Efficient Information Extraction over Evolving Text Data

Fei Chen[1], AnHai Doan[1], Jun Yang[2], Raghu Ramakrishnan[3]

[1]University of Wisconsin-Madison

[2]Duke University

[3]Yahoo! Research

Information Extraction (IE)

Group Meeting Schedule

Jun 21: We'll discuss CIM and IR in room CS 310 at 4pm.

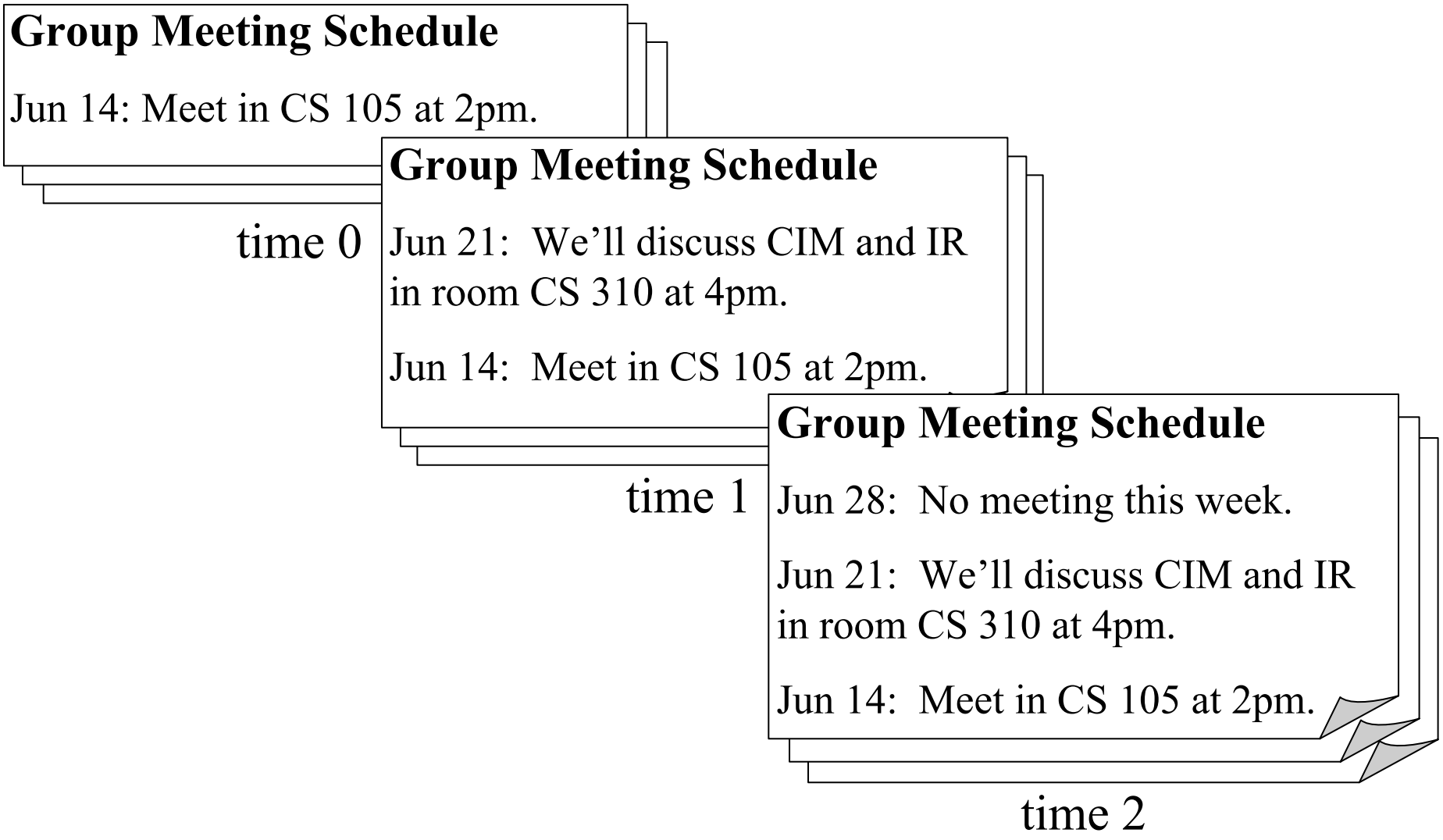
Jun 14: Meet in CS 105 at 2pm.

