Wang, Michael J. Franklin,
Minos Garofalakis, Joseph M. Hellerstein

VLDB, Singapore, 16th September, 2010

Outline

- Information Extraction Systems
  - Information Extraction (IE)
  - "Extract-then-Query" – *Standard* IE System
  - "Query-Time-Extraction" – *BayesStore IE* System
- Primer on CRF
- Query-Driven Extraction
  - Select-over-Top1 Queries
- Probabilistic SPJ Queries
  - Probabilistic Join Queries
- Experimental Results
- Conclusion

Information Extraction (IE)

- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference Wednesday, where Steve Jobs is expected to announce the birth of new stars in his product galaxy, including (probably) new iPods and (possibly) a successor to Apple TV.

  --- From WIRED August 30, 2010

Information Extraction (IE)

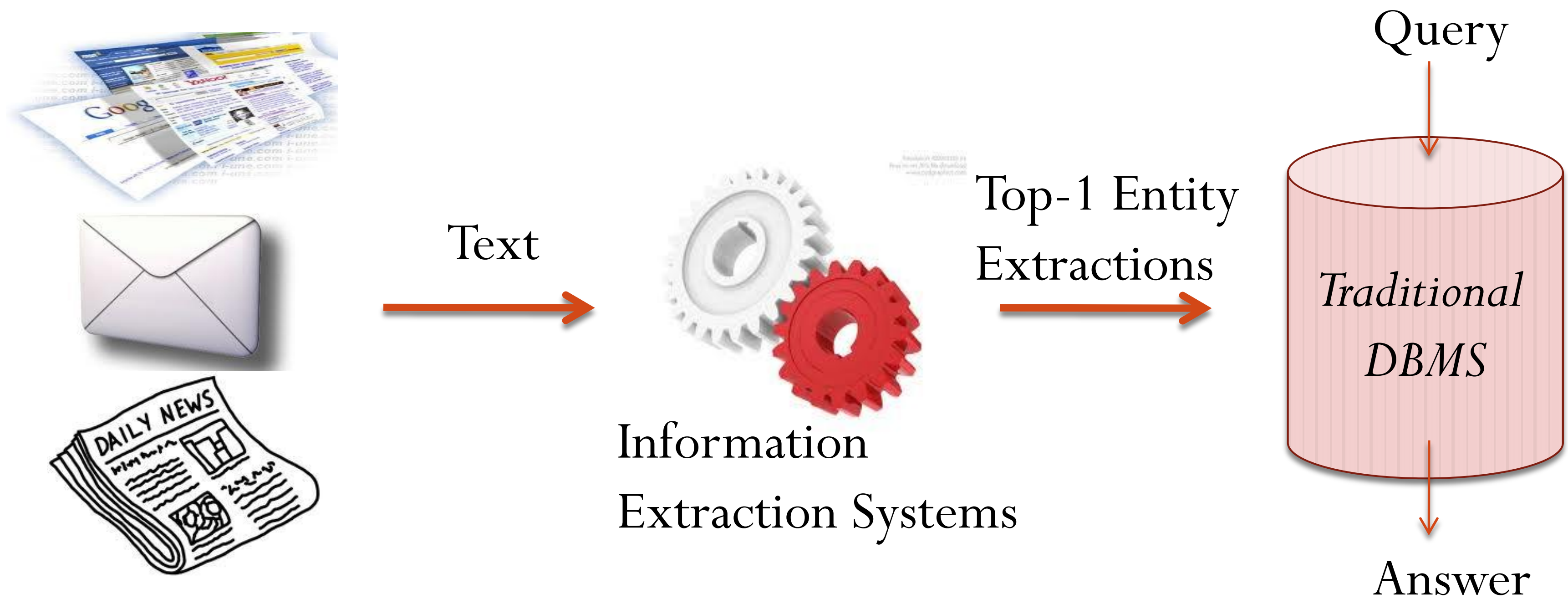
- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference Wednesday, where Steve Jobs is expected to announce the birth of new stars in his product galaxy, including (probably) new iPods and (possibly) a successor to Apple TV.

  --- *From WIRED August 30, 2010*

Labels:

Person   Company   Product   Event   Other

"Extract-then-Query" – Standard IE Systems



Problems:

1. Exhaustive extraction for all entities over all in-coming documents
2. Loses uncertainties and probabilities which are inherent in IE

Exhaustive vs. Query-Driven Extraction Example

Example Query:

SELECT *persons* FROM *blog* articles

WHERE company = "*Apple*"

- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference...
- The Big Apple lands '14 Super Bowl. Giants co-owner Jonathan Tisch said: "The greatest game will be played on the greatest stage!"...
- Apple Soufflé recipe by Julia Child: ... Pare, cut up, and stew ...

