Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections

Douglass R. Cutting[1] David R. Karger[1,2] Jan O. Pedersen[1]

Abstract

The Scatter/Gather document browsing method uses fast document clustering to produce table-of-contents-like outlines of large document collections. Previous work [1] developed linear-time document clustering algorithms to establish the feasibility of this method over moderately large collections. However, even linear-time algorithms are too slow to support interactive browsing of very large collections such as Tipster, the DARPA standard text retrieval evaluation collection. We present a scheme that supports constant interaction-time Scatter/Gather of arbitrarily large collections after near-linear time preprocessing. This involves the construction of a *cluster hierarchy*. A modification of Scatter/Gather employing this scheme, and an example of its use over the Tipster collection are presented.

1 Background

Our previous work on Scatter/Gather [1] has shown that document clustering can be used as a first-class tool for browsing large text collections. *Browsing* is distinguished from *search* because it is query-free. We posit situations in which the user has an information need that is either too general or too vague to be usefully expressed as a query is some search language. For example, the user may not be familiar with the vocabulary appropriate for describing a topic of interest, or may not wish to commit himself to a particular choice of words. Indeed, the user may not be looking for anything specific at all, but rather may wish to explore the general information content of the collection. In this context an information access system can still provide useful assistance by providing a navigable collection outline that suggests to the user both overall contents and a method for focussing attention on some coherent subset.

1.1 Scatter/Gather

In the Scatter/Gather browsing paradigm attention is always directed towards a *focus set* of documents potentially interesting to the user. Initially the focus set may be an entire document collection. The focus set is clustered into a small number of topic-coherent subsets. These clusters are summarized to form a "table of contents" which outlines the focus set. The user can then identify and select those clusters which appear most interesting, defining a new, smaller focus set which is the union of the selected clusters. The indicated subcollection becomes the focus set, and the process repeats.

Cluster summaries are not single-phrase labels, as one might expect to see in a hand-built outline, but rather suggestive text computed automatically from documents in the cluster. The current implementation offers two types of information. Both are based upon the *profile* of the cluster, a vector of weights reflecting the words appearing in the cluster's documents. The first is a list of "topical" words, those with high weights in the cluster's profile. The second is the titles of a few "typical" documents in the cluster.

Scatter/Gather is not envisioned as a stand-alone information access tool. Rather it is best used in tandem with search methods, such as a boolean search or similarity search. This is in analogy to paper reference books, which offer two access modes, a table of contents in the front for browsing and an index in the back for more directed searches. We anticipate that Scatter/Gather will not necessarily be used to find particular documents, but will instead, by giving exposure to the vocabulary presented in cluster summaries, help guide complementary search methods. For example, a cluster's profile may be used as a similarity search vector in a query against the entire collection. Conversely, Scatter/Gather may also be used to organize the results of

[1] Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304
[2] Stanford University

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery.
16th Ann Int'l SIGIR'93/Pittsburgh PA USA-6/93
©1993 ACM

word-based queries which retrieve too many documents.

1.2 A Scatter/Gather Session

Figure 1 summarizes a sample Scatter/Gather session over a text collection consisting of about 5000 articles posted to the *New York Times News Service* during the month of August 1990. Here, to simplify the figure, we manually assigned single-word labels based on the cluster summaries.

Suppose the user's information need is to determine generally what happened that month. It would clearly be difficult to construct a word-based query that effectively represents this information need because no specific topic description is available. The user might consider general topics, such as "international events", but again that topic description would not be effective because articles concerning international events need never use those words.

With Scatter/Gather, rather than being forced to provide search terms, the user is presented with a set of clusters—an outline of the collection. He need only select those clusters which seem potentially relevant to the topic of interest. In the example, the big stories of the month are immediately obvious from the initial scattering: Iraq invades Kuwait, and Germany considers reunification. This leads the user to focus on international issues: he selects the 'Kuwait' and 'Germany' and 'Oil' clusters. These three clusters are gathered together to form a smaller focus set.

