Enhanced Topic Distillation using Text, Markup Tags, and Hyperlinks

Soumen Chakrabarti* Mukul Joshi Vivek Tawde

IIT Bombay

ABSTRACT

Topic distillation is the analysis of hyperlink graph structure to identify mutually reinforcing *authorities* (popular pages) and *hubs* (comprehensive lists of links to authorities). Topic distillation is becoming common in Web search engines, but the best-known algorithms model the Web graph at a coarse grain, with whole pages as single nodes. Such models may lose vital details in the markup tag structure of the pages, and thus lead to a tightly linked irrelevant subgraph winning over a relatively sparse relevant subgraph, a phenomenon called *topic drift* or *contamination*. The problem gets especially severe in the face of increasingly complex pages with navigation panels and advertisement links. We present an enhanced topic distillation algorithm which analyzes text, the markup tag trees that constitute HTML pages, and hyperlinks between pages. It thereby identifies subtrees which have high text- and hyperlink-based coherence w.r.t. the query. These subtrees get preferential treatment in the mutual reinforcement process. Using over 50 queries, 28 from earlier topic distillation work, we analyzed over 700 000 pages and obtained quantitative and anecdotal evidence that the new algorithm reduces topic drift.

Topic areas: Citation and Link Analysis, Machine Learning for IR, Web IR.

1 Introduction

In the last several years, the Web has been evolving in fascinating ways, apart from just getting larger. Web content is migrating from static pages and files to dynamic views generated from complex templates and backing semi-structured databases. A variety of new hyperlink idioms such as navigation panels, advertisement banners, link exchanges, and Web-rings, have also been emerging.

The *document* has been a fundamental unit of analysis in traditional Information Retrieval (IR) [18, 21], as well as in more recent work on citation and link analysis [13, 4, 2], which have been shown to add significant value for certain broad classes of queries in careful benchmarking experiments [20].

On the Web, documents have typically been single, static HTML files. But with Web content migrating towards 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/`), that document-level view is now in jeopardy. Document and site boundaries are not what they used to be.

Furthermore, with the proliferation of hand-held devices with small or no screens, Web search engines and topic distillation tools must respond not with whole pages but with *snippets* extracted from a bigger context. Therefore the internal model of hypertext and the Web that is used in search and distillation algorithms must evolve to a finer level of detail, capturing element-level structure through the Document Object Model (DOM, see `http://www.w3.org/DOM/`) of coarse-grained pages and the links between them.

We call this detailed view the *fine-grained model*. The new model may warrant revisiting several problems in hypertext information retrieval. Here we focus on topic distillation: using hyperlink structure to identify mutually reinforcing *authorities* (popular pages) and *hubs* (comprehensive lists of links to authorities).

1.1 The problem

Hyperlink and citation analysis have significantly influenced hypertext search and ranking on the Web. Although there have been several extensions to traditional IR systems to handle hypertext [19], we know of no earlier effort to extend the more recent topic distillation work to the fine-grained model.

In our continual experiments with topic distillation systems through the last two years, we have seen the quality of output steadily deteriorate. Most often, we have traced the problem to the following two observations:

- Pages are more complex and have more links, many of which are 'noisy' from the perspective of the query, such as in banners, navigation panels, and advertisements. The assumption that human editorial judgment endorses hyperlink targets breaks down with dynamic, composite pages generated from templates.
- Topic distillation algorithms treat whole pages as atomic, indivisible nodes with no internal structure. This leads to false rank reinforcements and resulting contamination of the query responses.

Specifically, known distillation algorithms are vulnerable to "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`. The HTML presentation makes it clear to a human reader that some links are more coherent than others, but distillation algorithms are unable to exploit this distinction.

*Contact author, email `soumen@cse.iitb.ac.in`.

SIGIR'01, September 9–12, 2001, New Orleans, Louisiana, USA.

1.2 Our contributions

We propose a fine-grained model for Web content at the HTML DOM element level, and a new technique for topic distillation in a setting where page and site boundaries are blurred.

We argue that rather than accumulate hub scores over entire pages, they should be accumulated over a certain cut or *frontier* across the DOM tree of the page, which will *disaggregate* the hub score of the page into DOM subtrees which may have diverse densities of useful links in them.

We propose a new algorithm to locate regions and subtrees of pages which should get favorable treatment in propagating link-based popularity, implicitly suppressing propagation of popularity to regions with noisy links. We combine two sources of information to achieve this goal: the vector-space representation of the text contained in the subtree and the distribution of hub scores at the leaves of the subtree. The modified distillation algorithm thus guides the hub-and-authority reinforcement to work on a selected, highly relevant subgraph of the Web.

We evaluate our ideas using 50 queries of which 28 have been used before for topic distillation research. These queries lead our algorithm to collect and analyze over 700 000 pages with tens of millions of fine-grained DOM-level links. Quantitative measurements as well as anecdotal evidence suggest that the new algorithm is effective in avoiding topic drift and contamination problems and in identifying from composite pages the regions and snippets that are relevant to the query.

1.3 Related work

In the IR domain, document segmentation based on discourse and term features is well explored; Reynar's PhD thesis [16] includes a comprehensive survey. Some well-known segmentation systems are by Beeferman et al [1], Hearst (TextTiling, [11]), Ponte and Croft [15], and Richmond et al [17]. These systems are intended for text only and use distributions of terms and derived features, not hypertextual features, for the segmentation task. Amitay [17] describes a technique to annotate the target page of a URL by segmenting the text on source pages. We know of no earlier attempt to integrate information from text, hyperlinks and DOM structure for topic distillation. In our case, segmentation is not the end goal but falls naturally out of the analysis. Our work is most closely related to work on hyperlink induced topic search (HITS) [13], topic distillation [2, 7], and the PageRank algorithm used in Google [4]. These algorithms work at the coarse-grained hypertext graph level and are reviewed in §2.

