

# Querying Probabilistic Information Extraction

*Daisy Zhe Wang*

University of California, Berkeley

Database Seminar, MIT, Nov. 17, 2010

# Probabilistic Data Analytics

## Information Extraction Systems



**Extracted Entities** (e.g. names,

Which NYTimes articles most-likely mention people in Turing Award page From Wikipedia?

## Sensor Networks



**Sensor readings** (e.g. light,

What's the **Gaussian distribution** of average pressure of the area?

## Data Integration Systems



**Integration Results** (e.g. schema

What is **top-10 probable** records of an employee called "Bond"?

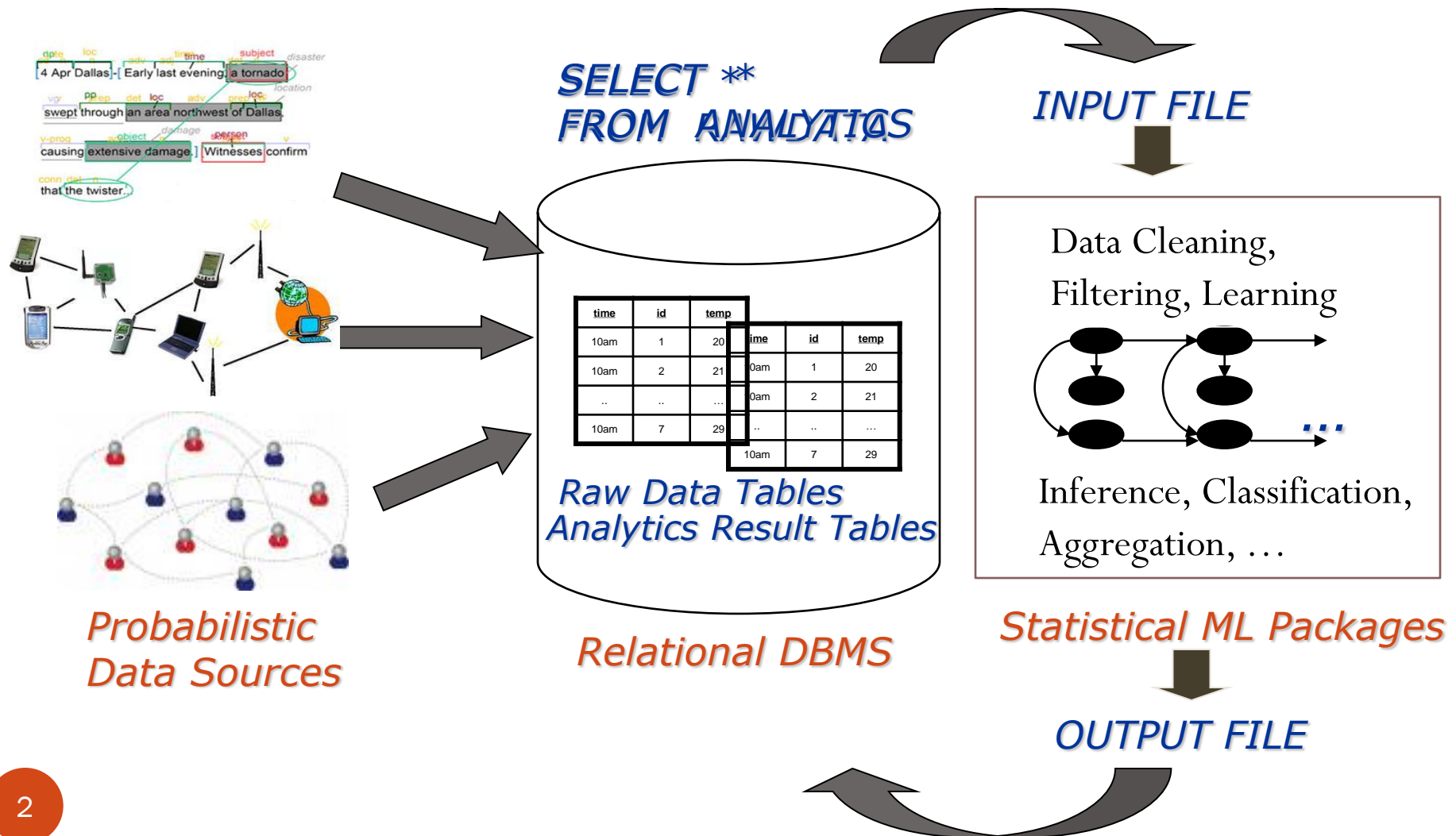
## Social Networks



**Predictive Analysis Results**

**How many** of soccer fans would be interested in the Ad for traveling to Argentina?

# State-of-the-art --- DB meets ML



# Problems and Solutions

- Problems
  - Scalability of SML packages
  - Expensive Data Transfer Cost
  - DB only as a Data Storage
  - $Q(\text{DB}(\text{SML Result})) \neq Q(\text{DB}+\text{SML})$
- Probabilistic Databases
- BayesStore Solutions – **Efficient and Scalable In-Database ML**
  - Represent Model and Probabilistic Data as First-class Objects
  - Implement Inference as Native Operators
  - Support Probabilistic Queries
  - Optimize across Inference and Relational Operations

# Part 1: Viterbi-based Probabilistic IE

- 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

# 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)

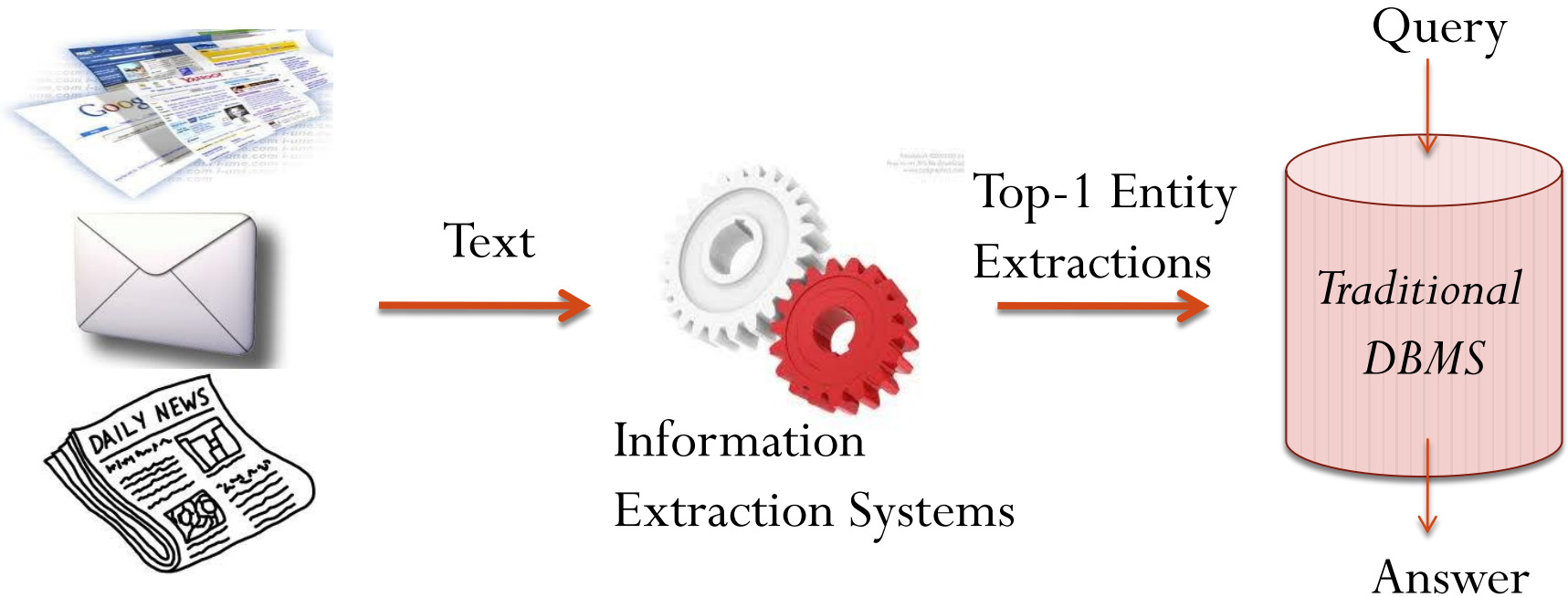
- Steve Jobs introduced the iPhone 4's videoconferencing feature FaceTime at WWDC 2010. Apple will hold a press conference Wednesday, where Steve Jobs is expected to announce the birth of new stars in his product galaxy, including (probably) new iPods and (possibly) a successor to Apple TV.

