Research Report

CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING

Dharmendra S. Modha
W. Scott Spangler

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099

Research Report RJ 10160 (95035)
Proceedings of ACM Hypertext Conference, May 30-June 3, 2000, San Antonio, Texas

IBM Research Division
Yorktown Heights, New York • San Jose, California • Zurich, Switzerland

CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING

Dharmendra S. Modha
W. Scott Spangler

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099

Research Report RJ 10160 (95035)
Proceedings of ACM Hypertext Conference, May 30-June 3, 2000, San Antonio, Texas
email: {dmodha,spangles}@almaden.ibm.com

October 7, 1999

ABSTRACT: Clustering separates unrelated documents and groups related documents, and is useful for discrimination, disambiguation, summarization, organization, and navigation of unstructured collections of hypertext documents. We propose a novel clustering algorithm that clusters hypertext documents using words (contained in the document), out-links (from the document), and in-links (to the document). The algorithm automatically determines the relative importance of words, out-links, and in-links for a given collection of hypertext documents. We annotate each cluster using six information nuggets: *summary*, *breakthrough*, *review*, *keywords*, *citation*, and *reference*. These nuggets constitute high-quality information resources that are representatives of the content of the clusters, and are extremely effective in compactly summarizing and navigating the collection of hypertext documents. We employ web searching as an application to illustrate our results.

Keywords: cluster annotation, feature combination, high-dimensional data, hyperlinks, sparse data, vector space model, toric k-means algorithm

1. INTRODUCTION

The World-Wide-Web has attained a gargantuan size [16] and a central place in the information economy of today. Hypertext is the *lingua franca* of the web. Moreover, scientific literature, patents, and law cases may be thought of as logically hyperlinked. Consequently, searching and organizing unstructured collections of hypertext documents is a major contemporary scientific and technological challenge.

Given a "broad-topic query" [13], a typical web search engine may return a large number of relevant documents. Without effective summarization, it is a hopeless and enervating task to sort through all the returned documents in search of high-quality, representative information resources. In this paper, we cluster the set of hypertext documents that are returned by a search engine in response to a broad-topic query into various clusters such that documents within each cluster are "similar" to each other. Clustering provides a way to organize a large collection of unstructured, unlabeled hypertext documents into labeled categories that are discriminated and disambiguated from each other. Furthermore, we capture the gist of each cluster using a scheme for cluster annotation that provides useful starting points for navigating/surfing in and around each cluster.

Ignoring the semantic information present in various HTML tags, a hypertext document has three different features: (i) the words contained in the document, (ii) out-links, that is, the list of hypertext documents that are *pointed to* or *cited* by the document, and (iii) the in-links, that is, the list of hypertext documents that *point to* or *cite* the document. We exploit all the three features to cluster a collection of hypertext documents. If two documents share one or more words, then we consider them to be semantically similar. Extending this notion to links, if two documents share one or more out-links or in-links, then we consider them to be similar as well. This simple observation is the key to the present paper. We propose a precise notion to capture the similarity between two hypertext documents along all the three features in an unified fashion. By exploiting our new similarity measure, we propose a geometric hypertext clustering algorithm: *the toric* k*-means* that extends the classical Euclidean k-means algorithm [12] and the spherical k-means algorithm [9, 19].

We annotate each cluster generated by the toric k-means algorithm using six information nuggets: *summary*, *breakthrough*, *review*, *keywords*, *citation*, and *reference*. The *summary* and the *keywords* are derived from words, the *review* and the *references* are derived from out-links, and the *breakthrough* and the *citations* are derived from in-links. These nuggets constitute high-quality, typical information resources, and are extremely effective in compactly summarizing and navigating the collection of hypertext documents.

The relative importances of the words, the out-links, and the in-links are tunable parameters in our algorithm. We propose an *adaptive* or *data-driven* scheme to determine these parameters with the goal of simultaneously improving the quality of all the six information nuggets for all the clusters.

Throughout the paper, we employ web searching as an application to illustrate our results. Anecdotally, when applied to the documents returned by AltaVista in responses to the queries *latex*, *abduction*, *guinea*, and *abortion*, our algorithm separates documents about "latex allergies" from those about "TeX& LaTeX," separates documents about "alien abduction" from those about "child abduction," separates documents about "Papua New Guinea," "Guinea Bissau," and "Guinea pigs" from each other, and separates documents about "pro-life" from those about "pro-choice", respectively.

We include directions for future work and a detailed literature survey at the end of the paper.

2. A GEOMETRIC ENCODING OF THE WEB

The Data Set Suppose we are given a collection of hypertext documents, say, $\mathcal{W}$. Let $\mathcal{Q}$ denote a subset of $\mathcal{W}$. In this paper, for example, $\mathcal{W}$ denotes the entire web, and $\mathcal{Q}$ denotes a small collection of hypertext documents retrieved by the search engine AltaVista (www.altavista.com) in response to a query. We are interested in clustering the hypertext documents in $\mathcal{Q}$. The situation of interest is depicted in Figure 1, where we have only shown those documents that are at most one out- or in-link away from the documents in $\mathcal{Q}$; in this paper, all other link information is discarded. The words contained in hypertext documents are not shown in Figure 1.

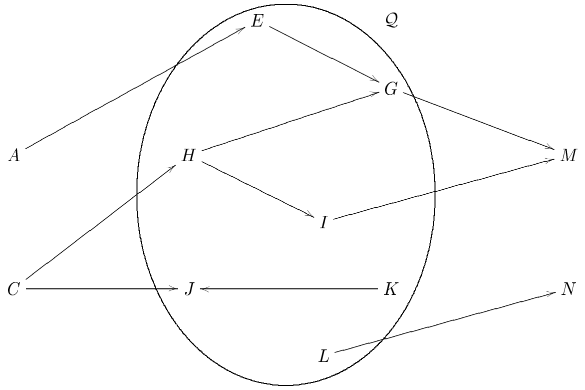

Figure 1: We are interested in clustering the set $\mathcal{Q} = \{\mathsf{E}, \mathsf{G}, \mathsf{H}, \mathsf{I}, \mathsf{J}, \mathsf{K}, \mathsf{L}\}$ of hypertext documents. The documents $\{\mathsf{A}, \mathsf{C}, \mathsf{M}, \mathsf{N}\}$ are not in $\mathcal{Q}$, but are hyperlinked to the documents in $\mathcal{Q}$.

We now extract useful features from $\mathcal{Q}$ and propose a geometric representation for these features. We will represent each hypertext document in $\mathcal{Q}$ as a triplet of unit vectors $(\mathbf{D}, \mathbf{F}, \mathbf{B})$. These component vectors are to be thought of as column vectors. The components $\mathbf{D}$, $\mathbf{F}$, and $\mathbf{B}$ will capture the information represented by the words contained in the document, the out-links originating at the document, and the in-links terminating at the document, respectively. We now show how to compute these triplets for each document in $\mathcal{Q}$.

Words The creation of the first component $\mathbf{D}$ is a standard exercise in text mining or information retrieval, see [21].