2 Preliminaries

Kleinberg's original HITS algorithm [13] started with a query q which was sent to a text search engine. The returned set of pages R_q (called the *root set* 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. The additional pages constitute the *expanded set* and the union of the root and expanded sets is the *base set* for the given query.

Nodes in the base set link to each other, but not all such hyperlinks are retained in HITS's representation of the graph. Links between pages on the same host or site are discarded because they are often seen to serve only a navigational purpose, or are 'nepotistic' in nature. We will revisit 'nepotism' in various forms in this paper.

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 (say, all 1's). Collectively, all hub (authority) scores are represented as the vector $\mathbf{h}$ (respectively, $\mathbf{a}$).

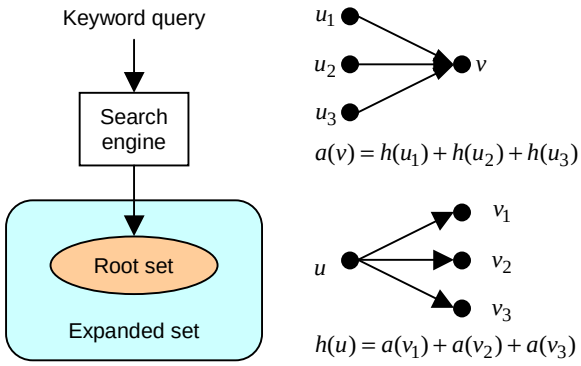

Figure 1: Hyperlink induced topics search (HITS).

Next the HITS algorithm alternately updates $\mathbf{a}$ and $\mathbf{h}$ 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 $\mathbf{a}$ and $\mathbf{h}$ so that $\sum_v h(v) = \sum_v a(v) = 1$. The iterations continue until the ranking of nodes by a and h stabilize (Figure 1). It is important to notice the bipartite reinforcement involved here: if $a(v_1)$ increases in some iteration, $h(u)$ will increase next, leading in turn to an increase in $a(v_2)$ and $a(v_3)$.

If E is represented in the adjacency matrix format ($E[i,j] = 1$ if there is an edge (i,j) and 0 otherwise), 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 $|\mathbf{h}|_1 = |\mathbf{a}|_1 = 1$. It can be shown that the scores *will* stabilize [10]: $\mathbf{a}$ will converge to the principal eigenvector of E^TE and $\mathbf{h}$ will converge to the principal eigenvector of EE^T.

Soon after HITS was published, Bharat and Henzinger (B&H) [2] found that the threat of nepotism was not necessarily limited to same-site links. Two-site nepotism (a pair of Web sites endorsing each other) was on the rise. In many trials with HITS, they found two distinct sites h_1 and h_2, where h_1 hosted a number of pages u linking to a page v on h_2, driving up $a(v)$ beyond what may be considered fair. B&H proposed a simple and effective edge-weighting scheme to prevent such problems: if k pages on h_1 point to v, they set the weight of each of these links be $1/k$, so that they added up to one, reflecting a belief that a site (not a page) is worth one unit of voting power. The entries in E may be changed appropriately so that the matrix notation is still valid. In the rest of the paper, we will use this edge-weighted version of HITS and call it HITS and B&H interchangeably.

B&H also noted that for some queries for which the relevant Web subgraph was well-connected to a denser subgraph (sometimes on a related broader topic), HITS output tended to *drift* towards the broader community. E.g., the response to *movie awards* may drift towards *movie studios*. More seriously, hubs with a small relevant region but large generic

Figure 2: Spammed or mixed hubs, a frequent contaminant in HITS. This page is `http://cheeze.qaz.com`.

navigation panels (Figure 2), which are rampant on the Web today, may lead to a clique completely taking over the top positions in both the hub and authority rankings (Figure 3).

To reduce drift, B&H fetched the documents in the root set and computed their vector space representation [18]; they then pruned from the expanded graph pages whose document vectors were "too far" from the root set centroid. Discarding authorities on such grounds seems reasonable, but hubs are often mixed, and the B&H heuristic may discard some valuable links based on the aggregate relevance of the entire page. Thus, B&H pruning may reduce the score of relevant authorities.

By the time the Clever system [7] was built, "mixed hubs" were already in evidence. Because HITS modeled a page as a single node with a single h score, high authority scores could diffuse from relevant links to less relevant links. In Clever, 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 were segmented using an ad-hoc set of prominent separators (such as `<UL>` or `<HR>`) into 'pagelets' with their own scores. Obviously, given the diversity of visual formatting tricks such as frames, tables, layers, etc., such a technique is completely at the mercy of the page author's choice of formatting style and tags.

Today, mixed hubs and clique attacks are pervasive, in the form of personal bookmarks on various topics, diverse businesses hosted by a common service which links up their homepages, and even clear spamming to foil link-based ranking algorithms. A recent study [3] has concluded that none of the common whole-page distillation algorithms are immune to topic drift and clique attacks. Therefore it is very important to bring in additional sources of information (tag tree structure) where possible.