This smaller focus set is then reclustered on the fly to produce eight new clusters covering the reduced collection. Since the reduced collection contains a subset of the articles, these new clusters reveal a finer level of detail than the original eight. The articles on the Iraqi invasion and some of the 'Oil' articles have now been separated into clusters discussing the U.S. military deployment, the effects of the invasion upon the oil market, and one which is about hostages in Kuwait.

The user feels his understanding of these large stories is adequate, but wishes to find out what happened in other corners of the world. He selects the 'Pakistan' cluster, which also contains other foreign political stories, and a cluster containing articles about Africa. This reveals a number of specific international situations as well as a small collection of miscellaneous international articles. The user thus learns of a coup in Pakistan and about hostages being taken in Trinidad, stories otherwise lost among the major stories of that month.

2 The Problem

Essential for the Scatter/Gather browsing paradigm are fast document clustering and effective cluster summarization. Previous work in document clustering generally concentrated on algorithms whose running time is quadratic in the collection size (*e.g.* the classic SLINK single-linkage clustering algorithm [4]). Quadratic running time is clearly too costly for interactive manipulation of the collections we envision, containing thousands of documents, possibly requiring days or even months to perform a single clustering. In contrast, the linear time algorithms previously presented[1] reduce the time required to only a few minutes (approximately 3000 documents per minute on a Sun Microsystems SPARCStation 2 [1]), fast enough for moderately large collections and the results of most broad word-based queries. However, linear-time clustering is too slow to support interactive browsing of very large document collections. This is particularly apparent when one considers applying Scatter/Gather to the Tipster collection[2], a DARPA standard for text retrieval evaluation, which contains about three-quarters of a million documents. At 3000 documents per minute, this requires around 4 hours to scatter, which is far too long to be considered interactive.

To achieve good interactive performance, a small constant time bound is required for each Scatter/Gather step. Clearly this cannot be accomplished without preprocessing the data to some extent, since any processing linear in the size of the collection becomes noninteractive on sufficiently large collections. Moreover, for large text collections even this preprocessing step must be reasonably efficient. Quadratic running times are still too slow. We therefore consider the task of constant interaction-time document clustering assuming near-linear time preprocessing.

3 Methodology

This section presents a method for accelerating Scatter/Gather.

3.1 The Hypothesis

Suppose one needs to cluster 10,000 documents into 10 groups of related documents. One expects that documents that are extremely similar to each other will usually end up in the same cluster. If an existing clustering of the same 10,000 documents into 500 groups were available, one would, by extension, be reasonably confident that all of the documents in a given one of

[1] Others in the information retrieval community have also proposed rectangular time cluster algorithms (i.e. order kn, where k is the number of desired clusters and n is the size of the document collection), *e.g.* [3, 5]. However, since the the traditional use of document clustering in retrieval (to broaden similarity search) calls for k proportional to n, no speed gain was realized from considering these algorithms; hence they have not been aggressively pursued. In our case k is a small fixed number and hence rectangular time algorithms are attractive.