Meetings

room	time
CS 310	4pm
CS 105	2pm

- **Many solutions in database/Web/AI communities with significant progress**
- **But most solutions have considered only static text corpora**

IE over Evolving Text Data

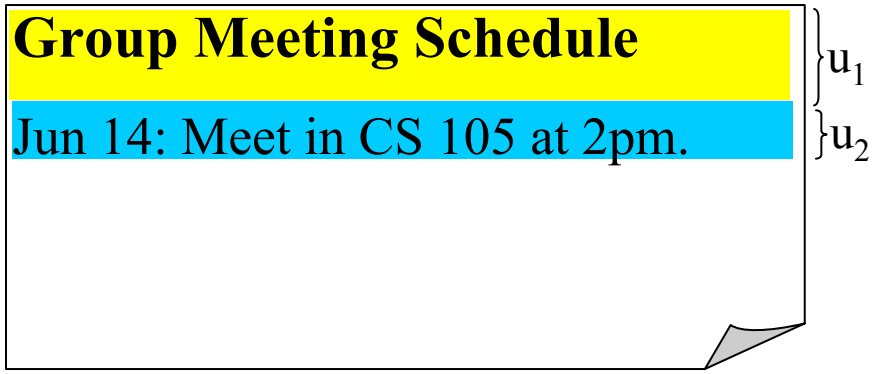


Current Approach and Its Limitations

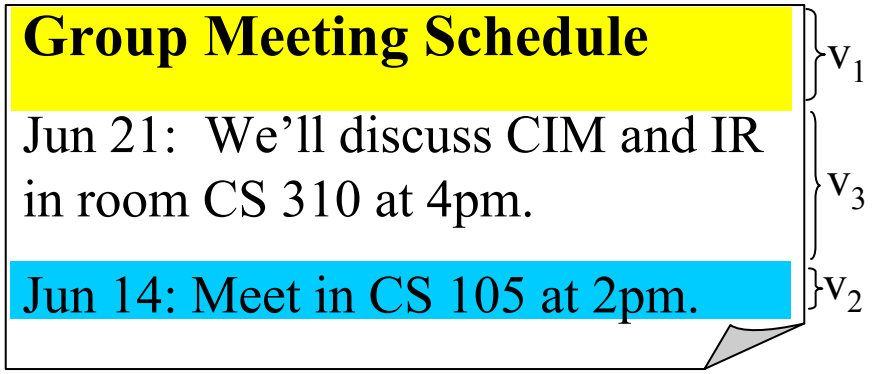
- **Apply IE to each corpus snapshot in isolation, from scratch**

- **Limitations:**
  - Inefficient: e.g., IE in DBLife
  - Unsuitable for time-sensitive applications: e.g., stock, auction
  - Unsuitable for interactive debugging over dynamic text corpora

Cyclex: Recycling Extraction



p



q

Meetings$_1$

room	time
CS 105	2pm

Meetings$_2$

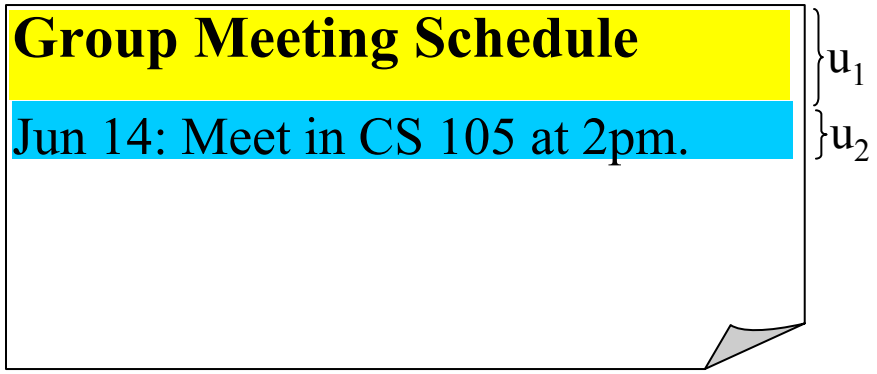
room	time
CS 105	2pm
CS 310	4pm

Challenges and Contributions

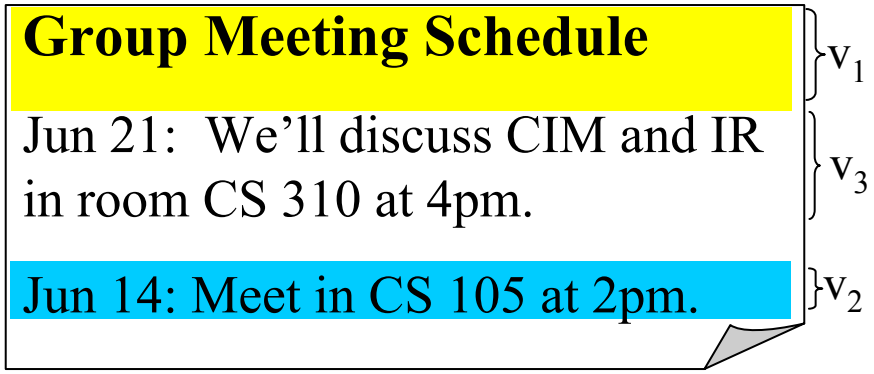
- **How to guarantee correctness?**
  - Model extractors using scope and context
- **How to choose a good way to match pages?**
  - Cost-based decisions using text specific cost model
- **How to efficiently execute the chosen plan given a large amount of disk-resident data?**
  - A way to scan data once

Why Guaranteeing Correctness Is Hard?