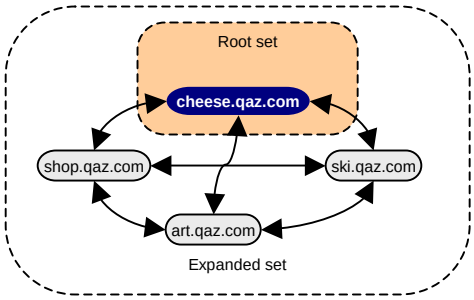

Figure 3: "Clique attack" on HITS for the query *cheese*.

3 Proposed model and algorithms

Existing topic distillation algorithms fail to mimic humans who are rarely foiled by clique attacks, because they carefully interpret HTML page idioms to locate content-bearing regions, assisted by text in those regions (Figure 4).

A key problem responsible for many of the aberrations discussed so far seems to be the *over-aggregation* of hub scores owing to the atomic, indivisible, single-node model for a hub. In our enhanced model, we assume that each HTML page is a DOM tree where some leaf nodes (elements) are `HREF`s to the *roots* of other DOM trees. (I.e., for simplicity, we remove location markers indicated by a `#`-sign from URLs, which occurs in a very small fraction of search engine responses.)

People can easily locate useful DOM subtrees from two kinds of locality clues:

1. The text in some region of the page looks more interesting from the perspective of the user's information need, so the user is encouraged to follow links appearing in those regions.

2. The user spots some familiar hyperlinks of known worth in some regions of the page, often clustered together.

In a recent paper [6] we isolated and explored the effect of exploiting the latter source of information using an algorithm we called DomHits. Our experience was mixed: DomHits did help to arrest topic drift, but was overly conservative at times. Lacking domain knowledge, DomHits was 'familiar' with a URL only if it belonged to the root set. If a good hub has only one link to a URL that is in the root set, DomHits would have no evidence to believe that the other URLs on that page could be any good. In this paper our main goal is to bring in textual information to reduce that shortcoming. The new algorithm presented in this paper is called DomTextHits.

3.1 A generative model for mixed hubs

Despite the mathematical symmetry of the HITS family of algorithms w.r.t. hubs and authorities, there is an essential functional asymmetry between hubs and authorities. Almost by definition, an authority on a topic[1] is expected not to digress, whereas the onus of exploring out from a hub seems to lie with the surfer. Both DomHits and DomTextHits deliberately introduce asymmetry into the HITS

[1]Unlike in ad-hoc retrieval, we can talk about topics and queries interchangeably in the distillation context.

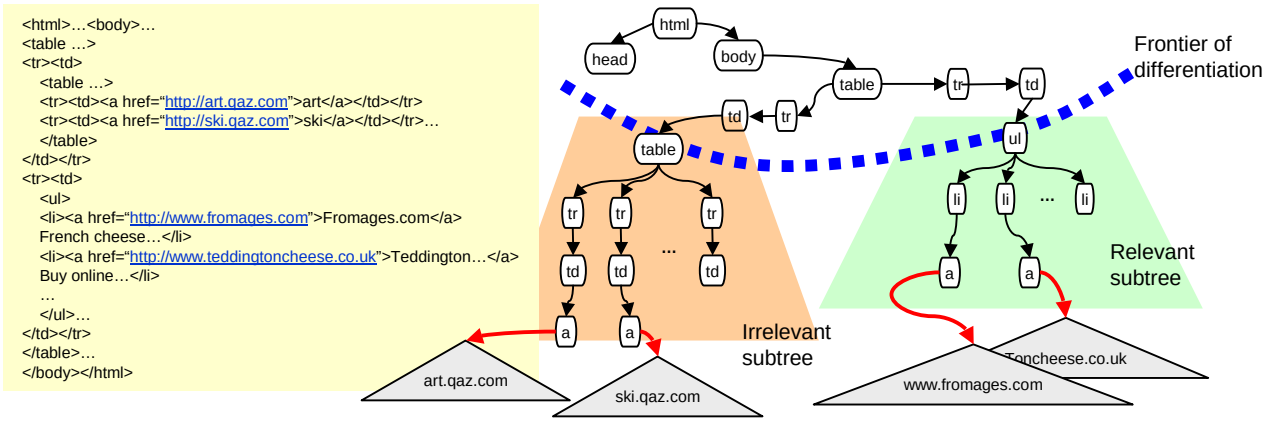

Figure 4: Humans are rarely misled by irrelevant links because they interpret HTML page idioms to locate content-bearing regions (i.e., HTML tag-subtrees), assisted by text in those regions. Pages are differentiated into relevant (green) and irrelevant (red) regions by their term distribution as well as links to known relevant or irrelevant sites.

framework, and work much harder on hubs than authorities, because detecting mixed hubs can reduce over-aggregation of hub scores and consequent leakage of authority scores.

How are mixed hubs created? The author picks a set of topics on which she wishes to compose the mixed hub page. (In this discussion we ignore subsequent modifications.) Unless the topics are almost inseparable, coherence in authorship dictates that outlinks relevant to a given topic are physically clustered on the page, and are accompanied by anchors and text that are indicative of that topic.

From the perspective of a user seeking content related to a specific topic, the author of a mixed hub page composes an HTML tag tree which is largely uninteresting, but for one or a few subtrees which seem to contain a significantly higher density of interesting text tokens and promising hyperlinks.

Suppose that w.r.t. a fixed topic (and an unchanging snapshot of the Web), all authors and users can agree about the vocabulary to use in and around the `HREF` anchors linking to the authority pages on that topic. If in addition they also agreed on a global ranking of authorities, all hubs would be identical or very similar: they would link to the same set of authorities, describing them in very similar terms. Let this (unknown) term distribution be Φ_0.

