Ranking Biomedical Passages for Relevance and Diversity

University of Wisconsin-Madison at TREC Genomics 2006

Andrew B. Goldberg, David Andrzejewski,
Jurgen Van Gael, Burr Settles, Xiaojin Zhu, Mark Craven*
Computer Science Department, * Biostatistics Department

Outline

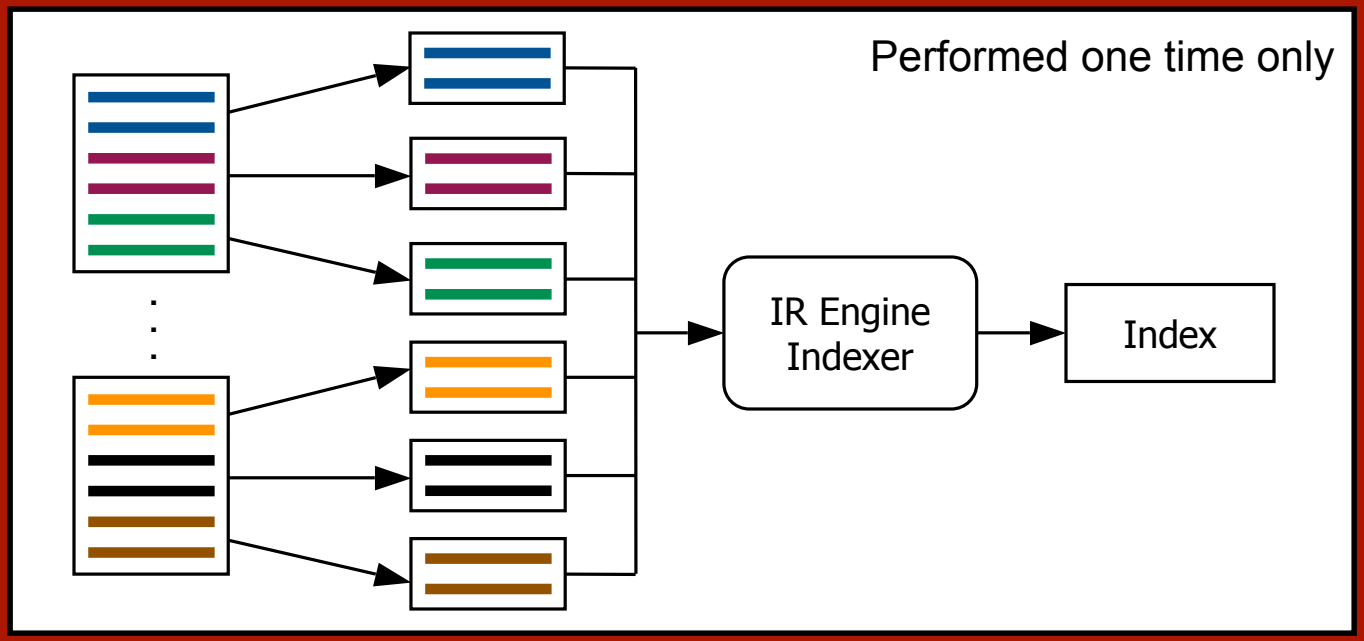
- Genomics Track Task Overview
- System Overview
- *Reranking using Absorbing Random Walks*
- Results
- Discussion

Genomics Track Task Overview

- Given 160,000 full-text biomedical articles
- Given a scientific query:
  - *What is the role of PrnP in mad cow disease?*
- Find and rank short passages about different aspects of the question
- Document, passage, and aspect-level evaluation metrics
- UW-Madison submitted 3 automatic runs