E extracts meetings only if a page has fewer than 4 lines



p



q

$Meetings_1$

room	time
CS 105	2pm

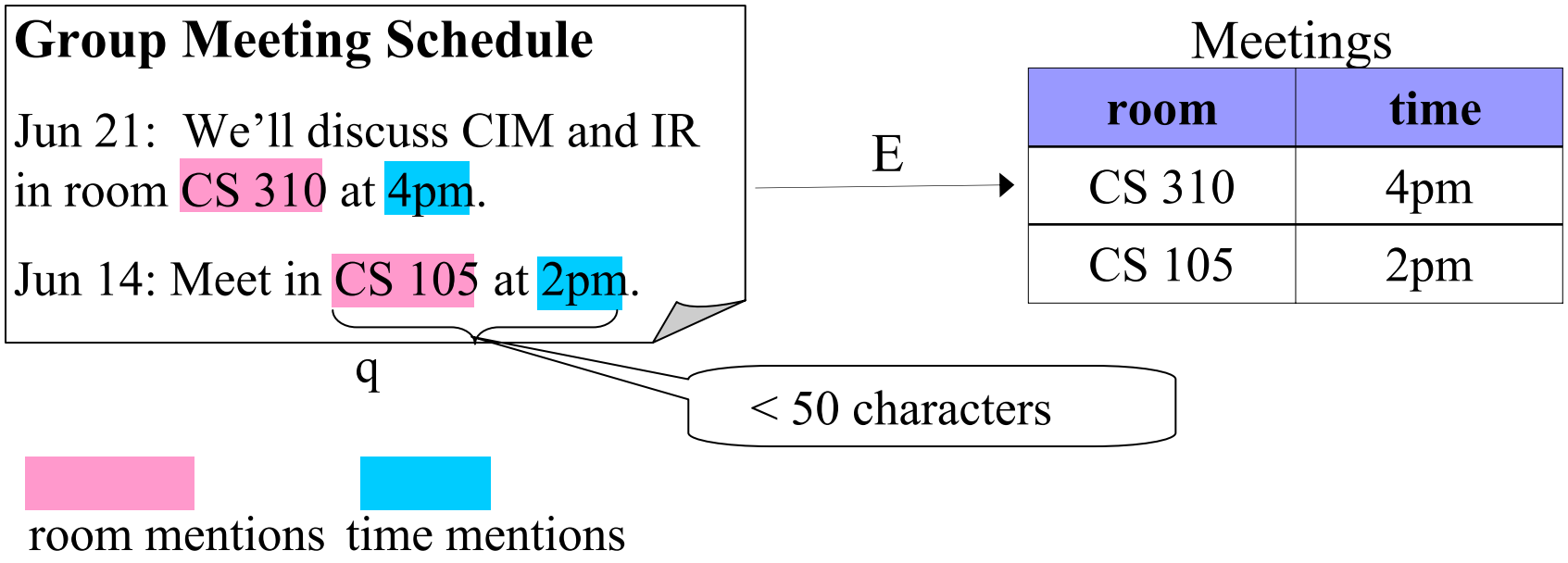
$Meetings_2$

room	time
CS 105	2pm
CS 310	4pm

Extractor Properties: Scope

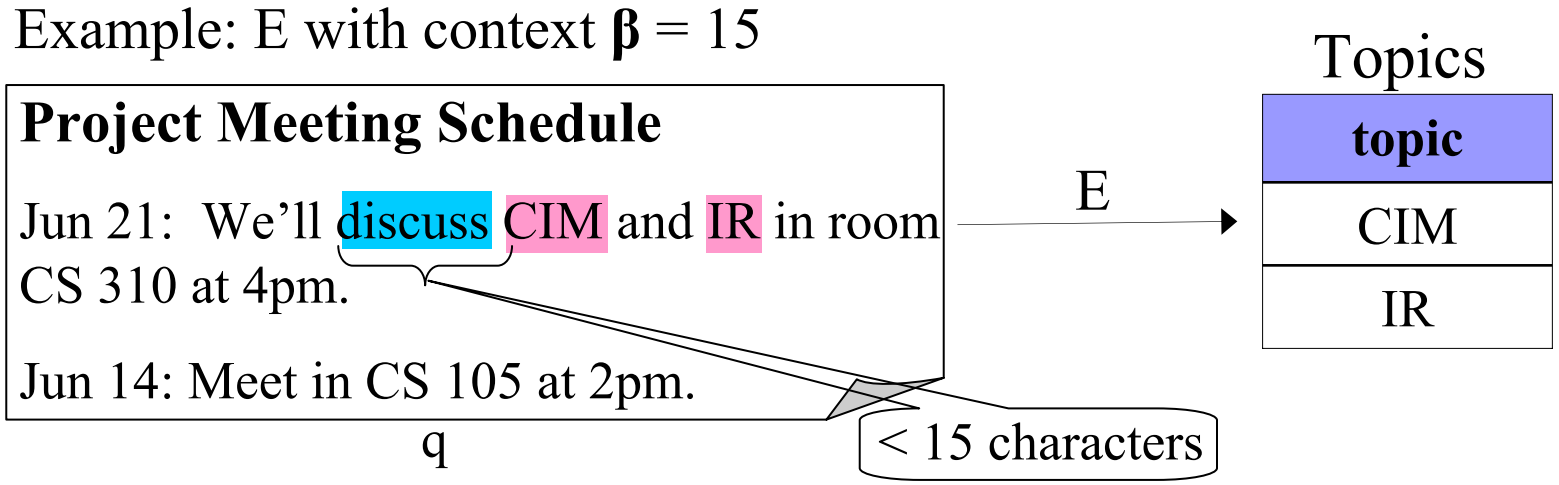
- **Attribute mentions of an entity often appear in close proximity in data pages.**
  - An extractor E has scope α iff any mention produced by E at most spans α characters.