In reality, not all hub page authors are aware of the best authorities, and their authorship styles are diverse. Moreover, the author of a mixed hub page deliberately chooses to dedicate not the entire page, but only a fragment of it, to URLs relevant to the given topic. The text "sub-document" d contained in any DOM subtree can be scored for its similarity w.r.t. the "ideal" distribution Φ_0. E.g., we may represent Φ_0 as a vector space model [18] and measure the similarity between d and Φ_0 using the standard cosine measure. We would expect relevant subtrees to have larger value of cosine similarity than irrelevant subtrees, although this signal may be accompanied by much noise (e.g., "click here" and "best viewed using Netscape").

3.2 Segmentation and smoothing

We wish to extract and use the signal in text and DOM trees to influence the HITS algorithm. A generic template for the new family of algorithms is given in Figure 5. The pseudocode differs in a number of important ways from earlier distillation algorithms.

First, we allow only the DOM tree roots of root set nodes to have a non-zero authority score when we start, unlike HITS/B&H which sets all scores to positive numbers. We believe that positive authority scores should diffuse out from the root set only if the connecting hub regions are trusted to be relevant to the query. Accordingly, the first half-iteration implements the $\mathbf{h} \leftarrow E\mathbf{a}$ transfer.

Second, for the transfer steps, the graph represented by E does not include any internal nodes of DOM trees. The asymmetry between hubs and authorities in our treatment leads to the new steps **segment** and **smooth**, which are the only steps that involve internal DOM nodes. Therefore, only DOM roots have positive authority scores, and only DOM leaves (corresponding to `HREF`s) have positive hub scores.

The role of the **segment** and **smooth** steps is to find the frontier of differentiation shown by the blue dotted line in Figure 4 earlier. The hub is then logically decomposed into one microhub per frontier node. E.g., in Figure 4, two microhubs would be created, one for the (unwanted) red subtree and one for the (favored) green subtree. We expect that the green microhub will take an active role in reinforcing good authorities, whereas the red microhub score will dwindle in comparison.

3.2.1 Text-based segmentation

Using term distribution to segment moderately long linear text documents is well-explored [11]. Organizing moderately long documents into topic-based clusters is also established [5]. Our application differs in that our "documents" are often very short snippets, we wish to use the additional structure present in the HTML tag tree, and we are interested in only those segments that are pertinent to the query.

A short query cannot be used in isolation for detecting the pertinence of text in a DOM subtree. E.g., for a candidate hub in the expanded set, the query terms *japanese*, *car* or *manufacturer* may not appear in a DOM subtree linking out to the Toyota or Mazda sites, and a cosine measure will be zero. The key insight here is that *Toyota* and *Mazda* are likely to be high-frequency terms in the root set documents. We thus regard the standard TFIDF-weighted vector space centroid of all root set documents as the **ground truth** term distribution for a given query. Our intuition is that in spite

```
construct the fine-grained graph for the given query
set all hub and authority scores to zero
for each page u in the root set do
    locate the DOM root r_u of u
    set a_{r_u} = 1
end for
while scores have not stabilized do
    perform the h ← Ea transfer
    segment hubs into "micro hubs"
    smooth their hub scores
    perform the a ← E^T h transfer
    normalize |a|
end while
```

Figure 5: DomHits and DomTextHits follows this general template. Note that the vertex set involved in E includes only DOM roots and leaves, and *not* other internal nodes. Internal DOM nodes are involved only in the **segment** and **smooth** steps.

```
compute TF and IDF for root set (whole) documents
compute IDF-scaled vectors for root set documents
compute centroid C of TFIDF vectors
for each root set document d do
    find cosine similarity between C and d
end for
let ρ be the median similarity score
for each DOM microhub node u in the base set do
    compute IDF-scaled vector d_u for text beneath u
    compute similarity between C and d_u
    if similarity is larger than ρ then
        mark u as a "must-prune" node
    end if
end for
```

Figure 6: Text-based pruning. IDF always refers to the root set. Any percentile can be used in place of the median.

of noise terms, consensus about the greatest content-bearing terms will emerge (Figure 13).

Given the ground truth vector, we need not invoke a general clustering or segmentation algorithm on the candidate hubs. Instead, for each root and internal DOM node u in the base set, we measure the cosine similarity between the ground truth vector and the vector corresponding to the text contained in the subtree of u. If the cosine similarity is "large enough" we decide that u is pertinent to the query and enforce that it must lie *at or below* the differentiation frontier. We call this operation *pruning*. Somewhat unintuitively, a pruned DOM subtree is a desired, trusted one for propagating authority. The judgment of "large enough" is elaborated in the pseudocode in Figure 6. If an ancestor and a descendant are both marked for pruning, the ancestor takes precedence. By convention all leaves are marked for pruning.

Once the frontier microhubs are determined, leaf hub scores are accumulated up the tree to the frontier nodes, and then the frontier aggregates are copied down to each leaf in their respective subtrees. This process is shown in Figure 7. We state the following claim without proof for lack of space.

CLAIM: The general template shown in Figure 5 together with the text-based pruning scheme illustrated in Figures 6 and 7 guarantee convergence of hub and authority scores.

