Predicting Web Actions from HTML Content

Brian D. Davison
davison@cse.lehigh.edu

Computer Science and Engineering
Computer Science and Engineering
Computer Science and Engineering
Computer Science and Engineering
CSE

LEHIGH UNIVERSITY

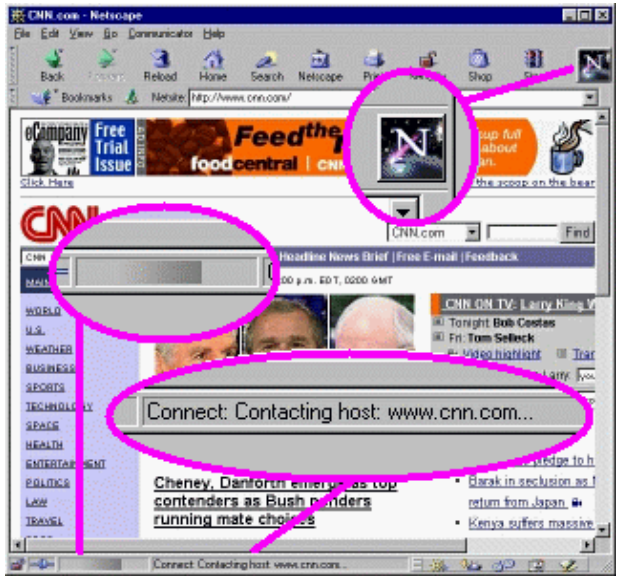
Outline

- Web Prefetching
- Prediction Using History
- Prediction Using Content
- Evaluation of Prediction
- Full-Content Data Set
- Experimental Results
- Summary

LEHIGH

Web Prefetching

- Perception of the 'World-Wide Wait' persists.
- Web caching in proxies and browsers helps.
  - But only useful for objects retrieved in the past.



- Prefetching has potential to help much more.
  - Need to predict user's request in advance.

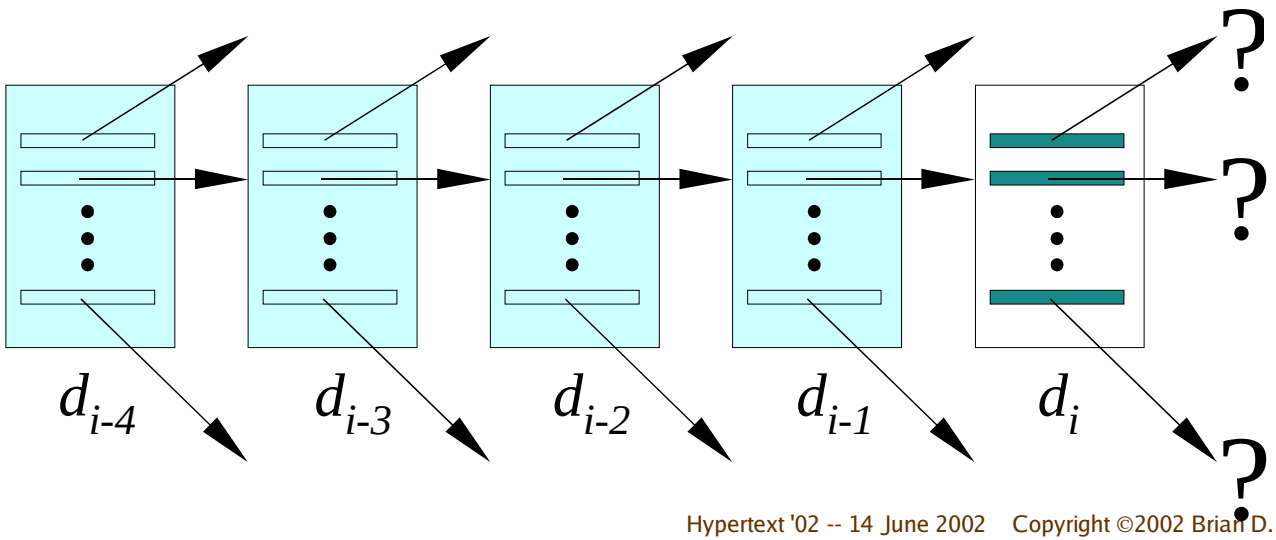
LEHIGH

Prediction Using History

- Primarily based on Markov models.
  - Calculates p(next|past)
  - Suggested by many researchers over years.
    - e.g., Using Markov Models for Web Site Link Prediction.
- But not always applicable:
  - Too little data, e.g., on first visit to a site.
  - Site-specific model can't predict off-site clicks.