Exhaustive vs. Query-Driven Extraction Example

<u>Example Query:</u>

SELECT *persons* FROM *blog* articles
WHERE company = "*Apple*"

- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference...
- The Big Apple lands '14 Super Bowl. Giants co-owner Jonathan Tisch said: "The greatest game will be played on the greatest stage!"...
- Apple Soufflé recipe by Julia Child: ... Pare, cut up, and stew ...

Exhaustive vs. Query-Driven Extraction Example

Example Query:

SELECT *persons* FROM *blog* articles

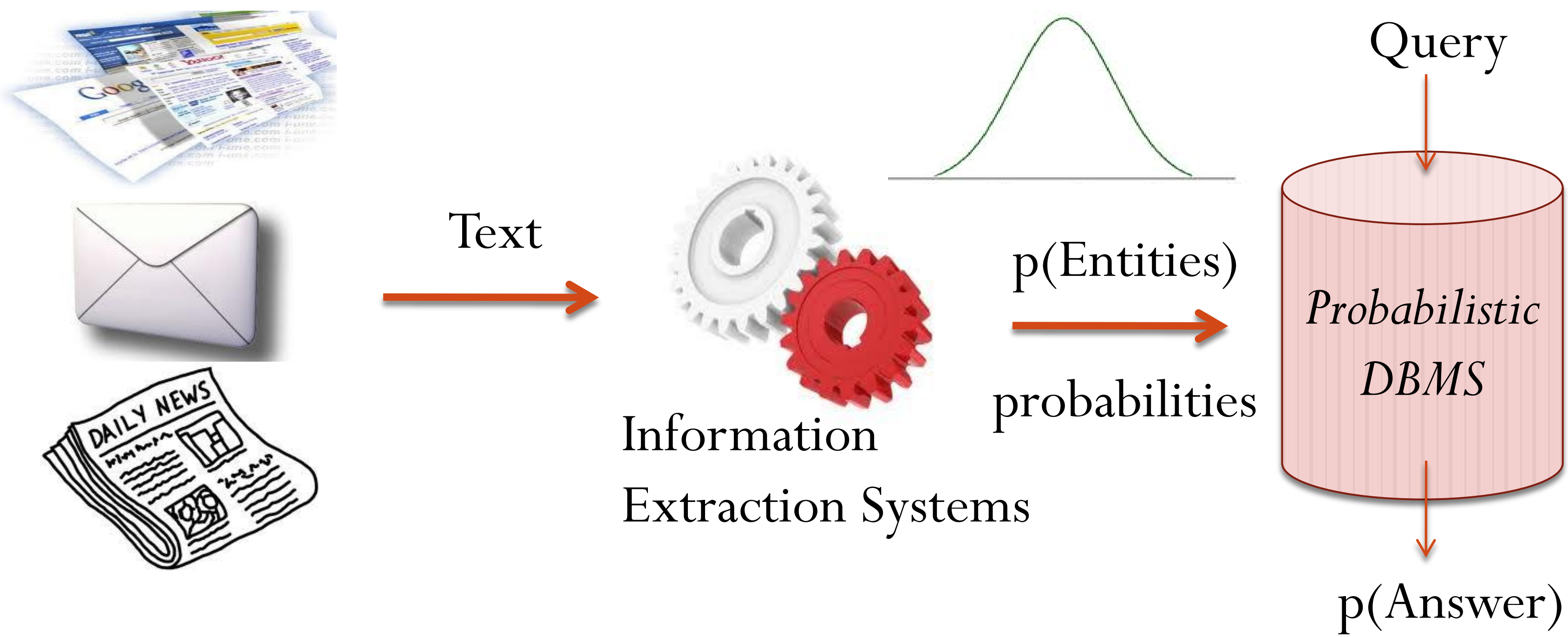
WHERE company = "*Apple*"

- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference...
- The Big Apple lands ✖
- Apple Soufflé recipes ✖

How to perform fast filtering without full inference?

