Integrating the Document Object Model with Hyperlinks for Enhanced Topic Distillation and Information Extraction[*]

Soumen Chakrabarti
Indian Institute of Technology Bombay

`http://www.cse.iitb.ernet.in/~soumen`

ABSTRACT

Topic distillation is the process of finding authoritative Web pages and comprehensive "hubs" which reciprocally endorse each other and are relevant to a given query. Hyperlink-based topic distillation has been traditionally applied to a macroscopic Web model where documents are nodes in a directed graph and hyperlinks are edges. Macroscopic models miss valuable clues such as banners, navigation panels, and template-based inclusions, which are embedded in HTML pages using markup tags. Consequently, results of macroscopic distillation algorithms have been deteriorating in quality as Web pages are becoming more complex. We propose a uniform fine-grained model for the Web in which pages are represented by their tag trees (also called their Document Object Models or DOMs) and these DOM trees are interconnected by ordinary hyperlinks. Surprisingly, macroscopic distillation algorithms do not work in the fine-grained scenario. We present a new algorithm suitable for the fine-grained model. It can *dis-aggregate* hubs into coherent regions by segmenting their DOM trees. Mutual endorsement between hubs and authorities involve these regions, rather than single nodes representing complete hubs. Anecdotes and measurements using a 28-query, 366000-document benchmark suite, used in earlier topic distillation research, reveal two benefits from the new algorithm: distillation quality improves, and a by-product of distillation is the ability to extract relevant *snippets* from hubs which are only partially relevant to the query.

Keywords: Topic distillation, Document Object Model, segmentation, Minimum Description Length principle.

1 Introduction

Kleinberg's Hyperlink Induced Topic Search (HITS) [14] and the PageRank algorithm [3] underlying Google have revolutionized ranking technology for Web search engines. PageRank evaluates the "prestige score" of a page as roughly proportional to the sum of prestige scores of pages citing it using hyperlinks. HITS also identifies collections of resource links or "hubs" densely coupled to authoritative pages on a topic. The model of the Web underlying these and related systems is a directed graph with pages (HTML files) as nodes and hyperlinks as edges.

Since those papers were published, the Web has been evolving in fascinating ways, apart from just getting larger. Web pages are changing from static files to dynamic views generated from complex templates and backing semi-structured databases. A variety of hypertext-specific idioms such as navigation panels, advertisement banners, link exchanges, and Web-rings, have been emerging.

There is also a migration of Web content from syntactic HTML markups towards richly tagged, semi-structured XML documents (`http://www.w3.org/XML/`) interconnected at the XML element level by semantically rich links (see, e.g., the XLink proposal at `http://www.w3.org/TR/xlink/`). These refinements are welcome steps to implementing what Berners-Lee and others call the *semantic Web* (`http://www.w3.org/1999/04/13-tbl.html`), but result in document, file, and site boundaries losing their traditional significance.

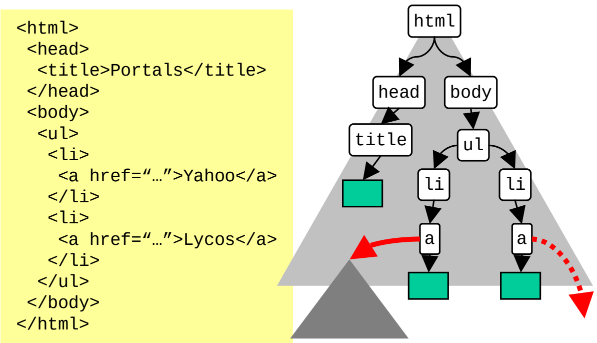
Continual experiments performed by several researchers [2, 15] reveal a steady deterioration of distillation quality through the last few years. In our experience, poor results are frequently traced to the following causes:

- Links have become more frequent and "noisy" from the perspective of the query, such as in banners, navigation panels, and advertisements. Noisy links do not carry human editorial endorsement, a basic assumption in topic distillation.
- Hubs may be "mixed", meaning only a portion of the hub may be relevant to the query. Macroscopic distillation algorithms treat whole pages as atomic, indivisible nodes with no internal structure. This leads to false reinforcements and resulting contamination of the query responses.

Thanks in part to the visibility of Google, content creators are well aware of hyperlink-based ranking technology. One reaction has been the proliferation of nepotistic "clique attacks"—a collection of sites linking to each other without semantic reason, e.g. `http://www.411fun.com`, `http://www.411fashion.com` and `http://www.411-loans.com`. (Figures 8 and 9 provide some examples.) Some examples look suspiciously like a conscious attempt to spam search engines that use link analysis. Interestingly, in most cases, the visual presentation clearly marks noisy links which surfers rarely follow, but macroscopic algorithms are unable to exploit it.

[*] (Note: To view the HTML version using Netscape, add the following line to your `~/.Xdefaults` or `~/.Xresources` file:
`Netscape*documentFonts.charset*adobe-fontspecific: iso-8859-1`
For printing use the PDF version, as browsers may not print the mathematics properly.)

WWW10, May 1–5, 2001, Hong Kong.
ACM 1-58113-348-0/01/0005.



```
<html>
 <head>
  <title>Portals</title>
 </head>
 <body>
  <ul>
   <li>
    <a href="…">Yahoo</a>
   </li>
   <li>
    <a href="…">Lycos</a>
   </li>
  </ul>
 </body>
</html>
```

Figure 1: In the fine-grained model, DOMs for individual pages are trees interconnected by ordinary hyperlinks. Each triangle is the DOM tree corresponding to one HTML page. Green boxes represent text.

Many had hoped that HITS-like algorithms would put an end to spamming, but clearly the situation is more like an ongoing arms-race. Google combines link-based ranking with page text and anchor text in undisclosed ways, and keeps tweaking the combination, but suffers an occasional embarrassment[1].

Distillation has always been observed to work well for "broad" topics (for which there exist well-connected relevant Web subgraphs and "pure" hubs) and not too well for "narrow" topics, because w.r.t. narrow topics most hubs are mixed and have too many irrelevant links. Mixed hubs and the arbitrariness of page boundaries have been known to produce glitches in the Clever system [6]: there has been no reliable way to classify hubs as mixed or pure. If a fine-grained model can suitably *dis-aggregate* mixed hubs, distillation should become applicable to narrow queries too.

Yet another motivation for the fine-grained model comes from the proliferation of mobile clients such as cell-phones and PDAs with small or no screens. Even on a conventional Web browser, scrolling through search results for promising responses, then scrolling through those responses to satisfy a specific information need are tedious steps. The tedium is worse on mobile clients. Search engines that need to serve mobile clients must be able to pinpoint narrow sections of pages and sites that address a specific information need, and limit the amount of extra matter sent back to the client [4].