Example: E with scope **α** = 50



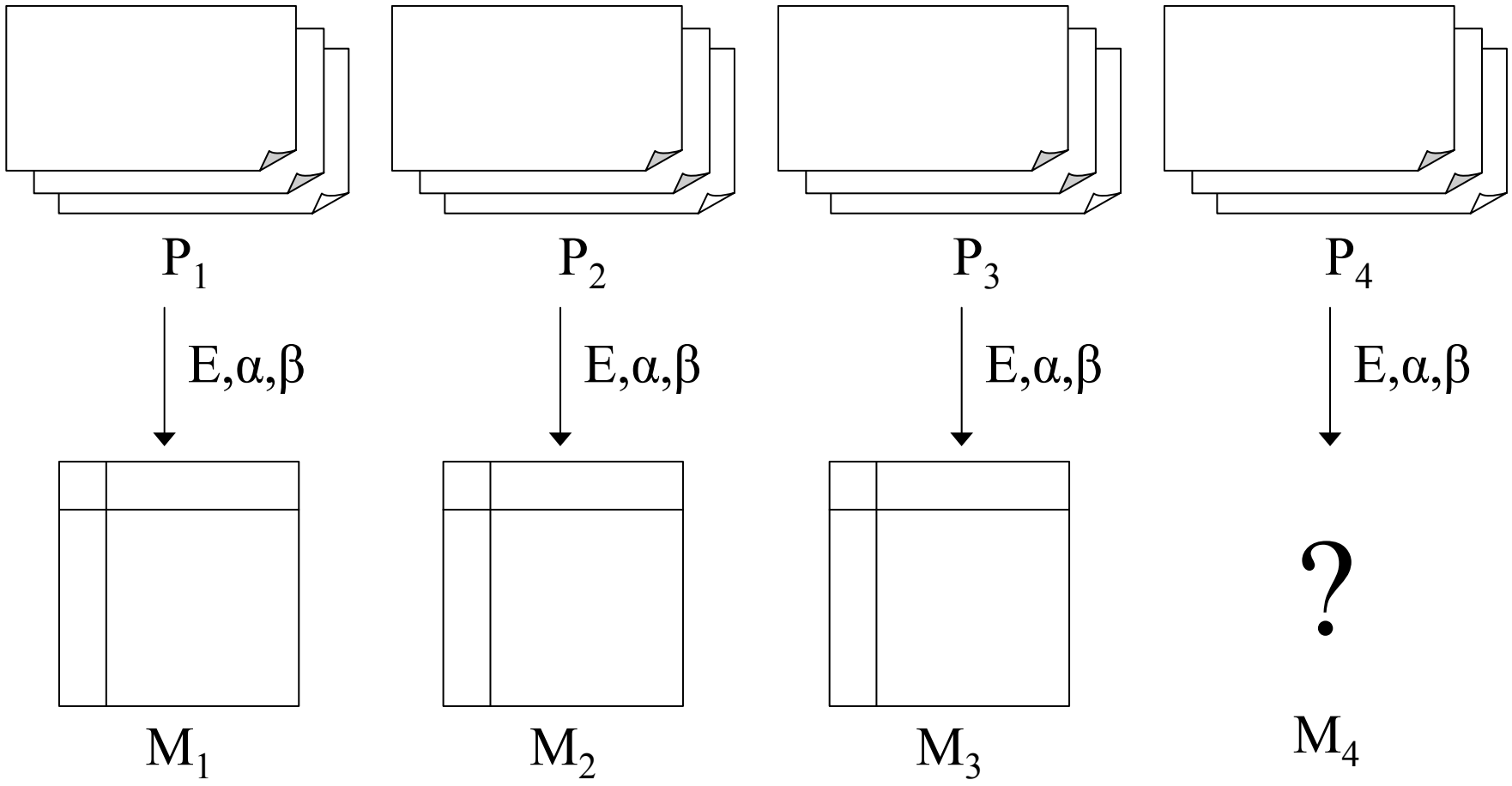
Extractor Properties: Context

- **Many extractors only examine small "context windows" on both sides of a mention to extract the mention.**