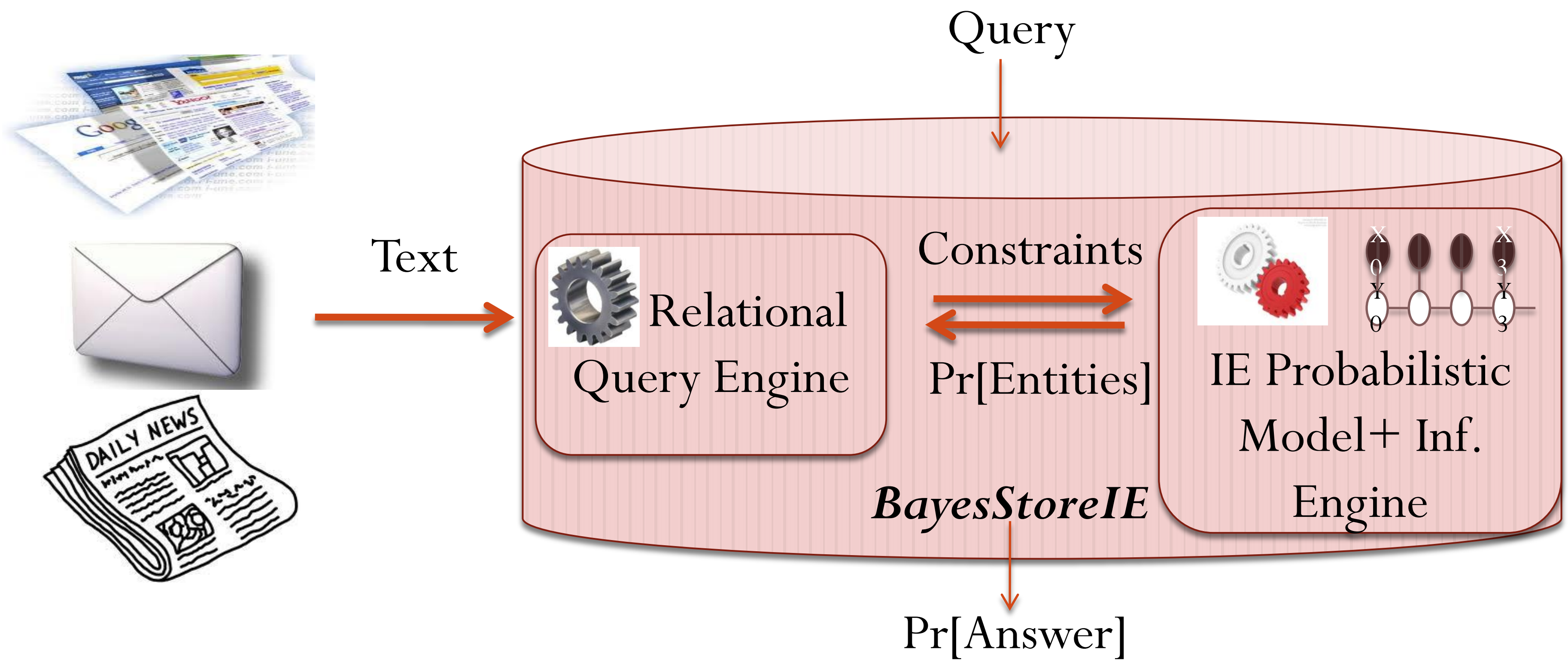
Challenge: Need to push condition *Label* = '*company*' into inference by deep integration of inference and relational ops.

“Extract-then-Query” – Storing Extractions and Probabilities



Still performs exhaustive extraction
Does not have the right representations to support IE probabilistic models inside of PDB *[Gupta, VLDB2005]*

"Query-Time-Extraction" – BayesStoreIE



Our Contributions:

- Deep Integration between Inference and Relational Operators
- Enable Query-Driven On-line Extraction
- Enable Probabilistic Queries over IE models