1.1 Our contributions

We initiate a study of topic distillation with a fine-grained model of the Web, built using the Document Object Model (DOM) of HTML pages. The DOM can model reasonably clean HTML, support XML documents that adhere to rigid schema definitions, and embed free text in a natural way. In our model, HTML pages are represented by their DOMs and these DOM trees are interconnected by ordinary hyperlinks (figure 1). The sometimes artificial distinction between Web-level, site-level, page-level, and intra-page structures is thereby blurred. Surprisingly, macroscopic distillation algorithms perform poorly in the fine-grained setting; we demonstrate this using analysis and anecdotes. Our main technical contribution is a new fine-grained distillation algorithm which can identify mixed hubs and segment their corresponding DOM trees into maximal subtrees which are "coherent" w.r.t. the query, i.e., each is almost completely relevant or completely irrelevant. The segmentation algorithm uses the Minimum Description Length (MDL) principle [16] from Information Theory [9]. Rather than collapse these diverse hub subtrees into one node, the new algorithm allocates a node for each subtree. This intermediate level of detail, between the macroscopic and the fine-grained model, is essential to the success of our algorithm. We report on experiments with 28 queries involving over 366000 Web pages. This benchmark has been used in previous research on resource compilation and topic distillation [5, 2, 6]. Our experience is that the fine-grained model and algorithm significantly improve the quality of distillation, and are capable of extracting DOM subtrees from mixed hubs that are relevant to the query.

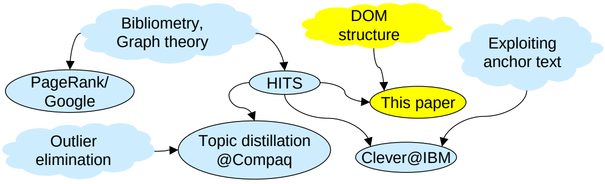


Figure 2: This work in the context of HITS and related research.

We note that in this study we have carefully and deliberately isolated the model from possible influences of text analysis. By controlling our experimental environment to not use text, we push HITS-like ideas to the limit, evaluating exactly the value added by information present in DOM structures. In ongoing work, we have added textual support to our framework and obtained even better results [7].

1.2 Benefits and applications

Apart from offering a more faithful model of Web content, our approach enables solutions to the following problems.

Better topic distillation: We show less tendency for topic drift and contamination when the fine-grained model is used.

Web search using devices with small or no screen: The ability to identify page snippets relevant to a query is attractive to search services suitable for mobile clients.

Focused crawling: Identification of relevant DOM subtrees can be used to better guide a focused crawler's link expansion [8].

Annotation extraction: Experiments with a previous macroscopic distillation algorithm (Clever [6]) revealed that volunteers preferred Clever to Yahoo! only when Yahoo!'s manual site annotations were removed in a blind test. Our work may improve on current techniques for automatic annotation extraction [1] by first collecting candidate hub page fragments and then subjecting the text therein to further segmentation techniques.

Data preparation for linguistic analysis: Information extraction is a natural next step after resource discovery. It is easier to build extractors based on statistical and linguistic models if the domain or subject matter of the input documents is suitably segmented [12], as is effected by our hub subtree extraction technique, which is a natural successor to resource discovery, and a precursor to linguistic analysis.

[1] `http://searchenginewatch.com/sereport/99/11-google.html` (local copy `GoogleDrEvil.html`) and `http://searchenginewatch.com/sereport/01/02-bush.html` (local copy `GoogleBush.html`) provide some samples.

1.3 Outline of the paper

In §2.1 we review HITS and related algorithms. This section can be skipped by a reader who is familiar with HITS-related literature. In §2.2 we illustrate some recent and growing threats to the continued success of macroscopic distillation algorithms. We show why the fine-grained model does not work with traditional HITS-like approaches in §3, and then propose our framework in §4. We report on experimental results in §5 and conclude in §6 with some comments on ongoing and future work.

2 Preliminaries

We review the HITS family of algorithms and discuss how they were continually enhanced to address evolving Web content.

2.1 Review of HITS and related systems

The HITS algorithm [14] started with a query q which was sent to a text search engine. The returned set of pages R_q was fetched from the Web, together with any pages having a link to any page in R_q, as well as any page cited in some page of R_q using a hyperlink. Links that connected pages on the same Web server (based on canonical host name match) were dropped from consideration because they were often seen to serve only a navigational purpose, or were "nepotistic" in nature.

Suppose the resulting graph is $G_q = (V_q, E_q)$. We will drop the subscript q where clear from context. Each node v in V is assigned two scores: the *hub score* $h(v)$ and the *authority score* $a(v)$, initialized to any positive number. Next the HITS algorithm alternately updates **a** and **h** as follows: $a(v) = \sum_{(u,v)\in E} h(u)$ and $h(u) = \sum_{(u,v)\in E} a(v)$, making sure after each iteration to scale **a** and **h** so that $\sum_v h(v) = \sum_v a(v) = 1$, until the ranking of nodes by a and h stabilize (see figure 3).

If E is represented in the adjacency matrix format (i.e., $E[i,j] = 1$ if there is an edge (i,j) and 0 otherwise) then the above operation can be written simply as $\mathbf{a} = E^T\mathbf{h}$ and $\mathbf{h} = E\mathbf{a}$, interspersed with scaling to set $|\mathbf{h}|_1 = |\mathbf{a}|_1 = 1$. The HITS algorithm effectively uses power iterations [11] to find **a**, the principal eigenvector of E^TE; and **h**, the principal eigenvector of EE^T. Pages with large a are popular or authoritative sources of information; pages with large h are good collections of links.

A key feature of HITS is how endorsement or popularity diffuses to siblings. If (u,v) and (u,w) are edges and somehow $a(v)$ becomes large, then in the next iteration $h(u)$ will increase, and in the following iteration, $a(w)$ will increase. We will describe this as "v's authority diffuses to w through the hub u." This is how sibling nodes reinforce each other's authority scores. We will revisit this property later in §3.

Google has no notion of hubs. Roughly speaking, each page v has a single "prestige" score $p(v)$ called its *PageRank* [3] which is defined as proportional to $\sum_{(u,v)\in E} p(u)$, the sum of prestige scores of pages u that cite v. Some conjecture that the prestige model is adequate for the living Web, because good hubs readily acquire high prestige as well. Our work establishes the value of a bipartite model like HITS, and indeed, the value of an *asymmetric* model where hubs are analyzed quite differently from authorities. Therefore we will not discuss prestige-based models any further.

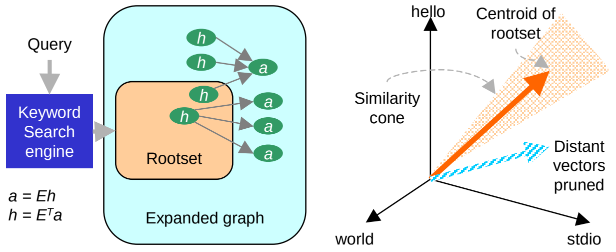


Figure 3: (a) HITS, a macroscopic topic distillation algorithm with uniform edge weights; (b) The B&H algorithm, apart from using non-uniform edge weights, discards pages in the expanded set which are too dissimilar to the rootset pages to prevent topic drift. Documents are represented as vectors with each component representing one token or word [17].

2.2 The impact of the evolving Web on hyperlink analysis

Elegant as the HITS model is, it does not adequately capture various idioms of Web content. We discuss here a slew of follow-up work that sought to address these issues.

