Scalable Identity Resolution in Email Collections

… Using MapReduce

Tamer Elsayed, Doug Oard, and Jimmy Lin
University of Maryland

April 12th, 2010

Identity Resolution in Email

54 Sheila's !!

Sheila …

Weisman	Maynes	Jarnot
Pardo	Nacey	Kirby
Glover	Ferrarini	Knudsen
Rich	Dey	Boehringer
Jones	Macleod	Lutz
Breeden	Howard	Wollam
Huckaby	Darling	Jortner
Tweed	Watson	Neylon
Mcintyre	Perlick	Qhanger
Chadwick	Advani	Nagel
Birmingham	Hester	Graves
Kahanek	Kenner	Mclaughlin
Foraker	Lewis	Venville
Tasman	Walton	Rappazzo
Fisher	Whitman	Miller
Petitt	Berggren	Swatek
Dombo	Osowski	Hollis
Robbins	Kelly	Chang

Da …ST 2000
Fr …enron.com>
To …adams@enron.com>
Su …**Call** has be rescheduled

Di…icipate? I…
ca…

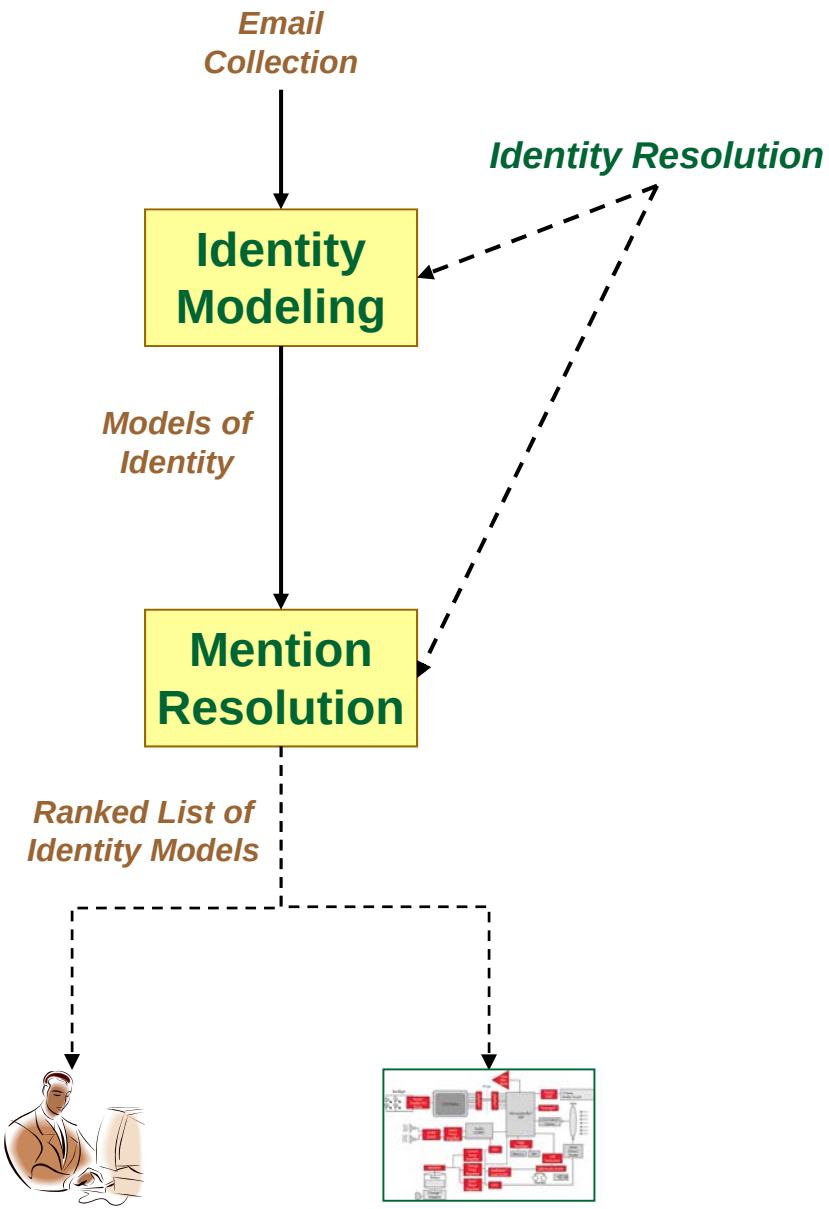
→ ***Rank Candidates***

Why is That Needed?

- ***Users*** **unfamiliar with discussions**
  - **Lawyers**
  - **Historians**
  - **Police investigators**

- ***Downstream process***
  - **Expanding ambiguous names at indexing time**
  - **Expert finding**
  - **Social network analysis**

Structure of the Problem



Generative Model

1. Choose "person" c to mention

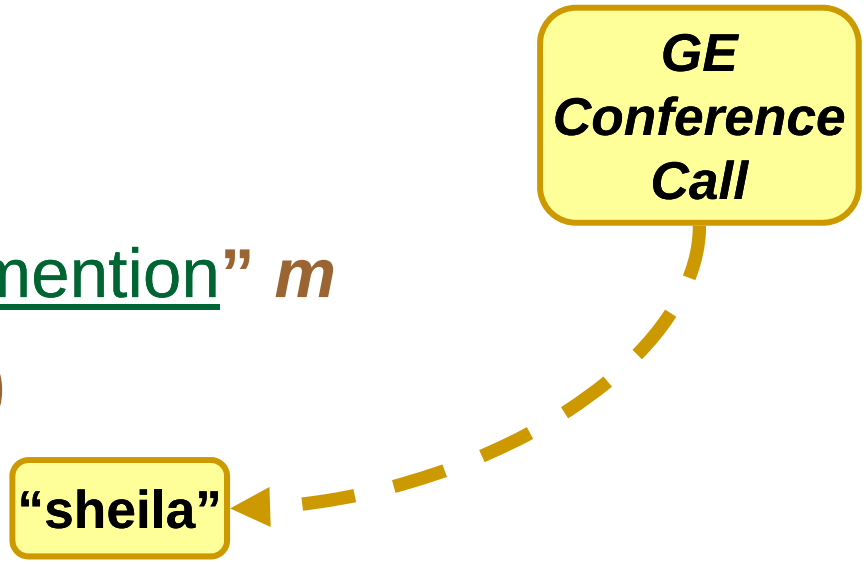
   $p(c)$

2. Choose appropriate "context" X to mention c

   $p(X \mid c)$

3. Choose a "mention" m

   $p(m \mid X, c)$



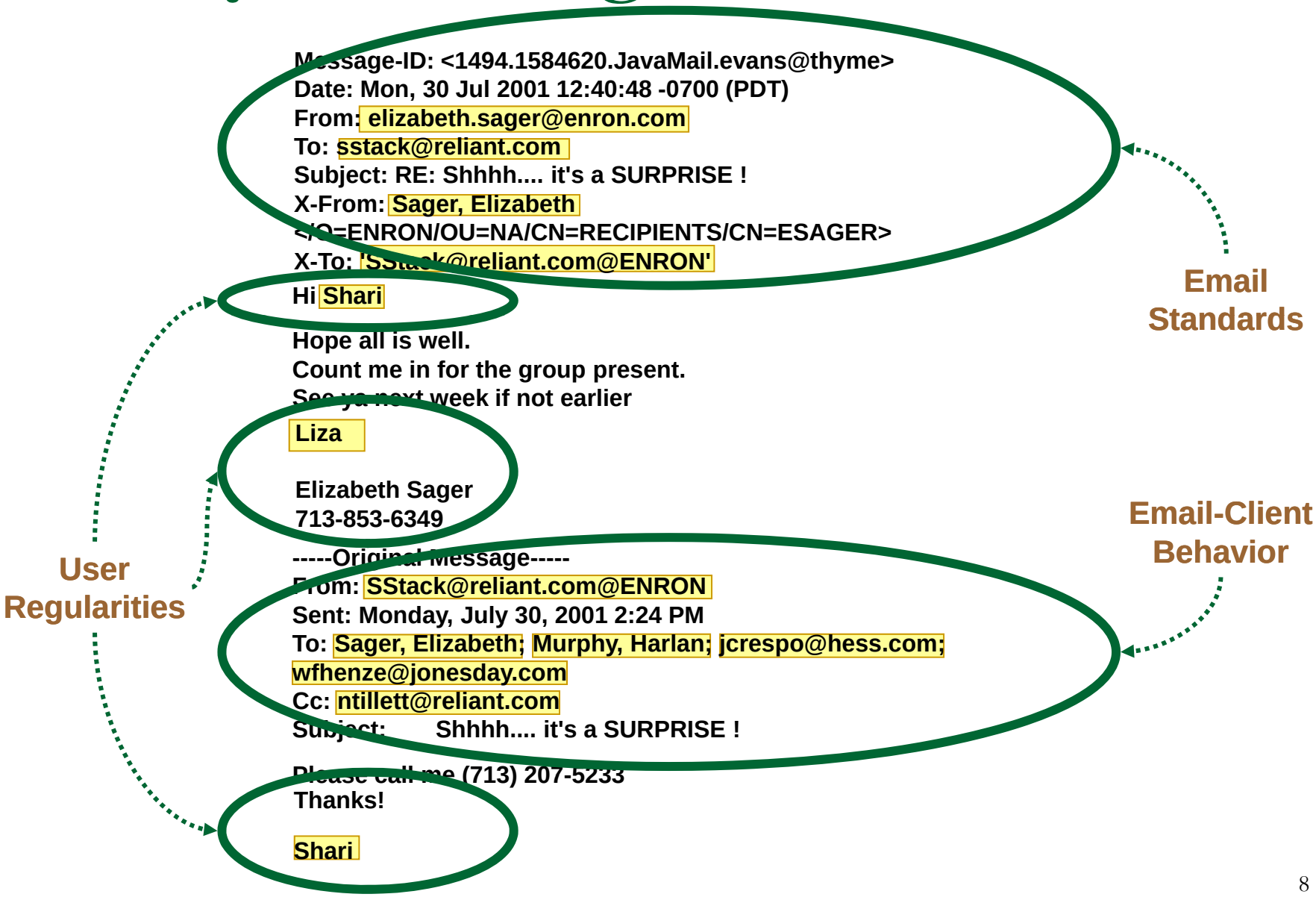
Outline

- Introduction and Approach Overview
- Identity Models and Mention Resolution
- Scalable MapReduce Solution
  - Pairwise Document Similarity
  - Mention Resolution
- Evaluation
- Conclusion