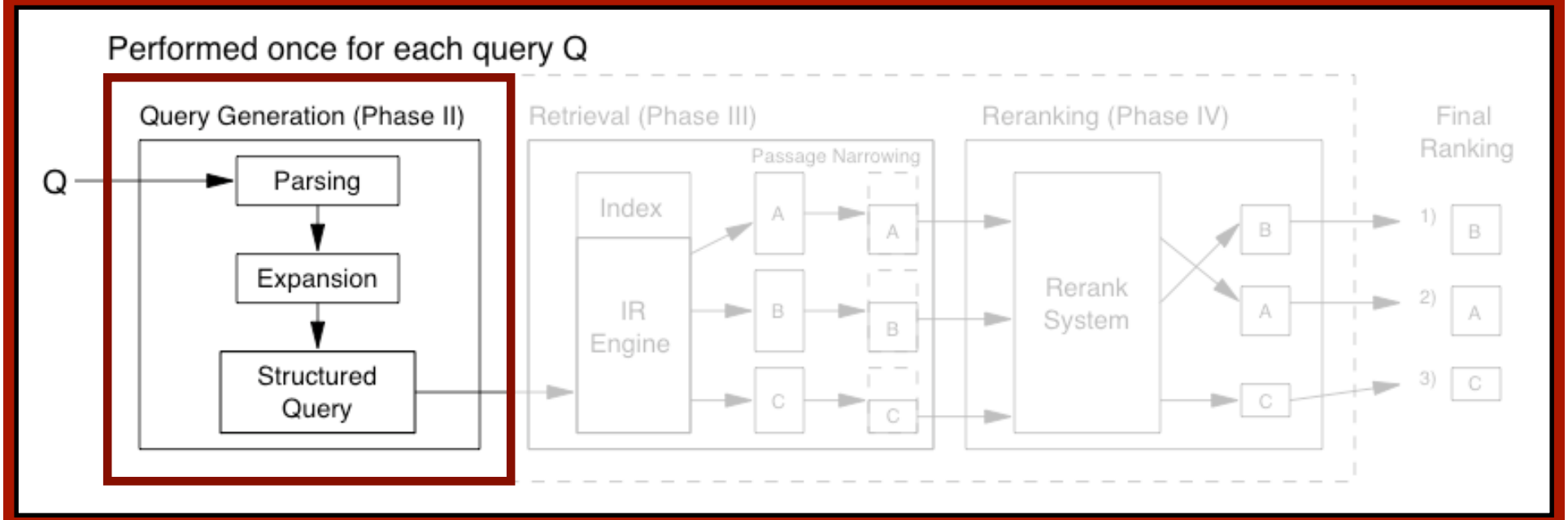
Phase I: Indexing

1. Split documents into legal spans
2. Build index using off-the-shelf IR engine
   - Lemur toolkit with Indri index



Phase II: Query Generation

1. Parse natural language topic questions
2. Expand queries using online resources
3. Automatically generate structured queries



Phase II: Query Generation

- Example parse (before stop word removal):

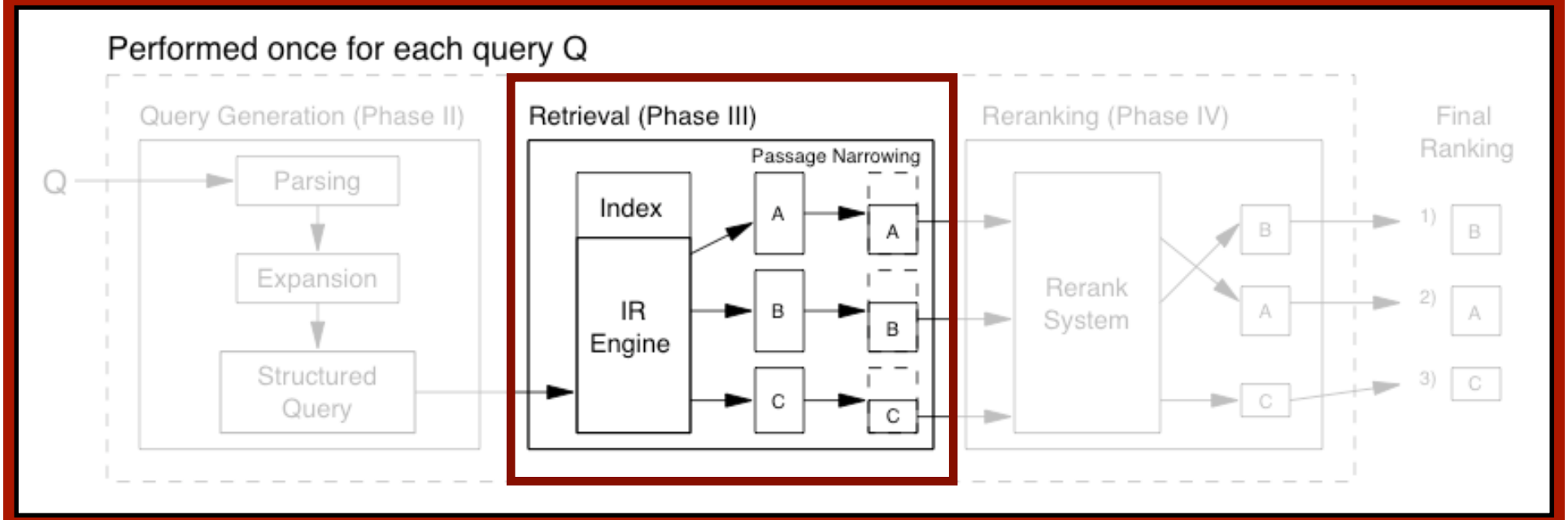
What is the role of PrnP in mad cow disease ?