Kleinberg dropped links within the same Web-site from consideration because these were often found to be navigational, "nepotistic" and noisy. Shortly after HITS was published, Bharat and Henzinger (B&H [2]) found that nepotism was not limited to same-site links. In many trials with HITS, they found two distinct sites s_1 and s_2, where s_1 hosted a number of pages u linking to a page v on s_2, driving up $a(v)$ beyond what may be considered fair. B&H proposed a simple and effective fix for such "site-pair" nepotism: if k pages on s_1 point to v, let the weight of each of these links be $1/k$, so that they add up to one, assuming a site (not a page) is worth one unit of voting power.

Later work in the Clever system [6] used a small edge weight for same-site links and a larger weight for other links, but these weights were tuned empirically by evaluating the results on specific queries.

Another issue with HITS were "mixed hubs" or pages u that included a collection of links of which only a subset was relevant to a query. Because HITS modeled u as a single node with a single h score, high authority scores could diffuse from relevant links to less relevant links. E.g., responses to the query *movie awards* sometimes drifted into the neighboring, more densely linked domain of *movie companies.*

Later versions of Clever tried to address the issue in two ways. First, links within a fixed number of tokens of query terms were assigned a large edge weight (the width of the "activation window" was tuned by trial-and-error). Second, hubs which were "too long" were segmented at a few prominent boundaries (such as `<UL>` or `<HR>`) into "pagelets" with their own scores. The boundaries were chosen using a static set of rules depending on the markup tags on those pages alone.

To avoid drift, B&H also computed a vector space representation [17] of documents in the response set (shown in Figure 3) and then dropped pages that were judged to be "outliers" using a suitable threshold of (cosine) similarity to the vector space centroid. B&H is effective for improving precision, but may reduce recall if mixed hubs are pruned because of small similarity to the root set centroid. This

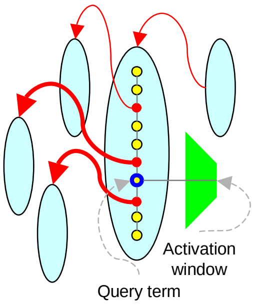


Figure 4: Clever uses a slightly more detailed page model than HITS. Hyperlinks near query terms are given heavier weights. Such links are shown as thicker lines.

may in turn distort hub and authority scores and hence the desired ranking. Losing a few hubs may not be a problem for broad queries but could be serious for narrower queries.

As resource discovery and topic distillation become more commonplace, we believe the quest will be for every additional resource than can possibly be harvested, not merely the ones that "leap out at the surfer." Our goal should therefore be to extract relevant links and annotations even from pages which are partially or largely irrelevant.

3 Generalizing hyperlinks to interconnected DOMs

HTML documents have always embedded many sources of information (other that text) which have been largely ignored in previous distillation research. Markups are one such source. From a well-formed HTML document, it ought to be possible to extract a tree structure called the Document Object Model (DOM). In real life HTML is rarely well formed, but using a few simple patches, it is possible to generate reasonably accurate DOMs. For XML sources adhering to a published DTD, a DOM is precise and well defined.

For simplicity, we shall work with a greatly pared-down version of the DOM for HTML pages. We will discard all text, and only retain those paths in the DOM tree that lead from the root to a leaf which is an `<A...>` element with an `HREF` leading to another page.

Hyperlinks always originate from leaf DOM elements, typically deep in the DOM tree of the source document. If same-site links are ignored, very few macro-level hyperlinks target an internal node in a DOM tree (using the "`#`" modifier in the URL). To simplify our model (and experiments) we will assume that the target of a hyperlink is always the root node of a DOM tree. In our experiments we found very few URLs to be otherwise.

A first-cut approach (which one may call *MicroHITS*) would be to use the fine-grained graph directly in the HITS algorithm. One may even generalize "same-site" to "same-DOM" and use B&H-like edge-weights. This approach turns out to work rather poorly.

To appreciate why, consider two simple example graphs shown in Figure 5 and their associated eigenvectors. The first graph is for the macro setting. Expanding out $\mathbf{a} \leftarrow E^T E\mathbf{a}$ we get

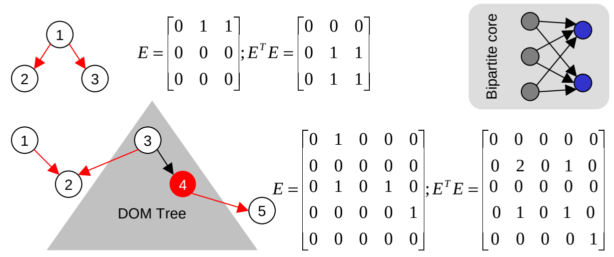
$$a(2) \leftarrow a(2) + a(3) \quad \text{and}$$
$$a(3) \leftarrow a(2) + a(3),$$

which demonstrates the mutual reinforcement. In the second example nodes numbered 3 and 4 are part of one DOM tree. This time, we get

$$a(2) \leftarrow 2a(2) + a(4) \quad \text{and}$$
$$a(4) \leftarrow a(2) + a(4),$$

but there is no coupling between $a(2)$ and $a(5)$, which we would expect at the macroscopic level. Node 4 (marked red) effectively *blocks* the authority from diffusing between nodes 2 and 5.

One may hope that bigger DOM trees and multiple paths to authorities might alleviate the problem, but the above example really depicts a basic problem. The success of HITS depends critically on reinforcement among *bipartite cores* (see figure 5) which may be destroyed by the introduction of fine-grained nodes.



$$E = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; E^T E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; E^T E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 5: A straight-forward application of HITS-like algorithms to a DOM graph may result in some internal DOM nodes blocking the diffusion of authority across siblings.

4 Proposed model and algorithm

At this point the dilemma is clear: by collapsing hubs into one node, macroscopic distillation algorithms lose valuable detail, but the more faithful fine-grained model prevents bipartite reinforcement.

In this section we present our new model and distillation algorithm that resolves the dilemma. Informally, our model of hub generation enables our algorithm to find a *cut* or *frontier* across each hub's DOM tree. Subtrees attached to these cuts are made individual nodes in the distillation graph. Thus the hub score of the entire page is *dis-aggregated* at this intermediate level. The frontiers are not computed one time as a function of the page alone, neither do they remain unchanged during the HITS iterations. The frontiers are determined by the current estimates of the hub scores of the leaf `HREF` nodes.

We will first describe the hub segmentation technique and then use it in a modified iterative distillation algorithm.

4.1 Scoring internal micro-hub nodes

Macroscopic distillation algorithms rank and report complete hub pages, even if they are only partially relevant. In this section we address the problem of estimating the hub score of each DOM node in the fine-grained graph, given an estimate of authority scores. Because inter-page hyperlinks originate in leaf DOM nodes and target root nodes of DOM trees, we will also assume that only those DOM nodes that are document roots can have an authority score.

At the end of the $\mathbf{h} \leftarrow E\mathbf{a}$ substep of MicroHITS, leaf DOM nodes get a hub score. Because leaf nodes point to exactly one page via an HREF, the hub score is exactly the authority score of the target page. Macroscopic distillation algorithms in effect aggregate all the leaf hub scores for a page into one hub score for the entire page. Reporting leaf hub scores in descending order would be useless, because they would simply follow the authority ranking and fail to identify good hub aggregates.