--- 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

## 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

## Example Query:

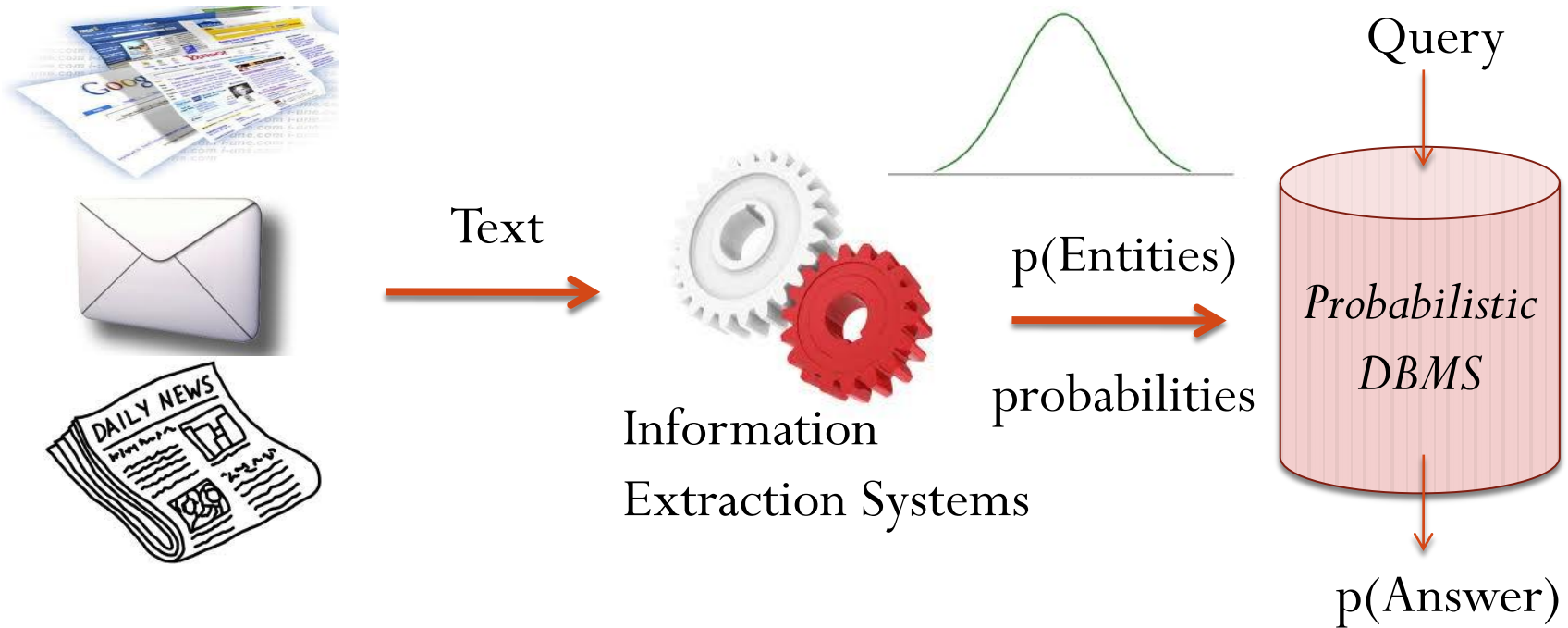
```
SELECT persons FROM blog articles  
WHERE company = "Apple"
```

- 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?

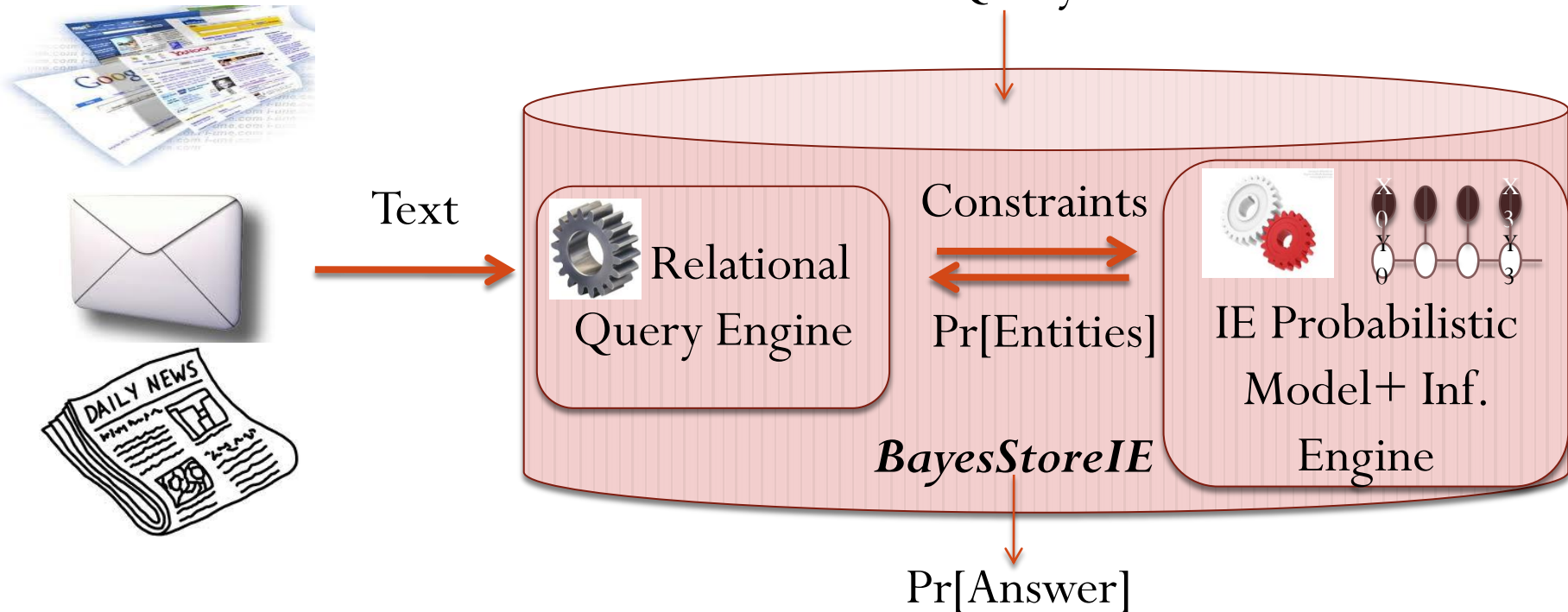
Challenge: Need to push condition *Label* = 'company' into inference by deep integration of inference and relational ops.

# “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



# 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

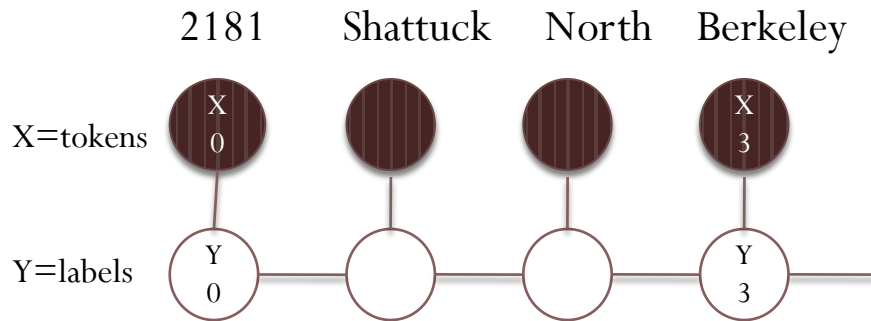
Where company like “%Apple%”

# Viterbi Top-1 Inference on CRF

Viterbi Dynamic Programming Algorithm:

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

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	32	47	46	46	50

# Text Data and CRF Representations

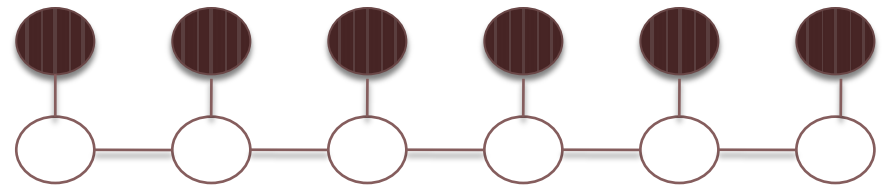
2181 Shattuck North Berkeley CA USA  
342 S Montezuma St Prescott AZ USA  
P.O. Box 210732 Minneapolis MN USA  
9330 Eastex Fwy Houston TX USA  
225 16th West Vancouver BC CAN  
1084 Salk Road Pickering ON CAN ...



docID	pos	token	Label
1	0	2181	
1	1	Shattuck	
1	2	North	
1	3	Berkeley	
1	4	CA	
1	5	USA	

Token Table

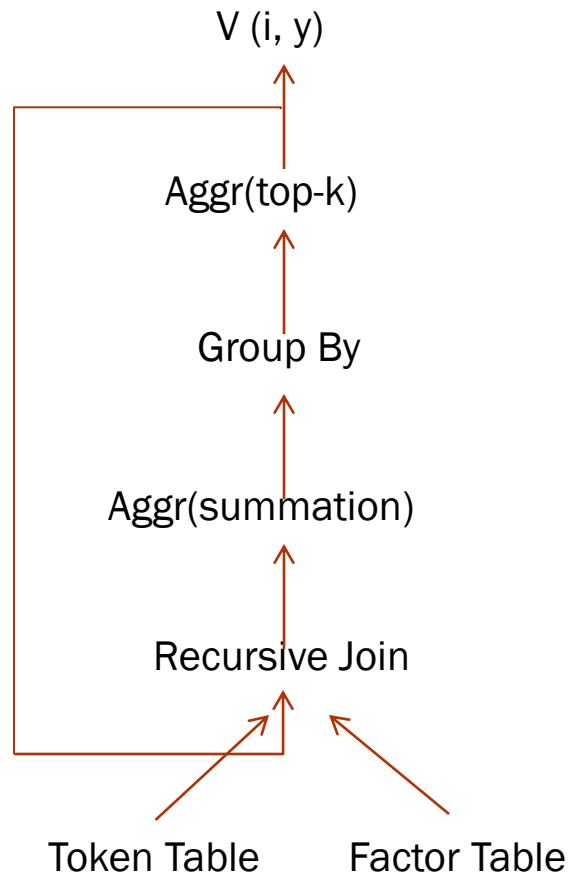
2181 Shattuck North Berkeley CA USA



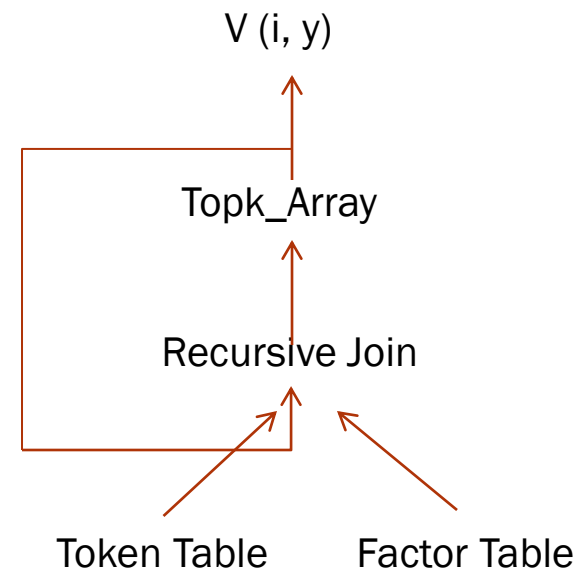
token	prevLabel	label	score
2181(DIGIT)	null	street num	22
2181(DIGIT)	null	street name	5
...	..	..	
Berkeley	streetname	street name	10
Berkeley	streetname	city	25
..	..	..	