- Structured query with expansion terms:

```
#filreq(
   #band(
      #syn( #1( PrnP ) #1( prion protein )...)
      #syn( #1( mad cow disease ) #1( BSE )
            #1( Bovine Spongiform Encephalopathy )...)
   )
   #combine( #1( PrnP ) #1( prion protein ) ...
             #1( mad cow disease ) #1( BSE )
             #1( Bovine Spongiform Encephalopathy )...
   )
)
```

Phase III: Retrieval

1. Run query using off-the-shelf IR engine
2. Trim paragraphs to create passages that include only the relevant sentences



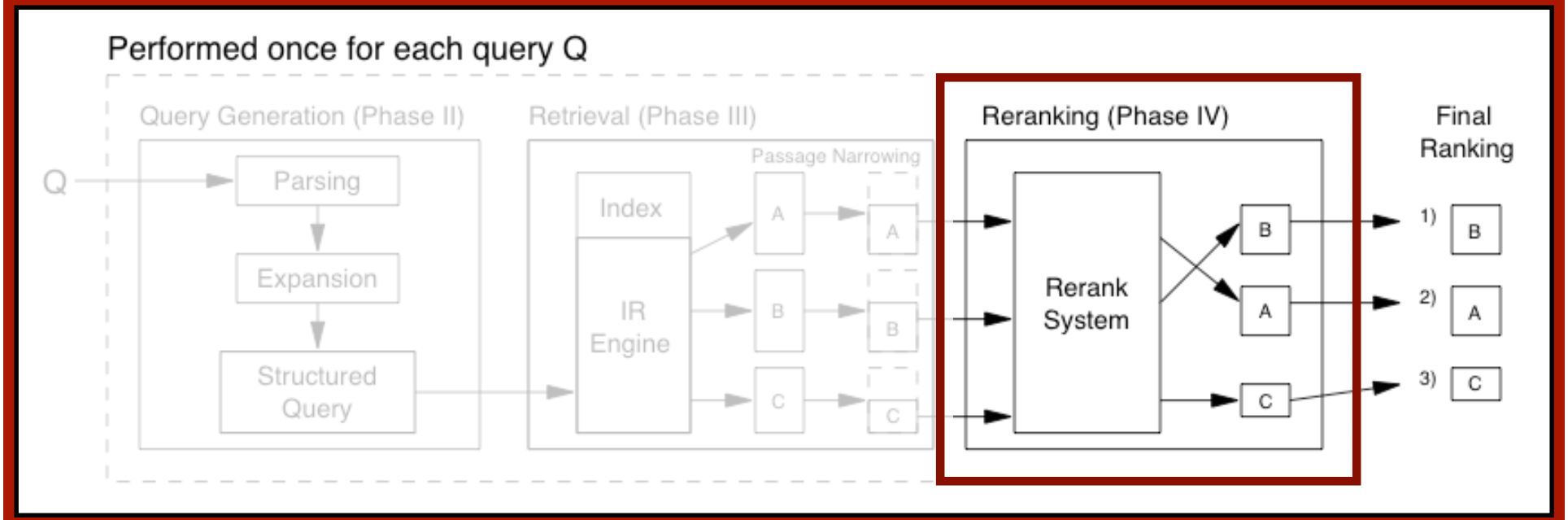
Phase III: Retrieval

- Example of passage narrowing:

In December 1984 a UK farmer called a veterinary surgeon to look at a cow that was behaving unusually. Seven weeks later the cow died. Early in 1985 more