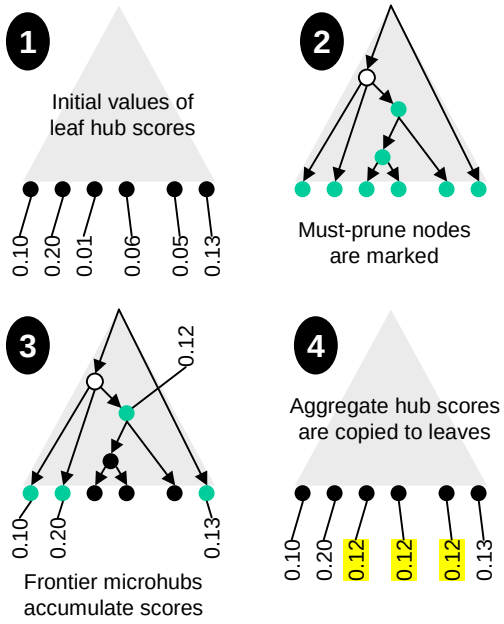

Figure 7: Microhub score **smooth**ing, which consists of aggregation and propagation.

3.2.2 Adding information from the link graph

In this section we describe how to use evidence from `HREF`s to known authorities to reinforce the text-based approach. Our arguments and models are similar to those for textual coherence. Just as Φ_0 represented the term distribution over interesting HTML segments (w.r.t. a fixed query), let Θ_0 represent the complete, global (but unknown) distribution of authority scores. If everyone knew and agreed about these scores, only one hub would ever be needed on any topic; all hub authors would compose identical hubs (modulo accompanying text).

As before, hub authors *will* differ in their choice of URLs to point to, and, as in the case of text, may choose to compose a mixed page with only a fraction of links that are relevant to the query. Therefore, the distribution of hub scores for pages (and regions of pages) composed by a specific author will be different from Θ_0.

We can regard this process as a progressive specialization of the hub score distribution starting from the global distribution Θ_0. For simplicity, assume all document roots are attached to a 'super-root' which corresponds to 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.

During topic distillation we observe pages which are the outcome of this generative process, and our goal is to reverse engineer or rediscover the frontier at which the score distributions were likely to have been fixed. The hub scores at the leaves can be explained excellently by including all leaves in the frontier, but the frontier models will be very different from the global model. Conversely, picking a frontier that is too shallow may save the cost of correction, but model the leaf scores poorly. This is a standard trade-off in fitting multiple models to data [8].

A formal model: More specifically, let the distribution associated with node w be Θ_w. The set of hub scores at the

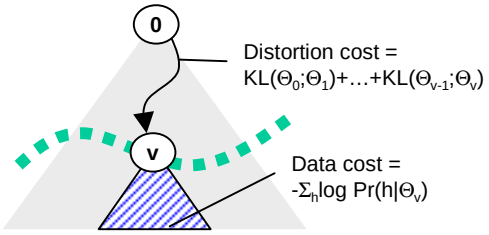

Figure 8: Optimal segmentation of mixed hubs.

leaves of a tree rooted at node w 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. By Shannon's theorem [8], there are efficient codes which can achieve a code length close to the entropy bound, i.e.,

$$\text{data encoding cost at } w = -\sum_{h \in H_w} \log \Pr_{\Theta_w}(h) \quad \text{bits.} \tag{1}$$

Here $\Pr_\Theta(h)$ is the probability of hub score h w.r.t. a distribution characterized by parameter Θ.

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 later.

Now consider a node u in the DOM graph with children $v_1, \ldots, v_k$. Suppose we decided to specialize the distribution Θ_v of some v away from Θ_u, the distribution of u. The cost for this is given by the well-known Kullback-Leibler (KL) distance from Θ_v to Θ_u, denoted $KL(\Theta_v; \Theta_u)$, and expressed as

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

where unlike in the case of entropy, the sum can be taken to an integral in the limit for continuous variable. 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., $KL(\Theta_w; \Theta_v) + KL(\Theta_v; \Theta_u)$. We will denote the cost of such a path as $KL(\Theta_u; \Theta_v; \Theta_w)$.

Given Θ_u and H_v, we should choose Θ_v so as to minimize the sum of the KL distance and data encoding cost:

$$KL(\Theta_v; \Theta_u) - \sum_{h \in H_v} \log \Pr_{\Theta_v}(h). \tag{3}$$

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

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

is minimized.

A practical search algorithm: The problem formulated 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 [12], 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.

```
1: Input: DOM tree for a candidate hub page
2: initialize frontier F to the DOM root node
3: while local improvement to code length possible do
4:    pick from F an internal node u with children {v}
5:    if u has only one child then
6:       replace u by v in F
7:    else
8:       if u is a must-prune node based on text then
9:          mark u as pruned
10:      else
11:         find the cost of pruning at u (see text)
12:         find the cost of expanding u (see text)
13:         mark u as pruned if prune cost
            is "significantly lower"
14:      end if
15:   end if
16: end while
```

Figure 9: **Segment** and **smooth** routines integrating information from text and DOM+link structure.

We use the obvious greedy expansion strategy. We 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, which is $-\sum_h \log \Pr_{\Theta_u}(h)$.
- The cost of expanding u to its children, plus the cost of encoding the subtrees H_v with respect to Θ_v. These add up to $\sum_v KL(\Theta_u; \Theta_v) - \sum_v \sum_h \log \Pr_{\Theta_v}(h)$.

If the latter cost is less, we expand u, otherwise, we prune it, meaning that u becomes a frontier node. See Figure 9 for details.