- **The text outside the context of a mention m is irrelevant for E to extract m.**

Problem Definition



Match Pages To Find Overlapping Regions

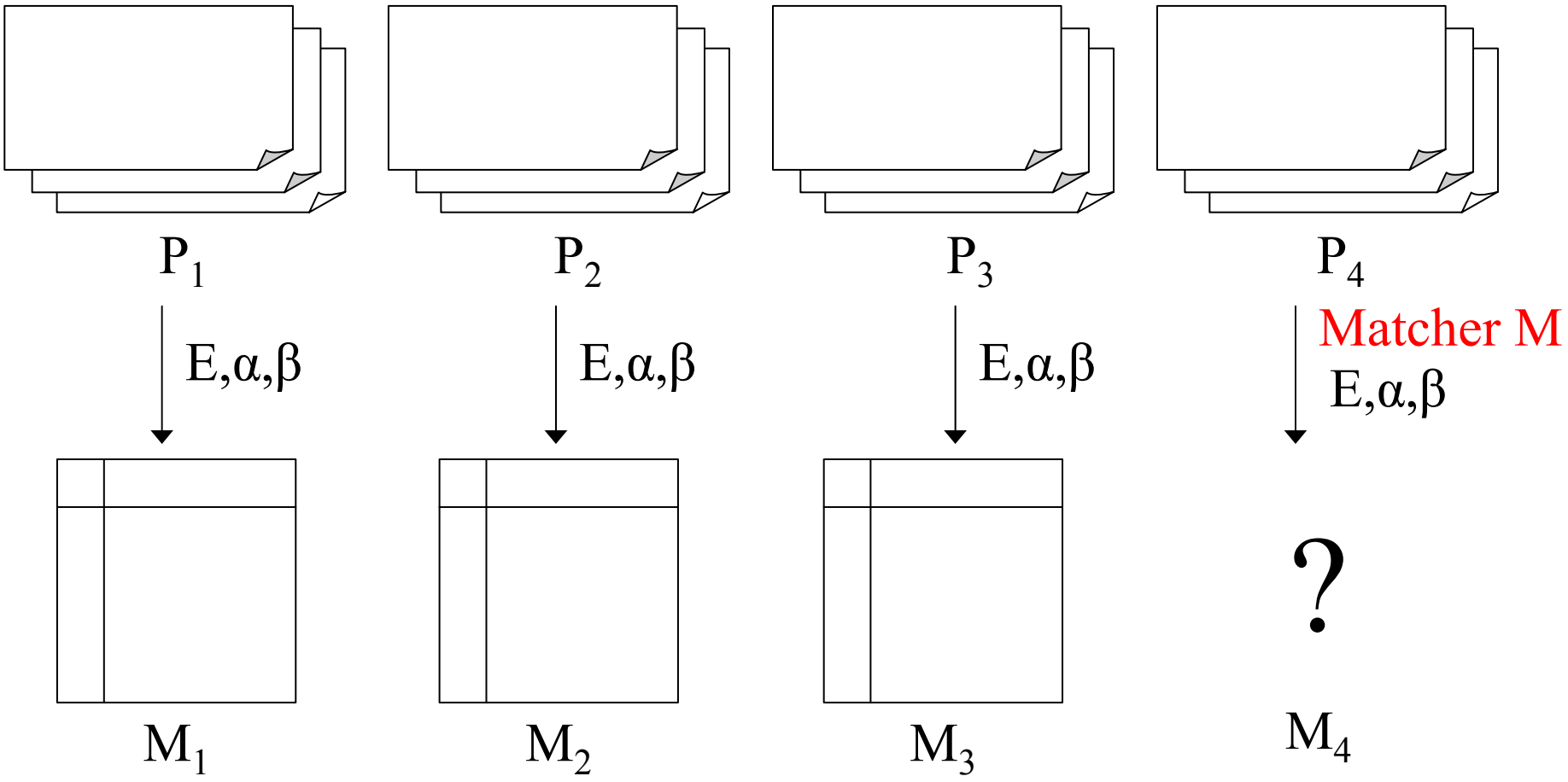
- **Consider 3 matchers (more can be added)**
  - DN (Doing Nothing) : immediately declares no overlapping regions are found
    - 0 runtime and no overlapping regions
  - UD (Unix Diff): a Unix-diff-command like algorithm
    - relatively fast runtime and some overlapping regions
  - ST (Suffix Tree): a novel suffix-tree based algorithm we developed
    - linear in the length of pages runtime and all overlapping regions
- **Matchers trade off runtime with result completeness**

(See paper for more details)