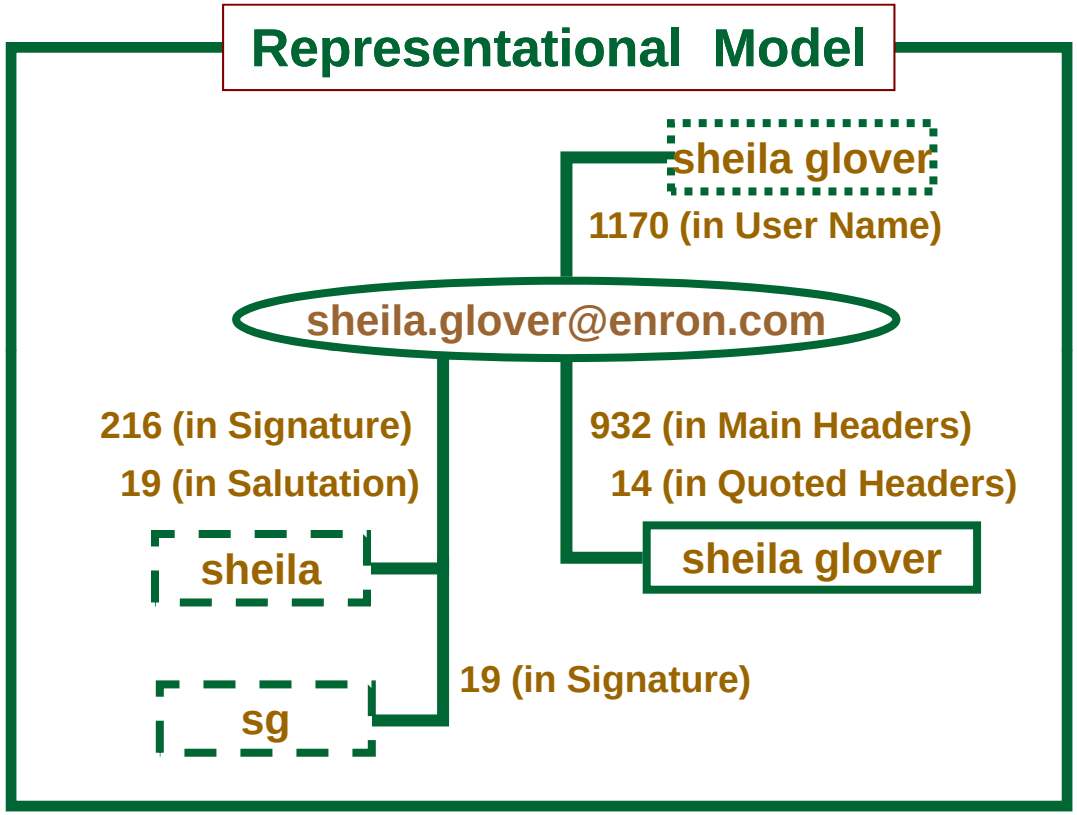
Outline

- Introduction and Approach Overview
- *Identity Models and Mention Resolution* 
- **Scalable MapReduce Solution**
  - **Pairwise Document Similarity**
  - **Mention Resolution**
- **Evaluation**
- **Conclusion**

"Easy/Unambiguous" References



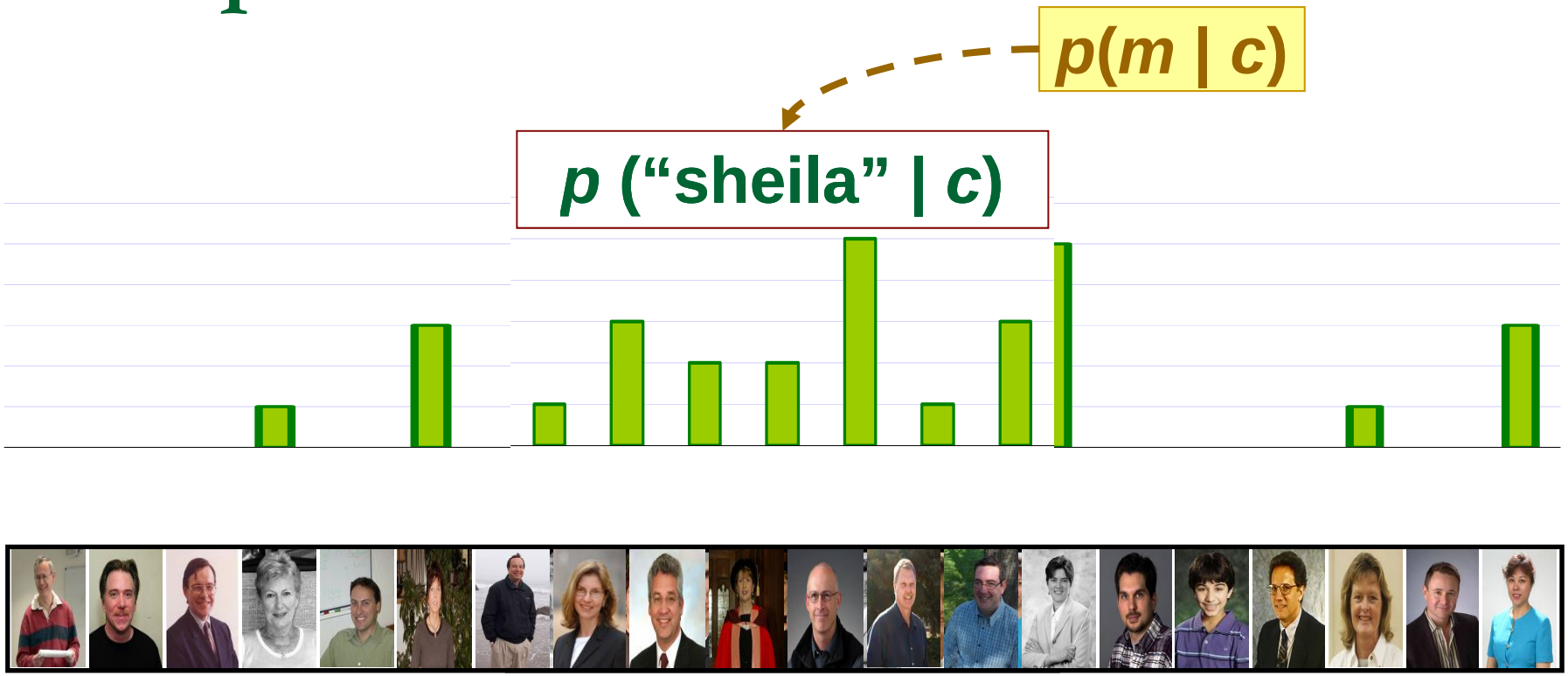
Representational Model of Identity



77,240 models

96.7% overall accuracy

Computational Model



***Goal*: estimate p(c | m "in context")**

Topical Context

***Date*: Wed Dec 20 08:57:00 EST 2000**
***From*: Kay Mann <kay.mann@enron.com>**
***To*: Suzanne Adams <suzanne.adams@enron.com>**
***Subject*: Re: GE Conference Call has be rescheduled**

Did Sheila want Scott to participate? Looks like the call will be too late for him.

Date: Fri Dec 15 05:33:00 EST 2000
From: david.oxley@enron.com
To: vince j kaminski <vince.kaminski@enron.com>
Cc: sheila walton ***sheila.walton@enron.com***
Subject: Re: Grant Masson

Great news. Lets get this moving along. Sheila, can you work out GE letter?

Vince, I am in London Monday/Tuesday, back Weds late. I'll ask Sheila to fix this for you and if you need me call me on my cell phone.

Social Context

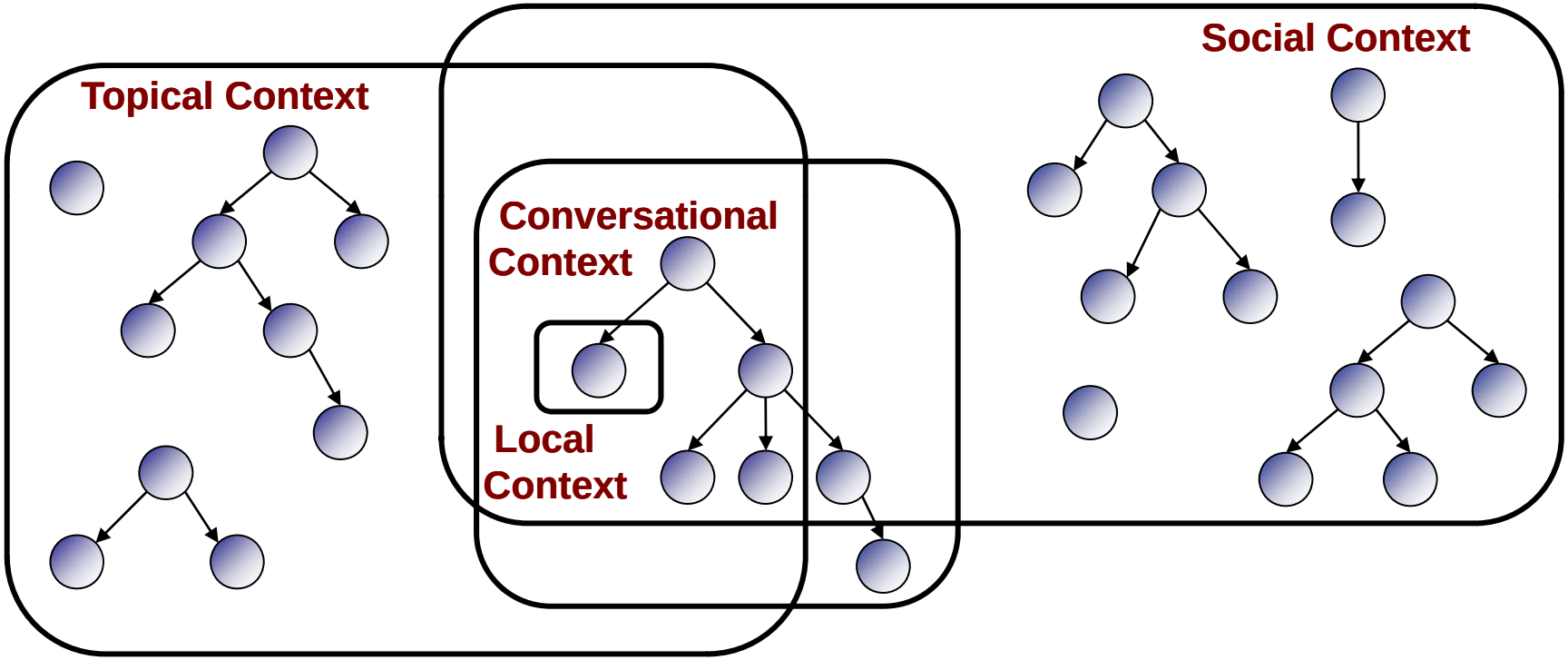
***Date*: Wed Dec 20 08:57:00 EST 2000**
***From*: Kay Mann <kay.mann@enron.com>**
***To*: Suzanne Adams <suzanne.adams@enron.com>**
***Subject*: Re: GE Conference Call has be rescheduled**

Did Sheila want Scott to participate? Looks like the call will be too late for him.