The basic idea is to construct a *word dictionary* of all the words that appear in any of the documents in $\mathcal{Q}$, and to prune or eliminate "function" words from this dictionary that do not help in semantically discriminating one cluster from another. For the present application, we eliminated those words which appeared in less than 2 documents, standard stopwords [10], and the HTML tags.

Suppose d unique words remain in the dictionary after such elimination. Assign an unique identifier from 1 to d to each of these words. Now, for each document x in $\mathcal{Q}$, the first vector $\mathbf{D}$ in the triplet will be a d-dimensional vector. The jth column entry, $1 \leq j \leq d$, of $\mathbf{D}$ is the number of occurrences of the jth word in the document x.

Out-links We now outline the creation of the second component $\mathbf{F}$. The basic idea is to construct an *out-link dictionary* of all the hypertext documents in $\mathcal{W} \setminus \mathcal{Q}$ that are pointed to by any of the documents in $\mathcal{Q}$. We also add each document in $\mathcal{Q}$ to the out-link dictionary. For example, in Figure 1, the out-link dictionary is $\{E, G, H, I, J, K, L, M, N\}$.

To treat nodes in $\mathcal{W} \setminus \mathcal{Q}$ and in $\mathcal{Q}$ in a uniform fashion, we add a self-loop from every document in $\mathcal{Q}$ to itself. Any document in the out-link dictionary that is not pointed to by at least two documents in $\mathcal{Q}$ provides no *discriminating* information. Hence, prune or eliminate all documents from the out-link dictionary that are pointed to by fewer than two documents (also counting the self-loops) in $\mathcal{Q}$. For example, in Figure 1, we eliminate the node N as it is pointed to by only L, but retain M as it is pointed to by both G and I. Similarly, we eliminate the nodes E, H, K, and L as they are not pointed to by any document in $\mathcal{Q}$ other than themselves, but retain G, I, and J as they are pointed to by at least one other document in $\mathcal{Q}$ and by themselves.

Suppose f unique nodes remain in the dictionary after such elimination. Assign an unique identifier from 1 to f to each of these documents. Now, for each document x in $\mathcal{Q}$, the second vector $\mathbf{F}$ in the triplet will be a f-dimensional vector. The jth column entry, $1 \leq j \leq f$, of $\mathbf{F}$ is the number of links to the jth retained node from the document x. We now present the out-link feature vectors for the example in Figure 1:

	E	G	H	I	J	K	L
G	1	1	1	0	0	0	0
I	0	0	1	1	0	0	0
J	0	0	0	0	1	1	0
M	0	1	0	1	0	0	0

In-links The creation of $\mathbf{B}$ is similar to that of $\mathbf{F}$; for completeness, we now briefly describe its construction. The basic idea is to construct an *in-link dictionary* of all the hypertext documents in $\mathcal{W} \setminus \mathcal{Q}$ that point to any of the documents in $\mathcal{Q}$. We also add each document in $\mathcal{Q}$ to the in-link dictionary.

To treat nodes in $\mathcal{W} \setminus \mathcal{Q}$ and in $\mathcal{Q}$ in a uniform fashion, we add a self-loop from every document in $\mathcal{Q}$ to itself. Any document in the in-link dictionary that does not point to at least two documents in $\mathcal{Q}$ provides no *discriminating* information. Hence, prune or eliminate all documents from the in-link dictionary that point to fewer than two documents (also counting the self-loops) in $\mathcal{Q}$.

Suppose b unique nodes remain in the dictionary after such elimination. Assign an unique identifier from 1 to b to each of these documents. Now, for each document x in $\mathcal{Q}$, the third vector $\mathbf{B}$ in the triplet will be a b-dimensional vector. The jth column entry, $1 \leq j \leq b$, of $\mathbf{B}$ is the number of links from the jth retained node to the document x.

Normalization Finally, for each document x in $\mathcal{Q}$, each of the three components $\mathbf{D}$, $\mathbf{F}$, and $\mathbf{B}$ is normalized to have a unit Euclidean norm, that is, their directions are retained and their lengths are discarded.

Torus We now briefly point out the geometry underlying our three-fold vector space models. Suppose that we have n documents in $\mathcal{Q}$. We denote each document triplet as

$$x_i = (\mathbf{D}_i, \mathbf{F}_i, \mathbf{B}_i), 1 \leq i \leq n.$$

Observe that, by construction, the component vectors $\mathbf{D}_i$, $\mathbf{F}_i$, and $\mathbf{B}_i$ all have unit Euclidean norm, and, hence, can be though of as points on the unit spheres S^d, S^f, and S^b in dimensions d, f, and b, respectively. Thus, each document triplet x_i lies on the product space of three spheres, which is a *torus*, see (www.treasure-troves.com/math/Torus.html). Furthermore, by construction, the individual entries of the component vectors $\mathbf{D}_i$, $\mathbf{F}_i$, and $\mathbf{B}_i$ are nonnegative, hence, the component vectors are in fact in the nonnegative orthants of R^d, R^f, and R^b, respectively. For notational convenience, we refer to the intersection of $(S^d \times S^f \times S^b)$ with the nonnegative orthant of R^{d+f+b} as $\mathcal{T}$.

AltaVista: Details Given a user query, we run it through AltaVista which typically returns a list of 200 URLs containing the keywords in the query. We crawl, and retrieve each of these 200 documents (those documents that could not be retrieved in 1 minute were discarded), and that becomes our set $\mathcal{Q}$. Next, we parse each of these documents, and construct the unpruned out-link dictionary. Finally, for each document in $\mathcal{Q}$, using queries of the form "link:URL" on AltaVista, we retrieve the URLs of top 20 documents that point to it. This constitutes our unpruned in-link dictionary. Observe that we do not need the actual documents in either the out- or the in-link dictionary. The set $\mathcal{Q}$ and the out- and the in-link dictionaries now become the inputs for the vector space model construction procedure described above.

query	n	d°	d	$d^{\diamond}$	n_d	f°	f	$f^{\diamond}$	n_f	$\hat{n}_f$	b°	b	$b^{\diamond}$	n_b	$\hat{n}_b$
latex	148	7059	2706	100.3	148	922	92	3.2	55	11	585	28	1.4	45	7
abortion	156	6205	2670	101.6	156	1286	144	3.3	67	10	662	58	1.8	64	9
guinea	146	6600	2814	100.0	146	1392	351	17.8	67	12	585	54	1.9	70	6
abduction	155	5967	2643	97.2	155	677	81	3.2	46	9	378	38	2.6	40	6
virus	146	6118	2627	111.6	146	2601	765	20.5	95	18	1191	100	3.8	70	14
"human rights"	157	7314	2800	113.6	157	1446	204	4.6	77	14	1369	99	2.9	71	12
dilbert	164	4584	1934	74.8	164	1257	173	3.8	80	4	385	15	1.3	45	5
terrorism	154	9824	4493	208.5	154	1762	242	6.1	74	18	675	47	2.0	51	14

Table 1: A note on notation: n represent the number of documents in $\mathcal{Q}$, d° and d are the number of words in the word-dictionary before and after elimination of function words, respectively, $d^{\diamond}$ is the average number of nonzero word counts per document, and n_d is the number of documents which contain at least one word after elimination. The symbols f°, f, $f^{\diamond}$, and n_f as well as the symbols b°, b, $b^{\diamond}$, and n_b have a similar meaning to their counterparts for the words. The symbols $\hat{n}_f$ and $\hat{n}_b$ are the number of documents in $\mathcal{Q}$ that are eventually retained in the final, pruned out-link and the in-link dictionaries, respectively.