Prediction Using Content

- Web content can be examined.
  - We can see the links within a page.
- Most page requests (perhaps 80%) are from clicking on a link in the current page.
- Knowing the links of the current page is a significant boost to finding the next page.



Prediction Ranking

- A naïve approach:
  - Use all links as predictions.
  - Prefetching all links typically requires too much time and/or bandwidth.
- We need to rank predictions by likelihood
  - Baseline ranking: Randomly order URLs.
    - (If we can't do better than random, we aren't doing anything)
  - Another simple approach:
    Rank the URLs in original page order.
- A more intelligent approach should be possible.

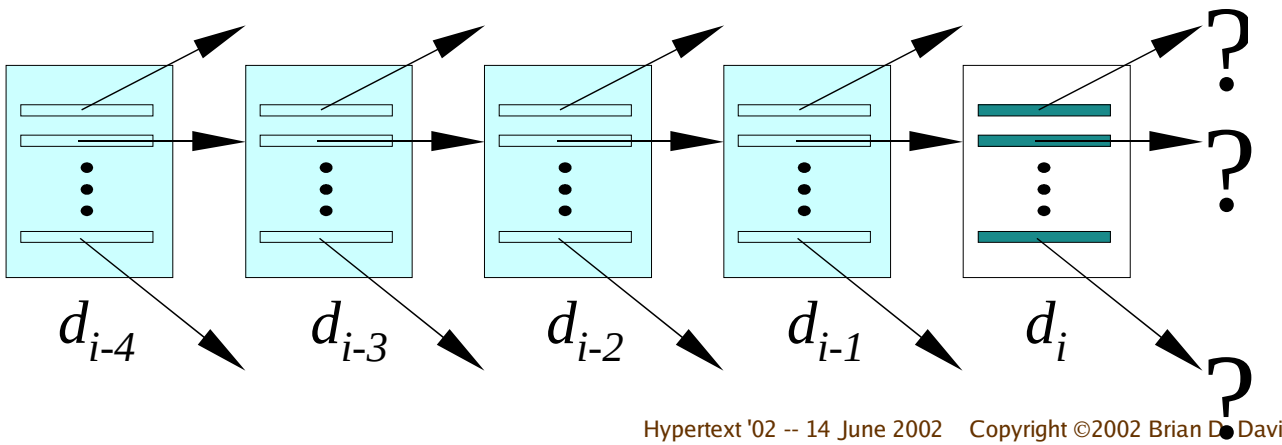
How does a user choose links?

- We suggest:
  - The user chooses the most interesting link.
- If we knew the user's interest, we could rank the links appropriately.
- One possibility:
  - Ask the user for their interests or learn and get feedback.
    - We would prefer something unintrusive.
    - We would also have to worry about multiple or changing interests.