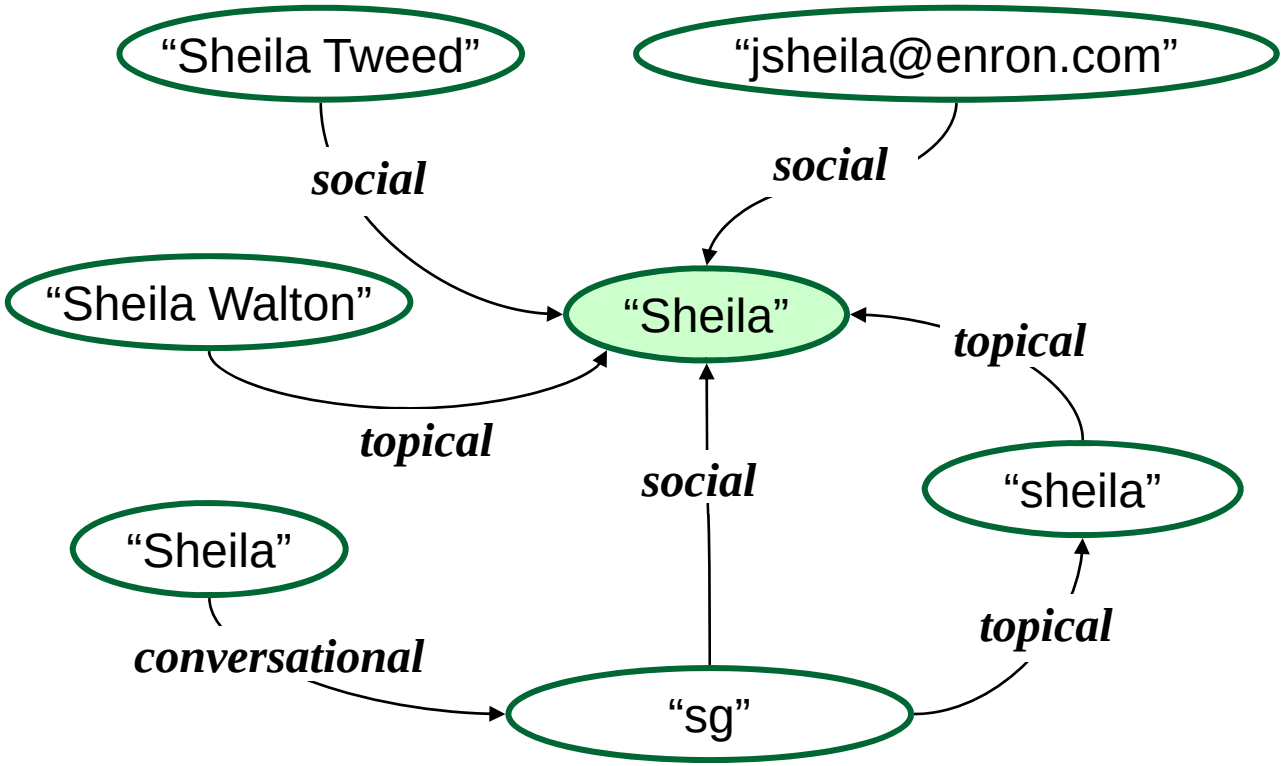
Date: Tue, 19 Dec 2000 07:07:00 -0800 (PST)
From: rebecca.walker@enron.com
To: **kay.mann@enron.com**
Subject: ESA Option Execution

Kay
Can you initial the ESA assignment and assumption agreement or should I ask ***Sheila Tweed*** to do it? I believe she is currently en route from Portland.
Thanks,
Rebecca

Contextual Space (Emails)



Contextual Space (Mentions)



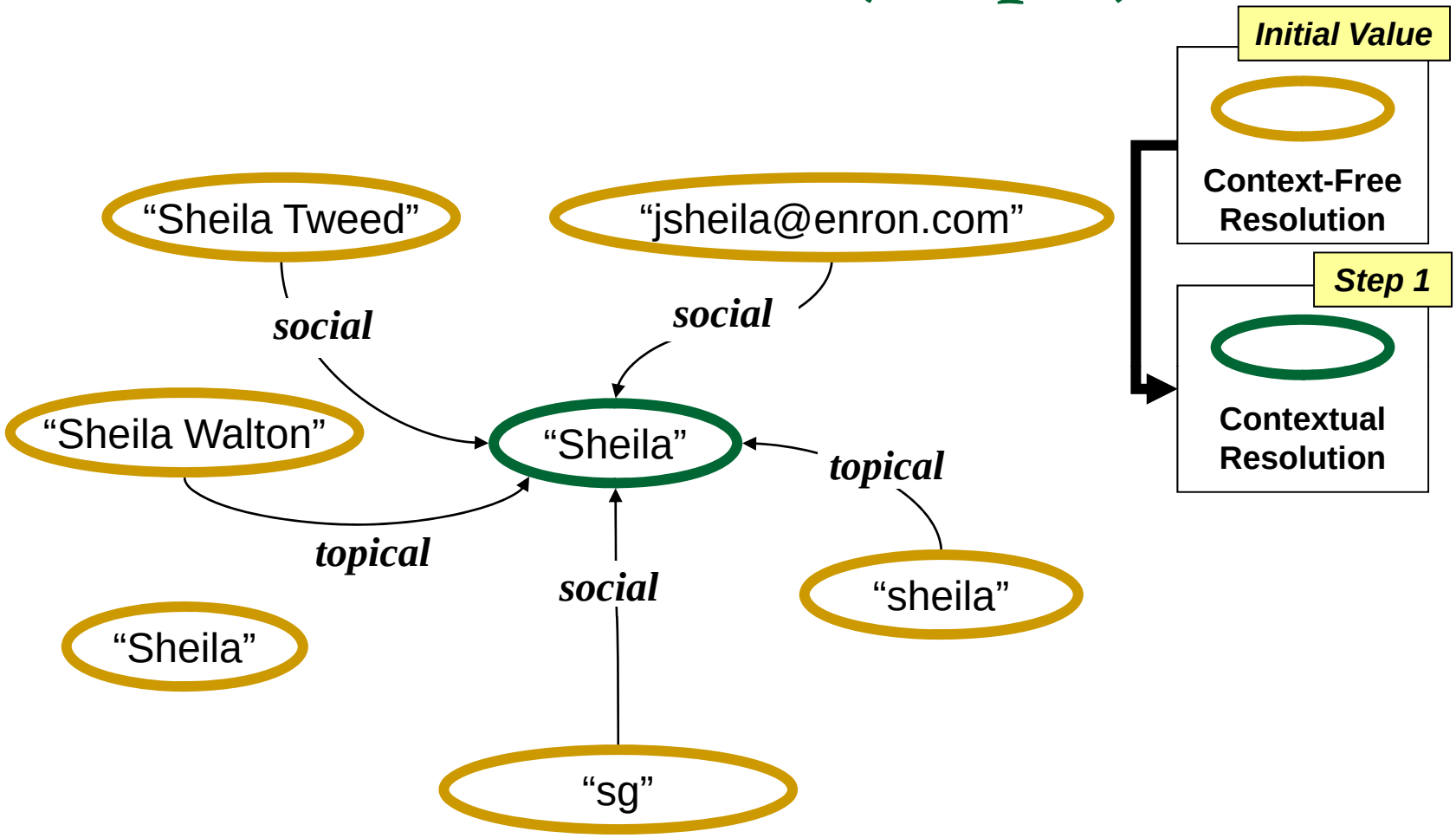
$$p(c \mid m, X(m)) = f(p(c \mid m')) \quad \forall m' \in X(m)$$

?

Context-Free Resolution (Step 0)

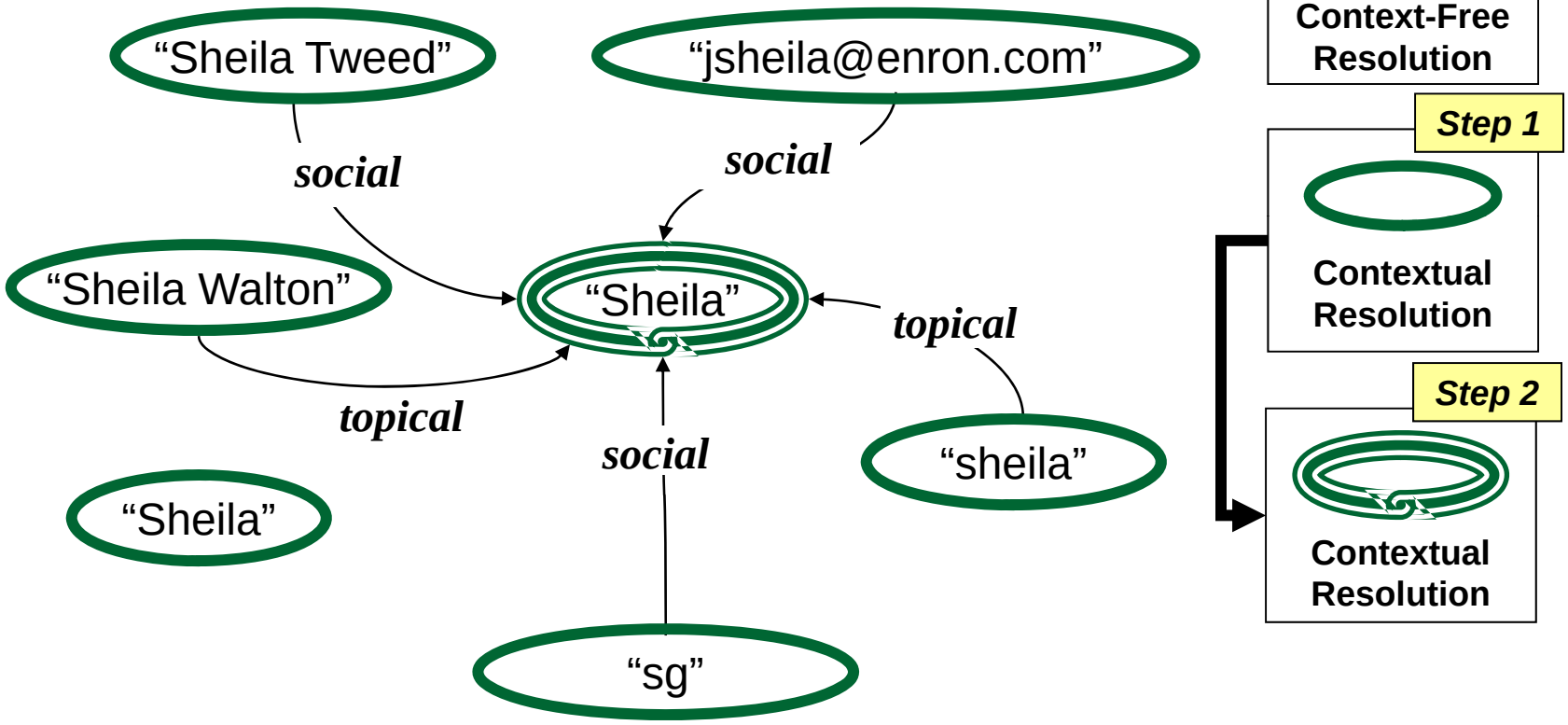
$$p(c \mid m, X(m)) \approx p(c \mid m) = \frac{p(m \mid c)p(c)}{p(m)}$$

Contextual Resolution (Step 1)