Factor Table

# Viterbi Implemented in SQL



**ViterbiPerDoc (ViterbiAllDoc)**



**ViterbiArray**

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

Example Query:

Select \*

From Viterbi-Top1 extractions of document set D

Where company like “%Apple%”

	pos	event	city	comp any	state	other
<b>Big</b>	0	5	1	0	1	1
<b>Apple</b>	1	2	15	7	8	7
<b>lands</b>	2	12	24	21	18	17
<b>`14</b>						
<b>Super</b>						
<b>Bowl</b>						

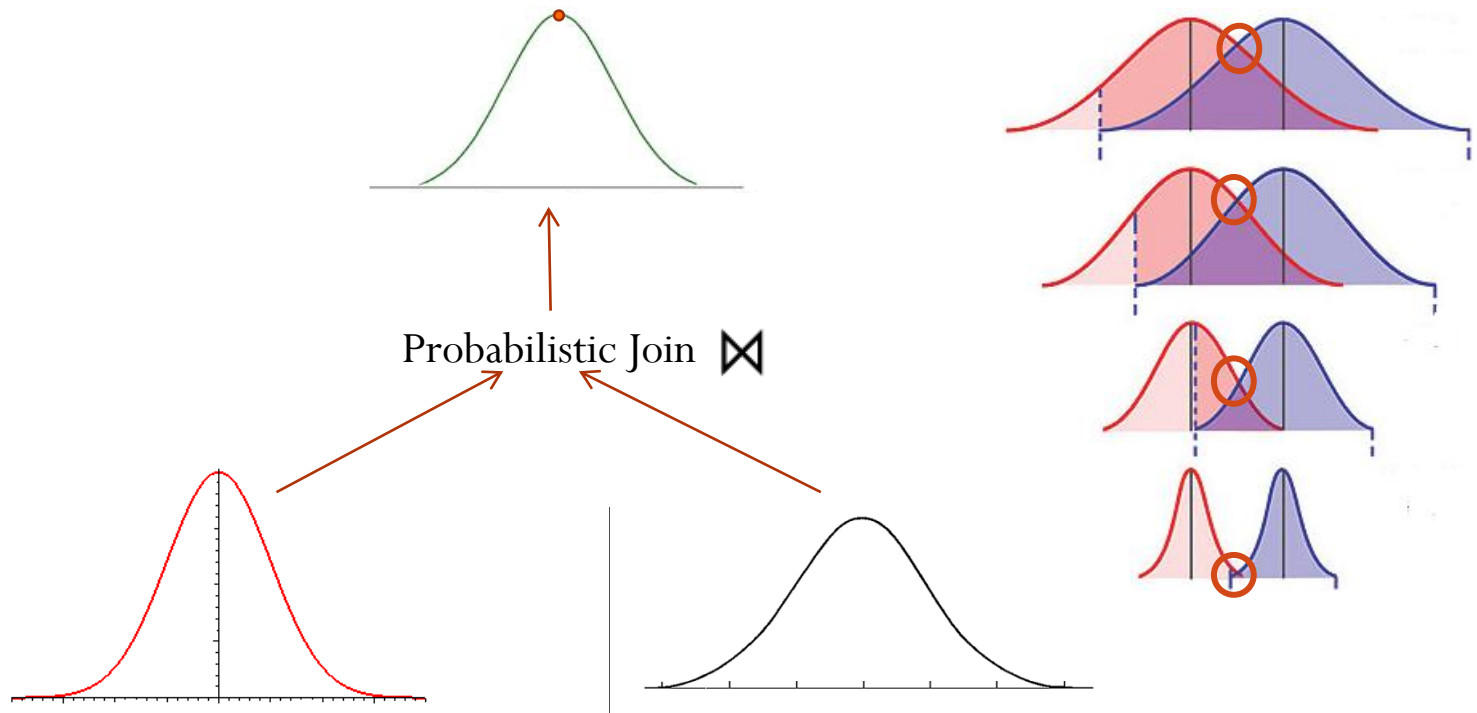
# Query Family 2: Probabilistic Join

Example Query:

Select *Top-1* results

From extraction distributions of documents in  $D1, D2$

Where  $D1.city = D2.city$



# Query Family 2: Probabilistic Join

## Example Query:

Select *Top-1* results

From extraction distributions of documents in  $D1, D2$

Where  $D1.city = D2.city$

## Naïve algorithm:

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

## Problem:

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

## Solution:

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*

**Sacramento  
Avenue  
San  
Francisco  
CA  
USA**

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

# 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*

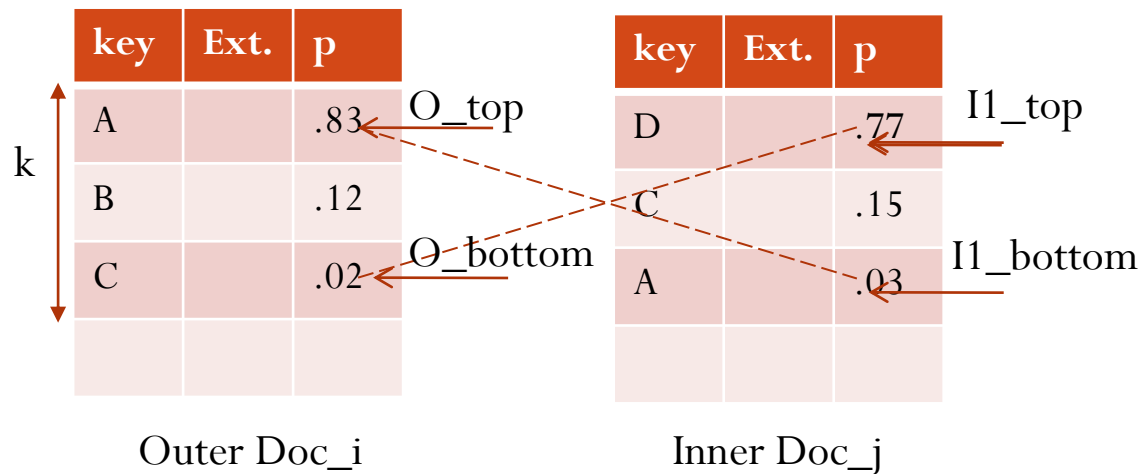
**Sacramento  
Avenue  
San  
Francisco  
CA  
USA**

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

3<sup>rd</sup> highest-probability extraction can be computed by another call...

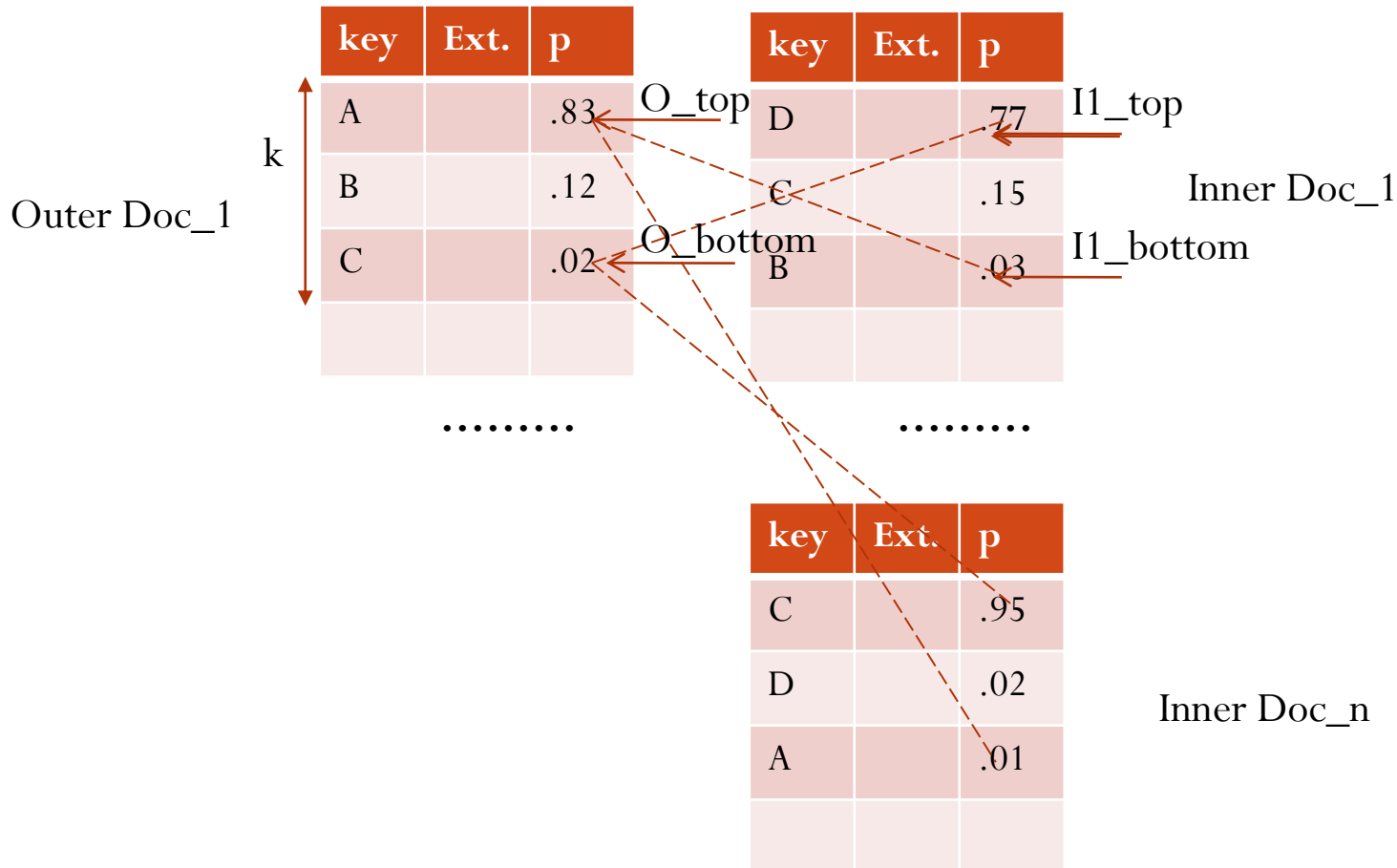
# Probabilistic Rank-Join

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



# Other Algorithms

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

For more details, please refer to Wang et. al. VLDB2010.