Outline

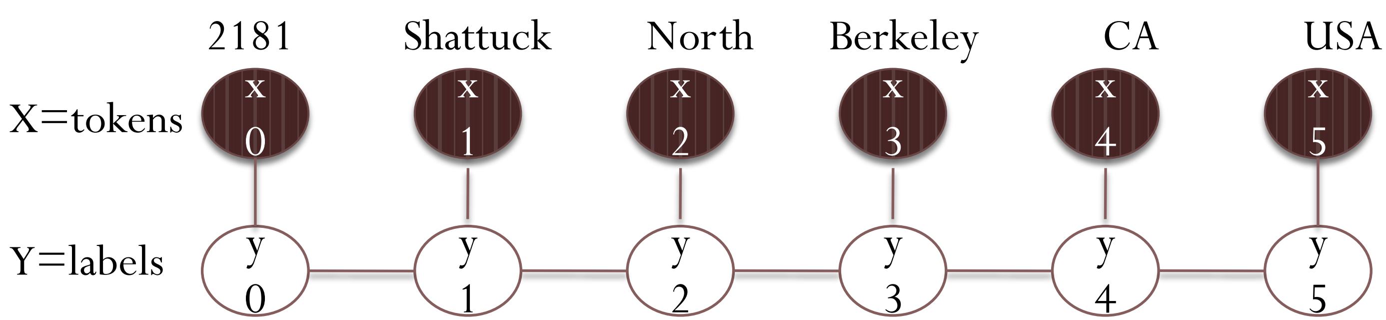
- Information Extraction Systems
  - Information Extraction (IE)
  - "Extract-then-Query" – *Standard* IE Approach
  - "Query-Time-Extraction" – *BayesStore IE* Approach
- Primer on CRF
- Query-Driven Extraction
  - Select-over-Top1 Queries
- Probabilistic SPJ Queries
  - Probabilistic Join Queries
- Experimental Results
- Conclusion

Conditional Random Fields (CRF)

Text (address string):

E.g., *"2181 Shattuck North Berkeley CA USA"*

CRF Model:



Possible Extraction Worlds:

x	2181	Shattuck	North	Berkeley	CA	USA	
y1	apt. num	street name	city	city	state	country	(0.6)
y2	apt. num	street name	street name	city	state	country	(0.1)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Two Query Families

Query Family 1: (SPJ-over-Top1)

Queries using only most-likely Extractions

Query Family 2: (Probabilistic SPJ)

Queries using probabilistic distributions

Query Family 1: Select-over-Top1

Example Query:

Select *

From Top-1 extractions of document set D

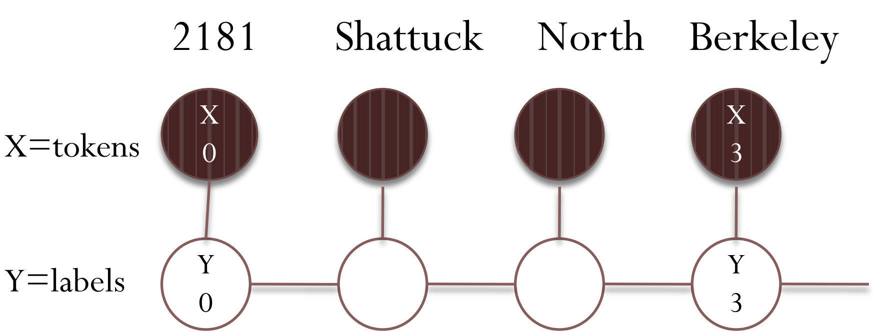
Where company like "%Apple%"

Viterbi Top-1 Inference on CRF

Viterbi Dynamic Programming Algorithm:

$$V(i,y) = \begin{cases} \max_{y'}(V(i-1,y') \\ \quad + \sum_{k=1}^{K} \lambda_k f_k(y,y',x_i)), & \text{if} \quad i \geqslant 0 \\ 0, & \text{if} \quad i = -1. \end{cases} \qquad (3)$$

CRF Model:



Dynamic Programming V matrix:

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	2	15	7	8	7
2	12	24	21	18	17
3	21	32	32	30	26
4	29	40	38	42	35
5	39	47	46	46	50

Query Family 1: Select-over-Top1 – Viterbi Early-Stopping Algorithm

<u>Example Query:</u>

Select *

From Viterbi-Top1 extractions of document set D

Where company like "%Apple%"