Statistics In Table 1, for a number of queries, we present statistical properties of the three-fold vector space models.

High-dimensional By observing the d, f, and b values in Table 1, we see that, even after pruning, the word, out-link, and in-link dictionaries are very high-dimensional. Also, typically, d is the much larger than both f and b.

Sparse By observing the ratios $d^{\diamond}/d$, $f^{\diamond}/f$, and $b^{\diamond}/b$ in Table 1, we see that the vector space models are very sparse. A sparsity of 96% is typical for words, that is, on an average each document contains only 4% of the words in the word dictionary. Similarly, sparsities of 95–98% and 91–97% are typical for out- and in-links, respectively.

By observing the n_f and n_b values, we see that not all documents have nonzero out-link and in-link features vectors. This points once again to the sparse link topology that is holding the web together. Also, the variations in n_f and n_b values point to the fact that some "topics" or "communities" are more tightly coupled than others.

Importance of $\mathcal{W} \setminus \mathcal{Q}$ Finally, by observing the $\hat{n}_f$ and $\hat{n}_b$ values, we see that the number of nodes from the original set $\mathcal{Q}$ retained in the final pruned out-link and in-link dictionaries is rather small. In other words, the interconnection structure of the set $\mathcal{Q}$ with the rest of the web, namely, $\mathcal{W} \setminus \mathcal{Q}$, contains the vast majority of the link information in our feature vectors. This justifies our decision to include links between the documents in $\mathcal{Q}$ and the documents $\mathcal{W} \setminus \mathcal{Q}$.

3. TORIC k-MEANS ALGORITHM

A Measure of Similarity Given document triplets $x = (\mathbf{D}, \mathsf{F}, \mathbf{B})$ and $\tilde{x} = (\tilde{\mathbf{D}}, \tilde{\mathsf{F}}, \tilde{\mathbf{B}})$ on the torus T, we define a measure of similarity between them as a weighted sum of the inner products between the individual components. Precisely, we write

$$S(x, \tilde{x}) = \alpha_d \mathbf{D}^T \tilde{\mathbf{D}} + \alpha_f \mathsf{F}^T \tilde{\mathsf{F}} + \alpha_b \mathbf{B}^T \tilde{\mathbf{B}}, \tag{1}$$

where *weights* α_d, α_f, and α_b are nonnegative numbers such that

$$\alpha_d + \alpha_f + \alpha_b = 1.$$

Observe that for any two document triplets x and $\tilde{x}$, $0 \leq S(x, \tilde{x}) \leq 1$. Also, observe that if we set $\alpha_d = 1$, $\alpha_f = 0$, and $\alpha_b = 0$, then we get the classical cosine similarity between document vectors that has been widely used in information retrieval [21]. The parameters α_d, α_f, and α_b are tunable in our algorithm to assign different weights to words, outlinks, and in-links as desired. We will later discuss, in detail, the appropriate choice of these parameters.

Concept Triplets Suppose we are given n document vector triplets $x_1, x_2, \ldots, x_n$ on the torus T. Let $\pi_1, \pi_2, \ldots, \pi_k$ denote a partitioning of these document triples into k *disjoint* clusters. For each fixed $1 \leq j \leq k$, the *concept vector triplet* or concept triplet, for short, is defined as

$$\mathbf{c}_j = (\mathbf{D}_j^\star, \mathsf{F}_j^\star, \mathbf{B}_j^\star) \tag{2}$$

$$\mathbf{D}_j^\star = \frac{\sum_{x \in \pi_j} \mathbf{D}}{\| \sum_{x \in \pi_j} \mathbf{D} \|}, \; \mathsf{F}_j^\star = \frac{\sum_{x \in \pi_j} \mathsf{F}}{\| \sum_{x \in \pi_j} \mathsf{F} \|}, \; \mathbf{B}_j^\star = \frac{\sum_{x \in \pi_j} \mathbf{B}}{\| \sum_{x \in \pi_j} \mathbf{B} \|}. \tag{3}$$

where $x = (\mathbf{D}, \mathsf{F}, \mathbf{B})$ and $\| \cdot \|$ denotes the Euclidean norm. Observe that, by construction, each component of the concept triplet has unit Euclidean norm. The concept triplet $\mathbf{c}_j$ has the following important property. For any triplet $\tilde{x} = (\tilde{\mathbf{D}}, \tilde{\mathsf{F}}, \tilde{\mathbf{B}})$ on the torus T, we have from the Cauchy-Schwarz inequality that

$$\sum_{x \in \pi_j} S(x, \tilde{x}) \leq \sum_{x \in \pi_j} S(x, \mathbf{c}_j). \tag{4}$$

Thus, in an average sense, the concept triplet may be thought of as being the closest in S to all the document vector triplets in the cluster π_j.

We shall demonstrate that concept triplets contain valuable conceptual or semantic information about the clusters that is important in interpretation and annotation.

The Objective Function Motivated by (4), we measure the "coherence" or "quality" of each cluster π_j, $1 \le j \le k$, as

$$\sum_{\mathbf{x} \in \pi_j} S(\mathbf{x}, \mathbf{c}_j).$$

Observe that if all documents in a cluster are identical, then the average coherence of that cluster will have the highest possible value of 1, while if the document vectors in a cluster vary widely, then the average coherence will be small, that is, close to 0. We measure the quality of any given partitioning $\{\pi_j\}_{j=1}^k$ using the following *objective function*:

$$\sum_{j=1}^{k} \sum_{\mathbf{x} \in \pi_j} S(\mathbf{x}, \mathbf{c}_j). \tag{5}$$

Intuitively, the objective function measures the combined coherence of all the k clusters.

The Algorithm We would like to find k disjoint clusters $\pi_1^\dagger, \pi_2^\dagger, \cdots, \pi_k^\dagger$ such that the following is maximized:

$$\{\pi_j^\dagger\}_{j=1}^k = \arg\max_{\{\pi_j\}_{j=1}^k} \left(\sum_{j=1}^{k} \sum_{\mathbf{x} \in \pi_j} S(\mathbf{x}, \mathbf{c}_j) \right). \tag{6}$$

Even when only one of the parameters α_d, α_f, or α_b is nonzero, finding the optimal solution to the above maximization problem is known to be NP-complete. We now discuss an efficient and effective approximation algorithm: the *toric* k*-means* that may be thought of as a *gradient ascent* method.

Step 1 Start with an arbitrary partitioning of the document vectors, namely, $\{\pi_j^{(0)}\}_{j=1}^k$. Let $\{\mathbf{c}_j^{(0)}\}_{j=1}^k$ denote the concept triplets associated with the given partitioning. Set the index of iteration $t = 0$. The choice of the initial partitioning is quite crucial to finding a "good" local minima; for recent work on this area, see [2].