Non-parametric evaluation of the KL distance is complicated, and often entails density estimates. Hence we used parametric distributions, specifically, the Poisson distribution for which the KL distance has closed form expressions. If Θ_i is a Poisson distribution with mean μ_i $(i = 1, 2)$, then

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

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 in discretizing positive values is to scale the smallest value to one, effectively allocating $\log \frac{h_{\max}}{h_{\min}}$ bits per value. This is not suitable in our case. One can easily construct graphs for which, as the HITS-style power iterations and scaling are performed, the score ratio may become unbounded. A reasonable compromise is possible by noting that the user is not interested in the precision of poor hubs. While evaluating $\Pr(h|\Theta)$ where distribution Θ has mean μ, We divide the range of hub scores into buckets whose width is the smaller of μ and the *median* hub score. As long as the bucket width was smaller than μ we found

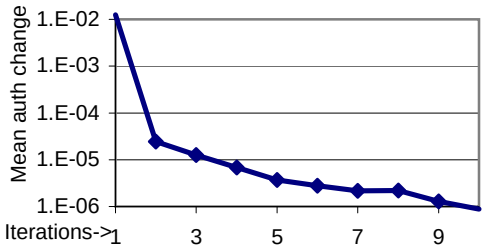

Figure 10: Rate of convergence is nearly exponential for all queries executed on the system.

the results to be insensitive to the specific choices. However the discretization warrants a fudge in comparing the cost of pruning vs. expanding: we prune only of the cost of pruning beats that of expanding by a margin (5%), to bias the system away from undue authority leaks.

3.3 The complete algorithm

Our overall algorithm is formed by plugging in the segmentation strategy shown in Figure 9 into the template in Figure 5.

If only text-based pruning (Figure 6) were used convergence of the scored would be guaranteed. However, with DOM-based pruning added in (Figure 9), the reader may observe that this is not a linear relaxation system any more, unlike HITS, Clever, or B&H. Depending on the leaf hub scores, the segmentation algorithm may find different frontiers in each iteration. Although convergence results for nonlinear dynamical systems are rare [9], in our experiments we never found convergence to be a problem (Figure 10).

However, unlike with linear relaxation, it *is* important to limit positive initial authority scores to root set documents alone, as shown in Figure 5. If, like HITS, we start from all a_i set to 1, the smoothing algorithm tends to prune too eagerly, resulting in excessive authority diffusion, as in HITS.

Since preventing drift is one goal of finding micro-hub aggregates, we also work another bias into the greedy expansion strategy: if there is a tie in the cost of expanding vs. pruning, or if a DOM node has only one child with positive hub scores, we expand the node. This stops authority diffusion across hubs that have only one link to a known authority.

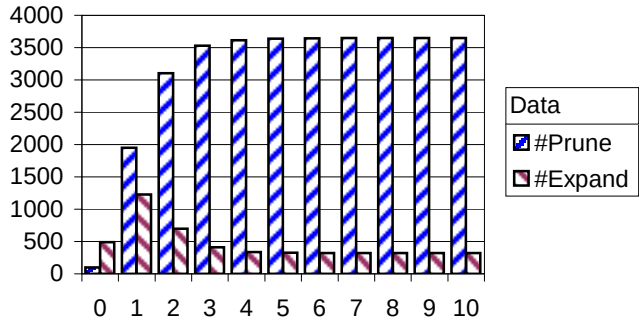

Figure 11: Our micro-hub smoothing technique is highly adaptive: the number of nodes pruned vs. expanded changes dramatically across iterations, but stabilizes in the end. The x-axis denotes iterations and the y-axis shows counts. There is also a controlled induction of new nodes into the smoothing operation owing to authority diffusion via relevant DOM subtrees.

4 Experimental study

We used two data sets (for a demo with precomputed results and anecdotes please visit `http://memex.cse.iitb.ac.in`):

ARC/Clever: This is the set of 28 queries used in the ARC/Clever system and also by B&H [2, 7]. We used up to 200 responses from RagingSearch (`http://raging.com`) as the root set. We report data on 24 queries.

DMoz: We picked 20 topics from a simplified version of the DMoz topic hierarchy (`http://dmoz.org`) with some 250 topics, and used a random sample of 200 URLs therefrom as the root set for each topic. We report data from four classes for lack of space.

In all cases, for each root set page, we used the top 200 backlinks from RagingSearch and Google. The total of about 50 queries required some one million page fetches of which about 700 000 succeeded[2]. We used `w3c-libwww` for crawling and `libxml2` for parsing. There were about 4 million non-local `HREF`s and over 15 million fine-grained nodes and links. Less than 1% of the `HREF`s 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.

All earlier topic distillation research had requested volunteers to judge output quality subjectively, but this was precluded given our larger scale of operation. (We plan to complete a user study shortly.) We therefore resorted to the evaluation methodology described in the rest of this section.

Convergence: For all of the queries, convergence was achieved within 10–20 iterations (Figure 10 is typical). Second, we wanted to ensure that convergence was not because of an effectively static selection of the sites of hub score accumulation. In Figure 11 we plot relative numbers of nodes pruned vs. expanded against the number of iterations. 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. The respective counts stabilize within 10–20 iterations in our experience.

Tuning ρ: We found that the effects of small variations of the prune threshold (Figure 6) on the relevance of answers is mild (Figure 12).

Root set centroid: We verified that the largest components of the root set centroid correspond to terms that are intuitive (Figure 13).

Hub relevance: We computed textual cosine similarity between top 40 microhubs and the root set ground truth for each algorithm and compared them. A typical example is shown in Figure 14; DomHits and DomTextHits clearly locate better hubs than HITS.

Authority relevance: We fetched all outlink (authorities) of the top 40 microhubs, eliminated authorities that were already in the root set, and for the others, computed their textual cosine similarity with the root set ground truth.