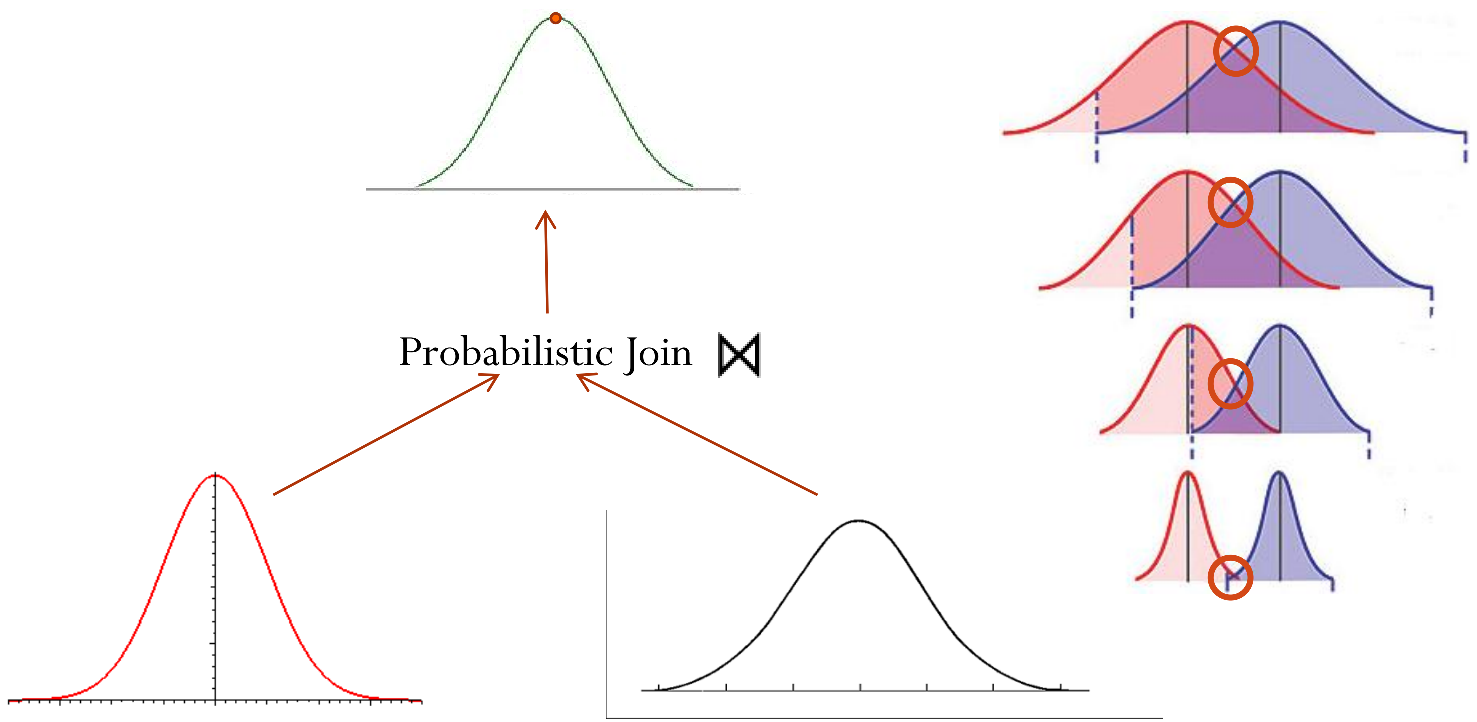
	pos	event	city	company	state	other
Big	0	5	1	0	1	1
Apple	1	2	15	7	8	7
lands	2	12	24	21	18	17
`14						
Super						
Bowl						

STOP!

Implemented in PostgreSQL using recursive queries and array functions

Query Family 2: Probabilistic Join

<u>Example Query:</u>
Select *Top-1* results
From extraction distributions of documents in $D1, D2$
Where $D1.city = D2.city$



Query Family 2: Probabilistic Join

<u>Example Query:</u>
Select *Top-1* results
From extraction distributions of documents in $D1, D2$
Where $D1.city = D2.city$

<u>Naïve algorithm:</u>

First compute *top-k* extractions for both input document sets, then compute join

<u>Problem:</u>

k needed to compute *Top-1* results varies for different documents

<u>Solution:</u>

Probabilistic Rank-Join algorithm based on Incremental Ranked Access to the List of Possible Extractions

Accessing Ranked List of Extractions – Incremental Viterbi Algorithm

- A novel variation of the Top-1 Viterbi algorithm, which computes the next highest-probability extraction *incrementally* and *more efficiently*

	pos	street num	street name	city	state	countr y
Sacramento	0	5	1	0	1	1
Avenue	1	2	15	7	8	7
San	2	12	24	21	18	17
Francisco	3	21	32,	32	30	26
CA	4	29	40	38	42	35
USA	5	39	47	46	46	50