LEHIGH

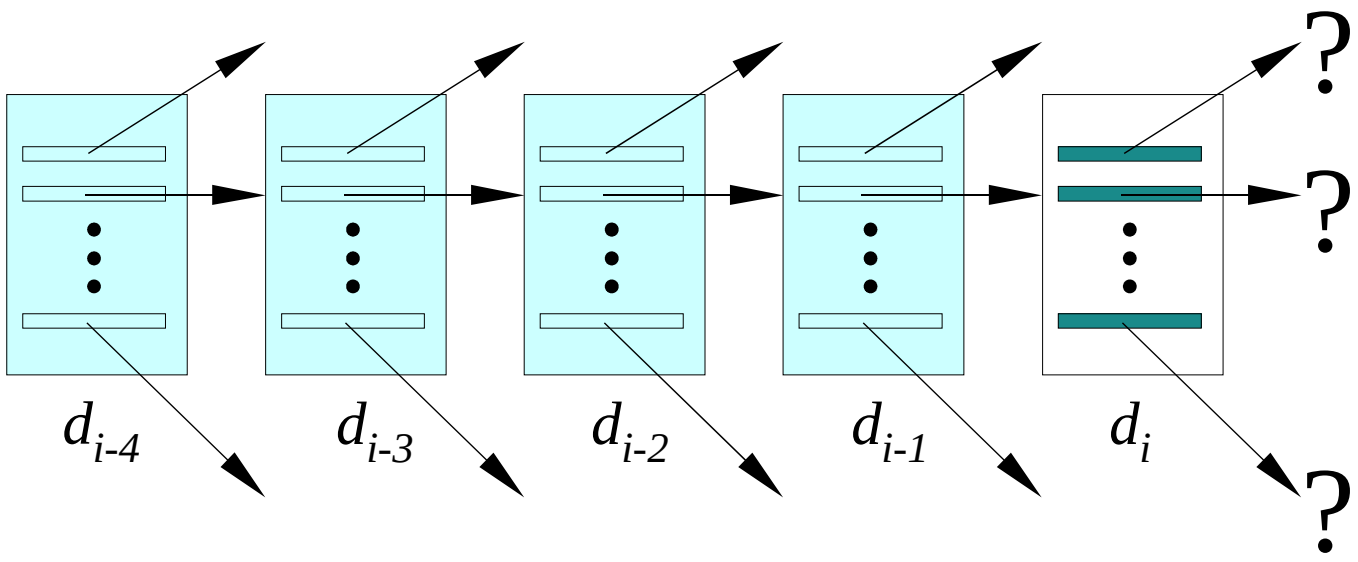
User Interest

- Our hypothesis:
  - The user is looking at his/her current interest.
  - (The set of pages recently seen corresponds to the current interest for the user.)
- Given a user interest, how to rank a set of links?
  - The text within and around a link provide a good description of the target document.
  - Therefore, look for link text that is similar to the interest.



Approach taken

- Combine text of previous four pages as a single document.
- Calculate the similarity (that is, essentially just the stemmed terms in common) to each of the potential links.
- Links with more terms in common get a larger score.



LEHIGH

Additional Details

- Many Web sites have repeated structure on every page (menu, disclaimer, etc.)
  - If you were to just concatenate the previous pages, you would emphasize the repeated text.
  - Effectively suggests that a "Terms of Use" link is highly desirable!
  - Instead, we only add the differences between pages so that such repeated text is no longer unduly emphasized.
- We use up to twenty additional terms both before and after each link in addition to all of the anchor text.

```
The <a href="http://www.acm.org/acm1/">ACM</a>
Conference will take you beyond cyberspace.
```

LEHIGH

Evaluation of Prediction

- Multiple URL ranking mechanisms:
  - Random order, Original order, and Similarity order
- Need to evaluate predictive accuracy over a real data set.
- Prefetching systems can use more than the top prediction.
  - It may have time/resources to prefetch more than one.
  - It may already have the top prediction in a local cache.
  - We test accuracy using the top one, three, and five predictions.
- We will also evaluate the case in which the prefetching system places objects into an infinite cache.
  - Current prediction failures may be useful later.