cows from the same herd developed similar clinical signs. In November 1986 **bovine spongiform encephalitis** (**BSE**) was first identified as a new disease, later reported in the veterinary press as a novel progressive **spongiform encephalopathy**. Later still the causal agent of **BSE** was recognized as an abnormal **prion protein**. Since the outset the story of **BSE** has been beset by problems.

Phase IV: Reranking

- Run 1: Indri ranking
- Run 2: Clustering-based reranking
- Run 3: Absorbing random walk reranking

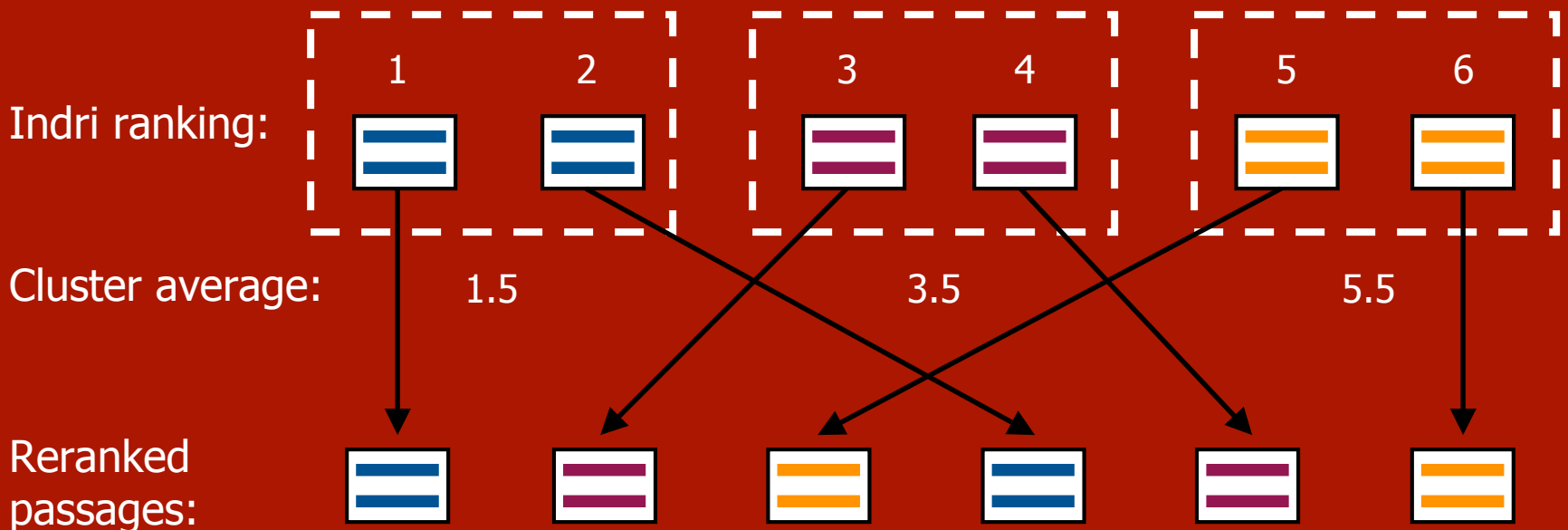


Clustering-Based Reranking

- Cluster passages using bag-of-words vectors and cosine similarity
- Assume clusters represent aspects
- Interleave results from clusters to achieve aspect diversity