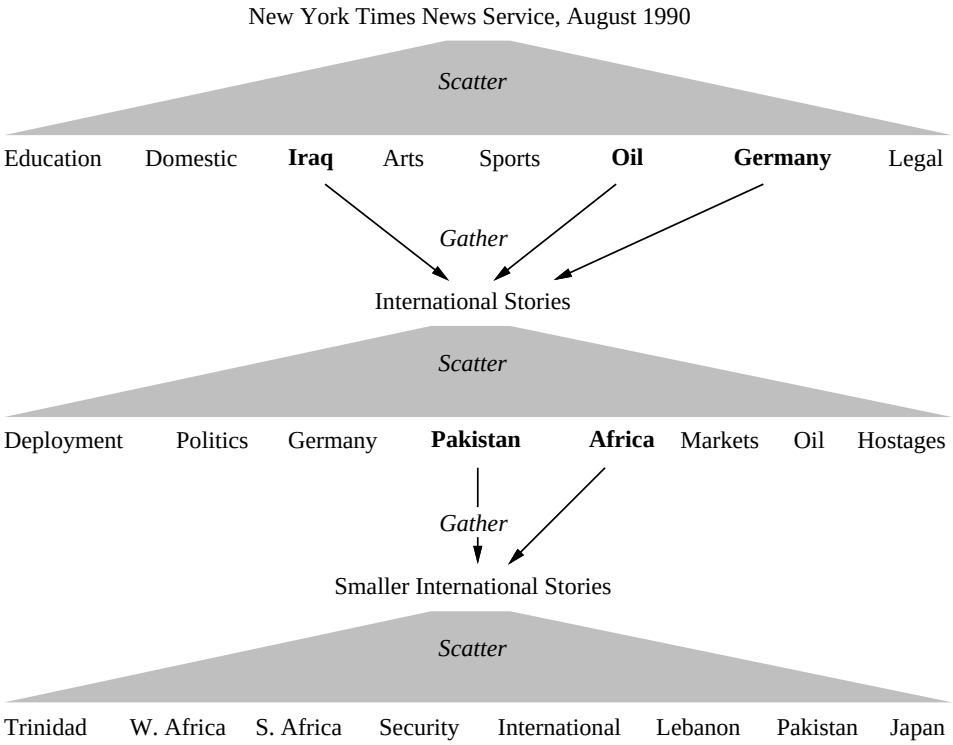

Figure 1: Illustration of Scatter/Gather

those 500 groups would end up in the same one of the desired 10 clusters.

More generally, we hypothesize that documents similar enough to be clustered together in a fine-grained clustering will be clustered together in a coarse-grained clustering. We call this the *Cluster Refinement Hypothesis*.

Consider treating each of the 500 intermediate clusters as a single *meta-document* consisting of the union (concatenation) of all the documents it contains. These meta-documents form a *condensed representation* of the collection. Instead of clustering the 10000 documents, we could cluster the 500 meta-documents. Each group of meta-documents would then correspond to the union of the groups of documents in those meta-documents. Each meta-document contains a collection of related documents, and meta-documents which are related cluster together. The Cluster Refinement Hypothesis thus implies that this will yield similar results to clustering the original 10,000 documents.

If we could always quickly produce a condensed collection of 500 meta-documents to act as a surrogate for the much larger collection of individual documents which we wish to cluster, we could guarantee that the running time to produce clusters would be bounded by the running time of the clustering algorithm on 500 items regardless of the size of the true document collection. Of course, this would be self-defeating if the only way to produce the meta-documents is by clustering the collection into 500 groups. We now present a data structure and a modification of the Scatter/Gather browsing method that provides the desired speedup. The browsing method uses the data structure to efficiently maintain a condensed representation of the focus set.

3.2 The Cluster Hierarchy

The data structure is a *cluster hierarchy*, such as that produced by classical agglomerative clustering algorithms. This tree is described recursively as either a leaf, corresponding to a single document, or a tree whose subtrees are cluster hierarchies. An agglomerative clustering algorithm produces such a tree, in which each internal node corresponds to the agglomeration of the documents represented by the children of that node. Intuitively, documents which are closely related have a common ancestor low down in the tree, because the agglomerative algorithm merges these documents relatively soon, whereas the common ancestors of unrelated documents are near the root. For our purposes, it is also desirable that the tree be relatively balanced.

An internal node in a cluster hierarchy corresponds to a meta-document containing the documents which are leaves of the subtree rooted at that node.

The hierarchy lets us view the collection at different granularities: we can represent it coarsely as a small

set of meta-documents (tree nodes) near the root of the hierarchy, or more finely as a large set of meta-documents near the leaves of the tree. More generally, given a meta-document which we wish to examine more closely, we can expand that meta-document, replacing it by its children. Since each of its children represents only a portion of the documents in the original meta-document, these children give a more detailed representation of the documents they contain than the original meta-document.