Step 2 For each document vector triplet $\mathbf{x}_i$, $1 \le i \le n$, find the concept triplet that is closest to $\mathbf{x}_i$. Now, for $1 \le j \le k$, compute the new partitioning $\{\pi_j^{(t+1)}\}_{j=1}^k$ induced by the old concept triplets $\{\mathbf{c}_j^{(t)}\}_{j=1}^k$:

$$\pi_j^{(t+1)} = \left\{ \mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^n : S(\mathbf{x}, \mathbf{c}_j^{(t)}) \ge S(\mathbf{x}, \mathbf{c}_\ell^{(t)}), 1 \le \ell \le k \right\}. \tag{7}$$

In words, $\pi_j^{(t+1)}$ is the set of all document vector triplets that are closest to the concept triplet $\mathbf{c}_j^{(t)}$. If it happens that some document triplet is simultaneously closest to more than one concept triplet, then it is randomly assigned to one of the clusters.

Step 3 Compute the new concept triplets $\{\mathbf{c}_j^{(t+1)}\}_{j=1}^k$ corresponding to the partitioning computed in (7) by using (2)-(3) where instead of π_j we use $\pi_j^{(t+1)}$.

Step 4 If some "stopping criterion" is met, then set $\pi_j^\dagger = \pi_j^{(t+1)}$ and set $\mathbf{c}_j^\dagger = \mathbf{c}_j^{(t+1)}$ for $1 \leq j \leq k$, and exit. Otherwise, increment t by 1, and go to step 2 above. An example of a stopping criterion is: Stop if the change in the objective function, between two successive iterations, is less than some specified threshold.

Shape of Clusters Clusters defined using (7) are known as *Voronoi* or *Dirichlet* partitions. The boundary between two clusters, say, $\pi_j^\dagger$ and $\pi_\ell^\dagger$, is the locus of all document triplets x on T satisfying:

$$S(x, \mathbf{c}_j^\dagger) = S(x, \mathbf{c}_\ell^\dagger).$$

If only one of the parameters α_d, α_f, or α_b is nonzero, then the above locus is a hypercircle on the corresponding sphere; when more than one parameters is nonzero, the locus is a hyperellipsoid. Thus, each cluster is a region on the surface of the underlying torus bounded by hyperellipsoids. In conclusion, the geometry of the torus plays an integral role in determining the "shape" and the "structure" of the clusters found by the toric k-means algorithm.

4. CLUSTER ANNOTATION AND INTERPRETATION

Suppose that we have clustered a hypertext collection $\mathcal{Q}$ into k clusters $\{\pi_j^\dagger\}_{j=1}^k$; let $\{c_j^\dagger\}_{j=1}^k$ denote the corresponding concept triplets. In this raw form, the clustering is of little use. We now use the concept triplets to interpret and annotate each cluster. *The process of seeking good cluster annotation will motivate the choice of the weights α_d, α_f, and α_b.*

Fix a cluster $\pi_j^\dagger$, $1 \leq j \leq k$. Let $\mathbf{c}_j^\dagger = (\mathbf{D}_j^\star, \mathbf{F}_j^\star, \mathbf{B}_j^\star)$ denote the corresponding concept triplet. We now show how to label the fixed cluster $\pi_j^\dagger$ using six different nuggets of information whose names have been inspired by their respective analogues in the scientific literature.

summary A *summary* is a document in $\pi_j^\dagger$ that has the most typical word feature vector amongst all the documents in the cluster. Formally, the *summary* is a document triplet $x = (\mathbf{D}, \mathbf{F}, \mathbf{B})$ whose word component $\mathbf{D}$ is closest in cosine similarity to $\mathbf{D}_j^\star$.

	query: virus, Cluster 1, size = 146	**query: “human rights,”** Cluster 1, size = 157
Keywords	viruse,anti,software,information,computer,update	human,international,unit,information,nation,report
Summary	Anti-Virus Tools (51)	Links To Other Human Rights Sources (40)
Review	SARC Virus EncyclopediaQ - Qm (19)	Derechos Human Rights - contact info (59)
Breakthrough	SARC Virus EncyclopediaXn - Xz (26)	United Nations Human Rights Website (22)
Reference	McAfee.com - The Place for Your PC	Derechos - Human Rights
Citation	Zaujimave linky	HUMAN RIGHTS REPORTING: Primary Web ···

	query: dilbert, Cluster 1, size = 165	**query: terrorism**, Cluster 1, size = 154
Keywords	adam,book,scott,comic,work,dogbert	terrorist,state,international,attack,bomb,security
Summary	DILBERT ZONE - scott adams past ··· (129)	US Policy on Terrorism..Part I* (21)
Review	DILBERT ZONE - dnrc sock puppets (103)	Terrorism Research Center: Counterterrorist ··· (34)
Breakthrough	July 1995: [BUBBA-L:26422] Re: Dilbert (121)	Terrorism Research Center: Terrorist Profiles (28)
Reference	Dilbert Zone	http://www.state.gov/www/global/terrorism/
Citation	Dilbert : On the Net 700 Sites!	Terrorism - U.S. News Net Links (116)

Table 2: By treating the entire set $\mathcal{Q}$ as one cluster, we present the corresponding six nuggets for each of the four queries: *virus*, *“human rights,” dilbert*, and *terrorism*. Every document that is in $\mathcal{Q}$ is followed by a parenthetic number that represents its rank in the documents returned by AltaVista. Every summary, review, and breakthrough is always followed by a parenthetic number, whereas the references or citations are followed by a parenthetic number only when applicable. Also, see: www.almaden.ibm.com/cs/people/dmodha/toric/toric.html

breakthrough A *breakthrough* is a document in $\pi_j^\dagger$ that has the most typical in-link feature vector amongst all the documents in the cluster. Formally, the *breakthrough* is a document triplet $x = (\mathbf{D}, \mathbf{F}, \mathbf{B})$ whose in-link component $\mathbf{B}$ is closest in cosine similarity to $\mathbf{B}_j^\star$.

review A *review* is a document in $\pi_j^\dagger$ that has the most typical out-link feature vector amongst all the documents in the cluster. Formally, the *review* is a document triplet $x = (\mathbf{D}, \mathbf{F}, \mathbf{B})$ whose out-link component $\mathbf{F}$ is closest in cosine similarity to $\mathbf{F}_j^\star$.

keywords *Keywords* for the cluster $\pi_j^\dagger$ are those words in the word dictionary that have the largest weight in $\mathbf{D}_j^\star$ compared to their respective weights in $\mathbf{D}_\ell^\star$, $1 \leq \ell \leq k, \ell \neq j$. *Keywords* are the most discriminating words in a cluster, and constitute an easy-to-interpret cluster signature.

citations *Citations* for the cluster $\pi_j^\dagger$ are those in-links in the in-link dictionary that have the largest weight in $\mathbf{B}_j^\star$ compared to their respective weights in $\mathbf{B}_\ell^\star$, $1 \leq \ell \leq k, \ell \neq j$. *Citations* represent the set of most typical links *entering* (the documents in) the given cluster.

references *References* for the cluster $\pi_j^\dagger$ are those out-links in the out-link dictionary that have the largest weight in $\mathbf{F}_j^\star$ compared to their respective weights in $\mathbf{F}_\ell^\star$, $1 \leq \ell \leq k, \ell \neq j$. *References* represent the set of most typical links *exiting* (from the documents in) the given cluster.