Accessing Ranked List of Extractions – Incremental Viterbi Algorithm

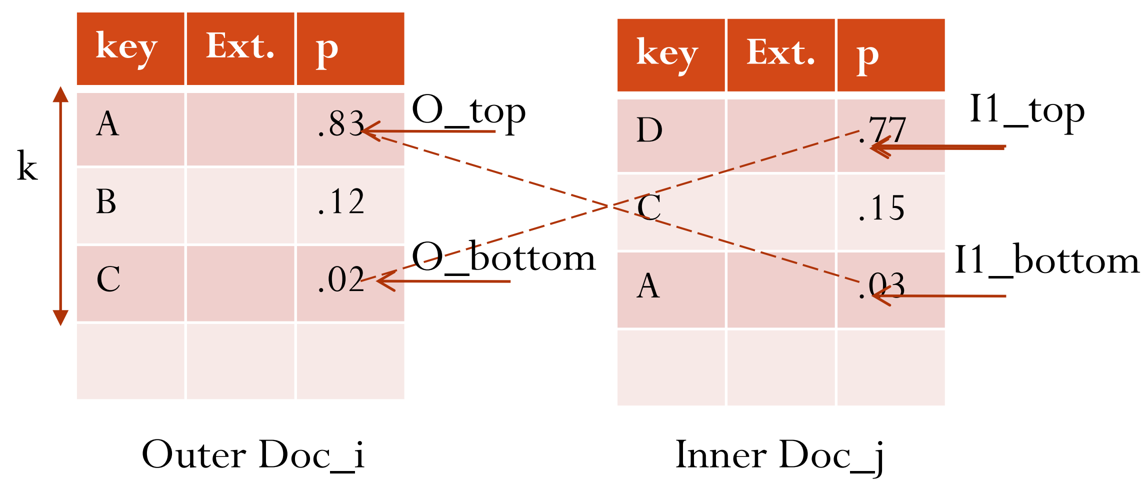
- A novel variation of the Top-1 Viterbi algorithm, which computes the next highest-probability extraction *incrementally* and *more efficiently*

	pos	street num	street name	city	state	countr y
Sacramento	0	5	1	0	1	1
Avenue	1	2	15,10	7	8	7
San	2	12	24,18	21	18	17
Francisco	3	21	32,	32,31	30	26
CA	4	29	40	38	42,38	35
USA	5	39	47	46	46	50,48

3rd highest-probability extraction can be computed by another call…

Probabilistic Rank-Join

Rank-join is applied to each pair of “joinable” document to compute *Top-1* join results



Probabilistic Rank-Join

A set of rank-joins are computed simultaneously for a set of outer documents and a set of inner documents

Outer Doc_1

k

key	Ext.	p
A		.83
B		.12
C		.02

O_top

O_bottom

....

Inner Doc_1

key	Ext.	p
D		.77
C		.15
B		.03

I1_top

I1_bottom

....

Inner Doc_n

key	Ext.	p
C		.95
D		.02
A		.01

Other Algorithms

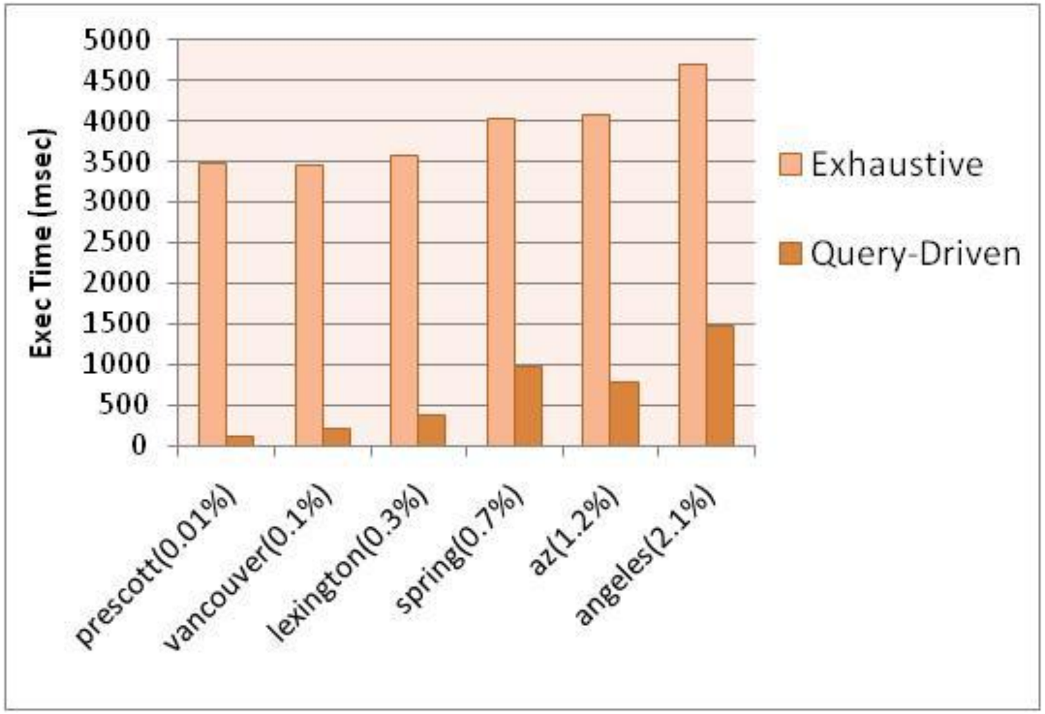
- Probabilistic Selection
- Probabilistic Projection
- Query-Driven Join-over-Top1

