Searching within structured, topic-focused web sites

Vince Tam & John Shepherd
CSE@UNSW
Context

- Systems to assist Question-Answering ...
  - e.g. help desk operators, course tutors, ...
- ... on (hierarchical) web sites devoted to one topic ...
  - e.g. technical documentation, lecture notes, assignments, ...
- ... needing to find specific info to answer a question
First-cut “Solution”

- feed question into Google, with site restriction
- will produce required pages in top 10 for QA’er
- but ...
  - Google doesn’t (seem to) index all pages within a site
  - query terms may not be present in relevant pages
Some Observations

- structured, topic-focused web sites tend to
  - use hierarchy (e.g. index pages + subordinates)
  - spread content for one topic across several pages
- examples: course web sites, documentation
- exploit this by considering page + neighbours to
  - extend set of terms for each page
  - compute augmented (improved) similarity score
Previous Work

- our work differs from previous work which attempts to use relationships between pages
  - link analysis ... authoritative pages (e.g. Kleinberg)
  - related pages ... similar content (e.g. Dean/Henzinger)
- above work treats pages as separate entities, and web as simply a large set of pages
- conjecture: exploiting the structure of topic-focused web sites allows us to incorporate info from nearby pages to improve search effectiveness (cf. Sugiyama)
Our Approach (summary)

- ignore “web-scale” aspects of search technology
- drive search via page + supplementary pages
- choose appropriate/best supp pages
- extend each page with terms from supp pages
- weight contribution of terms from supp pages
Terminology

- \( q = \) query (set of terms)
- \( Site = \) structured, topic-focussed web site
- \( D = \) any page examined for relevance
- \( T = \) target page (i.e. page relevant to query)
- \( N = \) page nearby \( D \) (a.k.a. neighbouring page)
- \( \text{supplementary page} = \) useful neighbouring page
Identifying Nearby Pages

- for strictly hierarchical web sites ... 
- use URL structure to identify parents/children/siblings 
  - http://mycourse/lectures/index.html
  - http://mycourse/lectures/week01/index.html
  - http://mycourse/lectures/week01/slide05.html
  - http://mycourse/lectures/week01/slide06.html
Identifying *Nearby Pages* (cont)

- **contextual path**: path from website root to page
- pages become more *general* as we approach root
- **sequential path**: ordered pages within a context
- e.g. *week7/slide01, week7/slide02, week7/slide03*
Identifying *Nearby* Pages (cont)

- for sites with no URL hierarchy (e.g. Wikis) ...
- use hyperlinks to identify “adjacent” pages
- with some effort, can often identify contextual and sequential paths
- since web-site structure is (relatively) static
  - can store *nearby page set* for each $D$
  - can store contextual/sequential path info
Augmented Similarity Score

$$Sim'(D,q) = Sim(D,q) \times (1 + Aug(D,q))$$

$$Aug(D,q) = \text{Avg}_{N \in \text{neighbours}(D)} S(N,D,q)$$

$$S(N,D,q) = \text{Supp}(N,D,q) \times Sim(N,q)$$

- measure $Sim(...)$ via standard similarity measure (0..1)
- augment similarity value by “incorporating” neighbours
- in fact, use only top-k $S(N,d,q)$ values for average
Computation of $Aug(D,q)$

- details of how $Aug(D,q)$ is calculated (we use $k=10$)

**Inputs:** $D$, $q$, $k$

supps = []

foreach $N$ in neighbours($D$) {
    s = $S(N,D,q)$
    if (s == 0) continue
    if (count(supps) > $k$ && s > min(supps))
        delete(supps, min(supps))
    insert(supps, s)
}

$Aug = \frac{\text{sum(supps)}}{\text{count(supps)}}$
Supplementary Power

\[ Supp(N,D,q) = \frac{\alpha Link + \beta BiDirect + \gamma Compl}{\alpha + \beta + \gamma} \]

- usefulness (power) of \( N \) as supplementary page for \( D \)
- how specific is the connection between \( N \) and \( D \)?
- are there bidirectional links between \( N \) and \( D \)?
- does content of \( N \) improve search term coverage?
- tuning parameters: \( \alpha, \beta, \gamma \)
Link/Specificity Measure

\[
Link(N, D) = \begin{cases} 
1 / inlinks(N) & \text{if } D \text{ links to } N \\
1 / outlinks(N) & \text{if } N \text{ links to } D \\
\text{Harmonic}(N) & \text{if 2-way links } N \leftrightarrow D 
\end{cases}
\]

- if more links into/out-of \( N \), \( N \) is less specific to \( D \)
- e.g. \( D \)'s parent links only to small number of pages
- e.g. \( D \)'s parent is an index with very many links
Bi-directionality Measure

\[ \text{BiDirect}(N, D) = \begin{cases} 
1 & \text{if bi-directional links between } N \text{ and } D \\
0 & \text{otherwise} 
\end{cases} \]

- pages that refer to each other are strongly related
- should carry more weight as supp pages
Complementarity Measure

\[ \text{Compl}(N, D, q) = \begin{cases} 
1 & \text{if } N \text{ contains query terms not in } D \\
0 & \text{otherwise} 
\end{cases} \]

- neighbouring pages that have terms from query
- if these terms do not exist in page \( D \)
- should have more weight as supp pages
Computing Matches

**Inputs:** Site, q, term index, k

```plaintext
res = []  // ordered list of (Page,Sim) pairs

foreach D in Site {
    s = Sim'(D,q)
    if (s < threshold) continue
    if (count(res) >= k && s > minSim(res))
        delete(res, minSim(res))
        insertInSimOrder(res, (D, s))
}
```

**Outputs:** list of “relevant” pages in Sim order
Computing Matches  (cont)

- if done naively, above computation is very expensive
- pre-computation is possible for some values ...

  - index mapping vocab terms to pages
    - augmented by including terms from neighbours
  - mapping tables for $Link(N,D)$ and $BiDirect(N,D)$

- compute $Sim(P,q)$ for all $P$, then cache $Sim$ values
Evaluation

- vector-space/tfidf/cosine similarity measure
- three data sets
  - two UNSW course web sites (similar structures)
    - each with ~2000 pages, strongly hierarchical
  - using 30 queries derived from course forums
  - TREC Enterprise 2007 data (via CSIRO)
    - with 350K documents (4GB), some hierarchy
  - using the 50 topics/queries from the TREC track
Evaluation (cont)

- measured prec@5 for query sets
- goal is to have relevant pages ranked close to top

<table>
<thead>
<tr>
<th>prec@5</th>
<th>CSIRO</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim(D)</td>
<td>0.60</td>
<td>0.28</td>
</tr>
<tr>
<td>Sim(D+N)</td>
<td>0.59</td>
<td>0.28</td>
</tr>
<tr>
<td>Sim'(D) $\alpha = \beta = \gamma = 1$</td>
<td>0.63</td>
<td>0.32</td>
</tr>
<tr>
<td>Sim'(D) $\alpha = \beta = 1, \gamma = 2$</td>
<td>0.63</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Sunday, 26 December 2010
Conclusions, ToDo, Q+A

- experiments promising but not conclusive
- other aspects to consider/expand
  - additional “linkage factors” between pages
  - further tuning of weights
  - using more distant neighbours
- using hierarchy implicit in chapter/section/paragraph?
- any (more) questions?