Instead of the total hub score, one may consider the *density* of hub scores in a subtree, which may be defined as the total hub score in the subtree divided by the number of HREF leaves. The maximum density will be achieved by the leaf node that links to the best authority. In our experience small subtrees with small number of leaves dominate the top ranks, again under-aggregating hub scores and pitting ancestor scores against descendant scores.

4.1.1 A generative model for hubs

To help us find suitable frontiers along which we can aggregate hub scores, we propose the following generative model for hubs.

Imagine that the Web has stopped changing and with respect to a fixed query, *all* Web pages have been manually rated for their worth as hubs. From these hub scores, one may estimate that the hub scores have been generated from a distribution Θ_0. (E.g., Θ_0 may represent an exponential distribution with mean 0.005.) If the author of a hub page sampled URLs at random to link to, the distribution of hub scores at the leaves of the page would approach the global distribution provided enough samples were taken.

However, authors differ in their choice of URLs. Hub authors are not aware of all URLs relevant to a given query or their relative authority; otherwise all hubs authored on a topic would be complete and identical, and therefore all but one would be pointless to author. (Here we momentarily ignore the value added by annotations and commentaries on hub pages.)

Therefore, the distribution of hub scores for pages composed by a specific author will be different from Θ_0. (E.g., the author's personal average of hub scores may be 0.002, distributed exponentially.) Moreover, the authors of mixed hubs deliberately choose to dedicate not the entire page, but only a fragment or subtree of it, to URLs that are relevant to the given query. (As an extreme case a subtree could be a single HREF.)

We can regard the hub generation process as a progressive specialization of the hub score distribution starting from the global distribution. For simplicity, assume all document roots are attached to a "super-root" which corresponds to the global distribution Θ_0. As the author works down the DOM tree, "corrections" are applied to the score distribution at nodes on the path.

At some suitable depth, the author fixes the score distribution and generates links to pages so that hub scores follow that distribution. This does not mean that there are no interesting DOM subtrees below this depth. The model merely posits that up to some depth, DOM structure is indicative of *systematic* choices of score distributions, whereas beyond that depth variation is *statistical*.

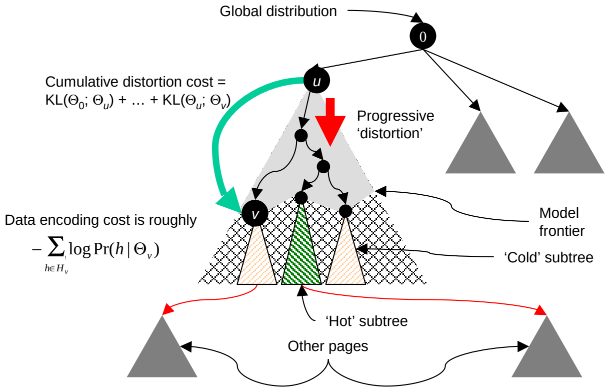

Global distribution
0
u
Cumulative distortion cost =
KL(Θ0; Θu) + … + KL(Θu; Θv)
Progressive
'distortion'
v
Data encoding cost is roughly
$-\sum_{h \in H_v} \log \Pr(h \mid \Theta_v)$
Model
frontier
'Cold' subtree
'Hot' subtree
Other pages

Figure 6: Our fine-grained model of Web linkage which unifies hyperlinks and DOM structure.

4.1.2 Discovering DOM frontiers from generated hubs

During topic distillation we observe pages which are the outcome of the generative process described above, and our goal is to discover the "best" frontier at which the score distributions were likely to have been fixed.

A balancing act is involved here: one may choose a large and redundant frontier near the leaves and model the many small, homogeneous subtrees (each with a different distribution Θ_w) attached to that frontier accurately, or one may choose a short frontier near the root with a few subtrees which are harder to model because they contain diverse hub scores. The balancing act requires a common currency to compare the cost of the frontier with the cost of modeling hub score data beneath the frontier.

This is a standard problem in segmentation, clustering, and model estimation. A particularly successful approach to optimizing the trade-off is to use the Minimum Description Length (MDL) principle [16]. MDL provides a recipe for bringing the cost of model corrections to the same units as the cost for representing data w.r.t a model, and postulates that "learning" is equivalent to minimizing the sum total of model and data encoding costs.

Data encoding cost: First we consider the cost of encoding all the h-values at the leaves of a subtree rooted at node w. Specifically, let the distribution associated with w be Θ_w. The set of HREF leaf nodes in the subtree rooted at node w is denoted L_w, and the set of hub scores at these leaves is denoted H_w. As part of the solution we will need to evaluate the number of bits needed to encode h-values in H_w using the model Θ_w. There are efficient codes which can achieve a data encoding length close to Shannon's entropy-based lower bound [9] of

$$-\sum_{h \in H_w} \log \Pr_{\Theta_w}(h) \quad \text{bits}, \qquad (1)$$

where $\Pr_{\Theta_w}(h)$ is the probability of hub score h w.r.t. a distribution represented by Θ_w. (E.g., Θ_w may include the mean and variance of a normal distribution.) We will use this lower bound as an approximation to our data encoding cost. (This would work if the h-values followed a discrete probability distribution, which is not the case with hub scores. We will come back to this issue in §4.2.)

Model encoding cost: Next we consider the model encoding cost. Consider node v in the DOM tree. We will assume that Θ_0 is known to all, and use the path from the global root to v to inductively encode each node w.r.t its parent. Suppose we want to specialize the distribution Θ_v of some v away from Θ_u, the distribution of its parent u. The cost for specifying this change is given by the well-known Kullback-Leibler (KL) distance [9] $KL(\Theta_u; \Theta_v)$, expressed as

$$KL(\Theta_u; \Theta_v) = \sum_x \Pr_{\Theta_u}(x) \log \frac{\Pr_{\Theta_u}(x)}{\Pr_{\Theta_v}(x)}. \qquad (2)$$

Intuitively, this is the cost of encoding the distribution Θ_v w.r.t. a reference distribution Θ_u. E.g., if X is a binary random variable and its probabilities of being zero and one are $(.2, .8)$ under Θ_1 and $(.4, .6)$ under Θ_2, then $KL(\Theta_2; \Theta_1) = .4 \log \frac{.4}{.2} + .6 \log \frac{.6}{.8}$. Unlike in the case of entropy, the sum can be taken to an integral in the limit for a continuous variable x. Clearly for $\Theta_u = \Theta_v$, the KL distance is zero; it can also be shown that this is a necessary condition, and that the KL distance is asymmetric in general but always non-negative.

If Θ_u is specialized to Θ_v and Θ_v is specialized to Θ_w, the cost is additive, i.e., $KL(\Theta_u; \Theta_v) + KL(\Theta_v; \Theta_w)$. We will denote the cost of such a path as $KL(\Theta_u; \Theta_v; \Theta_w)$. Moreover, the model encoding cost of v starting from the global root model will be denoted $KL(\Theta_0; \ldots; \Theta_v)$.