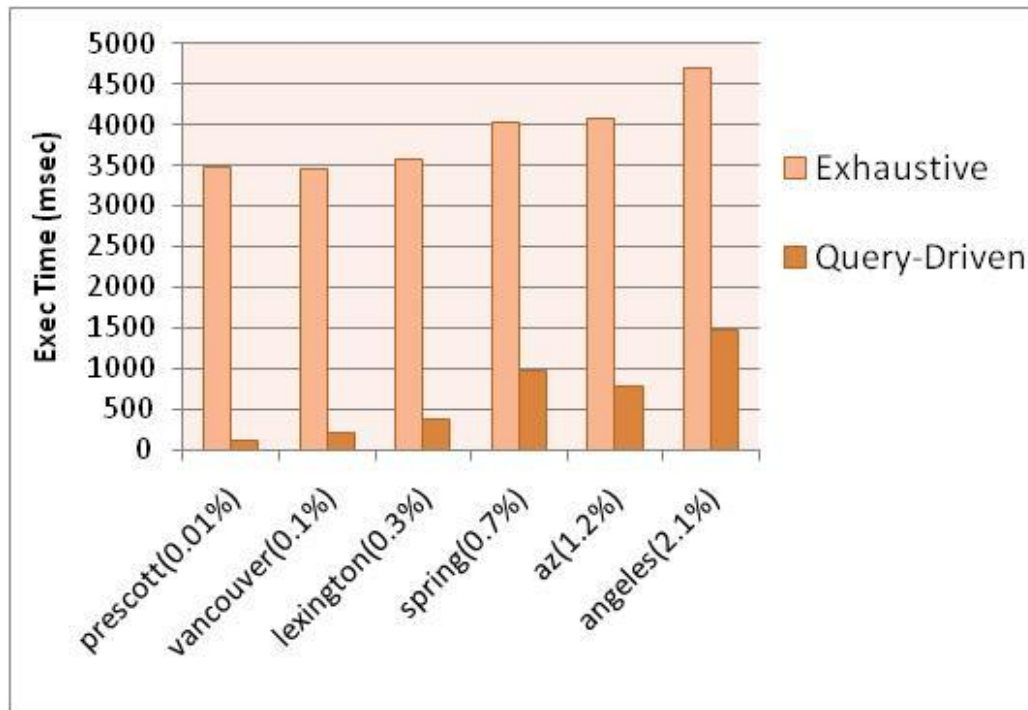
# Outline

- 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**

# Evaluation 1: [Efficiency Improvement]

## Exhaustive vs. Query-Driven Extraction with Inverted Index

Select-over-Top1 Queries

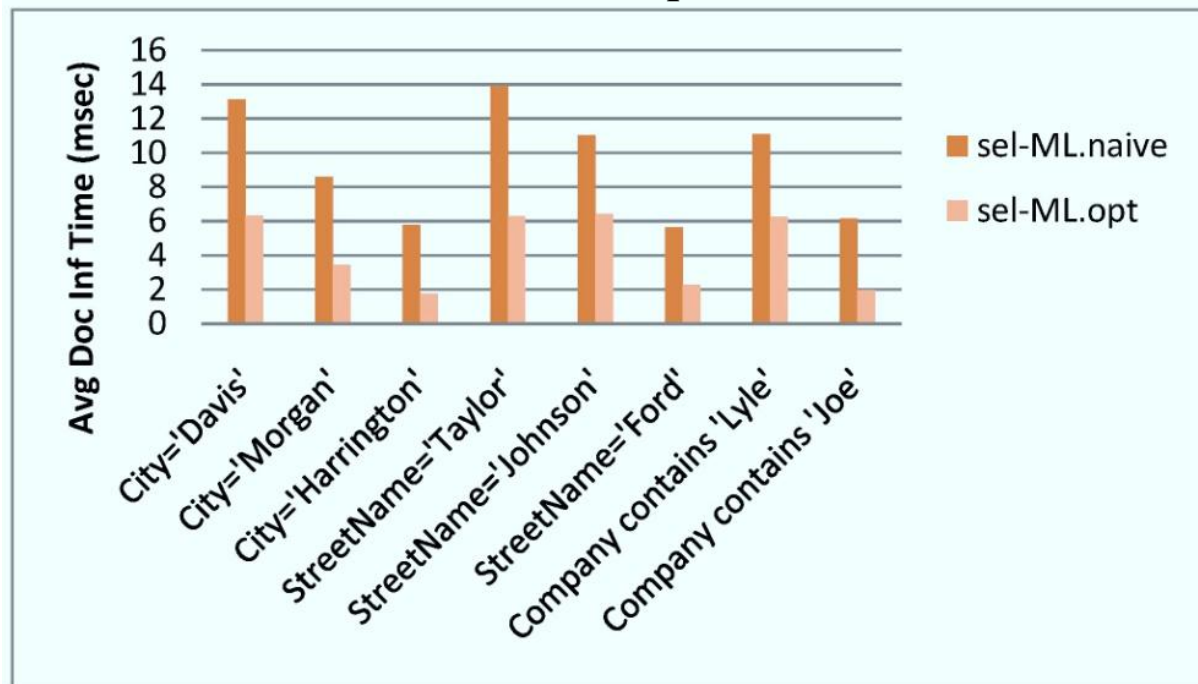


# Evaluation 2: [Efficiency Improvement]

## Query-Driven Extraction

### Inverted Index vs. Early-Stopping

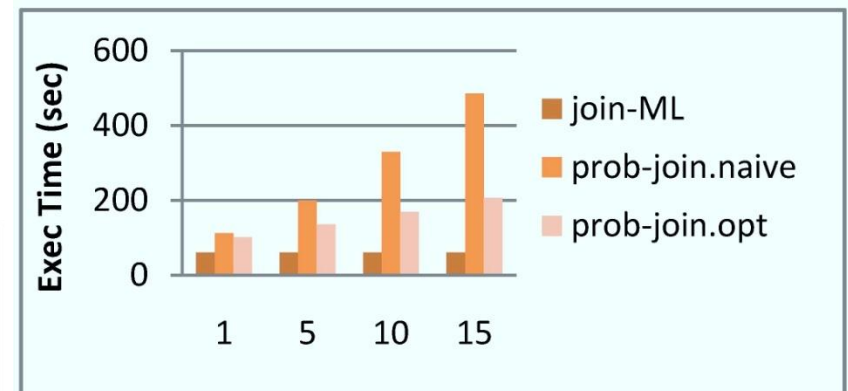
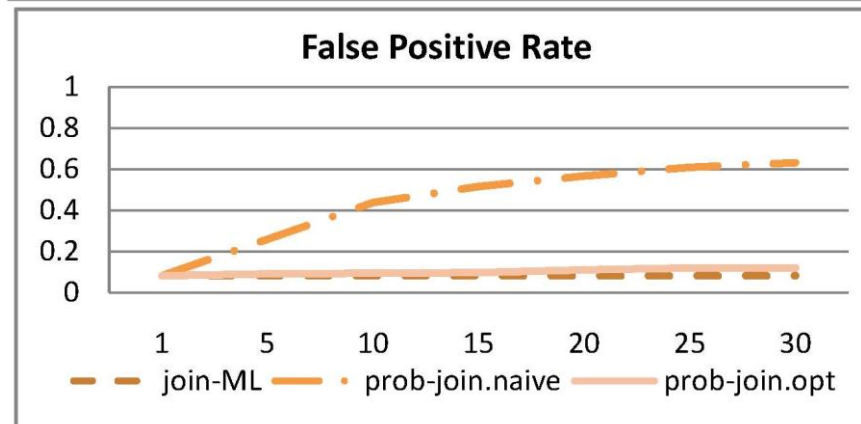
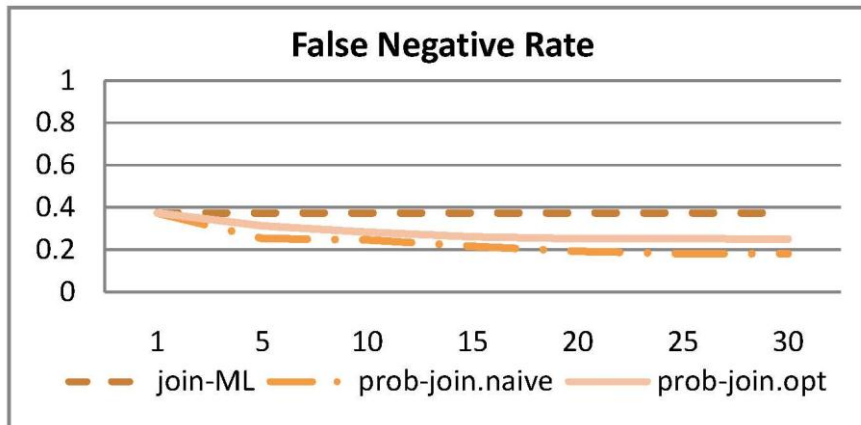
Select-over-Top1 Queries



Take-away: Query-Driven Extraction improves Efficiency.