Full-Content Data Set

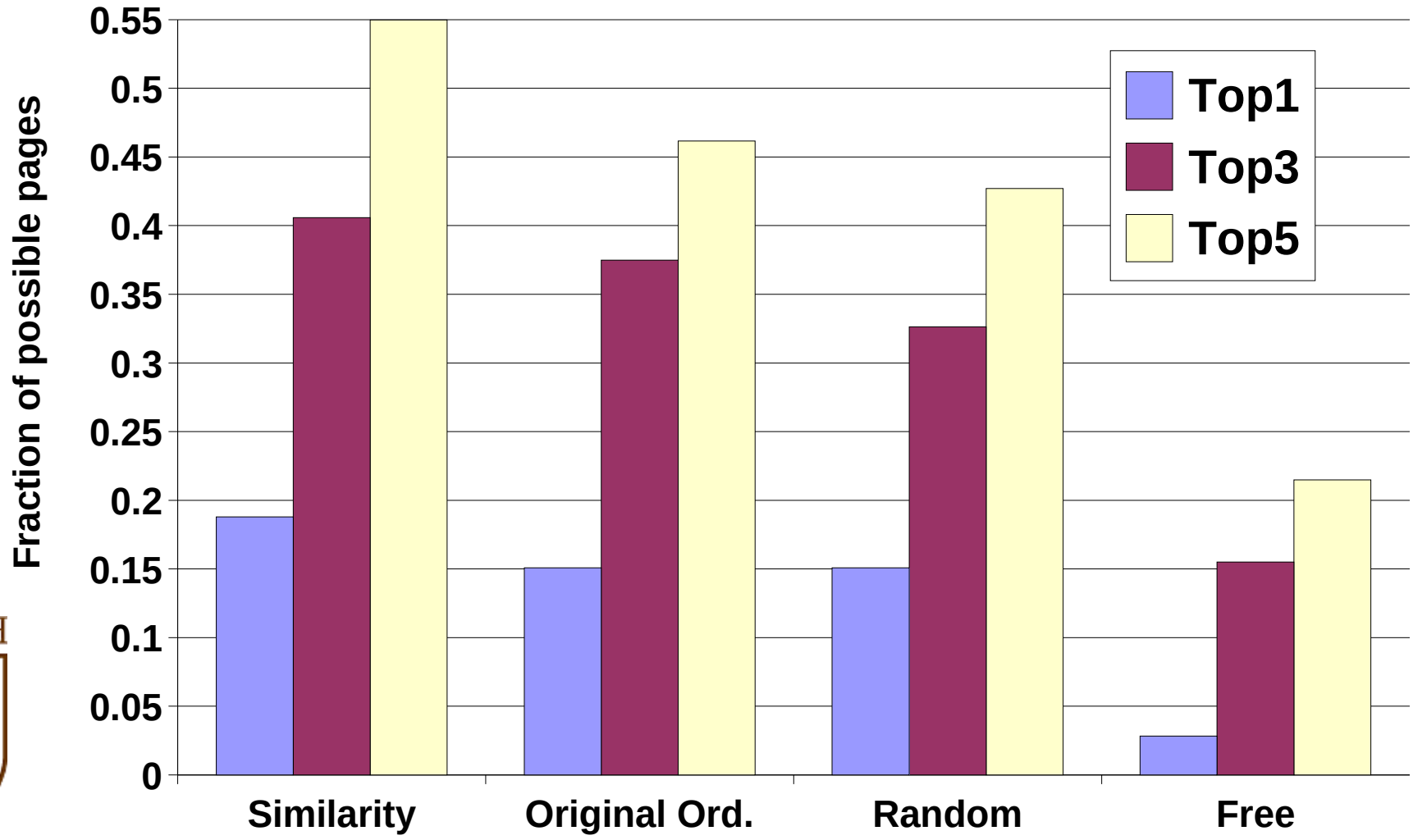
- Developed a custom Web proxy that recorded:
  - Web traffic for eight months in 1998-1999
  - Data from a relatively small set of volunteers
    - (mostly CS faculty and grad students at Rutgers Univ.)
  - Approximately 135,000 HTTP requests.
- This proxy captured full-content:
  - HTTP headers of all Web requests and responses.
  - Content of text and HTML objects.