Clustering-Based Reranking

- Example with 3 clusters



(Note: We arbitrarily used 10 clusters in Run 2)

Absorbing Random Walk Reranking

- Produces new ranking such that
  - A highly ranked passage is *central* to a local group in the set
  - Top ranked items cover many *diverse* groups
  - Initial ranking is included as prior knowledge
- Achieves these goals using *absorbing Markov chain random walks*

High-Level View of Algorithm

- Random walk on a graph over passages
- Ranked passages become absorbing states
- Absorbing states "drag down" importance of similar unranked states
- Newly ranked states differ from previously ranked states to promote diversity

Algorithm Input

- Graph W with n nodes (items to rank)
  - Represented by n x n weight matrix
  - Large weight means similar items
- Prior distribution r based on initial ranking
  - High initial ranks have high probabilities
  - No prior ranking = uniform distribution (all $1/n$)
- Weight $\lambda \in [0,1]$
  - Balances influence of W versus r

Finding the First Item to Rank

- Teleporting random walk
- Random walker moves around graph
  - With probability λ:
    - Walks to a neighbor state based on edge weights W
    - More likely to walk to similar state
  - Otherwise:
    - Teleports randomly according to r
    - More likely to walk to state with high initial rank

Finding the First Item to Rank

1. Create transition matrix by normalizing rows of W weight matrix

$$\tilde{P}_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}}$$

2. Transform into teleporting random walk by interpolating each row with prior r

$$P = \lambda \tilde{P} + (1 - \lambda)\mathbf{1}\mathbf{r}^{\top}$$

Finding the First Item to Rank

- Stationary distribution of random walk
  - Defines visiting probabilities of nodes
- Dense regions of graph (soft clusters) have high probabilities
- High probability states regarded as central, most important items
  - Like Google's PageRank algorithm