Combined optimization problem: Given the model at the parent node Θ_u and the observed data H_v, we should choose Θ_v so as to minimize the sum of the KL distance and data encoding cost:

$$KL(\Theta_v; \Theta_u) - \sum_{h \in H_v} \log \Pr{}_{\Theta_v}(h). \qquad (3)$$

If Θ_v is expressed parametrically, this will involve an optimization over those parameters.

With the above set-up, we are looking for a cut or frontier F across the tree, and for each $v \in F$, a Θ_v, such that

$$\sum_{v \in F} \left(KL(\Theta_0; \ldots; \Theta_v) - \sum_{h \in H_v} \log \Pr{}_{\Theta_v}(h) \right) \qquad (4)$$

is minimized. The first part expresses the total model encoding cost of all nodes v on the frontier F starting from the global root distribution. The second part corresponds to the data encoding cost for the set of hub scores H_v at the leaves of the subtrees rooted at the nodes v. Figure 6 illustrates the two costs.

4.2 Practical considerations

The formulation above is impractical for a number of reasons. There is a reduction from the knapsack problem to the frontier-finding problem. Dynamic programming can be used to give close approximations [13, 18], but with tens of thousands of macro-level pages, each with hundreds of DOM nodes, something even simpler is needed. We describe the simplifications we had to make to control the complexity of our algorithm.

We use the obvious greedy expansion strategy. We initialize our frontier with the global root and keep picking a node u from the frontier to see if expanding it to its immediate children $\{v\}$ will result in a reduction in code length, if so we replace u by its children, and continue until no further improvement is possible. We compare two costs locally at each u:

- The cost of encoding all the data in H_u with respect to model Θ_u.
- The cost of expanding u to its children, i.e., $\sum_v KL(\Theta_u; \Theta_v)$, plus the cost of encoding the subtrees H_v with respect to Θ_v.

If the latter cost is less, we expand u, otherwise, we prune it, meaning that u becomes a frontier node.

Another issue is with optimizing the model Θ_v. Usually, closed form solutions are rare and numerical optimization must be resorted to; again impractical in our setting.

In practice, if H_v is moderately large, the data encoding cost tends to be larger than the model cost. In such cases, a simple approximation which works quite well is to first minimize the data encoding cost for H_v by picking parameter values for Θ_v that maximize the probability of the observed data (the "maximum likelihood" or ML parameters), thus fix Θ_v, then evaluate $KL(\Theta_u; \Theta_v)$.

(As an example, if a coin tossed n times turns up heads k times, the ML parameter for bias is simply k/n, but if a uniform $\Theta_u = \mathcal{U}(0, 1)$ is chosen, the mean of Θ_v shifts slightly to $(k + 1)/(n + 2)$ which is a negligible change for moderately large k and n.)

Non-parametric evaluation of the KL distance is complicated, and often entails density estimates. We experimented with two parametric distributions: the Gaussian and exponential distributions for which the KL distance has closed form expressions. We finally picked the exponential distribution because it fit the observed hub score distribution more closely.

If Θ represents an exponential distribution with mean μ and probability density $f(x) = (1/\mu) \exp(-x/\mu)$, then

$$KL(\Theta_1; \Theta_2) = \log \frac{\mu_2}{\mu_1} + \left(\frac{\mu_1}{\mu_2} - 1 \right), \qquad (5)$$

where μ_i corresponds to Θ_i $(i = 1, 2)$.

The next issue is how to measure data encoding cost for continuous variables. There is a notion of the relative entropy of a continuous distribution which generalizes discrete entropy, but the relative entropy can be negative and is useful primarily for comparing the information content in two signal sources. Therefore we need to discretize the hub scores.

A common approach to discretizing real values is to scale the smallest value to one, in effect allocating $\log(h_{\max}/h_{\min})$ bits per value. This poses a problem in our case. Consider the larger graph in figure 5. If $\mathbf{h}$ is initialized to $(1, 1, 1, 1, 1)^T$, after the first few multiplications by EE^T which represents the linear transformation

$$(h(1), \ldots, h(5))^T \to (h(1) + h(3), 0, h(1) + 2h(3), h(4), 0)^T,$$

we get $(2, 0, 3, 1, 0)^T$, $(5, 0, 8, 1, 0)^T$, $(13, 0, 21, 1, 0)^T$, and $(34, 0, 55, 1, 0)^T$. Even if we disregard the zeroes, the ratio of the largest to the smallest positive component of $\mathbf{h}$ grows without bound. As scaling is employed to prevent overflow, $h(4)$ decays towards zero. This makes the $\log(h_{\max}/h_{\min})$ strategy useless.

A reasonable compromise is possible by noting that the user is not interested in the precision of *all* the hub scores. E.g., reporting the top α fraction of positive hub scores to within a small multiplicative error of ϵ is quite enough. We used $\alpha = 0.8$ and $\epsilon = 0.05$ in our experiments.

4.3 Distillation using segmented hubs

In this section we will embed the segmentation algorithm discussed in the previous section into the edge-weighted B&H algorithm. (Unlike the full B&H algorithm, we do no text analysis at this stage. We continue to call the edge-weighted version of HITS as "B&H" for simplicity.)

The main modification will be the insertion of a call to the segmentation algorithm after the $\mathbf{h} \leftarrow E\mathbf{a}$ step and before the complementary step $\mathbf{a} \leftarrow E^T\mathbf{h}$. It is also a reasonable assumption that the best frontier will segment each hub non-trivially, i.e., below its DOM root. Therefore we can invoke the segmentation routine separately on each page. Let the segmentation algorithm described previously be invoked as

$$F \leftarrow \mathbf{segment}(u)$$

where u is the DOM tree root of a page and F is the returned frontier for that page. Here is the pseudo-code for one iteration:

```
$\mathbf{h} \leftarrow E\mathbf{a}$
for each document DOM root $u$
    $F \leftarrow \mathbf{segment}(u)$
    for each frontier node $v \in F$
        $h(v) \leftarrow \sum_{w \in L_v} h(w)$
        for each $w \in L_v$
            $h(w) \leftarrow h(v)$
        reset $h(v) \leftarrow 0$
$\mathbf{a} \leftarrow E^T\mathbf{h}$
normalize $\mathbf{a}$ so that $\sum_u a(u) = 1$.
```

For convenience we can skip the hub normalization and only normalize authorities every complete cycle; this does not affect ranking.

The reader will observe that this is not a linear relaxation as was the case with HITS, Clever, or B&H, because **segment** may lead us to aggregate and redistribute different sets of hub scores in different iterations, based on the current leaf hub scores. (Also note that if F were fixed for each page for all time, the system would still be linear and therefore guaranteed to converge.) Although convergence results for non-linear dynamical systems are rare [10], in our experiments we never found convergence to be a problem (see §5).

However, we do have to take care with the initial values of $\mathbf{a}$ and $\mathbf{h}$, unlike in the linear relaxation situation where any positive value will do. Assume that the first iteration step transfers weights from authorities to hubs, and consider how we can initialize the authority scores. In contrast to HITS, we cannot start with all $a(v) = 1$. Why not? Because both good and bad authorities will get this score, resulting in many hub DOM subtrees looking more uniformly promising than they should. This will lead the **segment** algorithm to prune the frontier too eagerly, resulting in potentially excessive authority diffusion, as in HITS.