If we were interested in clustering a collection of not-hyperlinked text documents, then the *summary* and the *keywords* would constitute an adequate annotation. For hypertext collections, our annotation naturally extends the concepts of *summary* and the *keywords* from words to in-links and out-links as well. Observe that the *summary*, the *breakthrough*, and the *review* are meant to be primarily *descriptive* of the contents of the cluster, whereas the *keywords*, the *references*, and the *citations* are meant to be *discriminative* characteristics of the cluster. Also, observe that the *summary*, the *breakthrough*, and the *review* are, by definition, drawn from the set $\mathcal{Q}$; however, the *citations* and the *references* may or may not be in the set $\mathcal{Q}$.

Effectiveness of Annotation: Examples Suppose, for a moment, that we are not interested in clustering at all; in other words, suppose that we are interested in only one cluster, that is, $k = 1$. Even in this case, the six nuggets described above are meaningful, and often capture the top information resources present in $\mathcal{Q}$.

For example, in Table 2, by treating the entire set $\mathcal{Q}$ as one cluster, we present the six nuggets for each of the four queries: *virus*, "*human rights*," *dilbert*, and *terrorism.* As even a casual glance reveals, the annotation indeed captures the top information resources in every case, and provides a valuable starting point for navigating the documents surrounding the cluster.

Furthermore, note that, in Table 2, every document that is in $\mathcal{Q}$ is followed by a parenthetic number that represents its rank in the documents returned by AltaVista. For example, for the query *virus* the *summary* is "Anti-Virus Tools (51)" meaning that it was the fifty-first document returned by AltaVista. By observing these parenthetic numbers, we can conclude that, in almost every case, the top resources found by our annotation were not amongst the top documents returned by AltaVista. For example, for the query "*human rights*," our annotation finds the "United Nations Human Rights Website" as a *breakthrough*, while it is the twenty-second document returned by AltaVista. Thus, in its simplest form, our annotation provides a rearrangement of the results returned by AltaVista. Such rearrangements are important, since user studies have shown that the users rarely go beyond the top 20 documents returned by a web search engine [22].

5. CHOICE OF THE WEIGHTS

In the end, it is really the annotation of each cluster in terms of the above six nuggets that is presented to the end user. Hence, arguably, a natural goal of hypertext clustering is to obtain the most descriptive and discriminative nuggets possible. Clearly, if we use $\alpha_d = 1$ and $\alpha_f = \alpha_b = 0$, then we get a good discrimination amongst the resulting clusters in the feature space constituted by the words. Consequently, we obtain good *summary* and *keywords* for the resulting clusters. Similarly, if we use $\alpha_f = 1$ and $\alpha_d = \alpha_b = 0$, then we can obtain good *review* and *references* for the resulting clusters. Finally, if we use $\alpha_b = 1$ and $\alpha_d = \alpha_f = 0$, then we can obtain good *breaktrhough* and *citations* for the resulting clusters. To truly and completely exploit the hypertext nature of the given document collection, we would like all the six nuggets to be of good quality *simultaneously*. This can be achieved by judiciously selecting the parameters α_d, α_f, and α_b. We now provide a formal framework for this choice.

Throughout this section, fix the number of clusters $k \geq 2$. As before, let α_d, α_f, and α_b be nonnegative numbers that sum to 1. Geometrically, these parameters lie on a planar triangular region, say, Δ_0, that is shown in Figure 2. For brevity, we write $\boldsymbol{\alpha} = (\alpha_d, \alpha_f, \alpha_b)$. Let $\Pi(\boldsymbol{\alpha}) = \{\pi_j^\dagger\}_{j=1}^k$ denote the partitioning obtained by running the toric k-means algorithm with the parameter values α_d, α_f, and α_b. From the set of all possible clusterings $\{\Pi(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \Delta_0\}$, we would like to select a partitioning that yields the *best* cluster annotations. Towards this goal, we now introduce a figure-of-merit for evaluating and comparing various clusterings.

Fix a clustering $\Pi(\boldsymbol{\alpha})$. For the given clustering, the *summary*, which is a descriptive characteristics, for each of the clusters will be good if each cluster is as coherent as possible in the word feature space, that is, if the following is maximized:

$$\Gamma_d(\boldsymbol{\alpha}) \equiv \Gamma_d(\Pi(\boldsymbol{\alpha})) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in \pi_j} D^T D_j^\star,$$

where $x = (\mathbf{D}, \mathsf{F}, \mathbf{B})$. Furthermore, the *keywords*, which are a discriminative characteristics, will be good if the following is minimized:

$$\Lambda_d(\boldsymbol{\alpha}) \equiv \Lambda_d(\Pi(\boldsymbol{\alpha})) = \frac{1}{k-1} \sum_{j=1}^{k} \sum_{\mathbf{x} \in \pi_j} \sum_{\ell=1, \ell \neq j}^{k} D_j^T D_\ell^\star,$$

where $x = (\mathbf{D}, \mathsf{F}, \mathbf{B})$. Intuitively, $\Gamma_d(\boldsymbol{\alpha})$ and $\Lambda_d(\boldsymbol{\alpha})$ capture the *average within cluster coherence* and *average between cluster coherence*, respectively, of the clustering $\Pi(\boldsymbol{\alpha})$ in the word feature space. The *summary* and the *keywords* both will be good if the following ratio is maximized:

$$\mathcal{Q}_d(\boldsymbol{\alpha}) \equiv \mathcal{Q}_d(\Pi(\boldsymbol{\alpha})) = \begin{cases} \left(\frac{\Gamma_d(\boldsymbol{\alpha})}{\Lambda_d(\boldsymbol{\alpha})}\right)^{n_d/n} & \text{if } \Lambda_d(\boldsymbol{\alpha}) > 0, \\ 1 & \text{if } \Lambda_d(\boldsymbol{\alpha}) = 0, \end{cases} \tag{8}$$

where n_d denotes the number of document triplets in $\mathcal{Q}$ that have a non-zero word feature vector; see, for example, Table 1. In the case that $\Lambda_d(\boldsymbol{\alpha}) = 0$, the clusters are *perfectly separated* in the word feature space.

The quantities $\Gamma_f(\boldsymbol{\alpha})$, $\Lambda_f(\boldsymbol{\alpha})$, $\Gamma_b(\boldsymbol{\alpha})$, $\Lambda_b(\boldsymbol{\alpha})$, $\mathcal{Q}_f(\boldsymbol{\alpha})$, and $\mathcal{Q}_b(\boldsymbol{\alpha})$ are defined in a similar fashion. The quantity $\mathcal{Q}_f(\boldsymbol{\alpha})$ should be maximized to obtain good quality *review* and *references*, and the quantity $\mathcal{Q}_b(\boldsymbol{\alpha})$ should be maximized to obtain good quality *breakthrough* and *citations*.