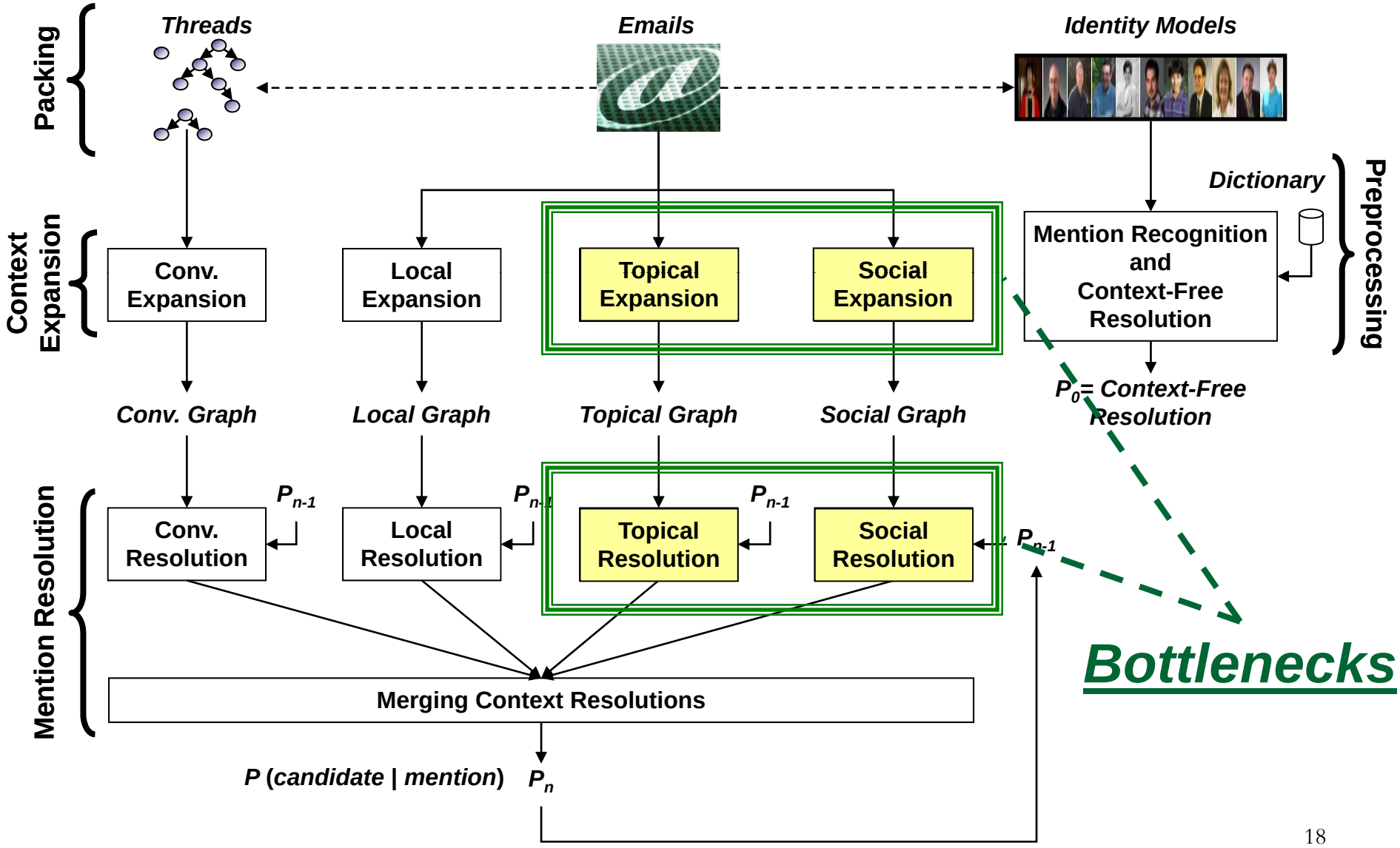
$$p(c \mid m, X(m)) = \frac{p(c, m, X(m))}{p(m, X(m))}$$

Contextual Resolution (Step 2)



$$p(c \mid m, X(m)) = \frac{p(c, m, X(m))}{p(m, X(m))}$$

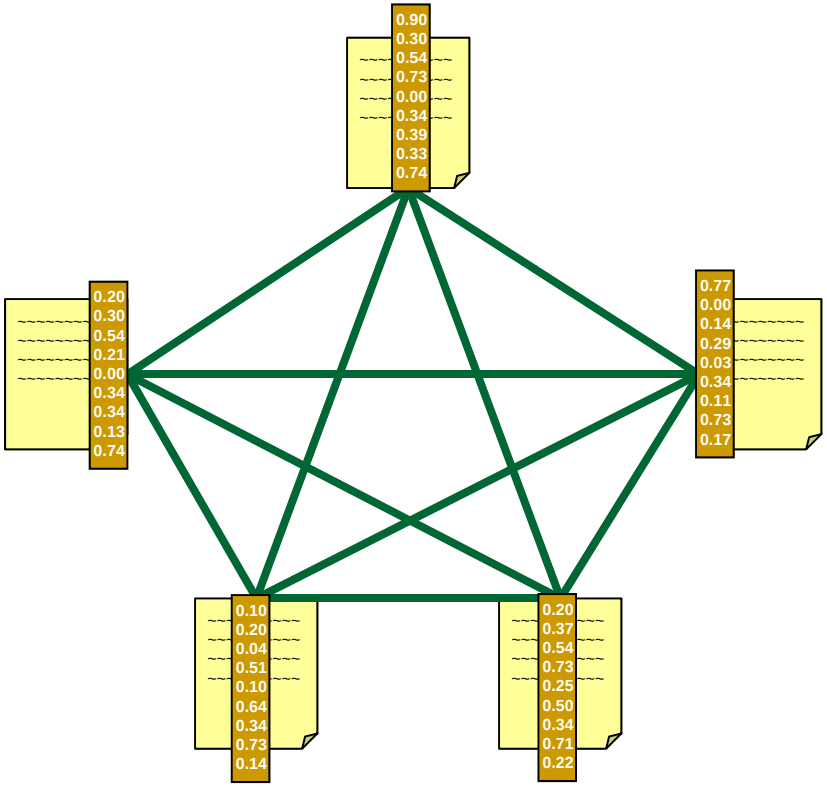
System Overview



Outline

- Introduction and Approach Overview
- Identity Models and Mention Resolution
- *Scalable MapReduce Solution* 
  - ***Pairwise Document Similarity***
  - ***Mention Resolution***
- **Evaluation**
- **Conclusion**

Context Expansion (Abstract): Computing Pairwise Similarity

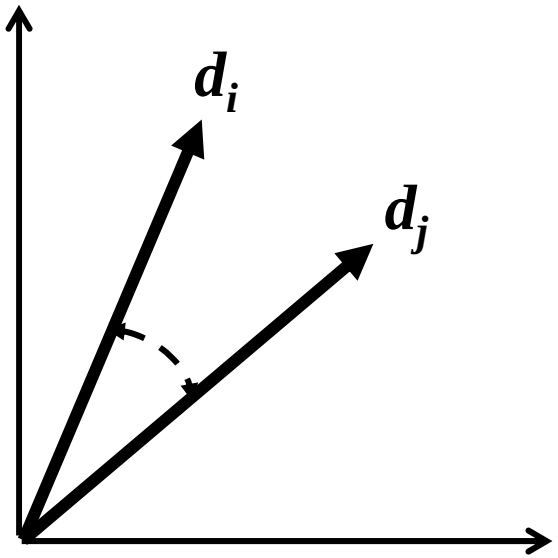


- *Applications*:
  - **Clustering**
  - **Coreference resolution**
  - **"*more-like-that*" queries**

Similarity of Documents

$$sim(d_i, d_j) = \sum_{t \in V} w_{t,d_i} w_{t,d_j}$$

- Simple inner product
- Cosine similarity
- Term weights
  - Standard problem in IR
  - tf-idf, BM25, etc.



Trivial Solution

$$sim(d_i, d_j) = \sum_{t \in V} w_{t,d_i} w_{t,d_j}$$

- load each vector o(*N*) times
- load each term o(df_t^2) times

Goal

scalable and efficient solution for large collections

Better Solution

Each term contributes only if appears in $d_i \cap d_j$

$$sim(d_i, d_j) = \sum_{t \in d_i \cap d_j} w_{t,d_i} w_{t,d_j}$$

$$sim(d_i, d_j) = \sum_{t \in d_i \cap d_j} term\_contrib(t, d_i, d_j)$$

- **Load weights for each term once**
- **Each term contributes o(df_t^2) partial scores**
- **Allows efficiency tricks**

Decomposition ➔ MapReduce

Each term contributes only if appears in $d_i \cap d_j$

$$sim(d_i, d_j) = \sum_{t \in d_i \cap d_j} w_{t,d_i} w_{t,d_j}$$

reduce

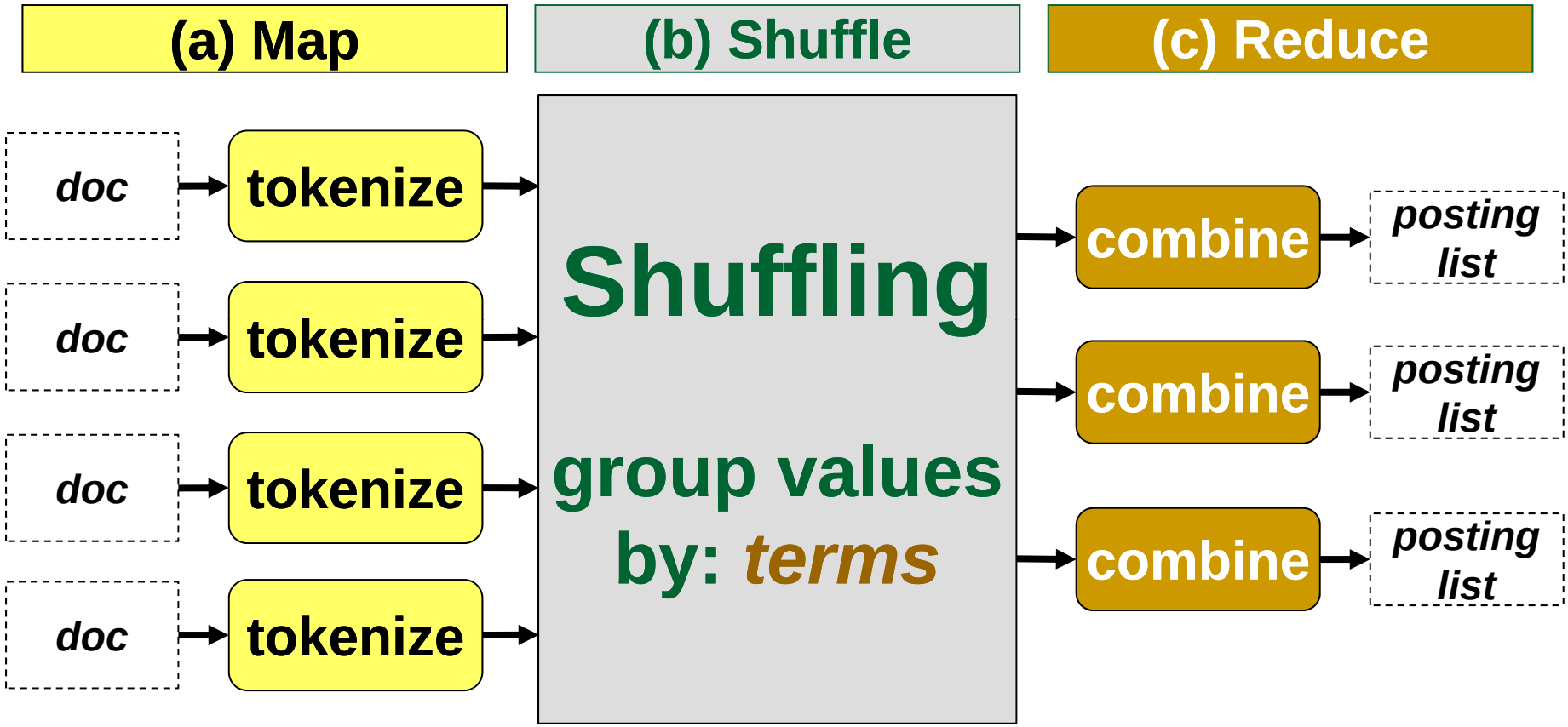
$$sim(d_i, d_j) = \sum_{t \in d_i \cap d_j} term\_contrib(t, d_i, d_j)$$