LEHIGH

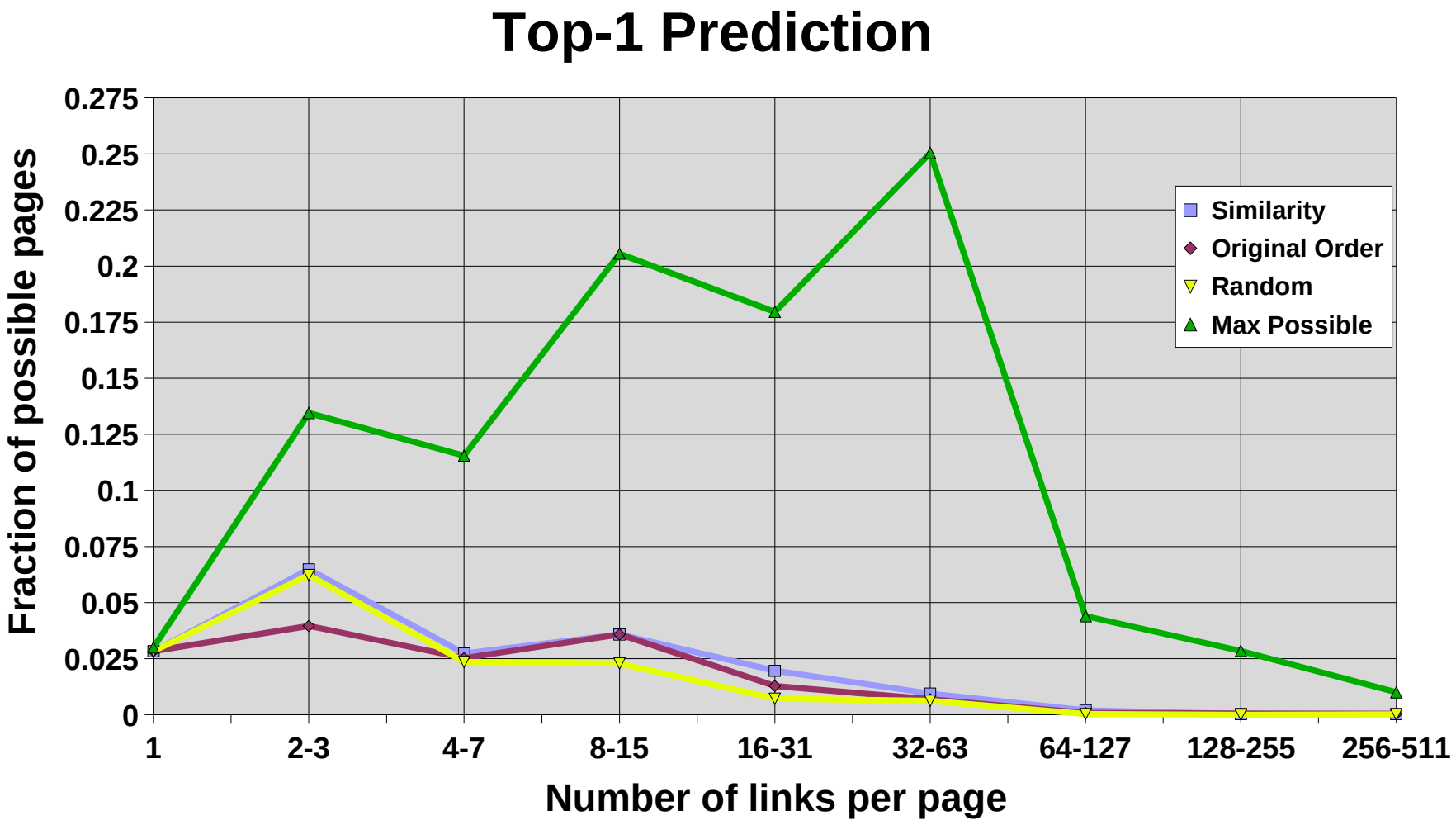
Non-prefetchable Content

- Many objects are not really prefetchable!
  - Uncacheable content.
  - Content whose retrieval causes side-effects.
- *Ideally these are handled by a revision to HTTP.*
- Most researchers identify URLs that look dynamic.
  - i.e., URLs with ? or cgi within the URL, or POST reqs.
  - Such URLs represent 28% of the pages in our dataset.
- In addition, when next URL is not in list of URLs, prediction scheme cannot succeed (another 47%).
- Remaining 25% may be predicted correctly.