Finding the First Item to Rank

3. Find unique stationary distribution

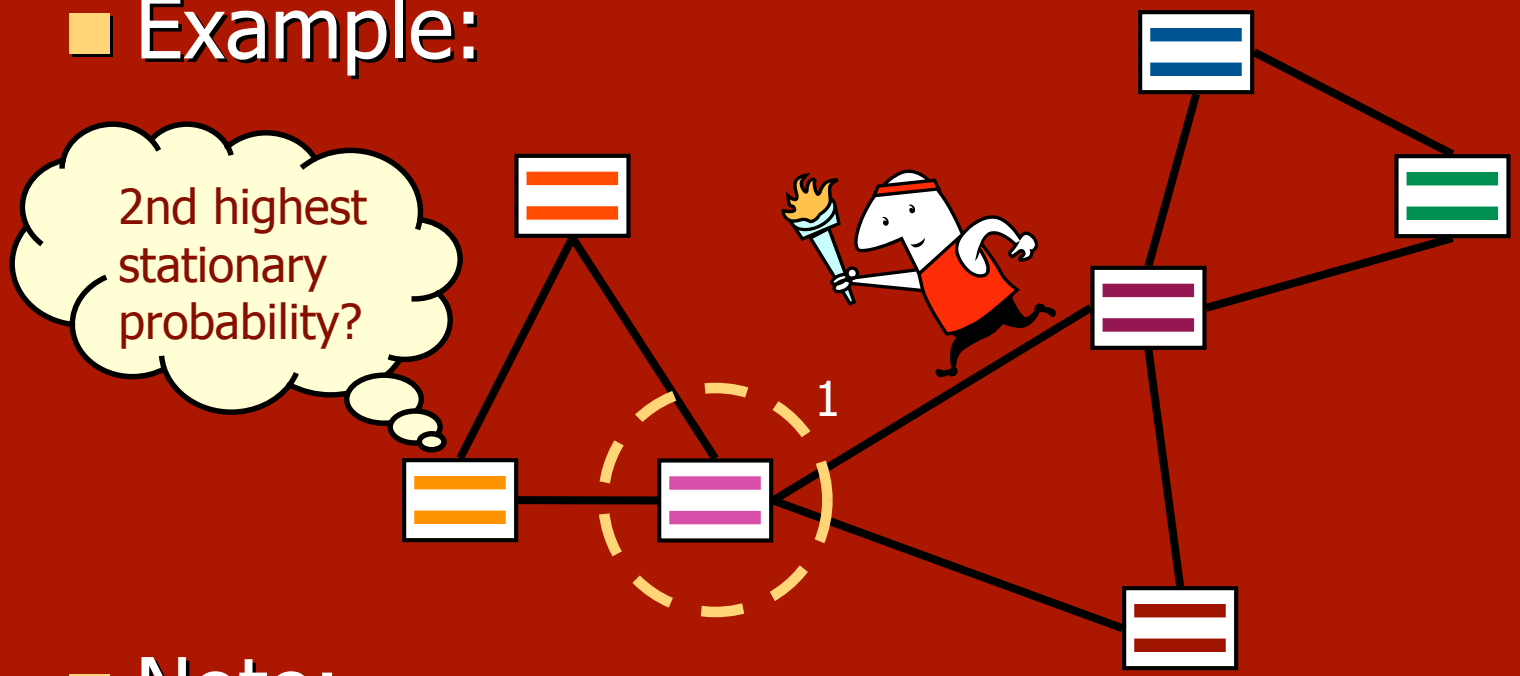
$$\pi = P^\top \pi$$

4. Select first item as the state with the largest stationary probability