index

map

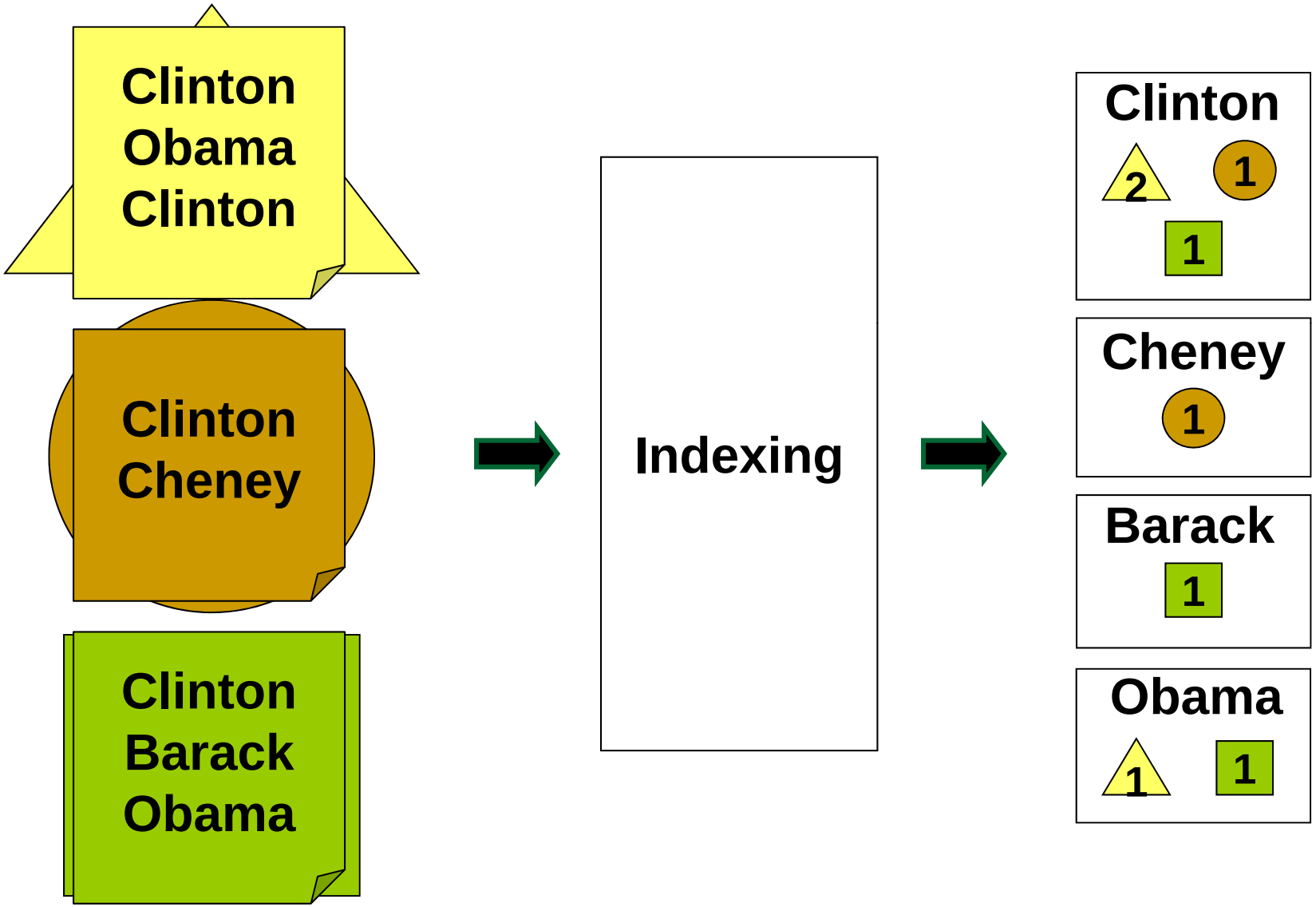
- Load weights for each term once
- Each term contributes o(df_t^2) partial scores

(a) Standard Inverted Indexing

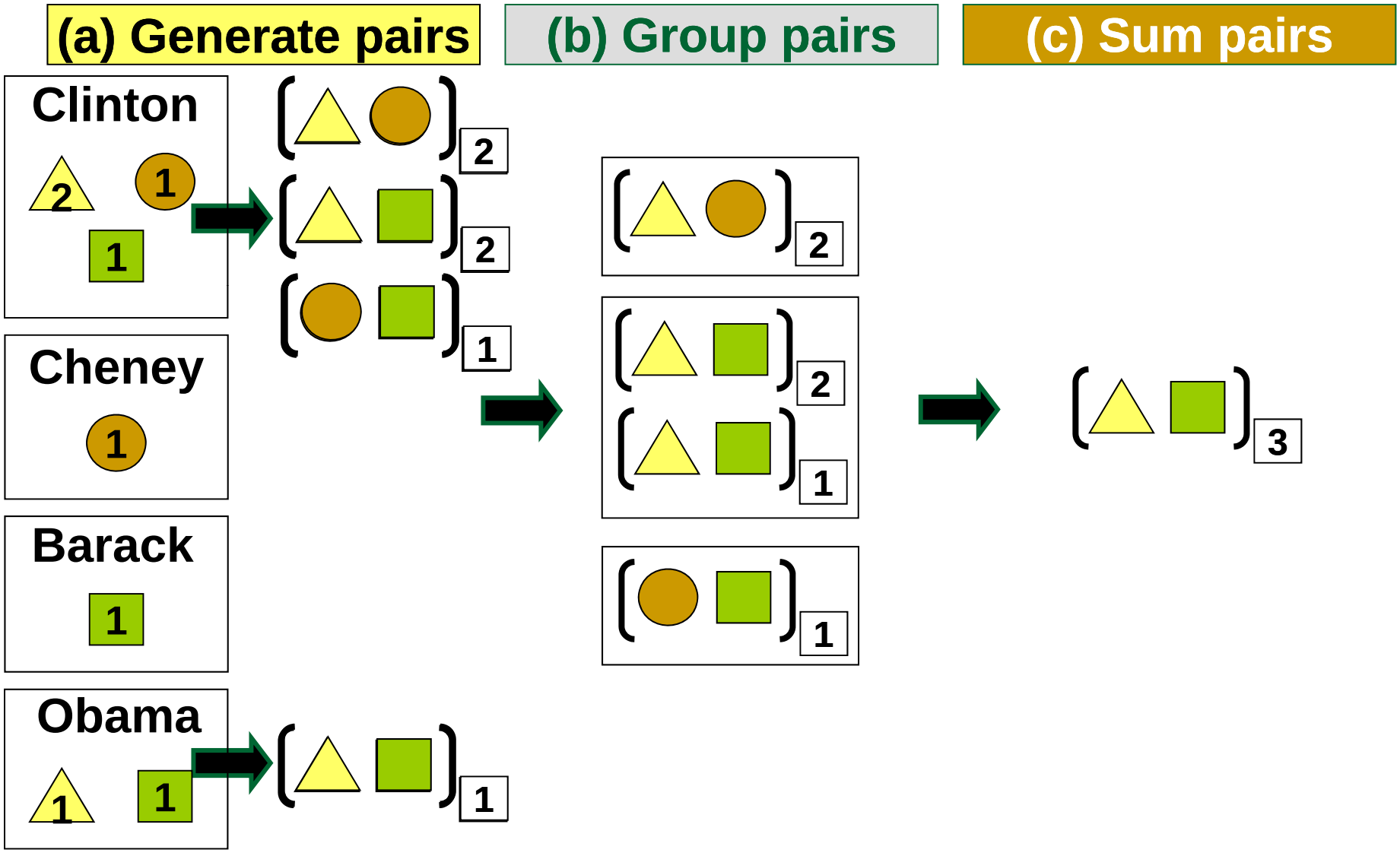


$$sim(d_i, d_j) = \sum_{t \in d_i \cap d_j} term\_contrib(t, d_i, d_j)$$

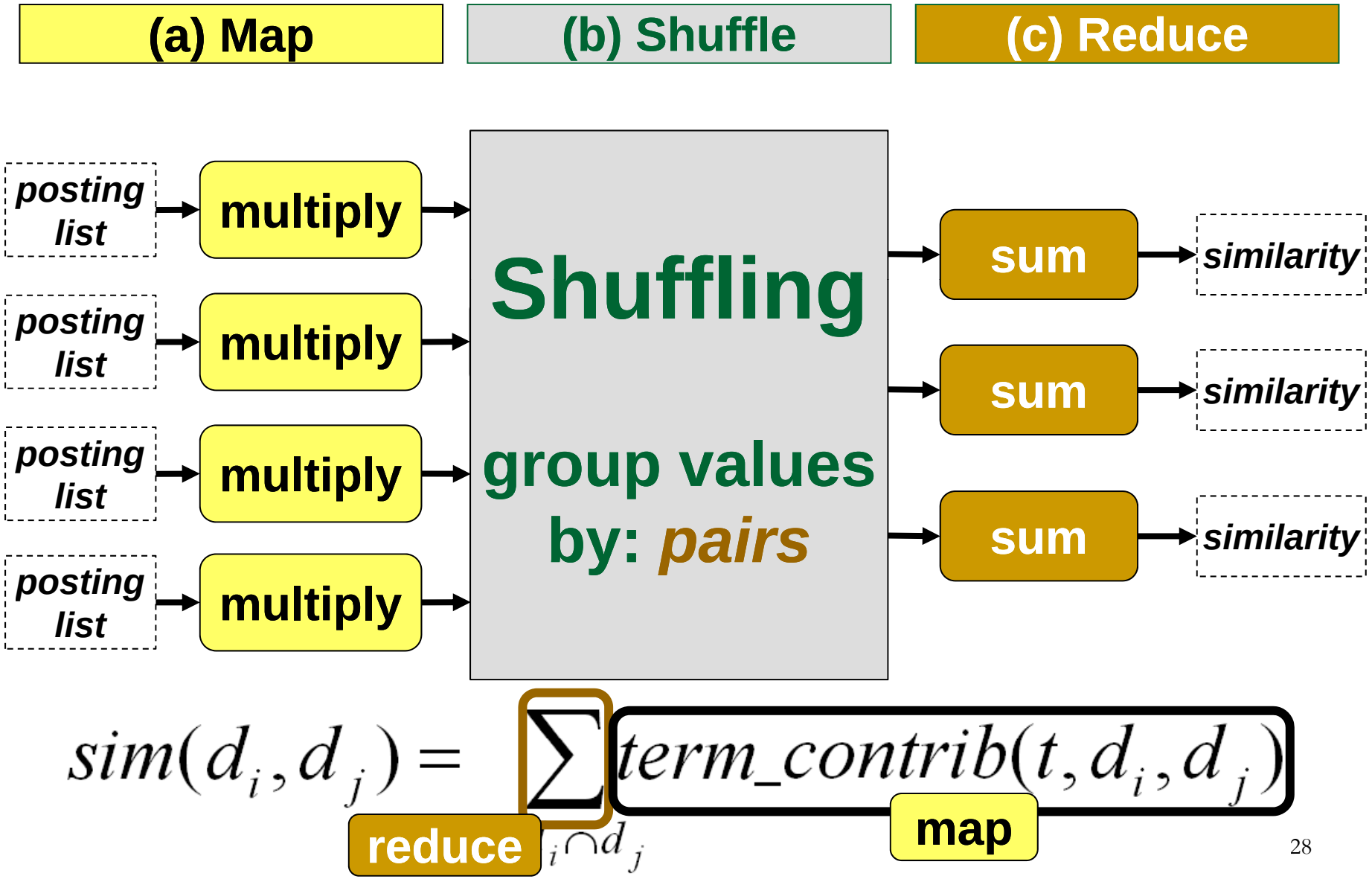
Indexing (3-doc toy collection)