3.3 Scatter/Gather on a Hierarchy

A cluster hierarchy can be used to accelerate the Scatter/Gather browsing method so that a single iteration of the process takes constant time regardless of the number of documents involved.

Recall the Scatter/Gather loop. The user begins with a focus set of documents of interest. This focus set is scattered into k clusters and presented to the user. The user selects a subset of these clusters, yielding a new, smaller and more detailed focus set. This process is then repeated.

Let M be the maximum number of items which can be clustered in the desired constant time bound ($M \gg k$). At each iteration, we begin with the focus set F, consisting of meta-documents. For the first iteration, the focus set is the single meta-document representing the entire collection.

F is first *expanded* using the following simple procedure:

while F has fewer than M meta-documents

> **Find** the meta-document D in F with the most leaves.
>
> **Replace** D by its children in the hierarchy.

This expanded focus set contains smaller, and thus more detailed, meta-documents than the original. By the Cluster Refinement Hypothesis, clustering an expanded F should yield similar results to that of clustering the individual documents in F. However, since the size of F is M, we know that F may be clustered in the desired constant time bound.

As before, summaries of the clusters are presented to the user. When the user then selects a subset of the resulting clusters, although notionally selecting a set of documents, he is in fact selecting a subset of the meta-documents of F. Thus, the new set of documents has a small condensed representation as meta-documents. The constant time bound is thus maintained through iterations.

3.4 Discussion

One might ask why we go to so much trouble. It might seem that the cluster hierarchy admits a much simpler browsing technique. Simply treat the hierarchy as a fixed categorization of the collection that can be navigated like a hierarchical menu system. That is, the user is presented with summaries of the children of the current node, one of which can be selected for further expansion. Since the entire tree is precomputed, no clustering at all would have to be performed at interaction time.

We find this search model too restrictive. It assumes that at each presentation only one cluster will be of interest to the user. However, the user's interests can easily span more than one cluster or be at a boundary between clusters. Scatter/Gather provides a more powerful interface without introducing substantial additional overhead.

It remains to describe the following:

- How to generate a cluster hierarchy in near-linear time.
- How to ensure that meta-documents are no harder to cluster than individuals.
- How to ensure that one may summarize a cluster composed of meta-documents, so as to accurately reflect the documents contained therein.

4 Implementation

4.1 Generating the Hierarchy

As was observed above, cluster hierarchies are the natural outputs of agglomerative clustering procedures. However, these procedures are typically quadratic time because at each step they merge the globally most similar pair of documents. On collections the size of Tipster, this is prohibitively slow even as a preprocessing step. We therefore propose a different approach.

Partitional clustering strategies such as Buckshot or Fractionation [1] cluster into k groups in $O(kn)$ time. Such partitional procedures can easily be used to generate a hierarchy: simply apply the algorithm recursively to each group in a partition, stopping at individual documents. Given a branching factor, k, we partition the entire collection into k groups in $O(kn)$ time. Each of these groups is now treated as a subcollection to be partitioned in turn into k subsubcollections, etc. At each level of the recursion there are a total of n items, since each item is in exactly one group at that level. Hence, the total cost to perform all clusterings at each level. is $O(kn)$. If each clustering is balanced, *i.e.* every cluster of a clustering contains a constant fraction of the items,

then there are $O(\log n)$ levels and hence the entire procedure can be performed in $O(kn \log n)$ time.

Although this procedure performs no global analysis, it is likely that any two similar documents will remain together through many levels of subclustering, and will therefore share a common ancestor much lower in the tree than that of two documents which are dissimilar.