# Evaluation 3: [Answer Quality Improvement]

## Probabilistic Join vs. Join-over-Top1



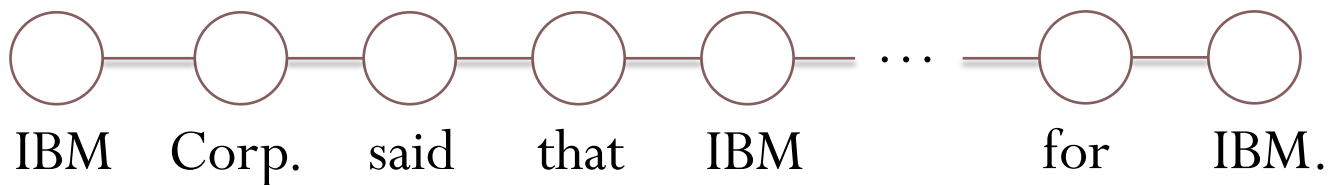
Take-away: Probabilistic SPJ improves answer quality at a computation cost  
A Query Design Space: efficiency vs. accuracy

# Part 2: Sampling-based Probabilistic IE

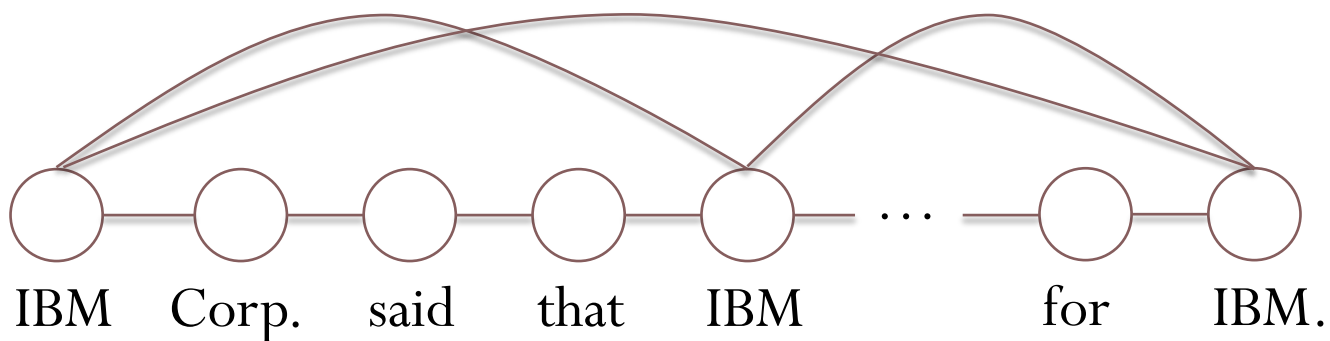
- Current Work
  - In-Database Gibbs Sampling Inference
  - Hybrid Inference

# Cycles From IE Models

Liner-chain CRF: (Top-1)

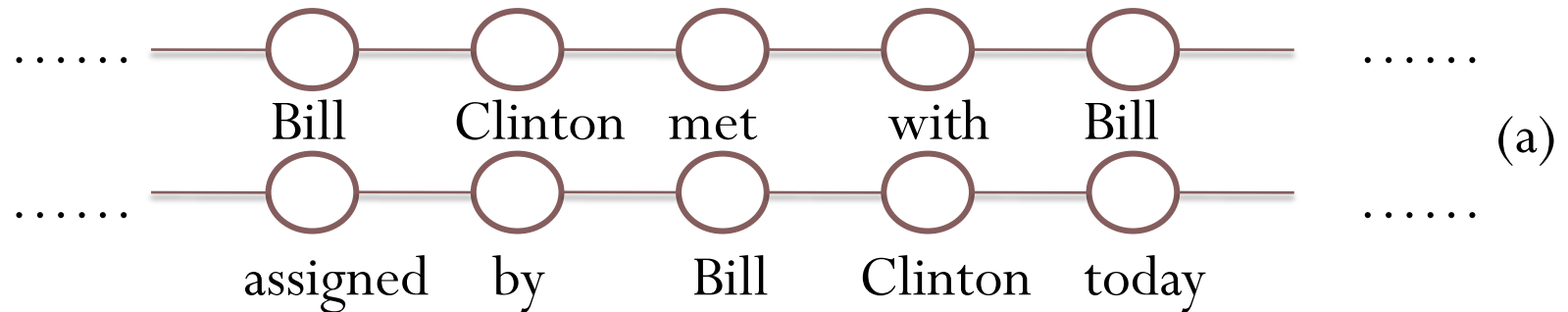


Skip-chain CRF: (Marginal)

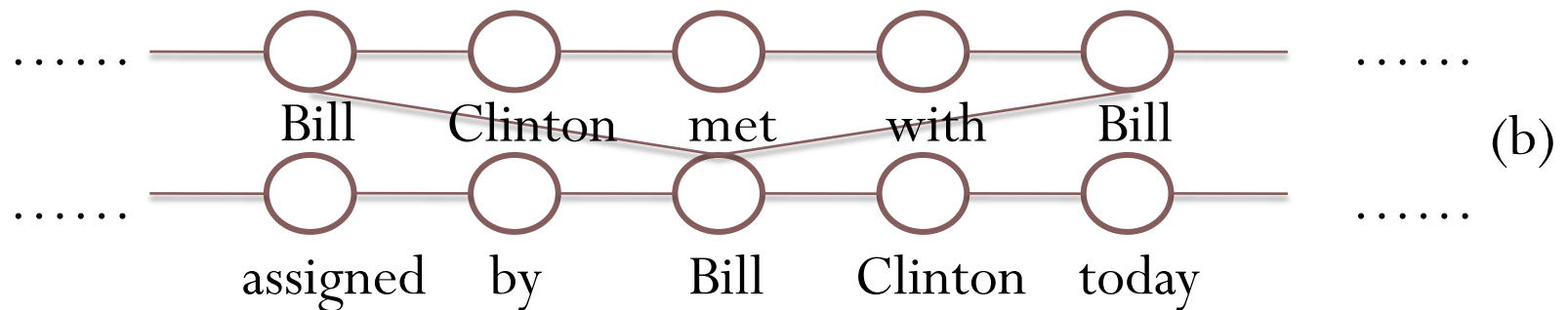


# Cycles induced From IE Queries

A pair of Liner-chain CRFs:



A join between Liner-chain CRFs:



# MCMC Sampling-based Inference

- General Inference algorithm
- Generate Samples of the result distribution
- Different variations
  - Metropolis Hastings
  - Gibbs Sampling

# Gibbs Sampling

## Algorithm (sketch):

### 1. Initialization:

- a. Select random labels for all tokens in TokenTbl:

$$w_0(y_1, \dots, y_N)$$

- b.  $w \leftarrow w_0$

### 2. Sampling Iterations: (For 1 ... M)

- a. Pick token  $x_i$  sequentially from TokenTbl

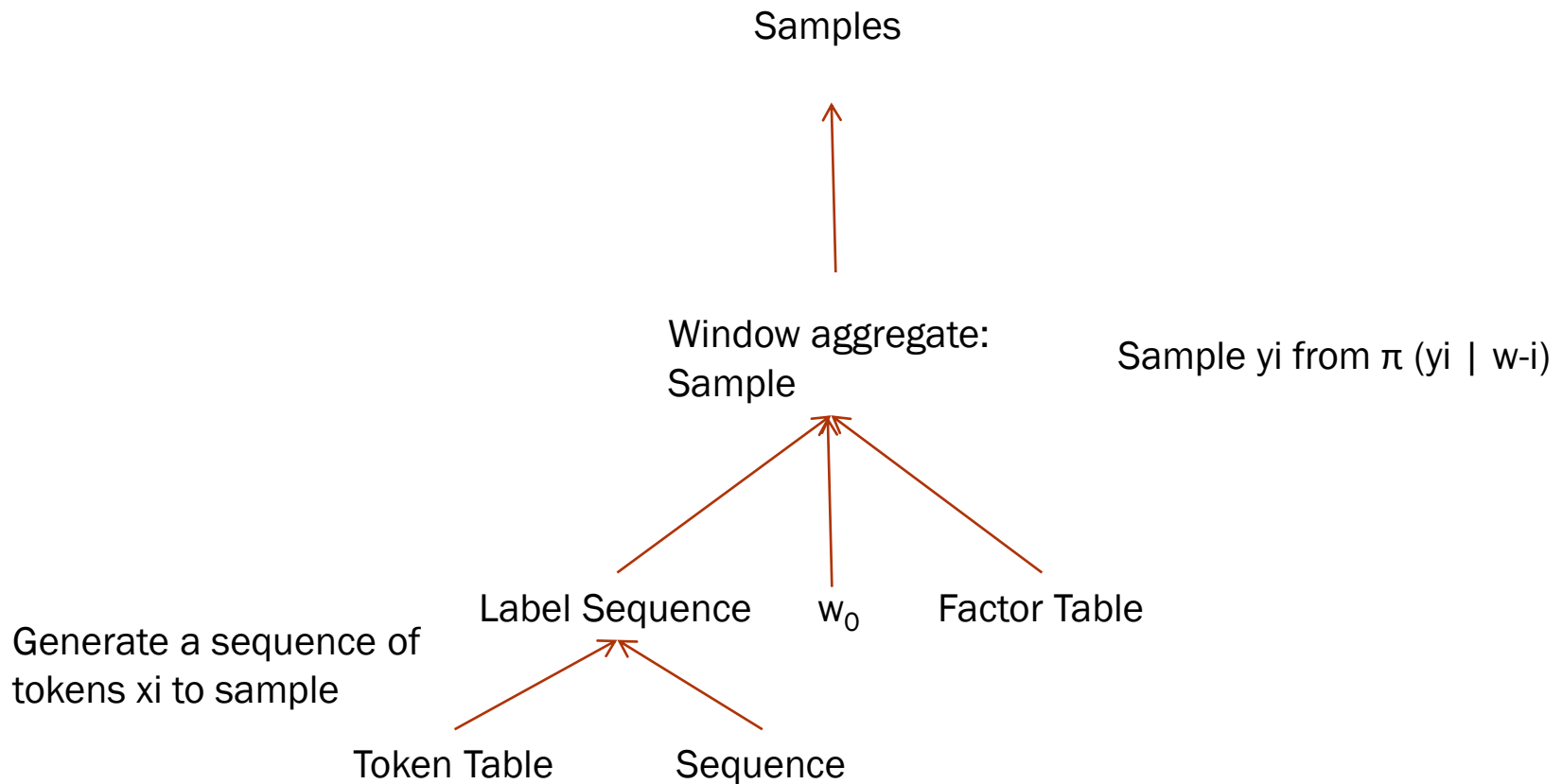
- b. Sample  $y_i$  from  $\pi(y_i | w_{-i})$