[2]This was a heroic effort given the poor reliability of our ISP (VSNL) and DNS services.

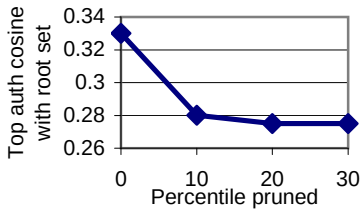

Figure 12: Relaxing the text-based threshold for pruning decreases the relevance of top authorities only mildly.

sushi: sushi, japanese, restaurant, page, bar, rice, roll
gardening: garden, home, plants, information, organic, click
bicycling: bike, bicycle, page, site, ride, tour, new, sports
alcoholism: alcohol, treatment, drug, addiction, recovery, abuse
blues: blues, site, festival, jazz, music, new, society

Figure 13: Despite a few Web-specific words, the largest components of root set centroid vectors are extremely intuitive; these serve reasonably well as "ground truth."

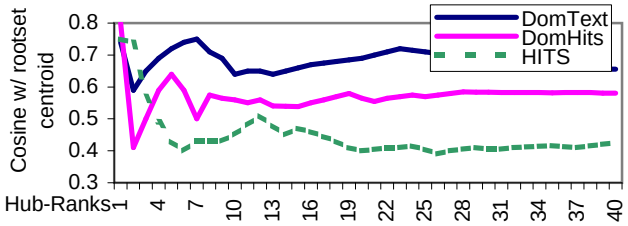

Figure 14: For query *Shakespeare* DomHits already beat HITS, but DomTextHits improved further upon DomHits in terms of proximity of text in top microhubs to the root set centroid.

This indicated if the most promising microhubs actually delivered (Figure 15). Authorities reachable from hubs found by DomTextHits are on an average 25% more relevant than those found by HITS, and those found by DomHits are almost 33% more relevant. Neither of DomTextHits and DomHits beats the other uniformly. For the query *cruises*, HITS is led astray by a clique attack from `http://cruisecyprus.com/` which leads it to a clique about Cyprus, whereas DomTextHits and DomHits do far better, DomTextHits being marginally betters than DomHits (Figure 17).

Expansion: The apparent clear superiority of DomHits may be misleading, because DomHits achieves this score by being extremely conservative in frontier pruning (Figure 16). Of the top 40 authorities, HITS takes only 10 from the root set, DomTextHits takes about 25, whereas DomHits takes as many as 33; so it is not too surprising that DomHits authorities have higher relevance.

DMoz and classification: For Dmoz data, no keyword query was involved. For a fixed topic c, the sample URLs form the root set. The Rainbow text classifier [14] is trained with all the DMoz topics ahead of time. The 40 top authorities from the distillation algorithm (HITS, DomHits or DomTextHits) are submitted to the classifier. For each authority document d, the classifier returns a Bayesian estimate of $\Pr(c|d)$. These are added up as the *expected number of relevant authorities* among the top 40. Details are omitted due to space constraints.

Obviously, an automatic classifier can make mistakes. Even so, we believe it is a reasonable surrogate for volunteers because (a) the root set is a coherent topical collection designed by human effort rather than keyword search, and (b) testers can make mistakes too. At least, we can interpret the classification result more reliably than an absolute cosine measure, because the 250 Dmoz topics cover most areas of

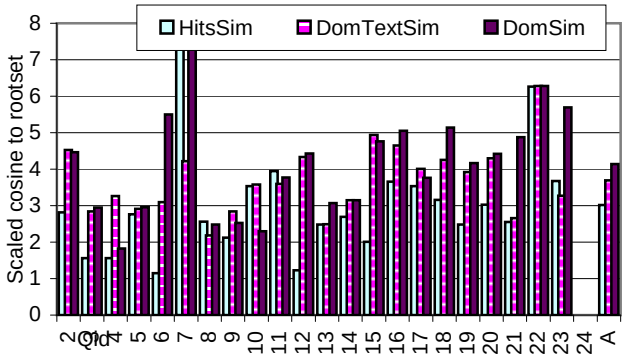

Figure 15: The authorities reported by DomHits and DomTextHits are 25–33% more similar to root set exemplars than HITS.

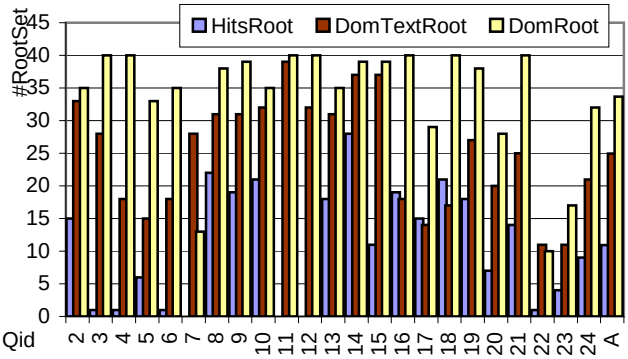

Figure 16: HITS admits many authorities from outside the root set; DomHits is too stringent; DomTextHits is in between.