(b) Pairwise Similarity (Example)



(b) Pairwise Similarity

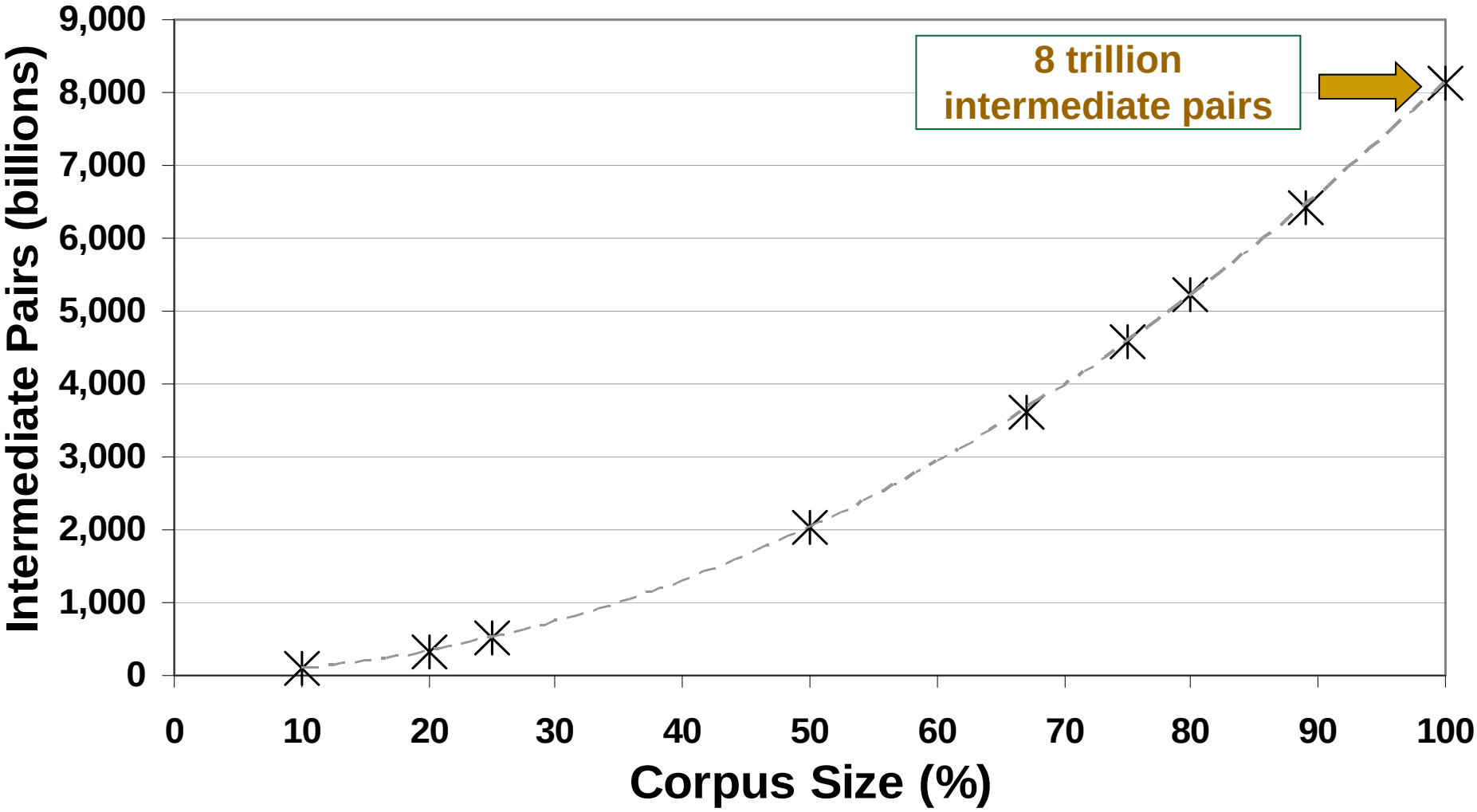


Experimental Setup

-  0.16.0
  - Open source MapReduce implementation
- Cluster of 19 machines
  - Each w/ two processors (single core)
- Aquaint-2 collection
  - 906K documents
- Okapi BM25
- Subsets of collection

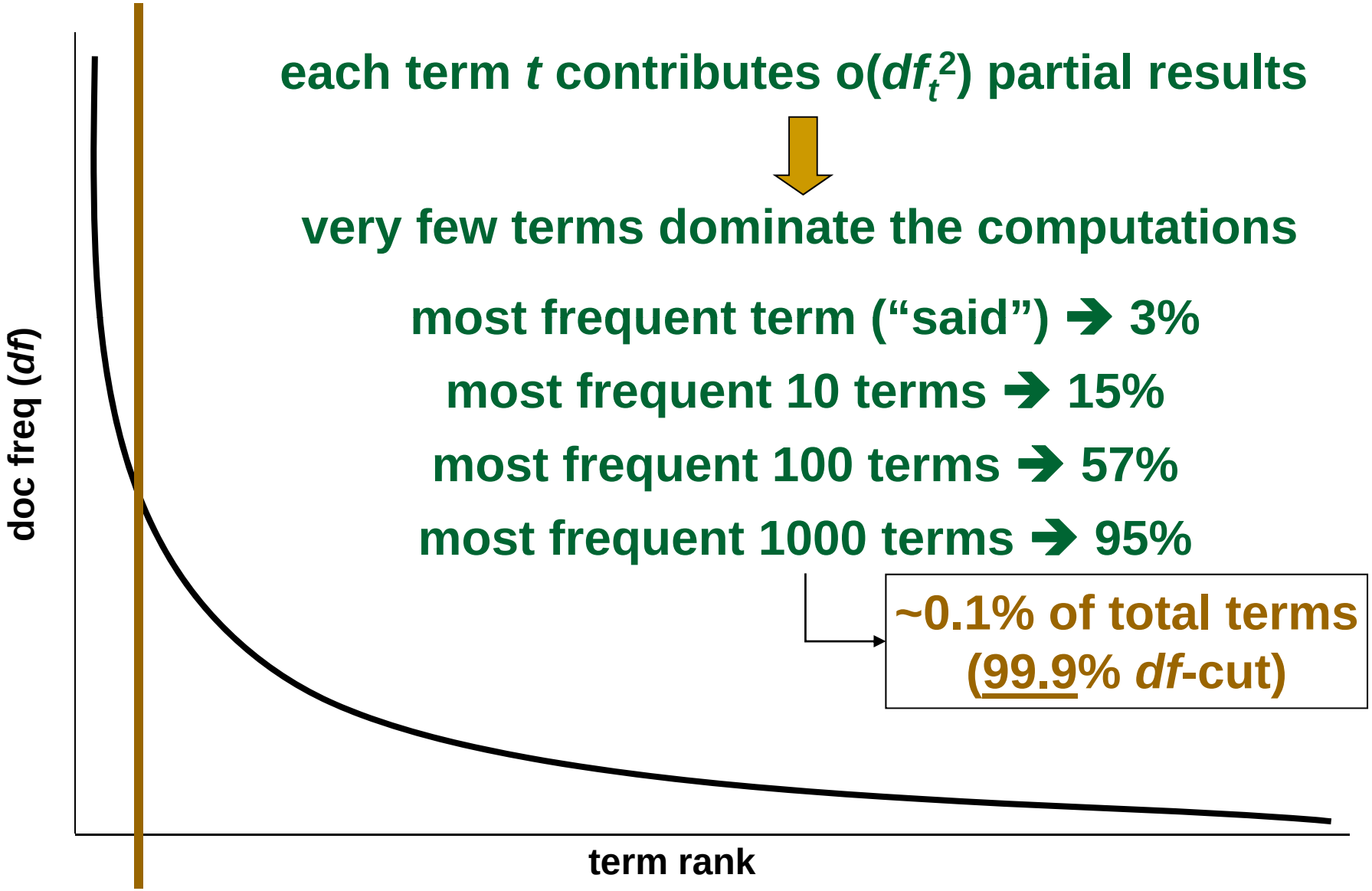
Efficiency (disk space)

Aquaint-2 Collection, ~ 906k docs



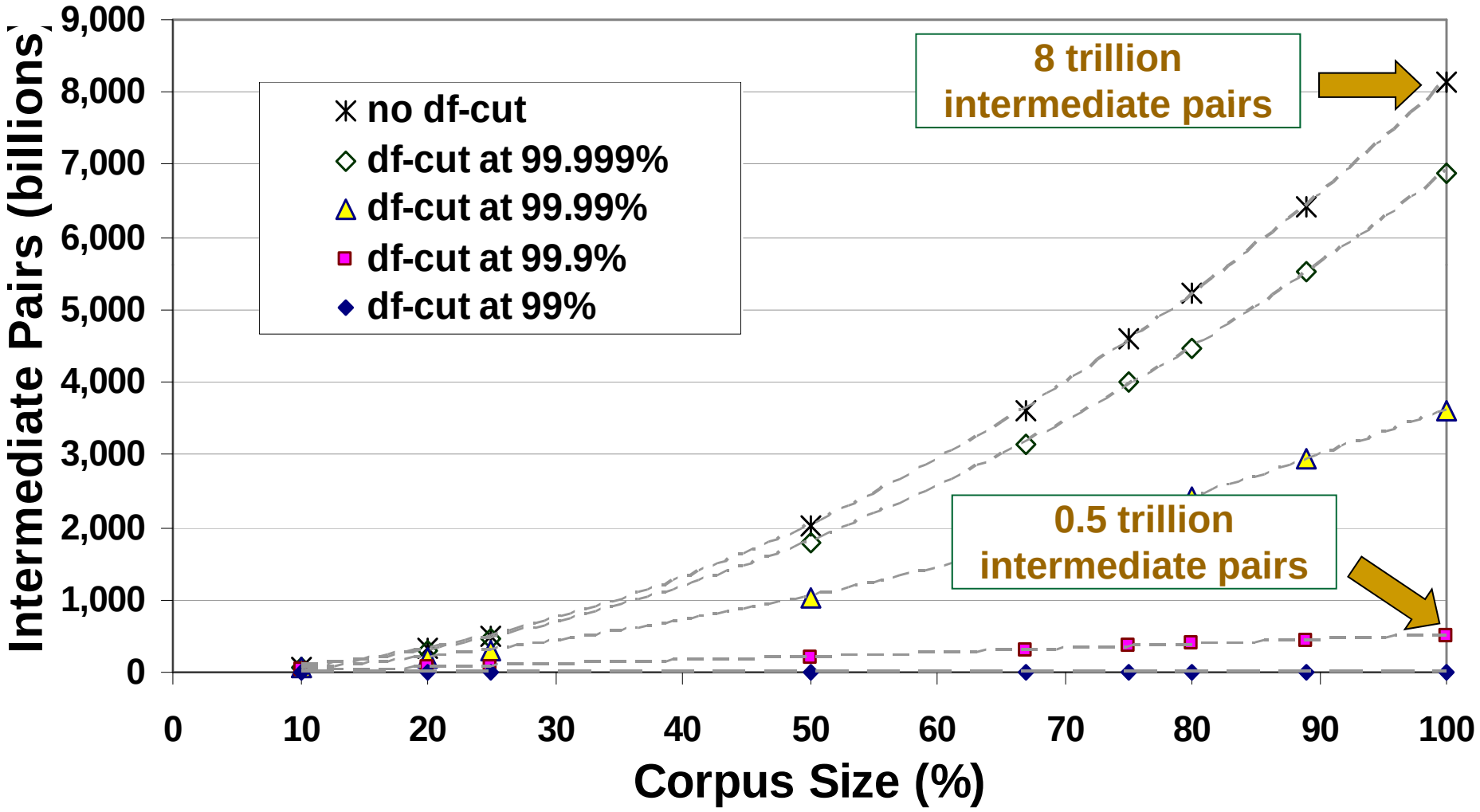
Hadoop, 19 PCs, each: 2 single-core processors, 4GB memory, 100GB disk