4.2 Meta-profiles and Truncation

Each internal node in the resulting hierarchy can be viewed as a meta-document; that is, as the union of its descendant leaves. For clustering, individual documents are represented as sets of word-weight pairs, *i.e.*, as sparse, high-dimensional vectors, or profiles.[2] Similarity between documents is defined as simply the cosine between vectors. Meta-documents may also be represented as profiles by simply summing the profiles of their descendant leaves. The notion of similarity clearly extends to meta-documents.

Hence, we can store in each node of the cluster hierarchy a profile which captures the word content of its subtree. Unfortunately, profiles of nodes high in the tree, formed by summing numerous individual document profiles, are less sparse; they contain many non-zero entries. Indeed, the root node's profile is a fully occupied vector since every word that occurs in the collection is seen in the root's subtree. This has implications for memory requirements, and more importantly, affects the time to cluster.

The measure of $O(kn)$ time for our partitional clustering algorithms is in fact a measure of $O(kn)$ similarity comparisons. The similarity measure we use is an inner product of document profiles, and can thus be computed in time linear in the profile sizes.[3] Thus a similarity computation could be treated as a constant time operation when we were considering only individual documents, because their profiles are all of bounded size. However, as profiles become increasingly large the time to compute similarities also increases. We have found that clustering meta-documents with their dense profiles takes almost as much time as clustering the documents of those meta-documents individually.

To solve this problem we *truncate* the meta-document profiles to include only the most topical (*e.g.* most highly weighted) entries. All profiles are then of the same length and we can guarantee that clustering n meta-documents will take the same time regardless of their actual size. We have found that these truncated profiles (we use the 50 most topical terms) are effective descriptions of their meta-documents. This is unsurprising, since, in fact, we present a cluster to a user in part by listing its most topical terms, *i.e.* a truncated profile.

A further benefit of truncation is that the total amount of storage required to represent the cluster hierarchy is linear in the size of the collection. This is because there can be no more internal nodes than there are leaves, and the storage required for each internal node is constant.

4.3 Meta-summarization

Scatter/Gather employs a *cluster digest* to summarize a cluster of individual documents [1]. We can extend this notion to clusters of meta-documents as follows. Recall that a cluster digest consists of two complementary components, a list of "topical" words, defined as those with highest weight in the cluster as a whole, and a list of "typical" titles, defined as titles of individuals most similar to the cluster profile. Since the computation of topical words depends only on the profiles of cluster members, it is trivially extended to clusters of meta-documents. Extending the computation of typical titles, on the other hand, requires us to define the notion of title for meta-documents. This can be accomplished by assigning to each node the title of the leaf (*i.e.* individual document) in its subtree most similar to the node's profile. Hence, in addition to a truncated profile, each node in the cluster hierarchy must also store a title to support fast cluster summarization.

The time needed to compute the profile and central document of a cluster of c items is $O(c)$; thus the running time to build the entire hierarchy remains $O(kn \log n)$ even with these additional computations.

5 Scatter/Gather over Tipster

In this section we provide a demonstration of the methods proposed in this paper operating over the very large Tipster collection.

The DARPA Tipster collection contains over 700,000 documents, occupying 2.1 gigabytes of text. There are over a million unique words in Tipster, with nearly a half a million occurring in more than one document.

The construction of a hierarchy for Tipster required forty hours of computation on a Sun SPARCStation 10. Around 210 megabytes of disk space were required to store the truncated profiles of the hierarchy, 10% of the size of the text. Using this hierarchy, with our default settings, Scatter/Gather steps require approximately 20 seconds.

According to the descriptions provided with the collection it contains articles from the AP Newswire and the Wall Street Journal, abstracts from the Department of Energy, entire Federal Register issues, and text from Ziff Davis' "Computer Select" disks. While this gives

[2] High-frequency function words are, of course, discarded through the use of a stop list.