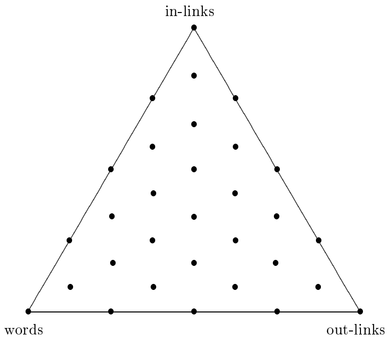

Figure 2: The triangular region Δ_0 formed by the intersection of the plane $\alpha_d + \alpha_f + \alpha_b = 1$ with the nonnegative orthant of R^3. The left-vertex, the right-vertex, and the top-vertex of the triangle corresponds to the points $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, respectively.

We are now ready to present a scheme to select the optimal parameter tuple $\boldsymbol{\alpha}^\dagger$ and the corresponding clustering $\Pi(\boldsymbol{\alpha}^\dagger)$.

Step 1 Theoretically, we would like to run the toric k-means algorithm for every parameter tuple $\boldsymbol{\alpha}$ in:

$$\Delta_0 = \{\boldsymbol{\alpha} : \alpha_d + \alpha_f + \alpha_b = 1, \alpha_d, \alpha_f, \alpha_b \geq 0\}. \tag{9}$$

In practice, we replace the region Δ_0 in (9) by a finite number of points on a discrete grid that are graphically shown using the symbol $\bullet$ in Figure 2.

Step 2 To obtain good cluster annotations in terms of all the six nuggets, we would like to simultaneously maximize $\mathcal{Q}_{\mathrm{d}}$, $\mathcal{Q}_{\mathrm{f}}$, and $\mathcal{Q}_{\mathrm{b}}$. Hence, we select the parameters $\boldsymbol{\alpha}^{\dagger}$ as the solution of the following maximization problem:

$$\boldsymbol{\alpha}^{\dagger} = \arg\max_{\boldsymbol{\alpha} \in \Delta} \left[\mathcal{Q}_{\mathrm{d}}(\boldsymbol{\alpha}) \times \mathcal{Q}_{\mathrm{f}}(\boldsymbol{\alpha}) \times \mathcal{Q}_{\mathrm{b}}(\boldsymbol{\alpha})\right], \tag{10}$$

where we now define the region Δ. First, we need some notation.

$$\begin{aligned}
\mathrm{R}_{\mathrm{d}} &= \{\boldsymbol{\alpha} \in \Delta_0 : \Lambda_{\mathrm{d}}(\boldsymbol{\alpha}) = 0\} \\
\mathrm{R}_{\mathrm{f}} &= \{\boldsymbol{\alpha} \in \Delta_0 : \Lambda_{\mathrm{f}}(\boldsymbol{\alpha}) = 0\} \\
\mathrm{R}_{\mathrm{b}} &= \{\boldsymbol{\alpha} \in \Delta_0 : \Lambda_{\mathrm{b}}(\boldsymbol{\alpha}) = 0\} \\
\Delta_3 &= \mathrm{R}_{\mathrm{d}} \cap \mathrm{R}_{\mathrm{f}} \cap \mathrm{R}_{\mathrm{b}} \\
\Delta_2 &= ((\mathrm{R}_{\mathrm{d}} \cap \mathrm{R}_{\mathrm{f}}) \cup (\mathrm{R}_{\mathrm{d}} \cap \mathrm{R}_{\mathrm{b}}) \cup (\mathrm{R}_{\mathrm{f}} \cap \mathrm{R}_{\mathrm{b}})) \setminus \Delta_3 \\
\Delta_1 &= (\mathrm{R}_{\mathrm{d}} \cup \mathrm{R}_{\mathrm{f}} \cup \mathrm{R}_{\mathrm{b}}) \setminus \Delta_2
\end{aligned}$$

We now define the region Δ as follows:

$$\Delta = \begin{cases} \Delta_3 & \text{if } \Delta_3 \neq \phi, \\ \Delta_2 & \text{elseif } \Delta_2 \neq \phi, \\ \Delta_1 & \text{elseif } \Delta_1 \neq \phi, \\ \Delta_0 & \text{otherwise.} \end{cases}$$

We now intuitively explain the reasoning behind the above definitions. The regions R_{d}, R_{f}, and R_{b} denote the set of parameters for which the corresponding clusterings perfectly separate the document triplets in the word, out-link, and in-link feature spaces, respectively. The region Δ_3 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets in *all* the three feature spaces. Clearly, if such clusterings are available, that is, if Δ_3 is not empty, then we would prefer them. Hence, we set $\Delta = \Delta_3$, if $\Delta_3 \neq \phi$. The region Δ_2 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets along *two*, but not all three, feature spaces. In the case that Δ_3 is empty, we prefer clusterings in Δ_2. Now, the region Δ_1 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets along *one and only one* of the three feature spaces. In the case that Δ_3 and Δ_2 are both empty, we prefer the clusterings in Δ_1. Finally, Δ_0 which is the entire triangular region in Figure 2 is the default choice when Δ_3, Δ_2, and Δ_1 are all empty. In practice, we have found that Δ_3 and Δ_2 are usually empty, and, hence, for most data sets, we expect the Δ to be either Δ_1 or Δ_0.

Step 3 Let $\Pi(\boldsymbol{\alpha}^\dagger)$ denote the optimal partitioning obtained by running the toric k-means algorithm with $\boldsymbol{\alpha}^\dagger$.

To illustrate the above scheme, we now present the Q_d, Q_f, Q_b, and $T = Q_d \times Q_f \times Q_b$ values for various parameter tuples, where $\mathcal{Q}$ is the the set of documents returned by AltaVista in response to the query *guinea* and $k = 3$.

α_d	α_f	α_b	Q_d	Q_f	Q_b	T
0.990	0.010	0.000	4.20	4.40	3.13	58.18
0.010	0.990	0.000	3.61	6.45	3.24	75.65
0.010	0.000	0.990	3.92	5.92	10.09	234.94
0.010	0.495	0.495	3.73	11.35	7.40	314.55

The first, second, and the third rows correspond to clustering primarily along words, out-links, and in-links, respectively, while the fourth row corresponds to the clustering corresponding to the optimal parameter tuple. It can be seen that the optimal clustering achieves significantly larger T value than clusterings which cluster only along one of the three features. In practice, the larger T value often translates into superior cluster annotation and a better clustering.