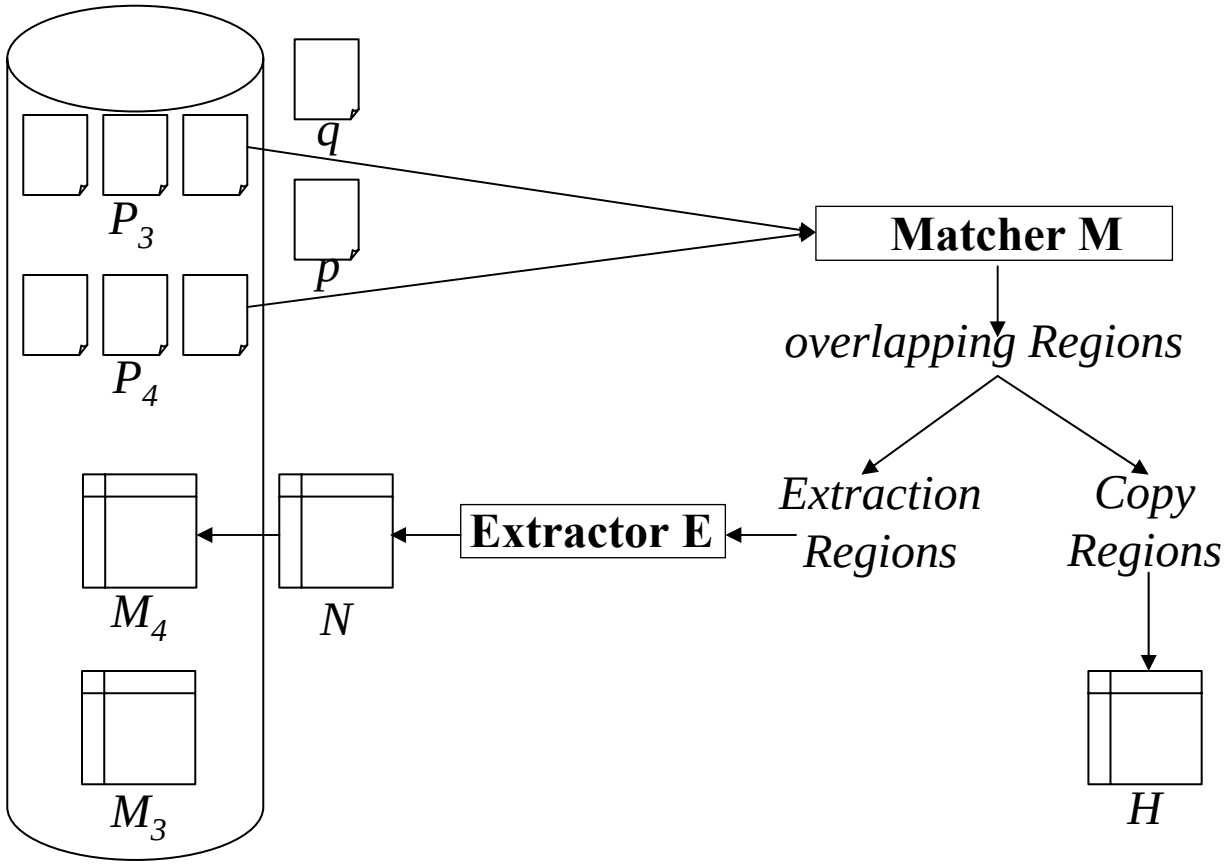
Choose the Optimal Matcher

- **Consider a plan space where plans differ in the matchers they use**
- **Use a cost model to estimate the completion time of each plan**
- **Text-specific cost model**
  - e.g., change rate of the text corpus, cost of the extractor, size of matching results and IE results, etc.
- **Collect statistics over past k snapshots**