[3] Here *size* is defined to be the number of non-zero entries

```
0 (77235) section, rule, public, office, agency, action, regulation, order, req
 FR: General Services Administration Acquisition (information, section, servic)
 FR: Community Development Block Grants          (section, federal, rule, regu)
 FR: Privacy Act of 1974; Notice of Systems of R (information, file, office, a)
1 (153421) official, house, soviet, country, leader, bush, american, police, un
 AP: After 10 Years, States Still Falter on Camp (state, percent, unite, natio)
 AP: URGENT                                      (state, u.s., unite, presiden)
 AP: Afghan President Asks America, Pakistan To  (government, official, u.s., )
2 (179334) share, stock, trade, sale, sell, business, exchange, york, buy, cent
 WSJ: Dividend News: Penn Central Sets Payout, W (company, share, million, sto)
 WSJ: Year-End Review of Bond Markets: Money Man (company, market, million, fi)
 WSJ: Year-End Review Of News Highlights: What W (company, million, bank, busi)
3 (112900) user, software, computer, network, ibm, technology, version, line, p
 ZF: Sun's NeWSprint: a new way to print. (Softw (user, program, software, net)
 ZF: 25 tough integration problems and solutions (software, network, user, pro)
 ZF: Forecast 1989.&M;                           (user, application, software,)
4 (220170) study, energy, present, temperature, test, describe, analysis, gas,
 DOE: This is a report on the development of a c (design, process, data, heat,)
 FR: Energy Conservation Voluntary Performance S (energy, fuel, development, p)
 DOE: Presents an experimental study of the wett (coal, present, level, study,)
```

Figure 2: Top-level Scattering of entire Tipster Collection

one some idea of what to expect, it would be useful to know more before one begins querying this collection so that one's expectations may be brought into accord with the contents of the corpus.

Pertinent questions include: what is the difference between the coverage in the Wall Street Journal and the AP newswire? What sort of things does "Computer Select" contain? What is a DOE abstract? What do they talk about in the Federal Registry? We could answer these by randomly selecting some articles from each collection and reading them, but Scatter/Gather provides a more thorough method.

Figure 2 shows a scattering of the entire collection. Five clusters are presented. The first line of each contains an identifier, the leaf-count of the cluster (in parentheses), and a list of topical terms from the cluster's profile. The succeeding three lines present the three most central meta-documents in the cluster. These each contain the title of the typical document in the meta-document, annotated by the source (one of AP, DOE, FR, WSZ or ZF), followed by a list of topical terms in the meta-document.

The top-level clustering roughly reproduces the division into five Tipster sources. This indicates both that the sources discuss disjoint topics, and that our clustering algorithms can discover this.

We immediately get answers to some of our questions: the federal registry contains government regulations (cluster 0); the AP newswire concentrates on political events (cluster 1), as opposed to the Wall Street Journal which concentrates on business news (cluster 2); the Ziff-Davis collection contains predominately press about computer-related products (cluster 3); and DOE abstracts are technical studies about energy (cluster 4).

Let us examine the newswire sources more closely by gathering clusters 1 and 2 and scattering this into ten new clusters. Figure 3 shows the results of this operation. Note that at this granularity, the Wall Street Journal and the AP newswire are shown to have a fair amount of overlap, though each predominates certain topics.

The clusters can be described as follows: (0) human interest and leisure; (1) legal affairs; (2) police actions; (3) markets; (4) companies; (5) finance; (6) foreign affairs; (7) congress; (8) presidential politics; and (9) regional news.