- c.  $w \leftarrow w_{-i} + y_i$

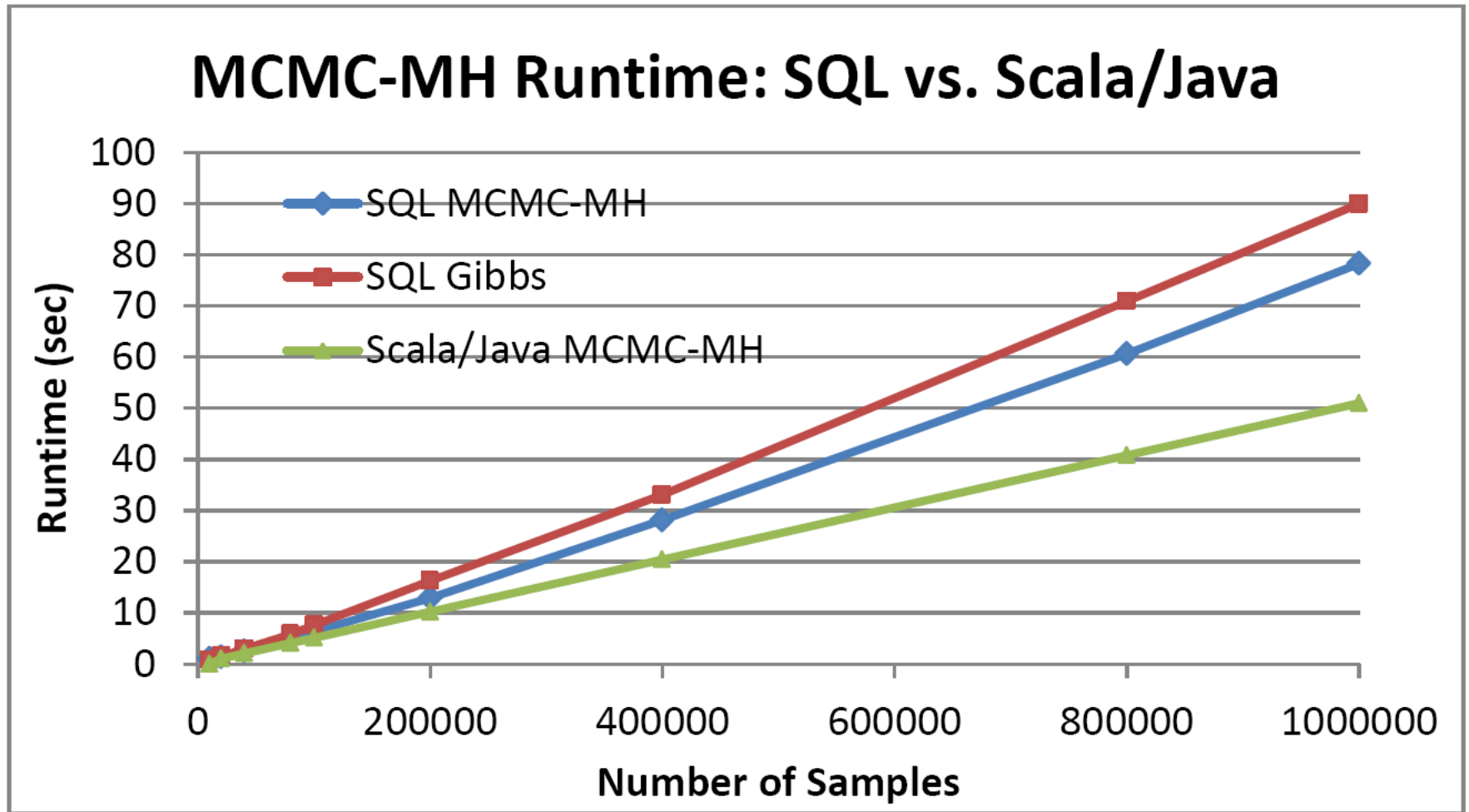
- d. return  $w$  as the next sample

This iterative algorithm can be implemented in a PL/pgSQL UDF.

# SQL Implementation of Gibbs



# Gibbs Runtime Comparison: Java/Scala vs. SQL Implementation



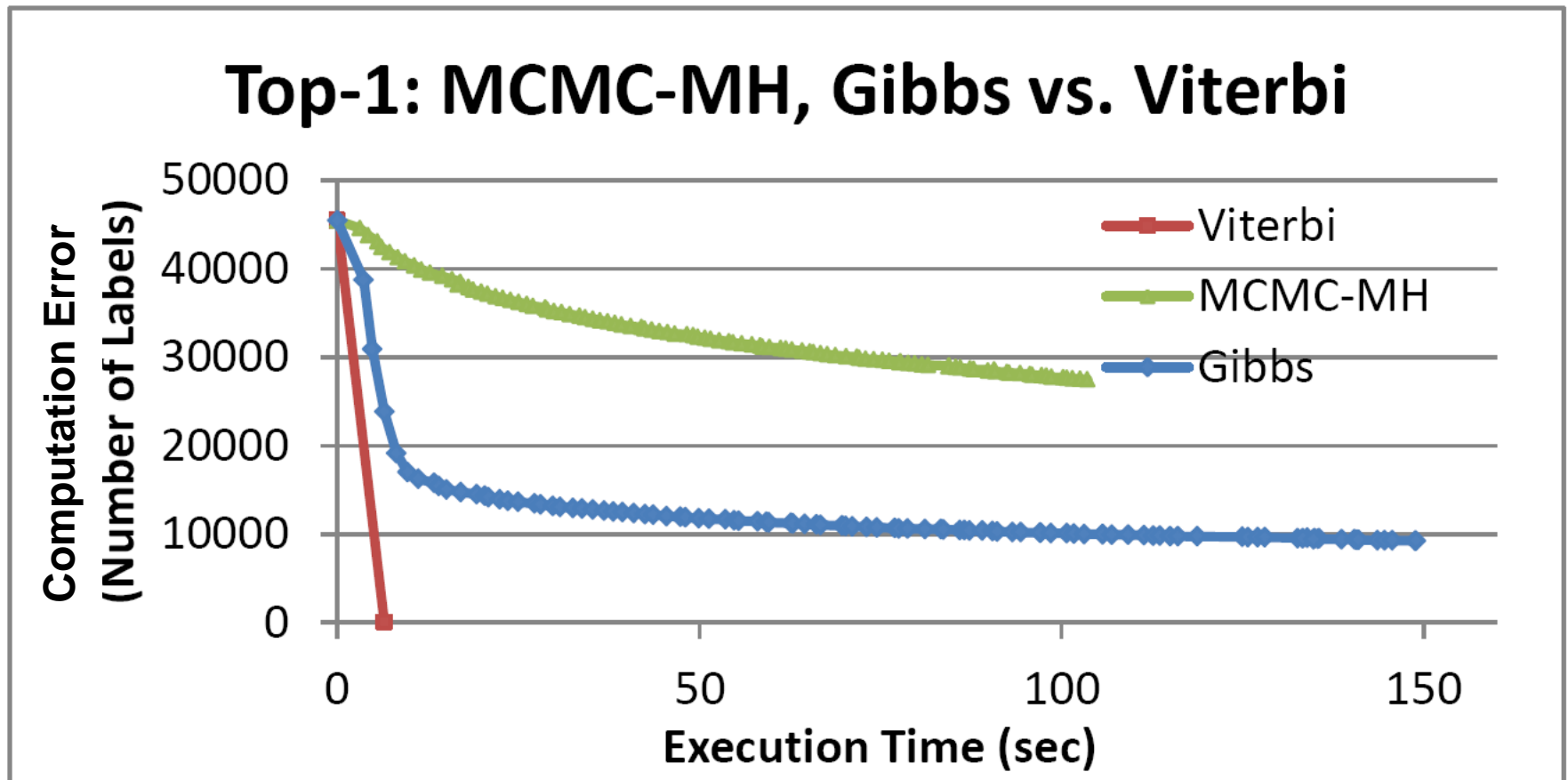
# Applicability of Inference Algorithms over Models and Queries

inference algorithm	Model			Query			Constraints	
	Top-1 Chain	Tree	Cyclic	Marginal Chain	Tree	Cyclic	Some	Arbitrary
Viterbi	✓						✓	
Sum-Product		✓		✓	✓		✓	
MCMC-MH	✓	✓	✓	✓	✓	✓		✓
Gibbs	✓	✓	✓	✓	✓	✓	✓	

Viterbi only works on Linear-Chain Models.

