Hierarchical Clustering and Tagging of Mostly Disconnected Data

S. Ingram[1], T. Munzner[1], and J.Stray[2]

[1]University of British Columbia, Vancouver, Canada
[2]Associated Press, New York City, USA

Abstract

We define the document set exploration task as the production of an application-specific categorization. Computers can help by producing visualizations of the semantic relationships between the documents, but the approach of directly visualizing the vector space representation of the document set via multidimensional scaling (MDS) algorithms fails to reveal most of the structure because such datasets are mostly disconnected, that is, the preponderance of inter-item distances are large and roughly equal. Interpreting these large distances as disconnection between items yields a decomposition of the dataset into distinct components with small inter-item distances, that is, clusters. We propose the Disconnected Component Tree (DiscoTree) as a data structure that encapsulates the hierarchical relationship between components as the disconnection threshold changes, and present a sampling-based algorithm for efficiently computing the DiscoTree in $O(N)$ time where N is the number of items. We present the MoDiscoTag application which combines the DiscoTree with an MDS view and a tagging system, and show that it succeeds in resolving structure that is not visible using previous dimensionality reduction methods. We validate our approach with real-world datasets of WikiLeaks Cables and War Logs from the journalism domain.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

We are interested in helping an analyst to understand a large set of documents of mostly unknown content, when indices, summaries, and other knowledge organization aids are not available. We assume that full text search is available but not completely helpful because the analyst does not know precisely what they are looking for. This problem arises in fields such as business, law, intelligence, and journalism. Noting that categorization and classification is a fundamental mode of human understanding [Bai94], we define the *document set exploration task* as the computer-assisted human construction of a categorization that is application-specific, that is, tailored to a specific use of a specific document set.

To do this we must encode documents in some way that preserves semantics. The *vector space representation* of documents [SWY75] yields very high dimensional spaces with thousands or tens of thousands of dimensions, corresponding to the natural language vocabulary of the document set. Many automatic clustering algorithms that carry out computations in these spaces have been proposed [Ber06] as ways to categorize large document collections. The output of these algorithms is a list of the documents in each cluster, which does not necessarily provide the analyst with a sense of the overall semantic structure of the document set. Nor is it obvious whether any particular clustering, out of the combinatorially huge number of partitions of objects into disjoint sets, captures the application-specific semantics of interest. For this reason, many sensemaking systems attempt to directly visualize the high-dimensional cluster structure of the documents through low-dimensional layouts created with dimensionality reduction (DR) techniques [WTP*95] [CWDH09].

However, we have found that many analysts who use DR for document set visualization have the persistent unease that there is often but not always structure in their datasets that is not revealed; that is, that they see false negatives in many cases but true negatives in others. Existing DR methods not only have the quality problem that layouts systematically contain false negatives, but also the speed problem that layouts take a long time to compute. We point out a critical dataset characteristic that underlies these problems: the vast majority of distances between items are large and nearly equal. We argue that these distances can be usefully interpreted as representing disconnection, breaking the set of points up into distinct components. We thus call these datasets *mostly disconnected*, or *modisco* for short.

We analyze a number of document sets and find that all of them have the modisco characteristic. This result is understandable given that a) the convergence to identical distances is a straightforward consequence of the geometry of high dimensions [RU11] and b) maximizing the distance between points was one of the original design goals of the vector space representation in order to increase precision in information retrieval applications [SWY75].

Unfortunately, obvious approaches to adapt dimensionality reduction techniques to handle modisco data gracefully by treating distant and nearby distances differently fail to solve the problem. Because adapting MDS to handle structure at multiple scales is difficult for fundamental mathematical reasons [BS02], our proposal is to exploit the modisco characteristic to efficiently build another data structure that can be used in conjunction with standard MDS.

We propose the Disconnected Component Tree, or the DiscoTree, as a data structure that represents the sequence of ever-finer decomposition of components into smaller ones as the disconnection threshold is changed from the maximum to the minimum distance between points. The DiscoTree is a particular instance of a hierarchical clustering, where the criterion used to cluster is strongly related to the distance metric used by dimensionality reduction methods. We introduce an efficient sampling-based approach that exploits the modisco property to run in $O(N)$ time.