Terms: Zipfian Distribution



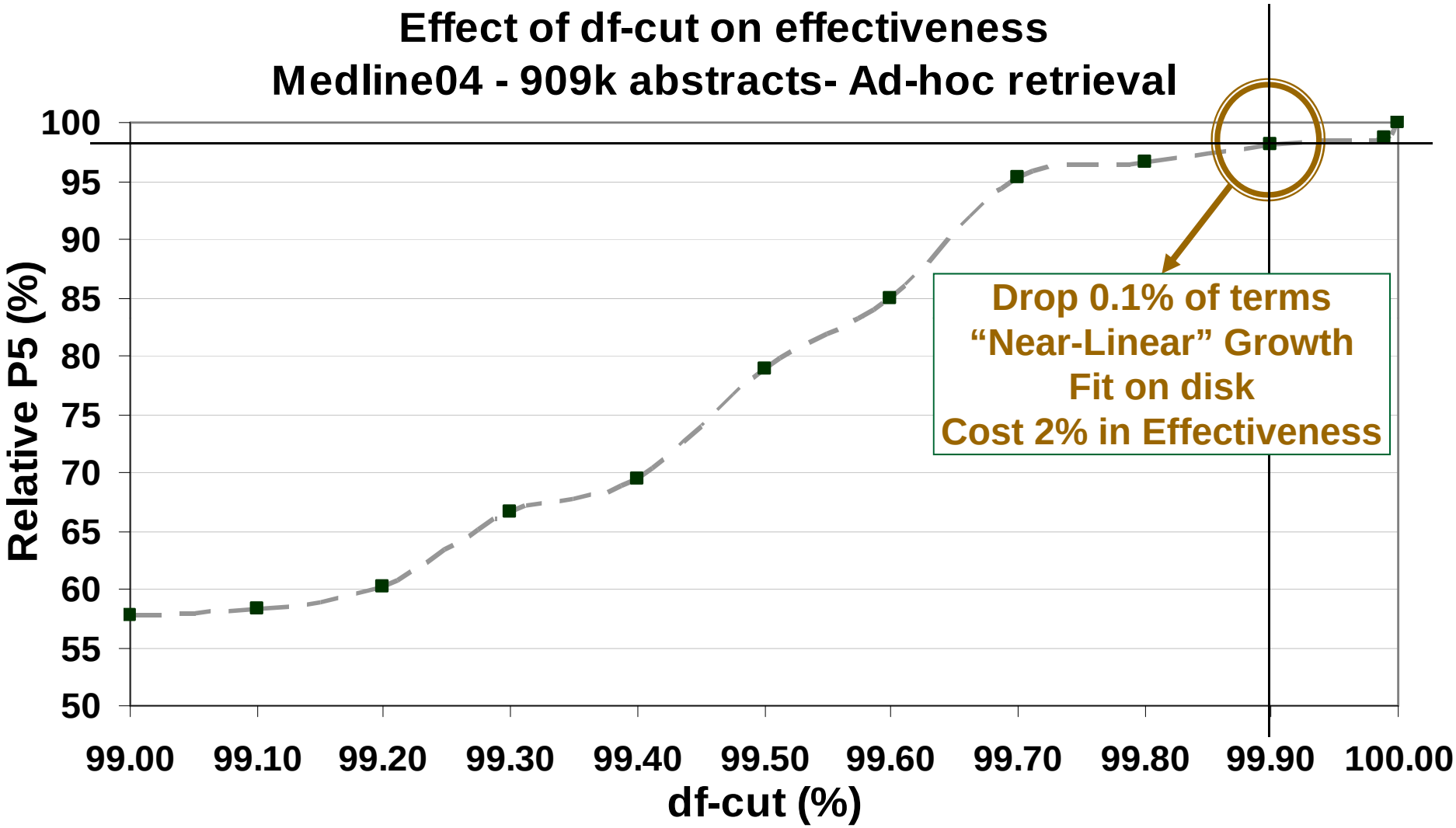
Efficiency (disk space)

Aquaint-2 Collection, ~ 906k doc



Hadoop, 19 PCs, each w/: 2 single-core processors, 4GB memory, 100GB disk

Effectiveness (recent work)



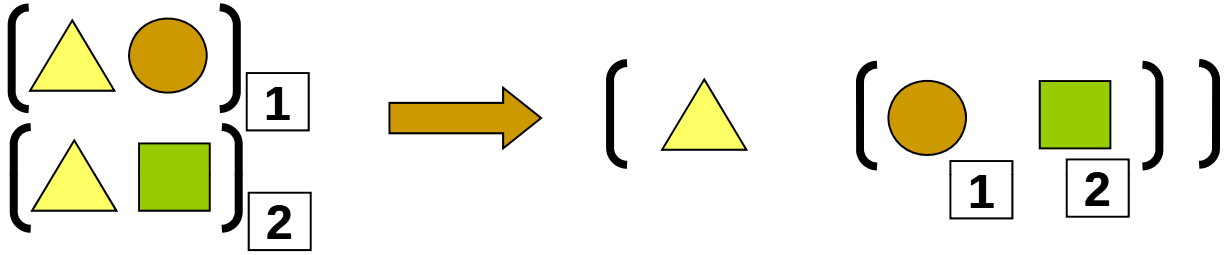
Hadoop, 19 PCs, each w/: 2 single-core processors, 4GB memory, 100GB disk

Other Approximation Techniques ?

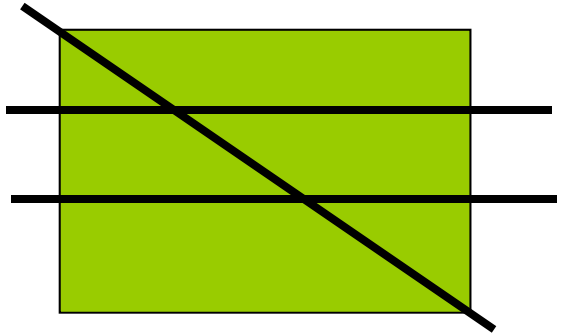
- Absolute *df*
  - Consider only terms that appear in at least *n (or %)* documents
- *tf*-cut
  - Consider only documents (in posting list) with tf > T ; T=1 or 2
  - OR: Consider only the top N documents based on tf for each term
- Similarity Threshold
  - Consider only partial scores > Sim_T
- Ranked List
  - Keep only the most similar N documents
    - In the reduce phase
  - Good for ad-hoc retrieval and “more-like this” queries

Space-Saving Tricks

- Stripes
  - Stripes instead of pairs & Group by doc-id not pairs

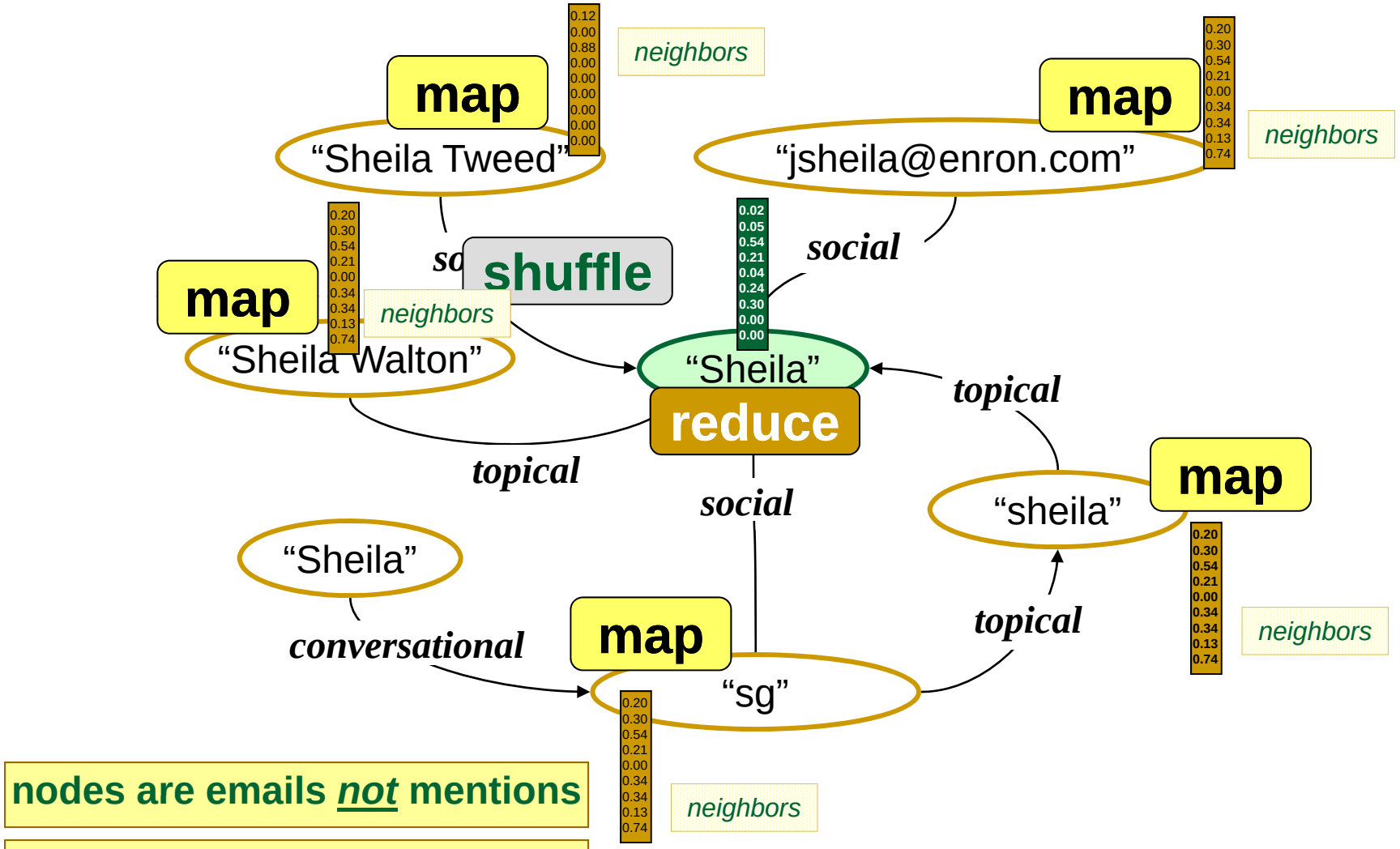


- Blocking
  - No need to generate the whole matrix at once
  - Generate different blocks of the matrix at different steps ➔ limit the max space required for intermediate results