6. RESULTS: THE PROOF IS IN THE PUDDING

In Table 3, we present the parameter tuples obtained by solving the maximization problem in (10) for each of the four queries: *latex*, *abduction*, *guinea*, and *abortion*.

query	k	Δ	$\alpha_d^\dagger$	$\alpha_f^\dagger$	$\alpha_b^\dagger$
latex	2	Δ_1	0.010	0.000	0.990
abduction	2	Δ_1	0.495	0.010	0.495
guinea	3	Δ_0	0.010	0.495	0.495
abortion	3	Δ_0	0.010	0.495	0.495

Table 3: The set of documents returned by AltaVista for each of the four queries: *latex*, *abduction*, *guinea*, and *abortion* are clustered into k clusters. For each query, we determine the optimal parameter tuple $\boldsymbol{\alpha}^\dagger = (\alpha_d^\dagger, \alpha_f^\dagger, \alpha_b^\dagger)$ by solving the maximization problem in (10). For queries *abduction* and *guinea*, all the three sets Δ_3, Δ_2, and Δ_1 turn out to be empty, and, hence, $\Delta = \Delta_0$. For queries *latex* and *abortion*, the two sets Δ_3 and Δ_2 turn out to be empty, but the set Δ_1 is not empty, and, hence, $\Delta = \Delta_1$.

In Table 4, we present the optimal clusterings corresponding to the optimal parameter tuples in Table 3 for the queries *latex*, *abduction*, *guinea*, and *abortion*. It can be seen from Table 4 that (i) the set of documents corresponding to *latex* is neatly partitioned into "latex allergies" cluster and into "$\TeX$& $\LaTeX$" cluster; (ii) the set of documents corresponding to *abduction* is neatly partitioned into "alien abduction" cluster and into "child abduction" cluster; (iii) the set of documents corresponding to *guinea* is neatly partitioned into "Papua New Guinea," "Guinea Bissau," and "Guinea pigs" clusters; and, finally, (iv) the set of documents corresponding to *abortion* is neatly partitioned into two "pro-life" cluster and one "pro-choice" clusters.

7. FUTURE WORK

Throughout this paper, we assumed that the number of clusters k is given; however, an important future problem is to automatically determine the number of clusters in an adaptive or data-driven fashion using information-theoretic criteria such as the MDL principle.

To determine the optimal parameter tuple $\boldsymbol{\alpha}^\dagger$, in this paper, we run the toric k-means algorithm for every $\boldsymbol{\alpha}$ on a certain discrete grid in the triangular region Δ_0. We are currently investigating a computationally efficient gradient ascent procedure for computing the optimal parameter tuple $\boldsymbol{\alpha}^\dagger$. The basic idea is to combine the optimization problems in (10) and in (6) into a single problem that can be solved using an iterative hill-climbing heuristic.

In this paper, we have employed the new similarity measure S in the k-means algorithm; it is also possible to use it with a graph-based algorithm such as the complete link method or with hierarchical agglomerative clustering algorithms.

8. LITERATURE REVIEW

Document clustering using only textual features such as words or phrases has been extensively studied; for a detailed review of various k-means type algorithms, graph theoretical algorithms, and hierarchical agglomerative clustering algorithms, see Rasmussen [19] and Willet [26].

By treating the references made by one scientific paper (or a patent or a law case) to another as a logical hyperlink, one can interpret scientific literature (or patents or law cases) as a hypertext document collection. Citation analysis was developed as a tool to identify core sets or clusters of articles, authors, or journals of particular fields of study by using the logical hyperlinks between scientific papers, see White and McCain [25] and Small [23]. Larson [15] has proposed using citation analysis with multidimensional scaling to identify clusters in the web. Recently, Kleinberg [13] has extended citation analysis to web searching. In response to a broad-topic query, his algorithm HITS produces two distinct but inter-related types of pages: *authorities* (highly-cited pages) and *hubs* (pages that cite many authorities). HITS only uses the link topology; CLEVER refines HITS to include query word matches within anchor text [5]. For a highly accessible treatment of the use of citation analysis in web searching, see [4]. The fundamental motivation behind this paper was to seek a synthesis of text-based clustering algorithms in [19, 26, 9] and links-based eigen-analysis in [13, 5, 4]. Conceptually, our *references*

	query: latex, Cluster 1, size = 82	**query: abduction**, Cluster 1, size = 85
Keywords	latex,glove,request,allergy,balloon,rubber	alien,ufo,story,experience,hip,generator
Summary	Latex Allergy Injuries - The Law Offices (122)	Wiendog's Alien Abduction Page (192)
Review	Enlanger Latex Mattresses - 1(800)FloBeds (188)	What is an alien abduction experience? (116)
Breakthrough	Latex Allergy Injuries - The Law Offices (122)	Alien Abduction Experience and Research (60)
Reference	www.FloBeds.com 1(800)FloBeds	ABIOGENESIS - POWER OF CREATION
Citation	LATEX ALLERGY	Orthopaedic Rehabilitation. Abduction Pillows (141)

	query: latex, Cluster 2, size = 66	**query: abduction**, Cluster 2, size = 71
Keywords	tex,document,package,command,math,postscript	child,children,parent,international,information,court
Summary	Intro to TeX; LaTeX; BibTeX and SliTeX (78)	England & Wales - International ··· Abduction (58)
Review	TeX and LaTeX (1)	A Halloween Abduction prevention page (105)
Breakthrough	Peter's TeX/LaTeX/LaTeX2e/LaTeX3 Page (38)	Iran - International Parental Child Abduction (159)
Reference	TeX Frequently Asked Questions	Islamic Family Law - International ··· Abduction (3)
Citation	PROGRAMMING: bookmarks	Child Abduction - Divorce Support Net Links

	query: guinea, Cluster 1, size = 92	**query: abortion**, Cluster 1, size = 72
Keywords	papua,country,png,weather,service,unit	life,pro,birth,partial,issue,request
Summary	Papua New Guinea Map (91)	Abortion (OU CALL) (79)
Review	Weather ··· Papua New Guinea Forecast (17)	Medical Misinformation About Abortion (103)
Breakthrough	@datec ··· Papua New Guinea (46)	Resource: Abortion-A Decision for Death (64)
Reference	@datec Internet - Papua New Guinea	National Right to Life Committee Main Page
Citation	PAPUA NEW GUINEA ORCHID NEWS	http://www.learnusa.org/articles/

	query: guinea, Cluster 2, size = 34	**query: abortion**, Cluster 2, size = 45
Keywords	pig,pigs,request,cavy,nance,live	women,cancer,baby,pregnancy,breast,heal
Summary	Guinea Pig Links (196)	Project Rachel; Post-Abortion Healing ··· (21)
Review	Todd's Guinea Pig Hutch (6)	Abortion; The Pontifical Academy for Life (135)
Breakthrough	Greg's Guinea Pigs (40)	Ohio Abortion Statistics (102)
Reference	Todd's Guinea Pig Hutch (6)	Life Institute-Proclaiming The Gospel of Life ···
Citation	OinkerNet & Guinea Pigs Worldwide!	Abortion References, Statistics; Study; Research ···

	query: guinea, Cluster 3, size = 20	**query: abortion**, Cluster 3, size = 39
Keywords	bissau,travel,information,embassy,island,world	reproductive,action,error,clinic,information,caral
Summary	Guinea Bissau @ Travel Notes (r). (70)	California Abortion & Reproductive (CARAL) (138)
Review	Papua New Guinea @ Travel Notes (r). (160)	California Abortion & Reproductive (CARAL) (35)
Breakthrough	Guinea-Bissau; with National Anthem ··· (23)	China: Abortion (43)
Reference	Country Information @ ··· Online Travel Guide.	Reproductive Health & Rights Center: Home Page
Citation	National Anthems of the World.	Dr. Pranikoff's Gyn Web Library - Abortion

Table 4: By running the toric k-means algorithm with the respective optimal parameter tuples in Table 3, we cluster the set of documents $\mathcal{Q}$ returned by AltaVista in response to the queries *latex*, *abduction*, *guinea*, and *abortion* into k = 2, 2, 3, and 3 clusters, respectively. Also, see: www.almaden.ibm.com/cs/people/dmodha/toric/toric.html

and *breakthrough* are analogous to *authorities*, and our *citations* and *review* are analogous to *hubs*.