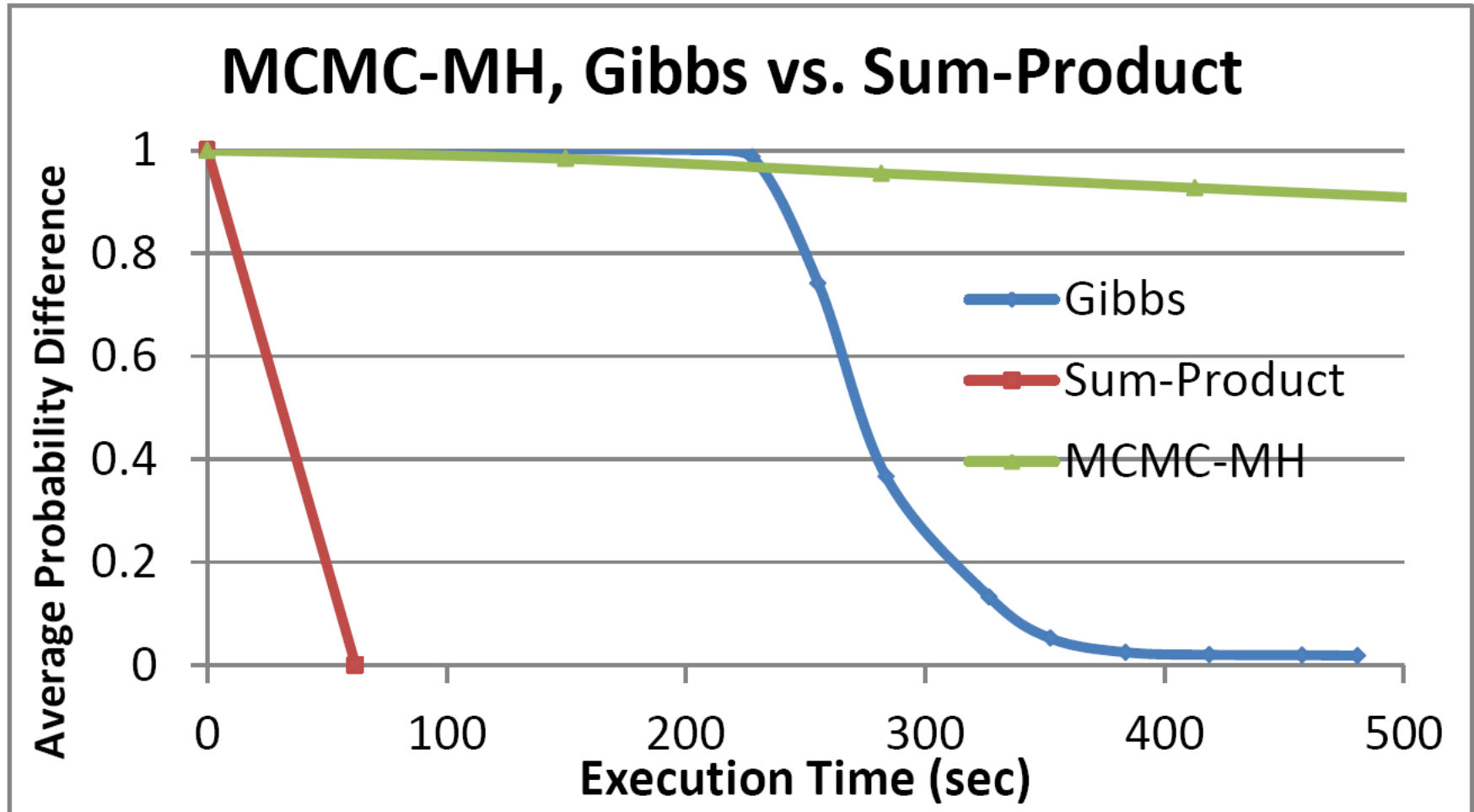
Exact Sum-Product algorithm can only apply to Tree-shaped models.

# Runtime-Accuracy Comparison for Top-1(MAP) Extraction: Gibbs vs. Viterbi



Viterbi is 10 times faster than Gibbs Sampling and is exact algorithm.

# Runtime-Accuracy Comparison for Marginal Inference: Gibbs vs. Sum-Product



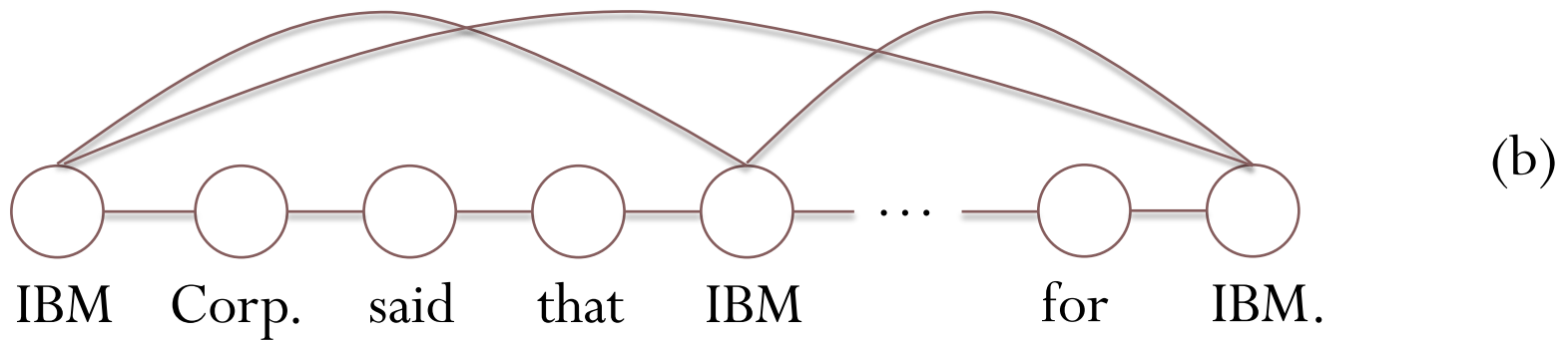
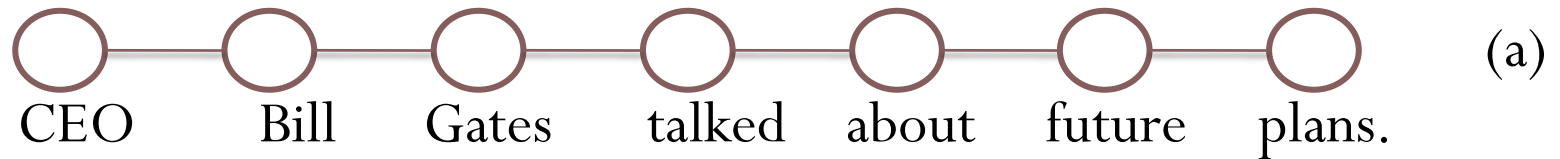
# Query-, Model-, Data-Dependent Inference Selection

- Applicability, Accuracy, Runtime
- Depends on the query (e.g., top-1 vs. marginal)
- Depends on the model structure (e.g., cyclic vs. chain)
- For CRF in IE, model structure depends on the text data!
- Hybrid Inference

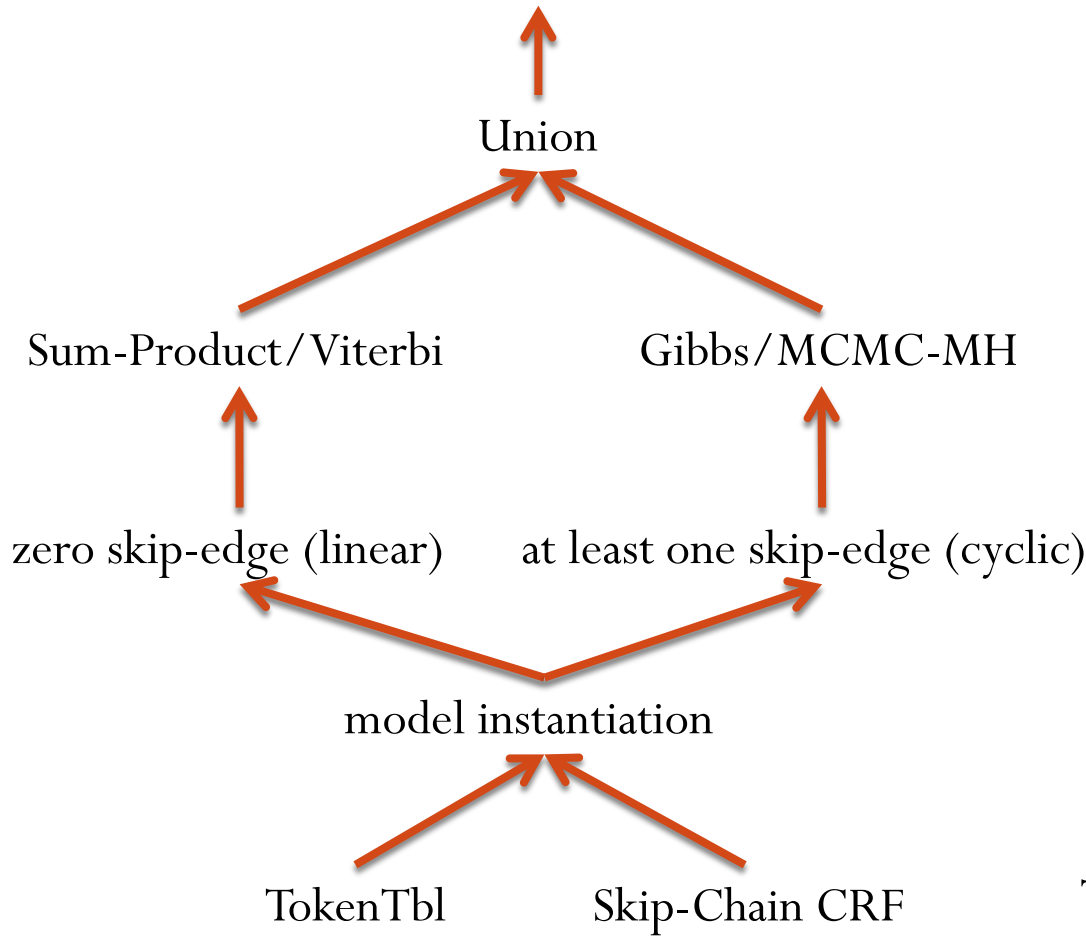
# Example 1: Skip-Chain IE Models

```
SELECT [Top-1(T1.docID) | Marginal(T1.pos | exist)]  
FROM   TokenTblT1
```

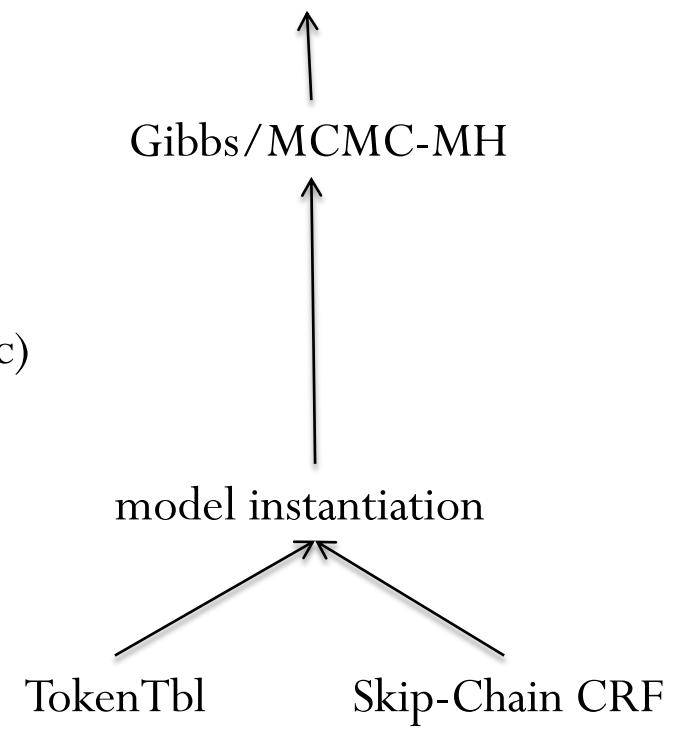
# Inference Selection & Hybrid Inference