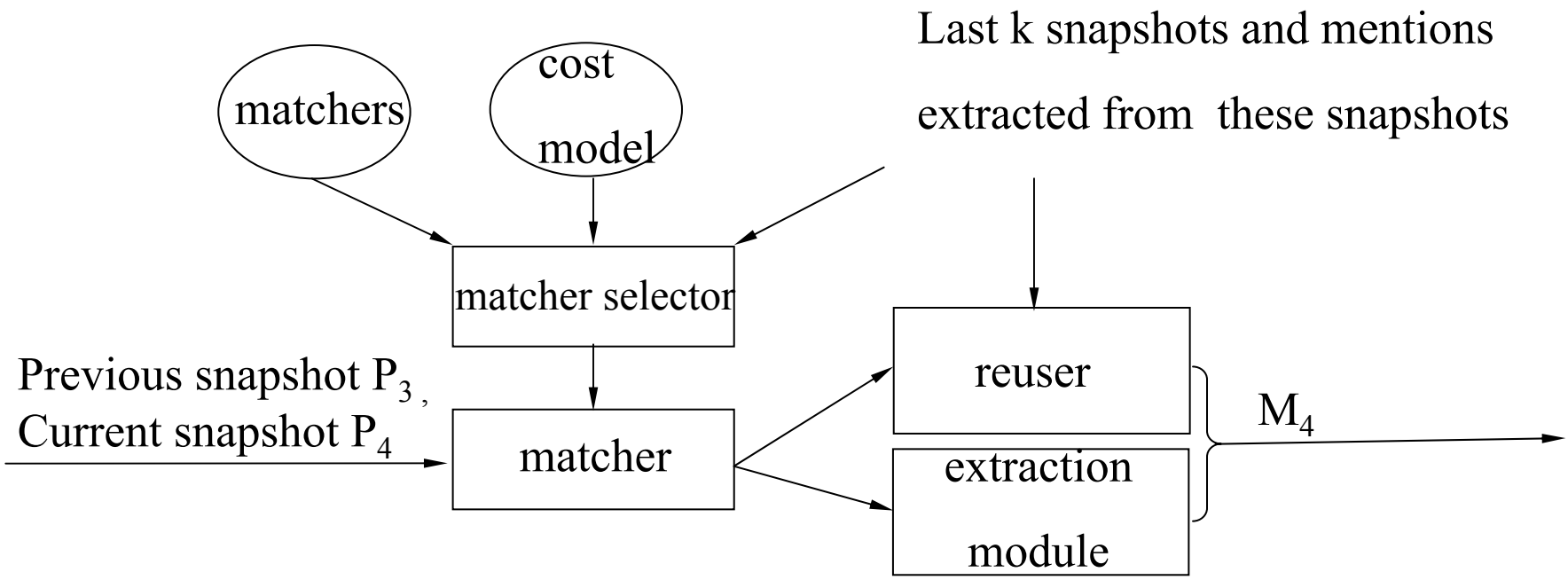
Challenge in Efficiently Executing the Chosen Plan



Interleave Matching, Extraction and Copy



Architecture



Experiment Setup

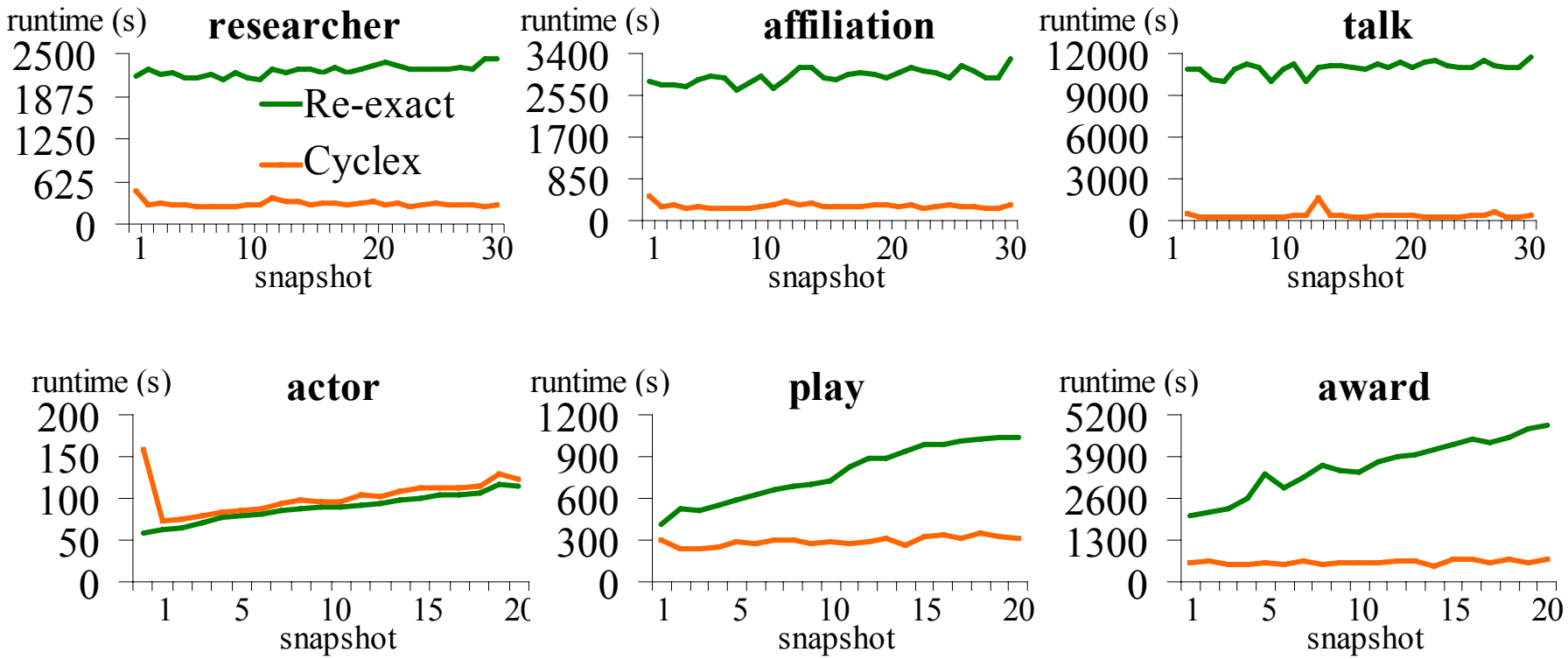
- **Datasets**

Data Sets	**DBLife**	**Wikipedia**
# Data Sources	980	925
# Snapshots	30	20
Time between snapshots	1 day	21 days
Avg # Page per Snapshot	10155	3038
Avg Size per Snapshot	180M	35M