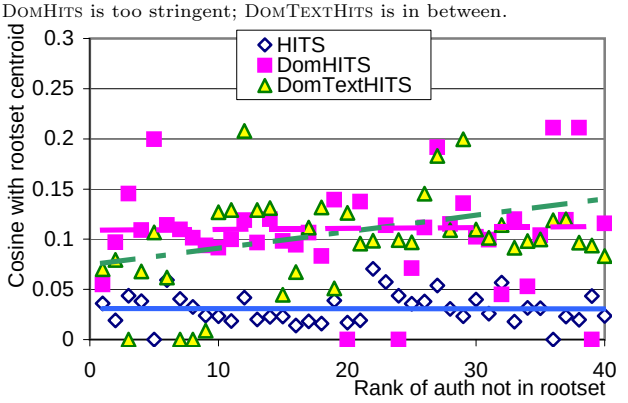

Figure 17: For *cruises* and several other queries, HITS drifts but DomHits and DomTextHits resist contamination. Here we take the top 1 through 40 authorities that were *not* in the root set and find their cosine similarity to the root set centroid. Higher dots are better.

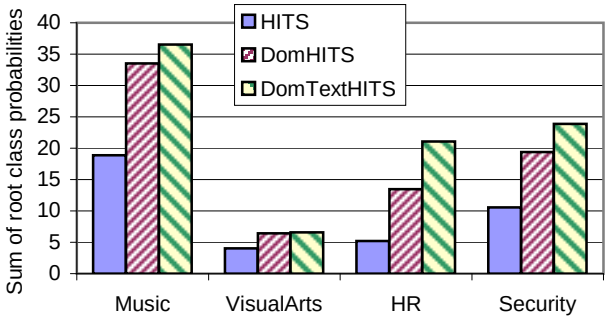

Figure 18: Authorities reported by DomTextHits have the highest probability of being relevant to the Dmoz topic whose samples were used as the root set, followed by DomHits and finally HITS. This means that topic drift is smallest in DomTextHits.

Query	HITS/B&H	DomHits	DomTextHits	Mixed hubs?
"affirmative action"				Y
alcoholism				Y
"amusement park*"				Y
architecture				
bicycling				
blues				Y
classical guitar				
cheese				Y
cruises				Y
computer vision				Y
field hockey				
gardening				Y
graphic design				
Gulf war				Y
HIV				Y
lyme disease				Y
mutual fund*				Y
parallel architecture				
rock climbing				
recycling can*				Y
stamp collecting				
Shakespeare				Y
sushi				Y

Drift: high=, medium=, low/none=

Figure 19: For most of the Clever queries, we inspected the extent of drift subjectively. The new algorithms reduce drift in a number of cases. The last column shows whether mixed hubs were easily found among the top 40. This was very often the case.

Web content, and any extraneous document pulled in owing to drift would escape to a different class. Figure 18 clearly shows that the topical "purity" of the rootset is best upheld by DomTextHits, followed by DomHits, HITS being the worst in this regard.

Drift anecdotes: Subjective judgment about drift for most of the Clever queries have been summarized in figure 19, which seem to indicate that DomHits and DomTextHits reduce drift in a number of cases. However only a detailed user study can provide the final word on this issue.

5 Conclusion and future work

We have proposed a fine-grained view of hypertext based on the Document Object Model and presented new algorithms for topic distillation that integrates text, markup tag structure, and regular hyperlinks. We propose a new technique for aggregating and propagating micro-hub scores at a level determined both by textual proximity to the query as well as the graph structure being analyzed. We show that the resulting procedure reduces topic drift and is moreover capable of identifying and extracting hub regions (DOM subtrees) relevant to the query. In ongoing work we are seeking a better unification of the similarity-based and code-length-based pruning strategies into a single statistical generative model which will let us automate tuning if any is needed. We also need better diagnostic and visualization tools to understand the structure of topic-based communities.

Acknowledgement: Thanks to Amit Singhal for helpful discussions and the anonymous reviewers for proposing clarifications and additional experiments.

References

[1] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1–3):177–210, 1999. Online at http://www.cs.cmu.edu/~lafferty/ps/ml-final.ps.

[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] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide Web. In *WWW 10*, Hong Kong, May 2001. Online at http://www10.org/cdrom/papers/314.

[4] 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.

[5] C. Buckley, M. Mitra, J. Waltz, and C. Cardie. Using clustering and SuperConcepts within SMART: TREC6. In *Proceedings of the Sixth Text Retrieval Conference (TREC6)*, Gaithersburg, MD, 1998. National Institute of Standards and Technology (NIST). Online at http://www.cs.cornell.edu/home/cardie/papers/trec6-ipm.ps.

[6] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *WWW 10*, Hong Kong, May 2001. Online at http://www10.org/cdrom/papers/489.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] A. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Software available from http://www.cs.cmu.edu/~mccallum/bow/, 1998.

[15] J. M. Ponte and W. B. Croft. Text segmentation by topic. In *First European Conference on Research and Advanced Technology for Digitial Libraries*, pages 120–129, 1997. Online at http://cobar.cs.umass.edu/pubfiles/ir-103.ps.gz.

[16] J. C. Reynar. *Topic Segmentation: Algorithms and Applications*. PhD thesis, University of Pennsylvania, 1998. Online at http://www.cis.upenn.edu/~jcreynar/research.html.

[17] K. Richmond, A. Smith, and E. Amitay. Detecting subject boundaries within text: A language independent statistical approach. In *Empirical Methods in Natural Language Processing*, volume 2, Providence, RI, 1997. Online at http://www.ics.mq.edu.au/~einat/publications.html.

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

[19] J. Savoy. An extended vector processing scheme for searching information in hypertext systems. *Information Processing and Management*, 32(2):155–170, Mar. 1996.

[20] A. Singhal and M. Kaszkiel. A case study in web search using TREC algorithms. In *WWW 10*, Hong Kong, May 2001. Online at http://www10.org/cdrom/papers/317.

[21] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979. Online at http://www.dcs.gla.ac.uk/Keith/Preface.html.