Outline

- Information Extraction Systems
  - Information Extraction (IE)
  - "Extract-then-Query" – *Standard* IE Approach
  - "Query-Time-Extraction" – *BayesStore IE* Approach
- Primer on CRF
- Query-Driven Extraction
  - Select-over-Top1 Queries
- Probabilistic SPJ Queries
  - Probabilistic Join Queries
- Experimental Results
- Conclusion

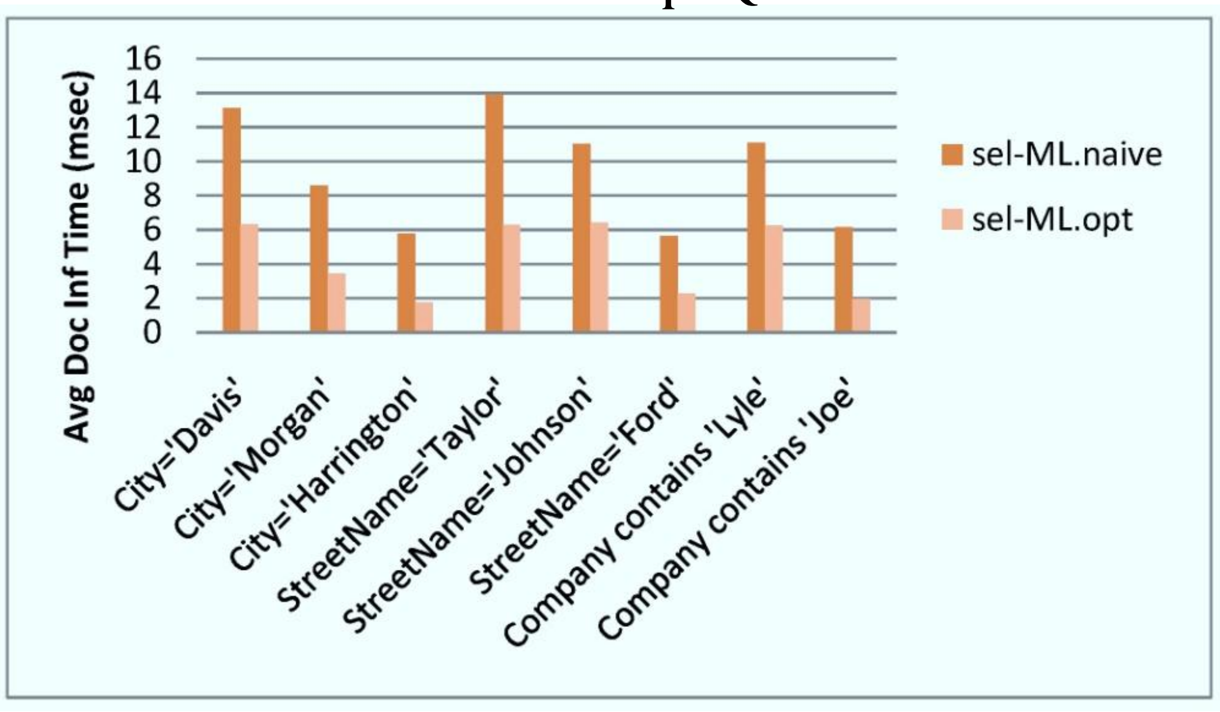
Evaluation 1: [Efficiency Improvement] Exhaustive vs. Query-Driven Extraction with Inverted Index