We propose a more conservative initialization policy. Similar to B&H, we assume that the textual content of the root-set documents returned by the text search engine is more reliably relevant than the radius-1 neighbors included for distillation. Therefore we start our algorithm by assigning only root-set authority scores to one. Of course, once the iterations start, this does not prevent authority from diffusing over to siblings, but the diffusion is controlled by hub segmentation.

There is one other way in which we bias our algorithm to be conservative w.r.t. authority diffusion. If a DOM node has only one child with a positive hub score, or if there is a tie in the cost of expanding vs. pruning, we expand the node, thereby pushing the frontier down and preventing the leaf hub score from spreading out to possibly irrelevant outlinks.

Taken together, these two policies may be a little too conservative, sometimes preventing desirable authority diffusion and bringing our algorithm closer to MicroHITS than we would like. For example, the graph being distilled may be such that page u has one DOM subtree clearly (to a human reading the text) dedicated to motorcycles, but only one link target v is in the expanded set. In ongoing work we are integrating text analysis into our fine-grained model to avoid such pitfalls [7].

5 Experiments and results

We used the 28 queries used in the Clever studies [5, 6] and by B&H [2] (shown in Figure 7). For each, RagingSearch returned at most 500 responses in the root set. These 500×28 pages were fetched and all their outlinks included in our database as well. RagingSearch and HotBot were used to get as many inlinks to the root set as possible; these were also included in our database. This resulted in about 488000 raw URLs.

After normalizing URLs and eliminating duplicates, approximately 366000 page fetches succeeded. We used the `w3c` command-line page fetching tool from `http://www.w3c.org` for its reliable timeout mechanism. We then scanned all these pages and filled a global (macro-)link table with 2105271 non-local links, i.e., links between pages not on the same hostname (as a lowercase string without port number).

We then proceeded to parse the documents into their DOMs in order to populate a different set of tables that represented the DOM nodes and the micro-links between them. We used the `javax.swing.text.html.parser` package and built a custom pared-down DOM generator on top of the SAX scanner provided. The total number of micro-links was 9838653, and the total number of micro-nodes likewise increased.

Out of the two million non-local links, less than 1% had targets that were not the root of the DOM tree of a page. Thus our introduction of the asymmetry in handling hubs and authorities seems to be not a great distortion of reality.

Even though our experiments were performed on a 700 MHz Pentium Xeon processor with 512 MB RAM and 60 GB of disk, handling this scale of operation required some care and compromise. In particular, to cut down the micro-graph to only about 10 million edges, we deleted all DOM paths that did not lead to an `<A...>...</A>` element. Otherwise, we estimated that the number of micro-links would be at least two orders of magnitude larger[2].

[2] In our ongoing work we are having to address this issue as we are also analyzing text.

#	Query	Drift	Mixed
1	``affirmative action''	large	
2	alcoholism		●
3	``amusement park*''	small	●
4	architecture		●
5	bicycling		
6	blues		
7	``classical guitar''	small	●
8	cheese		●
9	cruises		
10	``computer vision''		
11	``field hockey''		
12	gardening		●
13	``graphic design''	large	
14	``Gulf war''	large	
15	HIV		●
16	``lyme disease''	small	●
17	``mutual fund*''	small	
18	``parallel architecture''		●
19	``rock climbing''	large	
20	+recycling +can*		●
21	+stamp +collecting		
22	Shakespeare		●
23	sushi	small	●
24	telecommuting	large	
25	+Thailand +tourism	large	
26	``table tennis''	small	
27	``vintage cars''	small	●
28	+Zen +buddhism	large	

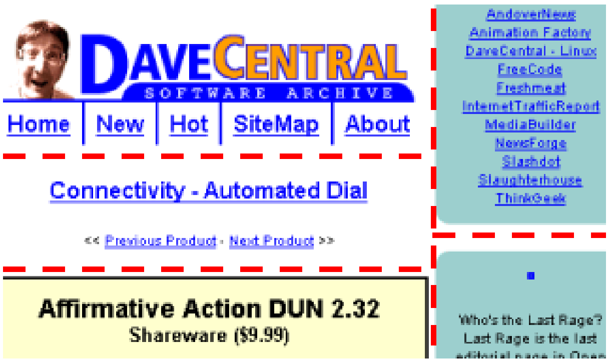
Figure 7: The set of 28 broad queries used for comparing B&H (without text analysis) and our system. The second column shows the extent of drift in the B&H response. The third column shows if mixed hubs were found within the top 50 hubs reported.

Figure 7 shows the 28 queries used by the Clever study and by B&H. As indicated before, our baseline was B&H with edge-weighting but without text-based outlier elimination, which we will simply call "B&H". We did not have any arbitrary cut-off for the number of in-links used as we did not know which to discard and which to keep. As B&H noted, edge-weighting improved results significantly, but without text analysis is not adequate to prevent drift. Of the 28 queries, half show drift to some extent. We discuss a few cases.

"Affirmative action" is understandably dominated by lists of US universities because they publicize their support for the same. Less intuitive was the drift into the world of software, until we found `http://206.243.171.145/7927.html` in the root set which presents a dialup networking software called Affirmative Action, and links to many popular freeware sites (figure 8). By itself, this page would not survive the link-based ranking, but the clique of software sites leads B&H astray.

Another example was "amusement parks" where B&H fell prey to multi-host nepotism in spite of edge-weighting. A densely connected conglomerate including the relevant starting point `http://www.411fun.com/THEMEPARKS/` (figure 9) formed a multi-site nepotistic cluster and misled macroscopic algorithms.

In both these cases there were ample clues in the DOM structure alone (leave alone text) that authority diffusion should be suppressed. We obtained several cases of reduced drift using our technique. (In ongoing work we are getting the improvement evaluated by volunteers.) One striking example was for the query "amusement parks" where our algorithm prevented `http://www.411...` from taking over the show (see figure 10; complete results are in `AP-macro.html` and `AP-micro.html`).



DaveCentral
SOFTWARE ARCHIVE

Home | New | Hot | SiteMap | About

Connectivity - Automated Dial

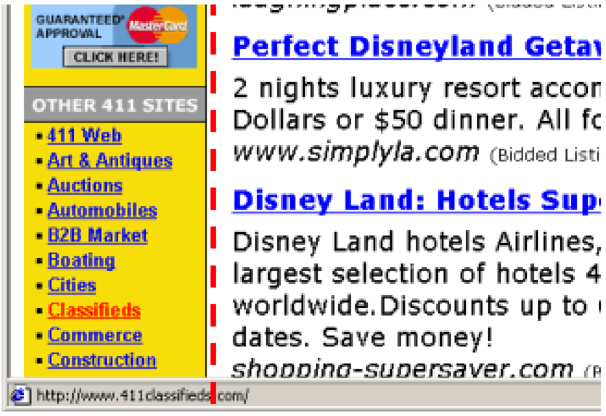
<< Previous Product - Next Product >>

Affirmative Action DUN 2.32
Shareware ($9.99)

AndoverNews
Animation Factory
DaveCentral - Linux
FreeCode
Freshmeat
InternetTrafficReport
MediaBuilder
NewsForge
Slashdot
Slaughterhouse
ThinkGeek

Who's the Last Rage?
Last Rage is the last

Figure 8: The part of this HTML page that contains the query *affirmative action* is not very popular, but adjoining DOM subtrees (upper right corner) create a dense network of software sites and mislead macroscopic distillation algorithms. Dotted red lines are drawn by hand.