# Execution Strategy for Skip-Chain Model



(a)

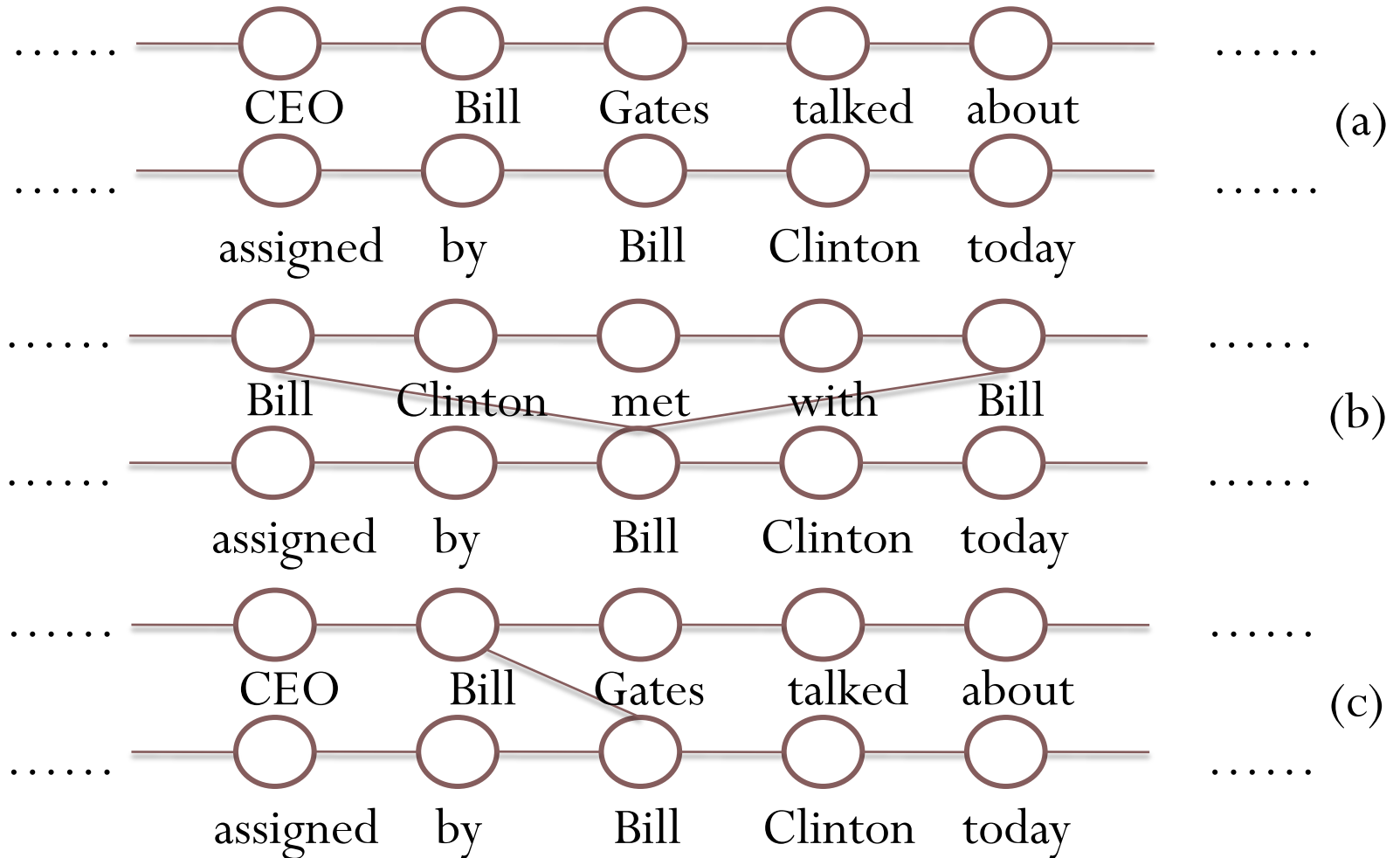


(b)

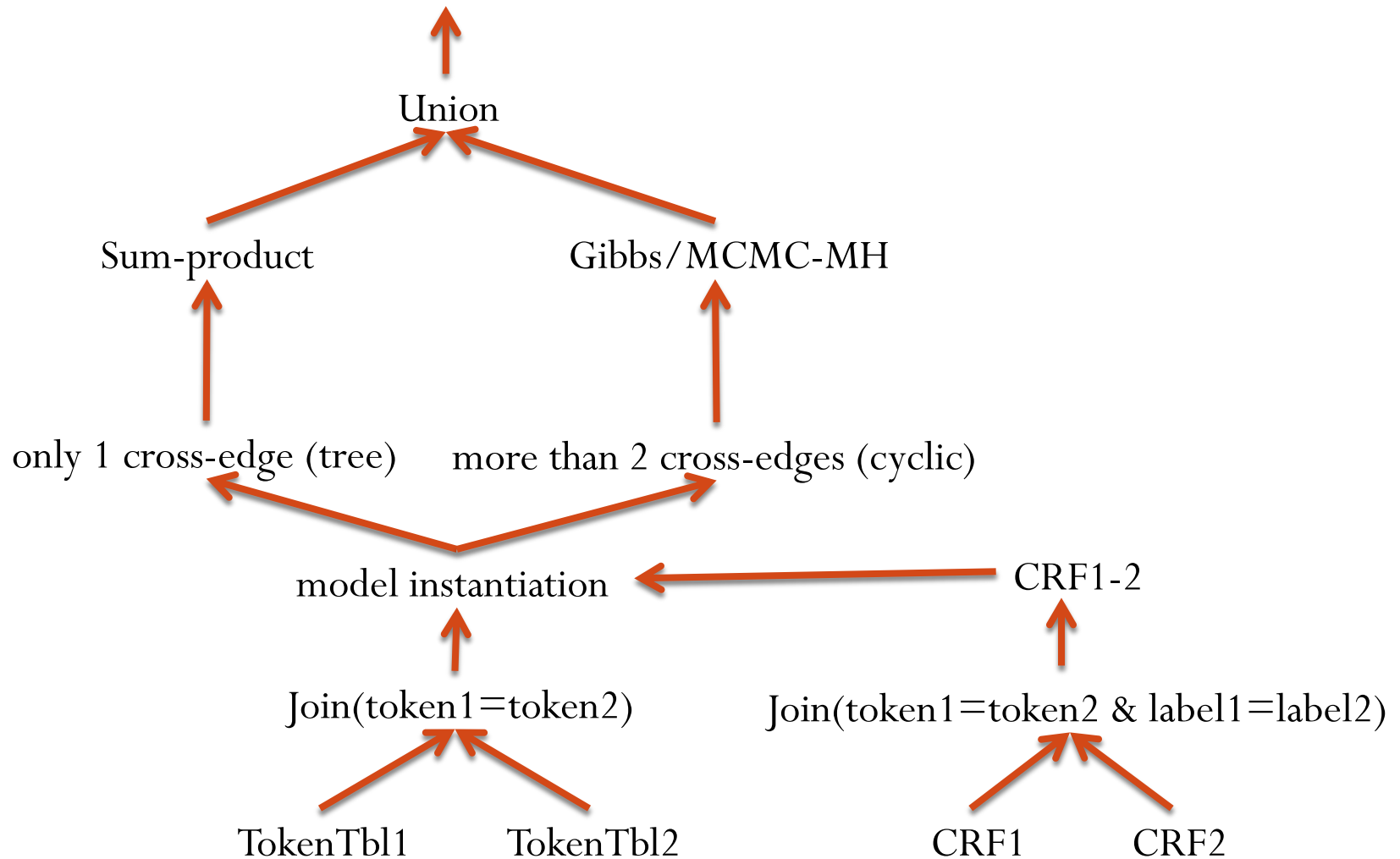
## Example 2: Probabilistic Join Queries

```
SELECT  Marginal(T1.docID,T2.docID,exist)
FROM    TokenTbl1 T1,TokenTbl2 T2
WHERE   T1.label = T2.label and T1.token = T2.token and
        T1.label = 'person';
```

# Inference Selection & Hybrid Inference



# Execution Strategy for Probabilistic Join



# Hybrid Inference Rewrite

- Apply Query over Model CRF  $\rightarrow$  CRF\*
- Instantiate Model CRF\* over Text  $\rightarrow$  {CRF\*[i]}
- Partition Data based on Model Structure
  - Cyclic, Linear-chain, Tree-shaped
- Choose Inference Algorithm
  - Based on the Query, the Model and the Text
  - Gibbs, Metropolis-Hastings, Viterbi, Sum-Product
- Execute Inference Algorithm

# Statistics from Real-Life Datasets and Potential Speed-up using Hybrid Inference

Data Corpora	Sentences with no Duplicate words	Potential speed-up for skip-chain model	Sentence-pairs share one word	Potential speed-up for join query
NYTimes	89.7%	×5	93.6%	×5
Twitter	90.0%	×5	74.4%	×3
DBLP	3.1%	small	1.0%	small

# Conclusion

- Part 1: Viterbi-based Probabilistic IE
  - BayesStoreIE framework
  - Deep Integration of Relational and Inference
  - Query-Driven Extraction
  - Probabilistic SPJ Queries
- Part 2: Sampling-based Probabilistic IE (Current Work)
  - Gibbs Sampling inference in DB
  - Hybrid Inference

# Thank you! ... Questions?

---

*BayesStore* Project Page:

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