$$g_1 = \text{argmax}_{i=1}^{n} \pi_i$$

Finding the First Item to Rank

- Example:



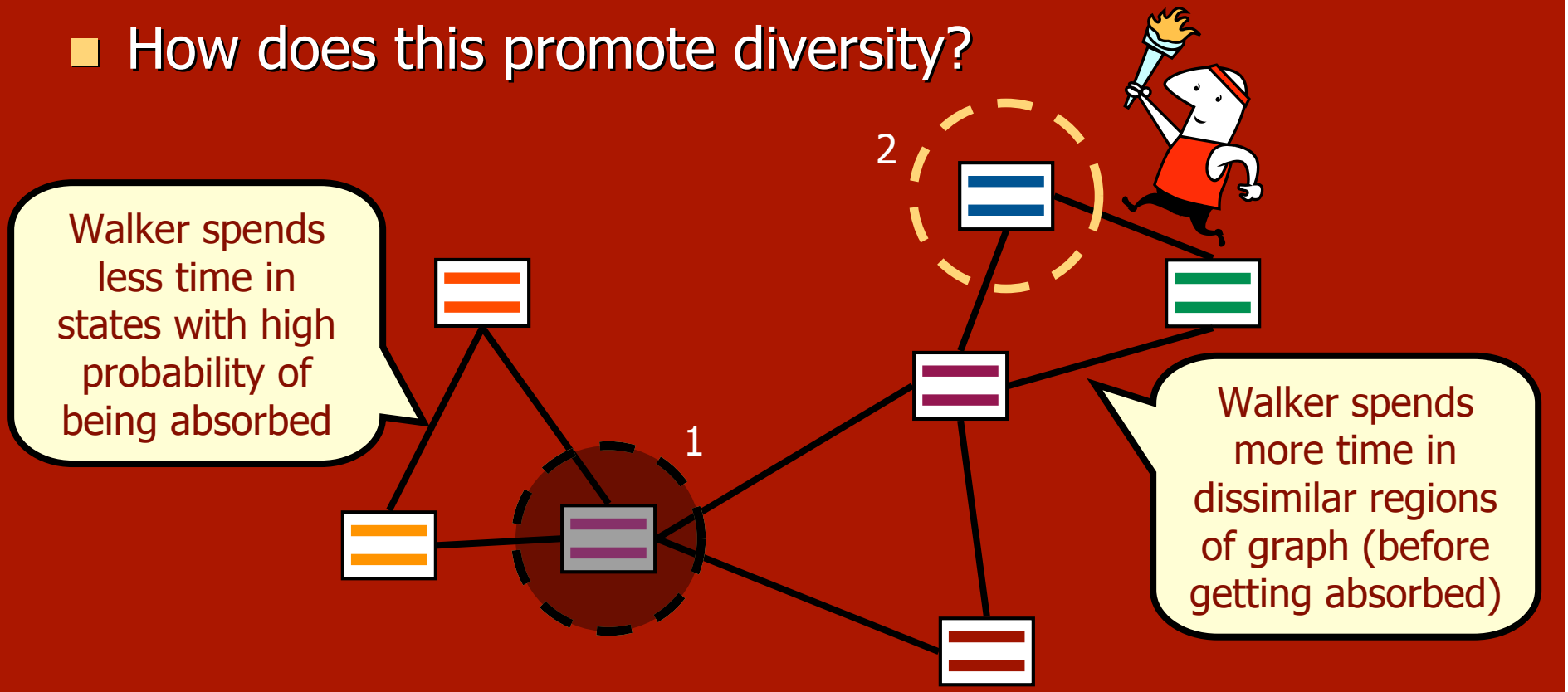
- Note:
  - Only the larger W edge weights are shown
  - Interpolation with r makes it fully connected