GUARANTEED APPROVAL
MasterCard
CLICK HERE!

OTHER 411 SITES
- 411 Web
- Art & Antiques
- Auctions
- Automobiles
- B2B Market
- Boating
- Cities
- Classifieds
- Commerce
- Construction

Perfect Disneyland Geta

2 nights luxury resort accor
Dollars or $50 dinner. All fc
www.simplyla.com (Bidded Listi

Disney Land: Hotels Sup

Disney Land hotels Airlines,
largest selection of hotels 4
worldwide.Discounts up to
dates. Save money!
shopping-supersaver.com

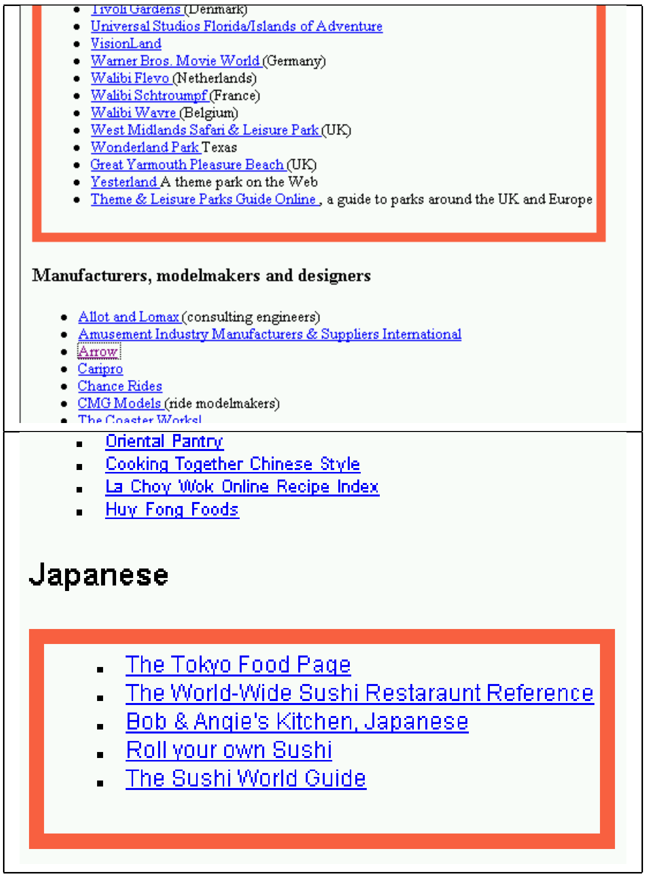
http://www.411classifieds.com/

Figure 9: The 411 "clique attack" comprises a set of sibling sites with different hostnames and a wide variety of topics linking to each other. A human can easily avoid paying attention to the sibling sites but macroscopic distillation will get misled. Dotted red lines are drawn by hand.

Figure 7 also shows that for almost half the queries, we found excellent examples of mixed hubs within the top 50 hubs reported. Given the abundance of hubs on these topics, we had anticipated that the best hubs would be "pure". While this was to some extent true, we found quite a few mixed hubs too. Our system automatically highlighted the most relevant DOM subtree; we present some examples in figure 11 and urge the reader to sample the annotated hubs packaged with the HTML version of this paper.

Macroscopic	Fine-grained
http://www.411boating.com	http://www.kennywood.com
http://www.411jobs.com	http://www.beachboardwalk.com
http://www.411insure.com	http://www.sixflags.com
http://www.411hitech.com	http://www.cedarpoint.com
http://www.411freestuff.com	http://www.pgathrills.com
http://www.411commerce.com	http://www.pki.com
http://www.411-realestate.com	http://www.valleyfair.com
http://www.411worldtravel.com	http://www.silverwood4fun.com
http://www.411worldsports.com	http://www.knotts.com
http://www.411photography.com	http://www.thegreatescape.com
	http://www.dutchwonderland.com

Figure 10: The fine-grained algorithm is less susceptible to clique attacks. The query here is *amusement parks*.



- Tivoli Gardens (Denmark)
- Universal Studios Florida/Islands of Adventure
- VisionLand
- Warner Bros. Movie World (Germany)
- Walibi Flevo (Netherlands)
- Walibi Schtroumpf (France)
- Walibi Wavre (Belgium)
- West Midlands Safari & Leisure Park (UK)
- Wonderland Park Texas
- Great Yarmouth Pleasure Beach (UK)
- Yesterland A theme park on the Web
- Theme & Leisure Parks Guide Online , a guide to parks around the UK and Europe

Manufacturers, modelmakers and designers

- Allot and Lomax (consulting engineers)
- Amusement Industry Manufacturers & Suppliers International
- Arrow
- Caripro
- Chance Rides
- CMG Models (ride modelmakers)
- The Coaster Works!

- Oriental Pantry
- Cooking Together Chinese Style
- La Choy Wok Online Recipe Index
- Huy Fong Foods

Japanese

- The Tokyo Food Page
- The World-Wide Sushi Restaraunt Reference
- Bob & Angie's Kitchen, Japanese
- Roll your own Sushi
- The Sushi World Guide

Figure 11: Two samples of mixed hub annotations: amusement parks amidst roller-coaster manufacturers and sushi amidst international cuisine.

Query	Annotated file
alcoholism	`AL1.html`
Amusement parks	`AP1.html`
Architecture	`AR1.html`
Classical guitar	`CG1.html`
HIV	`HI1.html`
Shakespeare	`SH1.html`
Sushi	`SU1.html`

We verified that our smoothing algorithm was performing non-trivial work: it was not merely locating top-scoring authorities and highlighting them. Within the highlighted regions, we typically found as many unvisited links as links already rated as authorities. In ongoing work we are using these new links for enhanced focused crawling.

A key concern for us was whether the smoothing iterations will converge or not. Because the sites of hub aggregation are data-dependent, the transform was nonlinear, and we could not give a proof of convergence. In practice we faced no problems with convergence; figure 12 is typical of all queries.

This raised another concern: was the smoothing subroutine doing anything dynamic and useful, or was convergence due to its picking the same sites for hub aggregation every time? In figures 13 and 14 we plot relative numbers of nodes pruned vs. expanded against the number of iterations. Queries which do not have a tendency to drift look like figure 13. Initially, both numbers are small. As the system bootstraps into controlled authority diffusion, more candidate hubs are pruned, i.e., accepted in their entirety. Diffused authority scores in turn lead to fewer nodes getting expanded. For queries with a strong tendency to drift (figure 14), the number of nodes expanded does not drop as low as in low-drift situations. For all the 28 queries, the respective counts stabilize within 10–20 iterations.