Similarity Matrix

Mention Resolution



nodes are emails *not* mentions

repeat for multiple iterations

Efficiency

- hadoop 0.17.2
  - Open source MapReduce implementation
- 200 processing nodes

Recognized References

from Main body	999,291
from Subject	51,386
from Main Header	1,642,923
from Quoted Body	442,099
from Quoted Header	522,716
Email-addresses	1,746,636
Single-token Names	1,331,375
Multi-token Names	580,407

Time Spent (minutes)

Packing	48	Social: Indexing	1.5	Topical: Indexing	1.5
Preprocessing	5	Social: Pairwise Sim.	5	Topical: Pairwise Sim.	5-13
Local: Total	9	Social: Resolution	13	Topical: Resolution	17-35
Conv.: Total	10	Social: Total	35	Topical: Total	45-75
Merging Scores	10				

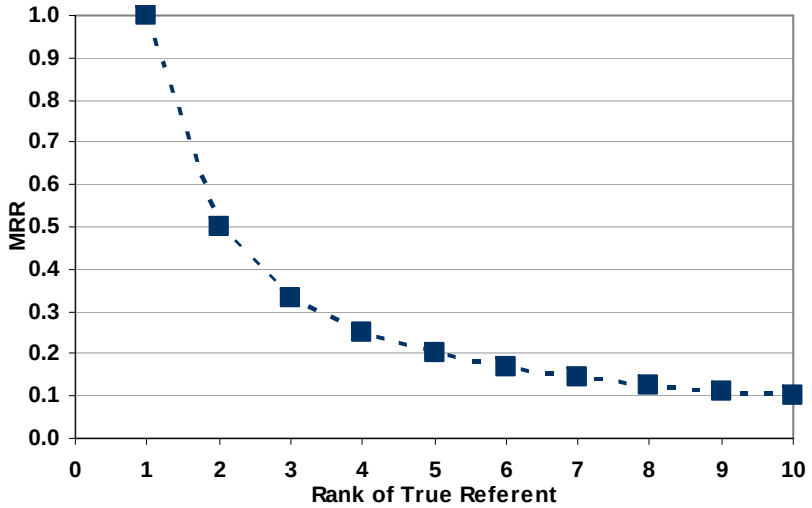
End-to-end runs: ~2-3 hours

Outline

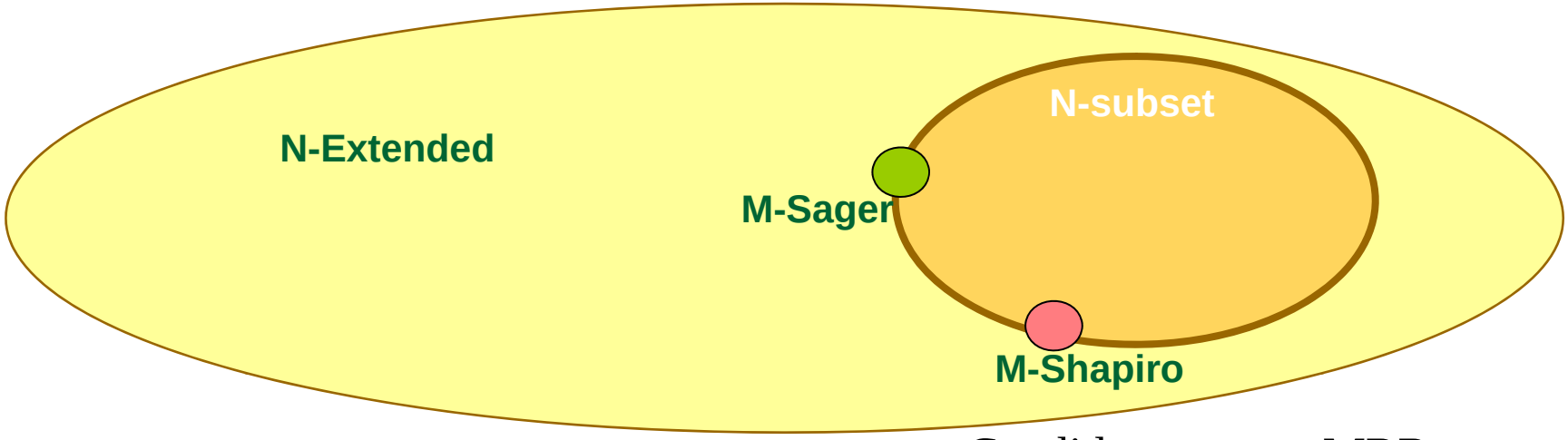
- Introduction and Approach Overview
- Identity Models and Mention Resolution
- Scalable MapReduce Solution
  - Pairwise Document Similarity
  - Mention Resolution
- *Evaluation* 
- **Conclusion**

Experimental Evaluation

- Repeatable and affordable
- Training and testing split
- Test Collection
  - Documents → emails
  - Queries → mentions in specific emails
  - Answers → true referents of those mentions (by humans)
- Evaluation Measure: Mean Reciprocal Rank (MRR)

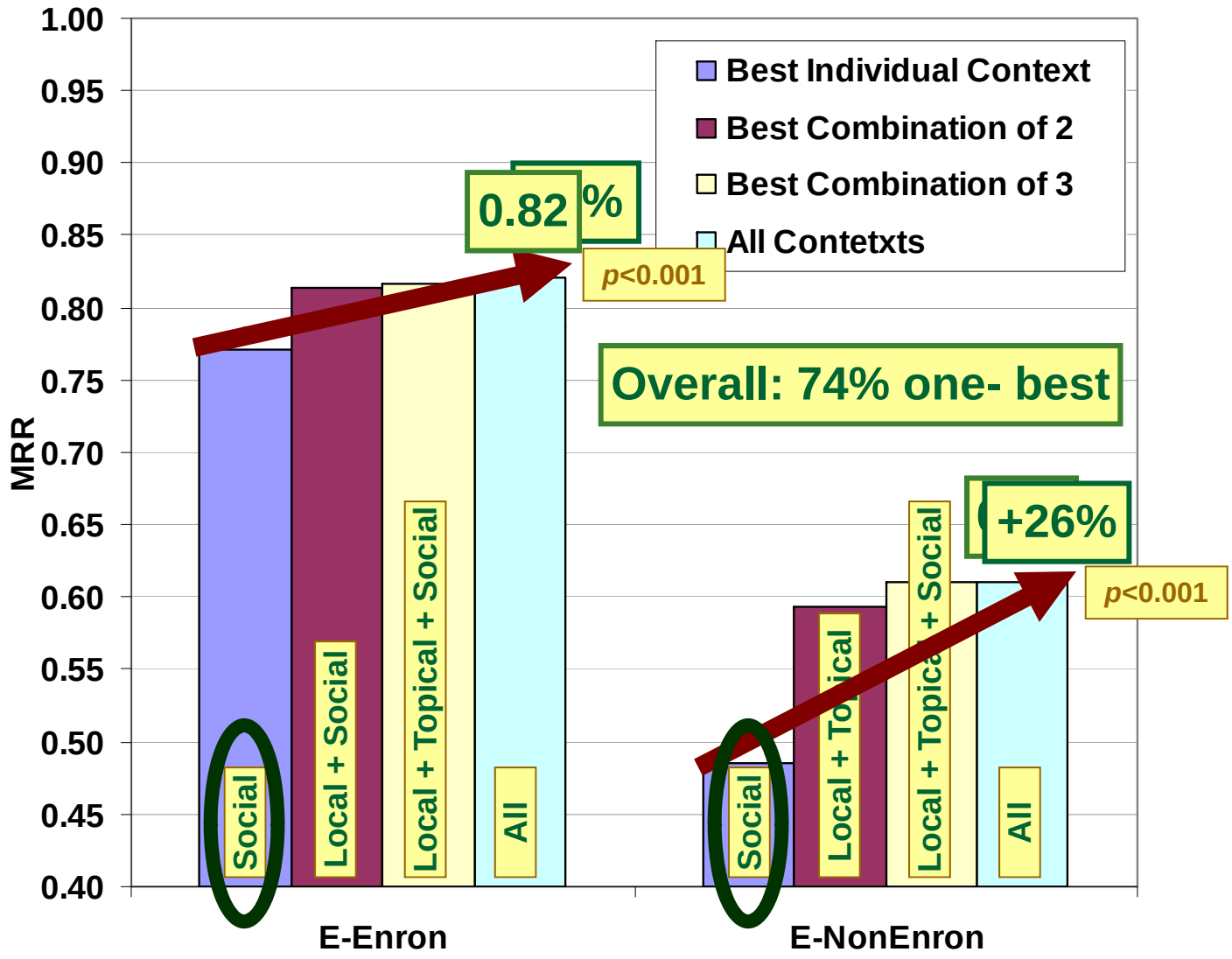


New Test Collection



Collection	Emails	Queries	Identities	Candidates Med	Candidates Range	MRR Mine	MRR Lit.
M-Sager	1,628	51	627	2	1-10	**0.905**	0.889
M-Shapiro	974	49	855	4	1-16	**0.894**	0.879
N-Subset	54,018	78	27,340	91	1-441	0.934	-
N-Extended	248,451	78	123,783	338	3-1,512	0.933	-
E-All	248,451	470	123,783	116	0-1,512	0.785	-
E-Enron	248,451	390	123,783	121	0-1,512	0.820	-
E-NonEnron	248,451	90	123,783	66	1-1,512	0.611	-

Testing on New Collection



Outline

- Introduction and Approach Overview
- Identity Models and Mention Resolution
- Scalable MapReduce Solution
  - Pairwise Document Similarity
  - Mention Resolution
- Evaluation
- *Conclusion* 

Conclusion

- **Simple and efficient MapReduce solution**
  - **applied to both topical and social expansion in "Identity Resolution in Email"**
  - **different tricks for approximation**
- **Shuffling is critical**
  - ***df*-cut controls efficiency vs. effectiveness tradeoff**
  - **99.9% *df*-cut achieves 98% relative accuracy**
- **Effective resolution algorithm**
  - **Compared favorably to previous work**
  - **Highlights importance of social context**
  - **Overall: 74% one-best**

Thank You!

Question?