- **Extractors**

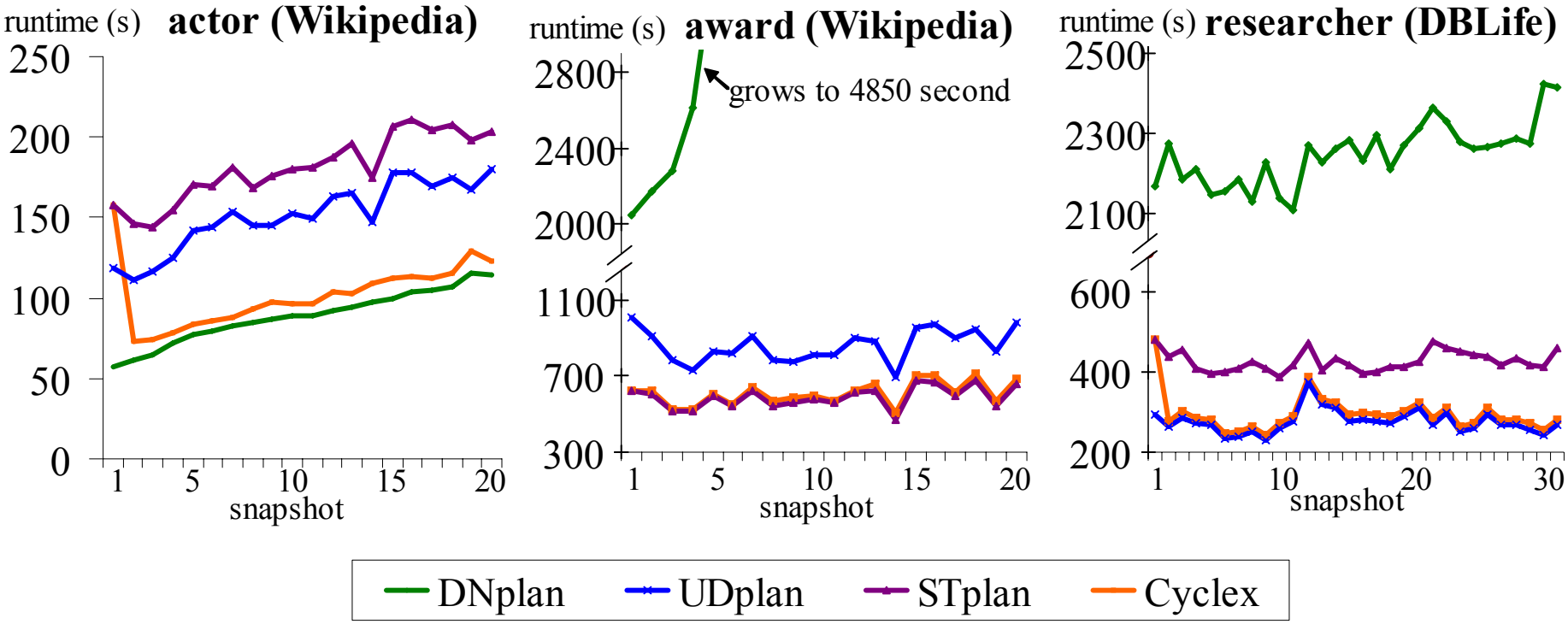
	DBLife			**Wikipedia**		
	researcher	affiliation	talk	actor	play	award
Scope α	32	93	400	35	96	250
Context β	3	7	10	3	4	10

Benefit of Recycling IE Results



- **In all cases except "actor", Cyclex drastically cut runtime of re-extraction from scratch by 50-90%**

Importance of Optimization



- **None of the matchers is uniformly optimal.**

  (See paper for more details)

Conclusion and Future Work

- **Proposed the first approach to speed up IE over evolving text data by recycling past IE results**

- **Defined challenges and provided initial solutions**
  - Model properties of extractors
  - Cost-based decisions in choosing an optimal matcher
  - Efficiently interleave matching, extraction, and copying

- **Future work**
  - Handle multiple extractors
  - Handle extractors that extract mentions across multiple pages

Related Work

- **Much work on IE**
  - Improve accuracy and efficiency
  - Recent work on scalable IE [tutorial in KDD06, SIGMOD06]

- **Evolving text data**
  - Repair wrappers as page templates change [McCann VLDB05]
  - Incrementally update an inverted index [Lim WWW03]

- **Exploiting overlapping text data in a document collection to compress indices** [Herscovici ECIR07, Zhang WWW07]

- **Optimizing IE programs and developing text-centric cost models** [Ipeirotis SIGMOD06, Jain ICDE07, Shen VLDB07]