We present the proof-of-concept MoDiscoTag application that combines the DiscoTree and an MDS view to support analysts in exploring and annotating document collections through tagging clusters. We validate that this approach helps journalists find semantically meaningful structure that is not visible in the MDS view alone using two complex, real-world datasets, subsets of the WikiLeaks Afghan War Logs and the WikiLeaks Cables.

The contributions of our work are:

- we propose combining hierarchical clustering, dimensionality reduction, and tagging for computer-supported human categorization of large document sets;
- we characterize the class of mostly disconnected data, describe how to identify it, and show that the vector space representation of document sets is mostly disconnected;
- we propose the DiscoTree hierarchical clustering as a useful data structure and provide a linear time algorithm for computing it;
- we show that our proof-of-concept MoDiscoTag application succeeds in helping journalists quickly find interesting structure in complex, real-world WikiLeaks datasets.

These four contributions are sufficiently diverse that we divide our discussion of previous work into the sections that pertain to each.

2. Clustering, Tagging, and DR for Sensemaking

We articulate the document set exploration task more precisely as supporting analysts in building an application-specific hierarchical categorization schema through tagging, starting from the scaffolding of a schema automatically created through hierarchical clustering. Dimensionality reduction fits into this workflow both to enable direct visualization of document set structure, where possible, and as a way to evaluate the relationship between the distance metric used in algorithmic clustering and the semantic content of the dataset.

2.1. Why Clustering?

Clusters of points in high-dimensional data sets are interesting because they often have meaning; that is, cluster structure frequently represents semantics of interest to the user. This statement posits a strong connection between a mathematical property and an abstract, high-level notion of human knowledge.

The idea of representing document collections as point sets in high-dimensional space began with the work of Luhn in the 1950s [Luh57] and was developed into the vector space model by Salton et al. [SWY75], originally designed for information retrieval tasks. In this model each dimension corresponds to a word in the vocabulary of the document corpus, so there are typically many thousands of dimensions. Each document is represented by a sparse vector with a non-zero entry for each unique word used in that document, typically derived the number of times that word appears in the document text, normalized and weighted by one of several standard formulas such as the term frequency / inverse document frequency (TF-IDF) method. Because this encoding discards all syntax and word ordering it is also called the *bag of words* document model, yet it has proven effective for information retrieval tasks and difficult to beat with other indexing schemes [SB88], which suggests that it captures important semantics of many types of document collections.

Information retrieval and dimensionality reduction applications rely on a similarity or distance function, defined over every pair of documents, which gives rise to a metric space and associated topology. Frequently the *cosine distance* method is used for document sets, essentially a dot product of document vectors, but there are many variations [SB88]. Spatially compact clusters of document vectors in this space were recognized by early information retrieval researchers as semantically interesting structures, giving rise to the *cluster hypothesis* [JvR71], a modern version of which is articulated as "documents in the same cluster behave similarly with respect to relevance to information needs" [MRS08]. The cluster hypothesis is widely assumed and has been shown to hold in the case of web-scale information retrieval [CW06].

Sensemaking differs from information retrieval in that the user does not know beforehand what type of information is sought. However, because the documents within a cluster are conceptually similar, representing a document corpus by its clusters may be a useful form of information reduction. The intuition is that if you have read a few documents in a cluster, you can assume that the rest will contain a similar type of information.

2.2. Why Tagging?

A cluster is, in the end, just a set of documents. While there is evidence that machine-extracted clusters capture interesting semantics, that does not help the user to understand what any given cluster means, much less a tree which may include hundreds of clusters and sub-clusters. Cluster labeling is the crucial next step in sensemaking.

There have been many more or less sophisticated attempts at automatic cluster labeling, ranging from displaying the most frequently occurring words to attempting to extract a single key sentence from a text corpus [Zha02]. A related problem is the naming of topics extracted by topic modeling algorithms such as Latent Dirichelet Analysis (LDA) [Ble11]. Each *topic* found by such an algorithm is a probability distribution over words, sometimes visualized with a word cloud as in [CLT*11]. Yet a word cloud is not a substitute for a good topic name, as researchers working with LDA-based methods tacitly acknowledge when they compose short, human-generated labels to refer to the semantics of extracted word distributions.