We now concentrate on international affairs by gathering clusters 2 and 6 and rescattering. The output of this is shown in Figure 4.

```
0 (18568) child, school, family, play, young, book, film, black, student, turn,
 WSJ: Missing in America: A Lost Brother Sends O (time, home, think, house, fa)
 AP: AP WEEKEND ENTERTAINMENT AND ARTS           (play, time, film, movie, thi)
 AP: Italian Women Getting Ahead, Say Men Must C (woman, man, time, think, cit)
1 (8196) court, case, judge, attorney, trial, sentence, prison, prosecutor, jur
 AP: Government Deciding Whether To Hold Second  (charge, case, trial, attorne)
 AP880307-0033                                   (court, sentence, case, judge)
 AP: Kidnapping or Extradition? Overseas Drug Ar (charge, court, u.s., arrest,)
2 (28954) police, fire, spokesman, army, officer, attack, israeli, soldier, pal
 AP: Radicals Kill Six Riot Policemen; Roh Warns (police, kill, official, gove)
 AP: URGENT                                      (city, report, government, of)
 AP: Police Deployment Stirs Racial Tensions In  (police, city, arrest, office)
3 (32778) index, fall, dollar, trader, future, yen, decline, volume, oil, board
 WSJ: World Stock Markets: Stronger Dollar and W (stock, market, share, price,)
 WSJ: Abreast of the Market: @ Upward Mobility:  (stock, market, share, price,)
 AP: Eds: SUBS 16th graf pvs bgng Standard & Poo (stock, market, index, trade,)
4 (128215) sale, business, executive, industry, quarter, unit, product, loss, r
 WSJ: Dividend News: Penn Central Sets Payout, W (company, share, million, sto)
 WSJ: Who's News: Johnson Products' Chief Quits; (company, million, executive,)
 ZF: Computer stocks fall led by IBM.&O;         (million, company, share, rev)
5 (29549) loan, term, debt, credit, reserve, mortgage, treasury, capital, asset
 WSJ: Credit Markets: Treasury Bonds Fall Again  (rate, bond, million, price, )
 WSJ: Financial Overhaul: Big Banks Would Get Va (bank, loan, federal, rate, f)
 WSJ: Year-End Review of Markets and Finance: Wh (billion, million, government)
6 (48287) soviet, country, unite, foreign, minister, union, war, communist, gor
 AP: Parties Meet To Salvage Government          (government, war, state, pres)
 WSJ: What's News -- World-Wide                  (government, south, u.s., sta)
 AP: URGENT                                      (state, u.s., unite, presiden)
7 (18251) senate, committee, congress, white, rep, budget, office, program, cut
 WSJ: Budding Issue: Bush's Schedule Shows He Sp (house, president, bush, sena)
 WSJ: Trade Measure Clears House By Big Margin - (bill, house, senate, vote, c)
 AP: Senate Putting Cheney Nomination on Fast Tr (house, federal, committee, c)
8 (5476) dukakis, presidential, jackson, vice, george, convention, michael, pol
 AP: Bush Says Dukakis Dividing Classes in Charg (dukakis, bush, campaign, pre)
 WSJ: Campaign '88 -- Democrat's Task: Dukakis M (jackson, dukakis, campaign, )
 AP: Bentsen Questions Quayle Qualifications; Qu (bush, quayle, president, vic)
9 (14481) fair, build, rain, northern, central, coast, southern, inch, temperat
 AP: Thunderstorms Follow Tornadoes; Record Cold (city, state, build, area, fa)
 AP: After 10 Years, States Still Falter on Camp (state, percent, unite, natio)
 AP880901-0082                                   (state, study, time, present,)
```