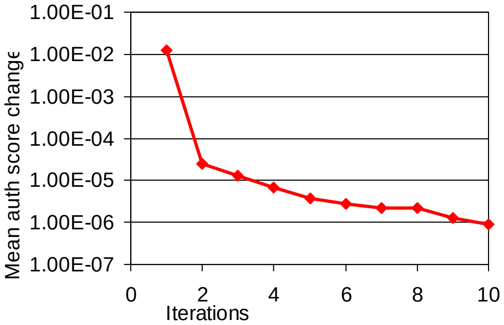


Figure 12: In spite of the non-linear nature of our relaxation algorithm, convergence is quick in practice. A typical chart of average change to authority scores is shown against successive iterations.

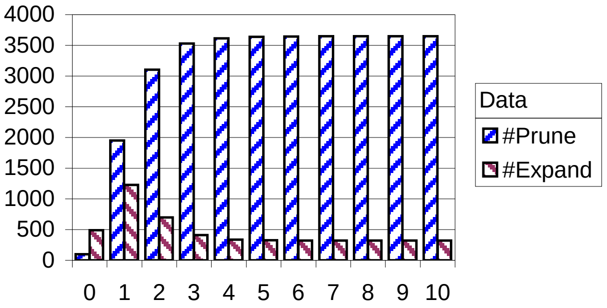


Figure 13: Our micro-hub smoothing technique is highly adaptive: the number of nodes pruned vs. expanded changes dramatically across iterations, but stabilizes within 10–20 iterations. There is also a controlled induction of new nodes into the response set owing to authority diffusion via relevant DOM subtrees (query: *bicycling*).

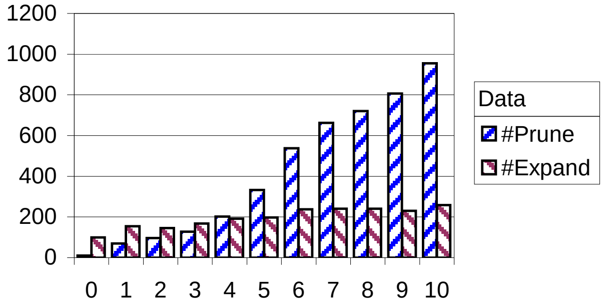


Figure 14: For some queries for which B&H showed high drift, our algorithm continues to expand a relatively larger number of nodes in an attempt to suppress drift (query: *affirmative action*).

Finally, we checked how close we were to B&H ranking. We expected our ranking to be correlated with theirs, but verified that there are meaningful exceptions. Figure 15 show a scatter plot of authority scores. It illustrates that we systematically under-rate authorities compared to B&H (the axes have incomparable scale; the leading straight line should be interpreted as $y = x$). This is a natural outcome of eliminating pseudo-authorities that gain prominence in B&H via mixed hubs.

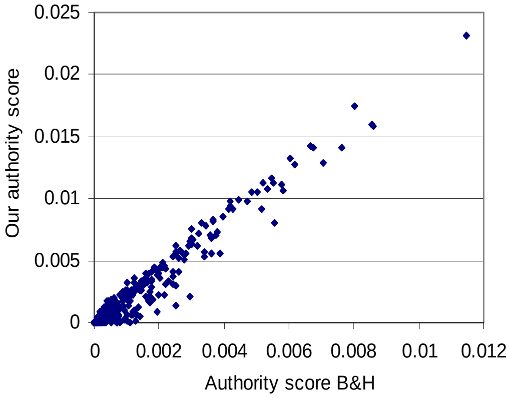


Figure 15: Our ranking is correlated to B&H, but not identical; we tend to systematically under-rate authorities compared to B&H.

6 Conclusion and future work

We have presented a fine-grained approach to topic distillation that integrates document substructure (in the form of the Document Object Model) with regular hyperlinks. Plugging in the fine-grained graph in place of the usual coarse-grained graph does not work because the fine-grained graph may not have the bipartite cores so vital to the success of macroscopic distillation algorithms. We propose a new technique for aggregating and propagating micro-hub scores at a level determined by the Minimum Description Length principle applied to the DOM tree with hub scores at the leaves. We show that the resulting procedure still converges in practice, reduces drift, and is moreover capable of identifying and extracting regions (DOM subtrees) relevant to the query out of a broader hub or a hub with additional less-relevant contents and links.

In ongoing work, apart from completing a detailed user study (as in the Clever project), we are exploring three more ideas. First, our algorithm depends on DOM branch points to be able to separate relevant hub fragments from irrelevant ones. We have seen some pages with a long sequence of URLs without any helpful DOM structure such as <LI> providing natural segment candidates. Second, we need to bring back some of the text analysis techniques that have improved HITS and integrate them with our model. Third, we are measuring if the link localization done by our system can help in faster resource discovery.

Acknowledgment: Thanks to Vivek Tawde and Hrishikesh Gupta for helpful discussions, to S. Sudarshan for stimulating discussions and generous support from the Informatics Lab, IIT Bombay, and the anonymous reviewers for helping to improve the presentation.

References

[1] E. Amitay and C. Paris. Automatically summarising web sites: Is there a way around it? In *9th International Conference on Information and Knowledge Management (CIKM 2000)*, Washington, DC, USA, 2000. ACM. Online at `http://www.mri.mq.edu.au/~einat/publications/cikm2000.pdf`.

[2] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Aug. 1998. Online at `ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf`.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th World-Wide Web Conference (WWW7)*, 1998. Online at `http://decweb.ethz.ch/WWW7/1921/com1921.htm`.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Focused web searching with PDAs. In *World Wide Web Conference*, Amsterdam, May 2000. Online at `http://www9.org/w9cdrom/195/195.html`.

[5] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *7th World-wide web conference (WWW7)*, 1998. Online at `http://www7.scu.edu.au/programme/fullpapers/1898/com1898.html`.

[6] S. Chakrabarti, B. E. Dom, S. Ravi Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the Web's link structure. *IEEE Computer*, 32(8):60–67, Aug. 1999.

[7] S. Chakrabarti, M. Joshi, and V. Tawde. Enhanced topic distillation using text, markup tags, and hyperlinks. Submitted for publication, Jan. 2001.

[8] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31:1623–1640, 1999. First appeared in the 8th International World Wide Web Conference, Toronto, May 1999. Available online at `http://www8.org/w8-papers/5a-search-query/crawling/index.html`.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[10] D. A. Gibson, J. M. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *VLDB*, volume 24, pages 311–322, New York, Aug. 1998.

[11] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, London, 1989.

[12] M. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM, June 1994. Online at `http://www.sims.berkeley.edu/~hearst/publications.shtml`.

[13] D. S. Johnson and K. A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research*, 8(1):1–14, 1983.

[14] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *ACM-SIAM Symposium on Discrete Algorithms*, 1998. Online at `http://www.cs.cornell.edu/home/kleinber/auth.ps`.

[15] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *WWW9*, pages 387–401, Amsterdam, May 2000. Online at `http://www9.org/w9cdrom/175/175.html`.

[16] J. Rissanen. Stochastic complexity in statistical inquiry. In *World Scientific Series in Computer Science*, volume 15. World Scientific, Singapore, 1989.

[17] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[18] S. Sarawagi. Explaining differences in multidimensional aggregates. In *International Conference on Very Large Databases (VLDB)*, volume 25, 1999. Online at `http://www.it.iitb.ernet.in/~sunita/papers/vldb99.pdf`.