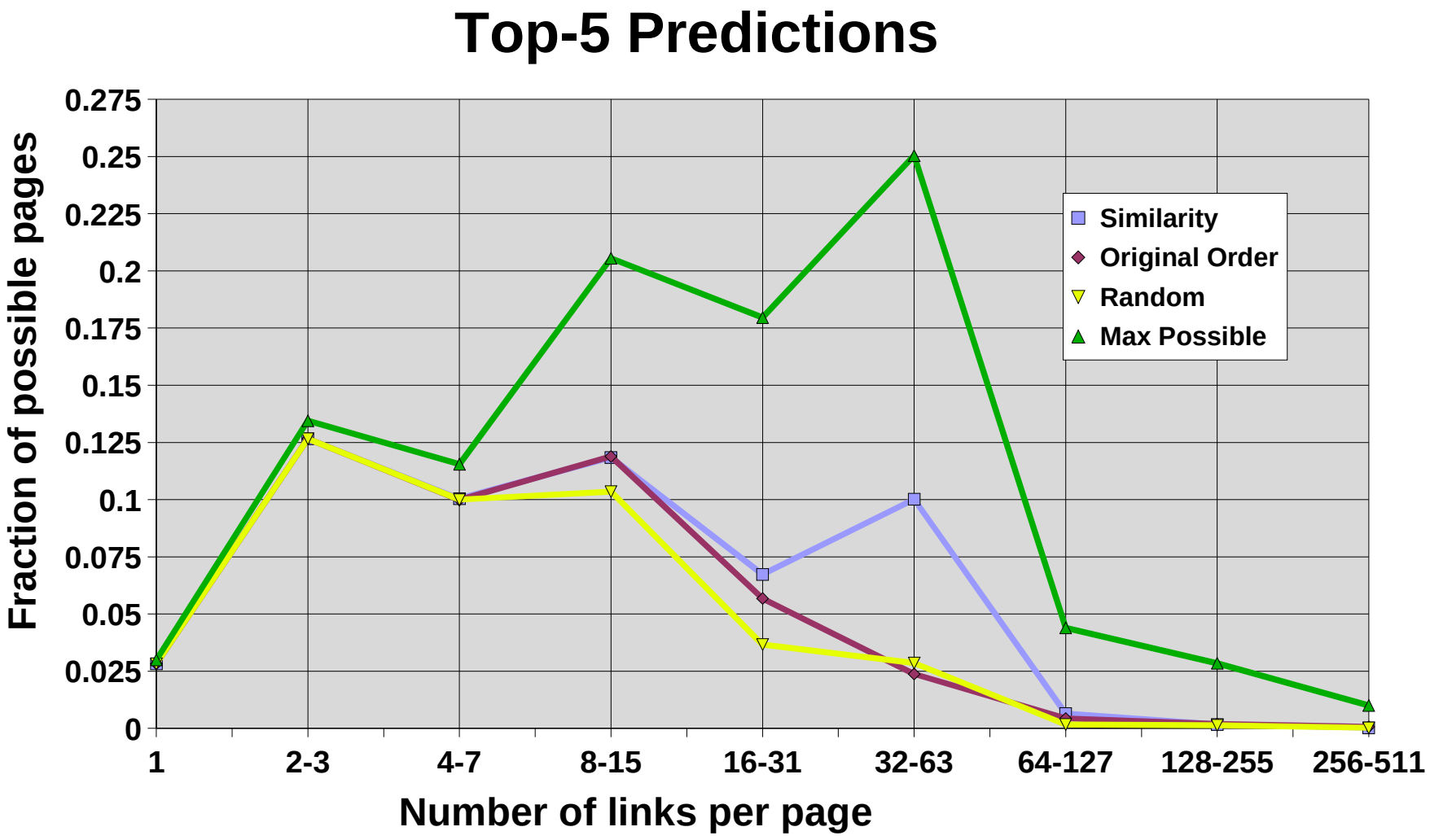
Experimental Results



When do approaches succeed?