Select-over-Top1 Queries



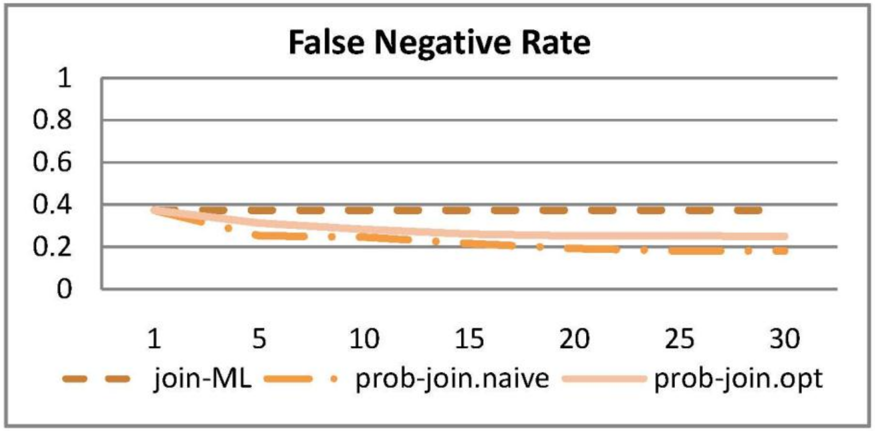
Evaluation 2: [Efficiency Improvement] Query-Driven Extraction Inverted Index vs. Early-Stopping

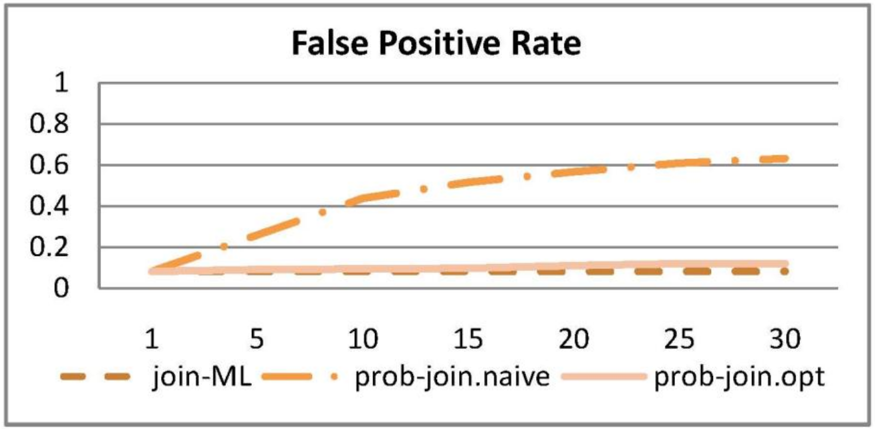
Select-over-Top1 Queries

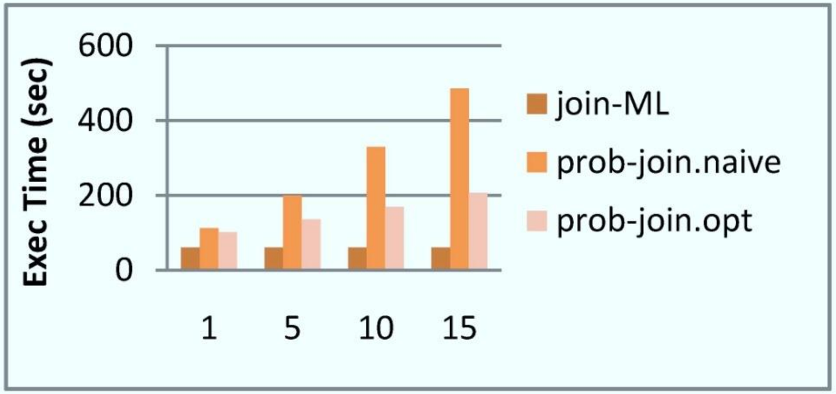


Take-away: Query-Driven Extraction improves Efficiency.

Evaluation 3: [Accuracy Improvement] Probabilistic Join vs. Join-over-Top1







Take-away: Probabilistic SPJ improves accuracy at a computation cost

A Query Design Space: efficiency vs. accuracy

Conclusion

- Querying Probabilistic IE
  - BayesStoreIE framework
  - Deep Integration of Relational and Inference
  - Query-Driven Extraction
  - Probabilistic SPJ Queries

- Current & Future Work
  - MCMC inference in DB
  - Conditional and Aggregation Queries in IE
  - Optimizer for Inference Operators (cost-accuracy co-optimization)

Thank you! … Questions?

BayesStore Project Page:

http://www.cs.berkeley.edu/~daisyw/BayesStore.html