A deeper problem is the suitability of the classification schema implied by the distance metric. In what sense are two documents really "similar"? In practice this depends on the context of the analysis. For example, should the `Warlogs` documents be grouped by location, specific event, type of incident, or actors involved? There is no reason to assume that the particular encoding and distance metric used to generate clusters necessarily partitions the documents in the most semantically useful way, especially given that, for the sensemaking task, the user may not know beforehand what ways are going to be interesting.

Grimmer and King approach this problem by visualizing the space of all possible clusterings, populated by executing a variety of clustering algorithms on the same data [GK10]. They are able to directly explore some of the different semantically interesting categorizations on the same set of documents.

In principle, the sensemaking loop should include adjustments to the vector encoding and distance metrics so as to explore different categorization schemas, but very little is known about how to do this. Instead, we consider the automatically-generated clusters as starting points for human classification. We argue for a tagging system that allows the user to summarize and annotate the content of a cluster, by applying a label to some or all of its documents. These tags allow the construction of a manual classification scheme that follows or cuts across the cluster structure as desired, and has greater or lesser resolution around particular concepts and subtrees.

2.3. Why Dimensionality Reduction?

Dimensionality reduction methods such as MDS also use distance metrics, just as k-means and other clustering methods do, to map items to a lower dimensional space. The promise of MDS and other DR methods is visual confirmation of the existence and size of clusters. MDS as a visualization technique is often used to verify a proposed clustering by coloring the points according to the clusters and checking if the spatial proximity relationships in the low dimensional view match up the coloring, or to visually check for cluster structure in unclassified data by noticing if there are any visually separated clusters in the view. The visual display of dimensionality reduction results allows the user to cross-check whether the relationships between the visible clusters that arise from the distance metric match their mental model of semantic content.

2.4. Why All Three?

In this framework, the sensemaking task is the construction of a set of tags which capture the concepts of interest — perhaps newly discovered — in the document collection. The tags act as an annotation layer to get human semantic understanding into the exploration loop, an automatically created clustering serves to accelerate the process, and dimensionality reduction provides a parallax view of the cluster structures. We thus argue for combining these three capabilities.

3. Mostly Disconnected Data

Buja and Swayne coined the term **indifferentiation** to describe equidistance between points in high-dimensional datasets [BS02] and showed that MDS algorithms are not well suited to such data. We have built on this by explicitly identifying the class of nearly indifferentiated data, which we term *mostly disconnected* data, as a major problem faced by real-world users exploring document collections and other very high dimensional datasets. We further propose the DiscoTree as a hierarchical clustering algorithm which exploits this structure for linear-time execution. The DiscoTree clustering is based on treating certain distances between points as disconnection, creating a decomposition into separate components.

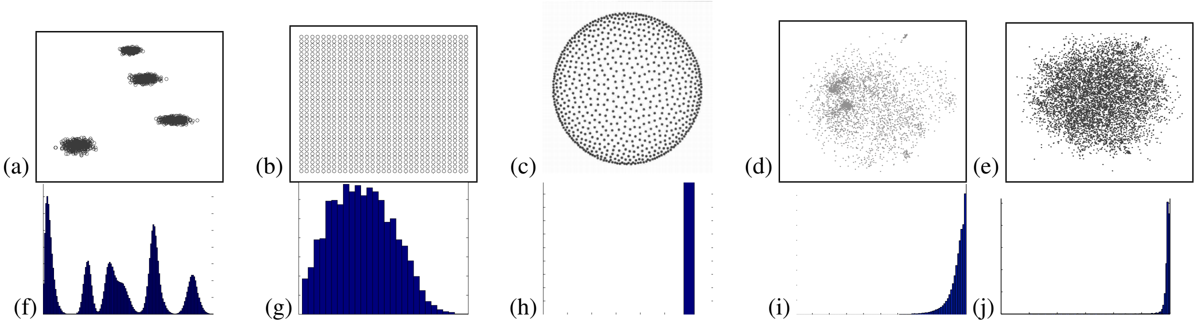


Figure 1: *Low-D layouts and histograms of high-D inter-point distances for five dataset types. The well-separated cluster dataset embedded with PCA (a) has a histogram with many smooth peaks and valleys (f). The grid dataset embedded with PCA (b), with points equally spaced on a manifold, has a histogram showing a single, unimodal distribution of distances (g). The simplex dataset (c) as embedded by Buja et al. [BS02] as a disc with sharp edges and a low-density center, and its histogram is a single spike at maximum distance (h). The mostly disconnected `Afghan` dataset embedded with Glimmer MDS [IMO09] (d) looks like a blob with few features in the embedding. Its histogram also shows a large spike at maximum distance, but also has a small subset of smaller-than-maximum distances (i). The modisco `Caracas` dataset has almost no visible features in the MDS blob (e) and an even sharper histogram curve (j).*

3.1. Distance Histograms and Embeddings

An important intuition about modisco datasets arises from understanding the relationship between histograms showing the distribution of distances between points in high-D space, and the embeddings in low-D space that various DR techniques produce for layouts.