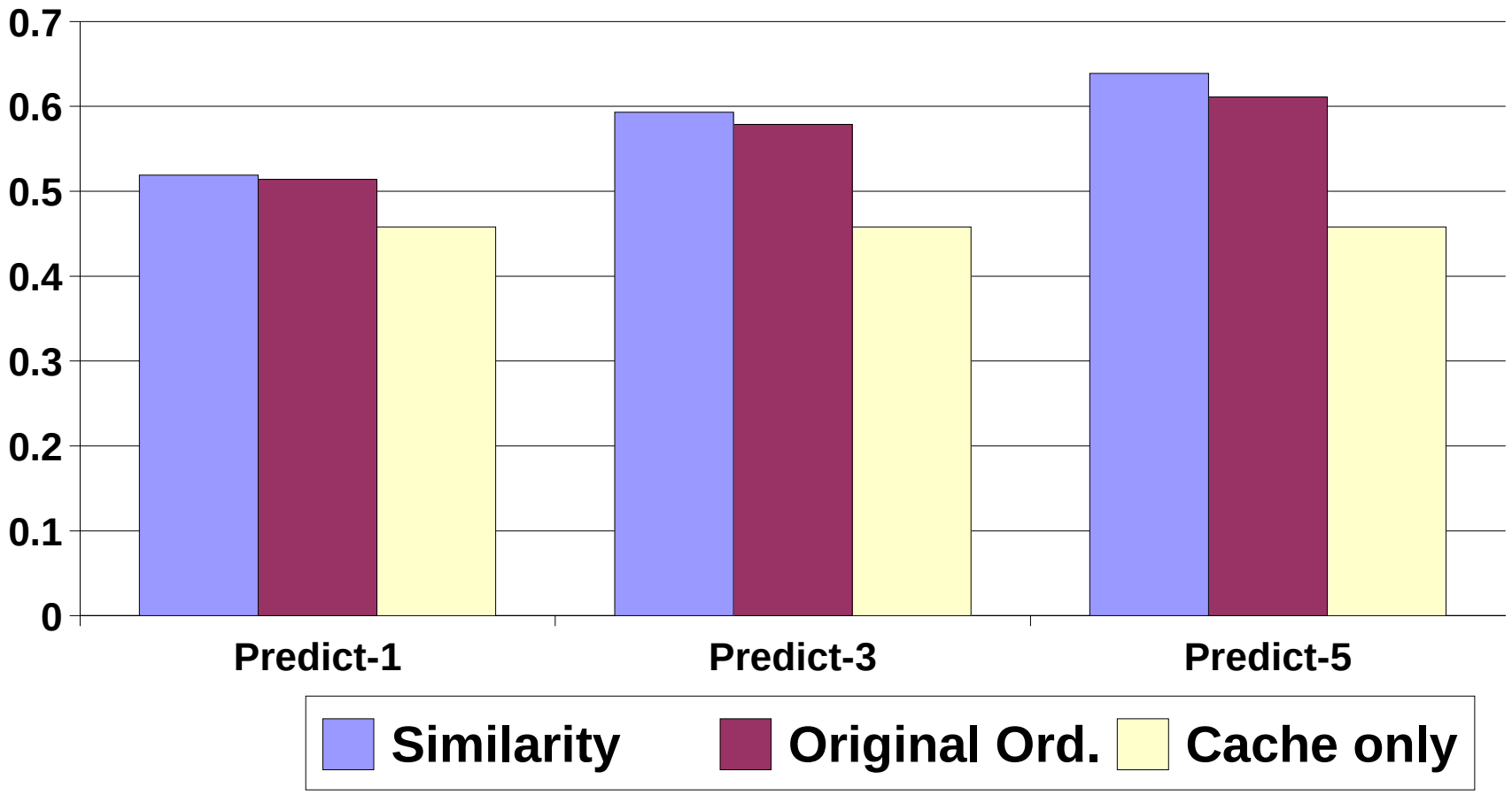
When do approaches succeed?



LEHIGH

When using an infinite cache

Predictive and/or cached performance over all pages.