Centrality versus Diversity

- Stationary distribution lacks diversity
  - High probability items from same local groups
- To ensure diversity:
  - First ranked item becomes an absorbing state
  - Walker can fall in "black hole" and walk ends
- Stationary distribution now uninformative
  - All walks will eventually get absorbed
- Need alternate way to select items

Ranking the Remaining Items

- New selection criterion:
  - Expected number of visits before absorption
- How does this promote diversity?



Ranking the Remaining Items

While more items to rank:

1. Turn ranked states into absorbing states
2. Compute expected number of visits per unranked item
3. Select the item with the maximum expected number of visits

Ranking the Remaining Items

Turn ranked states G into absorbing states

$$\text{For } g \in G, P_{gg} = 1, P_{gi} = 0, \forall i \neq g$$

Arrange P with ranked before unranked states:

$$P = \begin{bmatrix} \mathbf{I}_G & \mathbf{0} \\ R & Q \end{bmatrix}$$

Ranked items

Unranked items

Find *fundamental matrix*:

$$N = (\mathbf{I} - Q)^{-1}$$

N_{ij} = expected number of visits to state j before absorption, if the walk started in state i

Ranking the Remaining Items

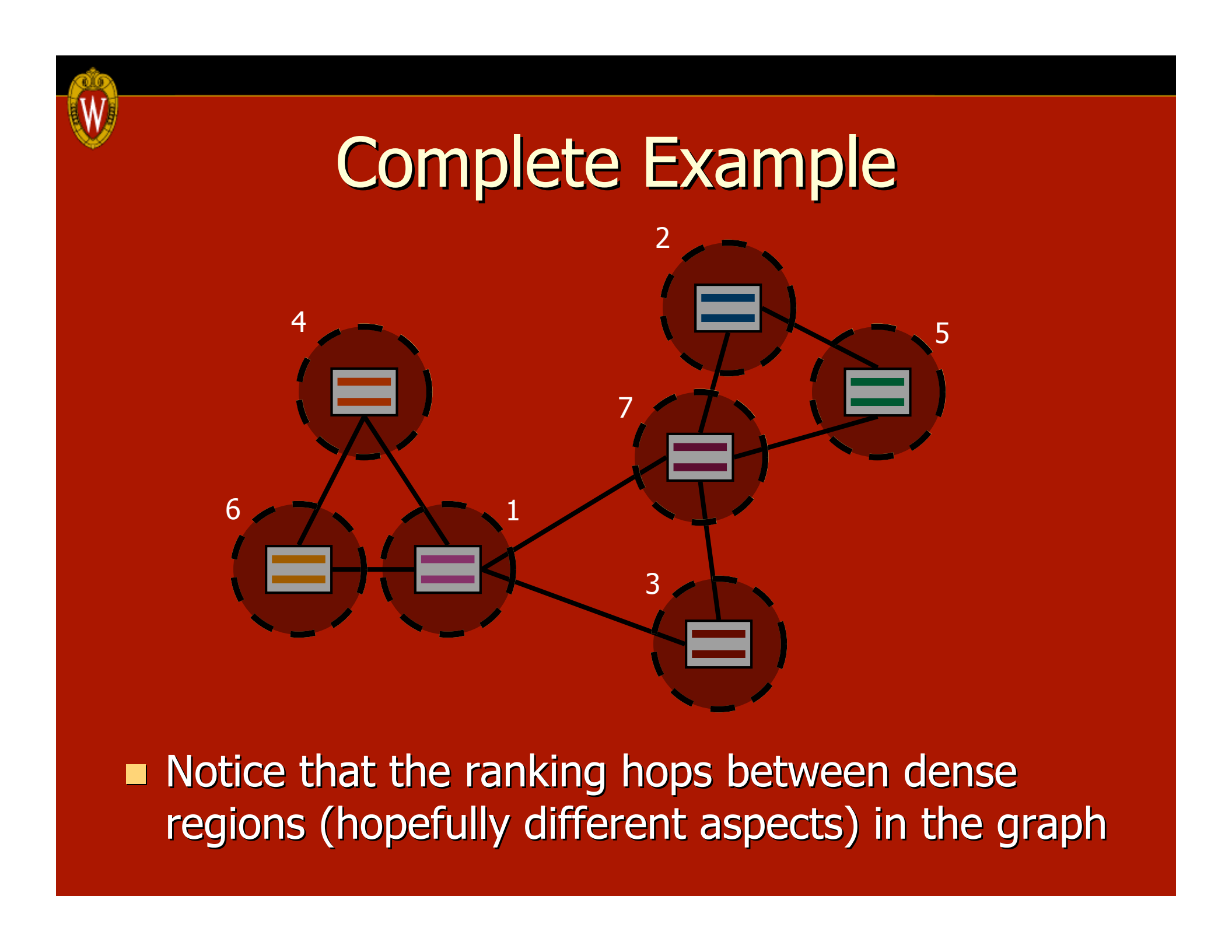
The expected number of visits per state:

$$\mathbf{v} = \frac{N^\top \mathbf{1}}{n - |G|}$$

v_i = expected number of visits to state j before absorption, regardless of starting state

Picking the next item:

$$g_{|G|+1} = \arg\max_{i=|G|+1}^{n} v_i$$



Complete Example

- Notice that the ranking hops between dense regions (hopefully different aspects) in the graph

Parameters in Run 3

- Weight matrix W
  - Symmetric 10-NN graph using cosine similarity of bag-of-words vectors
- Initial ranking distribution r

$$r_i \propto (n - initialRank_i + 1)$$

- Trade-off parameter λ
  - Arbitrarily set to 0.6 to put more emphasis on graph but still have influence of initial ranking

Results

Run	Document	Passage	Aspect
Indri Ranking	0.2368	0.0188	0.1516
Clustering	0.2030	0.0137	0.1319
Random Walk	0.2208	0.0159	0.1411

Mean average precision (MAP) scores

- Document and Passage seem mediocre
- Aspect appears competitive, but reranking methods fail to improve over baseline

Discussion

- Poor document and passage MAP scores
  - Query generation inadequate
  - No results produced for some topics
  - Perhaps exact matching in queries too strict
- Solutions?
  - Refine parsing technique
  - Use less restrictive query operators
  - Consult additional resources (GO, UMLS, etc)

Reranking Discussion

- Irrelevant documents appear diverse
  - Incorrectly placed even higher in ranks
- Similarity graph may be inappropriate
  - Needs to correlate with aspect similarity
  - Could use TF-IDF vectors (with IDF based on current set of passages)
  - Also, KL-divergence between language models

Thank You!

Questions?