Hypertext has been used to improve information retrieval. Salton [20] has proposed using bibliographic information, that is, out-links or references, for improving retrieval performance. The basic idea is to extract important terms from cited documents and to add these non-local terms to the citing document. This line of investigation and its variants has been explored in Kwok [14], Croft and Turtle [8], Frei and Steiger [11], and, most recently, in Chakrabarti, Dom, and Indyk [3]. Our work differs from this body of work in the important aspect that we consider the out-links and the in-links as first-class features in their own right and do not use non-local terms from either the cited or citing documents. Furthermore, this body of work has not focussed on hypertext clustering which is the problem of interest in this paper.

Botafogo [1] has proposed a graph-based algorithm for clustering hypertext that uses link information but no textual information; he proposed the number of independent paths between nodes as a measure of similarity. Mukherjea, Foley, and Hudson [17] have proposed using content- and structure-based algorithms for interactive clustering of hypertext. In their model, the user precisely specifies her information need, for example, all nodes containing some content or all graphical substructures, and, hence, unlike ours, theirs is not an automated clustering methodology.

Weiss et al. [24] combined information about document contents and hyperlink structures to automatically cluster hypertext documents. While our work is closest in spirit to [24], the two works are distinct in the choice of the algorithms, the underlying simiarity metrics, and the cluster naming or annotation scheme. In particular, [24] uses the complete link algorithm, while we develop a variant of the k-means algorithm. The complete link algorithm is quadratic-time complexity in the number of documents, while our method is linear-time complexity in the number of documents. Furthermore, their measure of similarity between two documents does not constitute a valid metric, and, hence, is not useful in a geometric setting like ours. Finally, our cluster annotation scheme has no analogue in [24].

Previously, Pirolli, Pitkow, and Rao [18] have combined both the link "topology and textual similarity between items as well as usage data collected by servers and page meta-information like title and size". [18] did not treat link topology and textual similarity differently as we do, but rather represented each hypertext document as a single vector of all these features. They left the problem of automatically categorizing hypertext documents using their feature space to future work. Chen [6] has proposed generalized similarity analysis that combines hypertext linkage, content similarity, and browsing patterns or usage. Chen and Czerwinski [7] have exploited generalized similarity analysis alongwith latent semantic indexing and pathfinder network scaling to develop an integrated framework for spatial organization of information and for browsing and searching. Their results are complementary to ours.

References

[1] Botafago, R. A. Cluster analysis for hypertext systems. In *ACM SIGIR* (1993).

[2] Bradley, P., and Fayyad, U. Refining initial points for k-means clustering. In *ICML* (1998), pp. 91–99.

[3] Chakrabarti, S., Dom, B. E., and Indyk, P. Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD* (1998).

[4] Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J. M., and Gibson, D. Hypersearching the web. *Scientific American* (June 1999).

[5] Chakrabarti, S., Dom, B. E., Raghavan, P., Rajagopalan, S., Gibson, D., and Kleinberg, J. Automatic resource compilation by analyzing hyperlink structure and associated text. In *WWW7* (1998).

[6] Chen, C. Structuring and visualizing the www by generalized similarity analysis. In *ACM Hypertext* (1997).

[7] Chen, C., and Czerwinski, M. From latent semantics to spatial hypertext–An integrated approach. In *ACM Hypertext* (1998).

[8] Croft, W. B., and Turtle, H. R. A retrieval model for incorporating hypertext links. In *ACM Hypertext* (1989).

[9] Dhillon, I. S., and Modha, D. S. Concept decompositions for large sparse text data using clustering. Tech. Rep. RJ 10147 (95022), IBM Almaden Research Center, July 8, 1999.

[10] Frakes, W. B., and Baeza-Yates, R. *Information Retrieval: Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[11] Frei, H. P., and Steiger, D. Making use of hypertext links when retrieving information. In *ACM European Conference on Hypertext* (1992).

[12] Hartigan, J. A. *Clustering Algorithms.* Wiley, 1975.

[13] Kleinberg, J. Authoritative sources in a hyperlinked environment. In *ACM-SIAM SODA* (1998).

[14] Kwok, K. L. A probabilistic theory of indexing and similarity measure based on cited and citing documents. *J. Amer. Soc. Inform. Sci.* (1985), 342–351.

[15] LARSON, R. Bibliometric of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting Amer. Soc. Info. Sci.* (1996).

[16] LAWRENCE, S., AND GILES, C. L. Searching the World Wide Web. *Science 280*, 5360 (1998), 98.

[17] MUKHERJEA, S., FOLEY, J. D., AND HUDSON, S. E. Interactive clustering for navigating in hypermedia systems. In *ACM Hypertext* (1994).

[18] PIROLLI, P., PITKOW, J., AND RAO, R. Silk from sow's ear: Extracting usable structures from the web. In *ACM SIGCHI Human Factors Comput.* (1996).

[19] RASMUSSEN, E. Clustering algorithms. In *Information Retrieval: Data Structures and Algorithms* (1992), W. B. Frakes and R. Baeza-Yates, Eds., Prentice Hall, Englewood Cliffs, New Jersey, pp. 419–442.

[20] SALTON, G. Associative document retrieval techniques using bibliographic information. *J. ACM* (1963), 440–457.

[21] SALTON, G., AND MCGILL, M. J. *Introduction to Modern Retrieval.* McGraw-Hill Book Company, 1983.

[22] SILVERSTEIN, C., HENZINGER, M., MARAIS, J., AND MORICZ, M. Analysis of a very large AltaVista query log. Tech. Rep. 1998-014, Compaq Systems Research Center, Palo Alto, CA, October 1998.

[23] SMALL, H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Amer. Soc. Inform. Sci.* (1973), 265–269.

[24] WEISS, R., VELEZ, B., SHELDON, M. A., NAMPREMPRE, C., SZILAGYI, P., DUDA, A., AND GIFFORD, D. K. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *ACM Hypertext* (1996).

[25] WHITE, H. D., AND MCCAIN, K. W. Bibliometrics. *Annual Review of Information Science and Technology 24* (1989), 119–186.

[26] WILLET, P. Recent trends in hierarchic document clustering: a critical review. *Inform. Proc. & Management* (1988), 577–597.