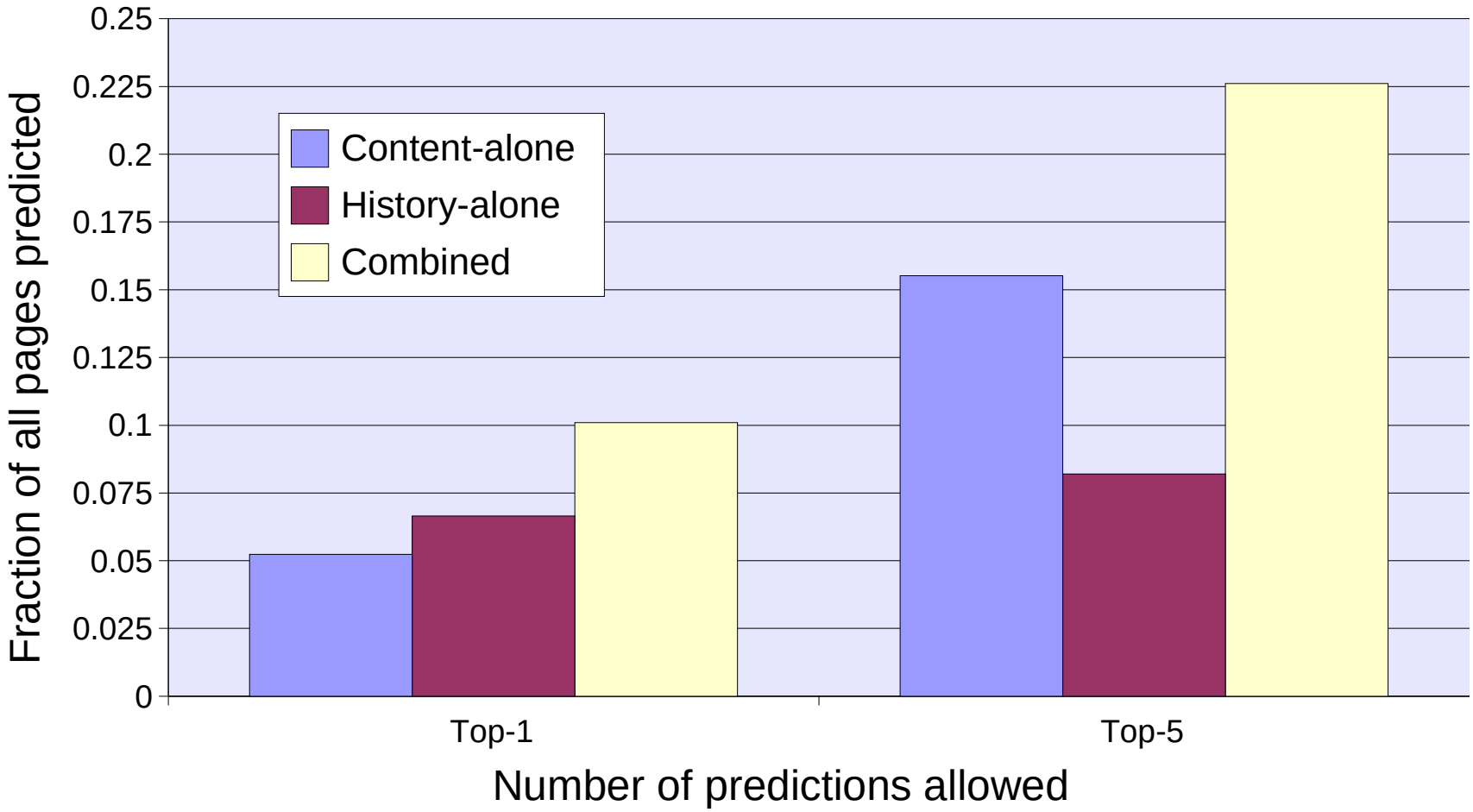


Discussion

- This talk discussed the prediction of Web page requests. Since we examine the content, it is trivial to prefetch the embedded resources as well.
- Caveat:
  - Small trace (few, mostly academic users)
- We believe:
  - Methods with stronger models of user interest are likely to perform better.
  - History-based methods are likely to perform better, when they have a sufficient model of past history.

LEHIGH

Combining History and Content



LEHIGH

Summary

- Primary results:
  - In the case of top-five predictions, similarity provided almost 30% improvement over random (without caching)
  - With an infinite cache, we are able to provide hits for 64% of all requests
    - (40% improvement over a non-prefetching system)
- When users view new pages, content-based methods are quite useful.
  - Users view new pages perhaps 40% of the time.
  - Certainly better than doing nothing!

LEHIGH

For more information

- Brian D. Davison

  davison@cse.lehigh.edu

  http://www.cse.lehigh.edu/~brian/

- Web Usage, Modeling, and Evaluation Lab

  http://wume.cse.lehigh.edu/

- Web Caching and Content Delivery Resources

  Tutorials, news, bibliography, tools, links

  http://www.web-caching.com/