Figure 3: Second-Level Scattering: Clusters 1 and 2 from the Top-level Scattering

```
0 (6130) man, newspaper, child, family, case, charge, air, death, federal, prod
 AP890507-0076                                (report, board, safety, air, )
 AP: U.N. Group Reports 400 'Disappearances' In  (report, country, group, offi)
 AP: Rescuers Continue Search For Survivors, Fif (official, report, news, agen)
1 (8341) trade, japan, market, import, export, billion, japanese, industry, eur
 WSJ: Major Nations Near an Accord On Capital Fl (u.s., country, export, gover)
 AP: U.S. Would Welcome Free-Trade Treaty Talks  (trade, u.s., state, unite, c)
 FR: Actions to Address Adverse Conditions Affec (state, u.s., unite, country,)
2 (15256) iran, rebel, contra, iranian, iraq, north, noriega, panama, security,
 WSJ: Panama Bungle: U.S. Tries to Salvage Its M (u.s., state, military, gover)
 WSJ: What's News -- World-Wide               (government, u.s., official, )
 AP: With AM-Philippines, Bjt                 (u.s., military, state, offic)
3 (11916) moscow, nuclear, weapon, missile, defense, europe, treaty, secretary,
 WSJ: --- President's Power Is Slipping; Soviet  (soviet, u.s., president, off)
 AP: US-Soviet Summit Not a Priority, White Hous (soviet, bush, president, sta)
 AP: Gorbachev Arrives In NY, Urges Expanded Sup (soviet, u.s., state, preside)
4 (8558) party, election, opposition, vote, lead, reform, german, rule, parliam
 AP: Premier Says He Favors Non-Communists in Go (government, party, leader, o)
 AP: Communists Give Themselves New Name; Plan N (party, communist, leader, re)
 AP: Opposition Labels Election A Farce       (party, election, vote, gover)
5 (2854) israeli, palestinian, israel, arab, bank, gaza, occupy, plo, uprise, s
 AP: Israeli Leaders See Movement in Baker Modif (palestinian, israeli, israel)
 AP: Palm Sunday Procession Is Canceled in Jerus (palestinian, israeli, israel)
 AP: Israeli Troops Kill Two Palestinians; Strik (palestinian, israeli, arab, )
6 (13383) police, man, officer, arrest, car, charge, death, protest, night, for
 AP: Man Shot At End of High-Speed Chase      (police, officer, report, man)
 AP: Police Confirm Arrest of Ex-Detective in Sl (police, government, official)
 AP: Bloody Night in Copenhagen               (police, man, house, woman, h)
7 (5314) damage, build, firefighter, burn, police, resident, blaze, service, ce
 AP: People Return To Homes As Fire Contained    (fire, official, firefighter,)
 AP: Fourteen More Deaths From Disease in Capita (report, state, official, new)
 AP: Fire Kills 16 at High-Rise Retirement Home; (fire, house, build, home, fi)
8 (3330) plane, flight, air, crash, airline, airport, pilot, passenger, jet, ai
 AP: Airliner Crashes in Iowa with 298 on Board, (plane, flight, airline, airp)
 AP: 13 Die But 94 Others Brave Smoke And Fire T (plane, crash, air, flight, o)
 AP: Death Toll At Five From U.S. Jet Crash      (air, plane, u.s., force, pil)
9 (2159) patient, study, metal, structure, contain, compound, theory, cell, dos
 DOE: The aim is to define representations of th (group, theory, coal, refs, g)
 ZF: People.&M;                               (group, president, state, pol)
 DOE: The aim of the present study was to determ (group, patient, cell, dose, )
```

Figure 4: Third-level Scattering: Clusters 2 and 6 from Second-Level Scattering

6 Conclusion

We have presented a method that extends the Scatter/Gather browsing paradigm to arbitrarily large corpora. This requires the precomputation of a cluster hierarchy, constructed using an $O(n \log n)$ divisive algorithm, with a linear storage overhead. The hierarchy enables construction of a concise representation of the focus set. This representation, a set of meta-documents, is by design of fixed size, and can hence be clustered in constant time, yielding constant-time interaction for each Scatter/Gather step.

References

[1] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of SIGIR'92*, June 1992. Also available as Xerox PARC technical report SSL-92-02.

[2] Donna Harman. The TIPSTER evaluation corpus. CDROM disks of computer readable text, 1992. Available from the Linguistic Data Consortium.

[3] G. Salton. *The SMART Retrieval System.* Prentice-Hall, Englewood Cliffs, N.J., 1971.

[4] R. Sibson. SLINK: an optimally efficient algorithm for the single link cluster method. *Computer Journal*, 16:30–34, 1973.

[5] P. Willett. Document clustering using an inverted file approach. *Journal of Information Science*, 2:223–231, 1980.