Figure 1 shows low-D embeddings of five different dataset types alongside histograms of their high-D distances. Figure 1a is a synthetic mixture of gaussians in four dimensions, which form clear clusters in the PCA layout. Its corresponding distance histogram is a multimodal distribution of smooth peaks. Figure 1b is a PCA layout of a simple 2D grid: a smooth and densely sampled manifold whose points are distributed uniformly. Its histogram of distances is a smooth unimodal distribution. The cluster example and the manifold example are instances where the low-D embedding produced by DR algorithms produces faithful representations of how the analyst mentally models the data: high-D clusters of points are represented as low-D clusters of dots and high-D uniformly spaced manifolds are represented as uniformly spaced dots.

Figure 1c is an exact k-D simplex; that is, it is a hypertetrahedron in k dimensions, so all of its vertices are the same distance apart. Its histogram is a single spike at this uniform distance value. This synthetic configuration was introduced by Buja and Swayne [BS02] as an example of perfectly *indifferentiated* data, meaning that all high-D distances are identical. They show both theoretically and experimentally that its low-D embedding has a characteristic circular disk shape with a sharp edge and low density in the center, and that a similar pattern holds when distances are randomly distributed around a narrow range.

Figure 1d shows the `Warlogs` dataset, which has the modisco property; that is, one where most of the high-D distances are nearly identical, but some are not. The low-D embedding looks like a blob with some apparent structure. However, the histogram shows that there are significant numbers of non-identical distances in the dataset as well as a large spike of near-identical ones. This MDS embedding is an example of a false negative where even though some structure is visible, significant additional structure is invisible because clusters are squished up next to others without intervening regions of lower density, as shown in Figure 4 where structures found with the DiscoTree are color coded against a similar layout. Figure 1e shows the `Caracas` dataset, which has an even steeper histogram and almost no apparent structure in the MDS layout; complex and interesting structure found with the DiscoTree can be seen in Figures 5, 6, and 7.

Modisco data sets have vastly more distances that are roughly equal than those that differ. Specifically, modisco datasets of N points have $O(N^2)$ high-D distances that are roughly the same and only $O(N)$ distances that differ, out of their total N^2 inter-point distances. Checking whether a dataset has the modisco property is easy to see by simply inspecting the distance histogram, because it will have characteristic pattern of a super-linear upward curve culminating in a spike for the $O(N^2)$ distances. Checking this property algorithmically is also straightforward.

We expect that very high dimensional datasets will be modisco under a variety of distance metrics, because a consequence of the well-known curse of dimensionality [Bel61] is that both Euclidian norms and inner-product angles converge to central values as dimensionality increases [RU11].

Österling et al. do point out that in very high dimensional spaces, distances between points become relatively uniform [OST*10], but they do not build further on this statement in terms of characterizing classes of data.

3.2. Vector Space Representations Are MoDisco

Salton et al. showed that the TF-IDF method maximizes the high-D distance between individual documents and that this correlates with improved information retrieval performance [SWY75]. Accordingly, we analyzed the histograms of a number of document sets in the vector space representation with TF-IDF weighting, including the WikiLeaks *Warlogs* and *Cables* discussed in this paper, the *NIPS*, *Enron*, *Kos*, and *NYTimes* datasets from the UC Irvine Machine Learning Repository [FA10], and the *REUTERS* and *MEDLINE* datasets used for previous document set visualization work such as [OST*10]. Visual inspection of the histograms showed that all of these have the modisco property.

Essentially, the vector space representation now at the core of many document handling systems was explicitly designed to have a property which makes it difficult to visualize with MDS methods.

3.3. DR on MoDisco Data

A vast number of dimensionality reduction techniques have been proposed, where later approaches strive to address the failure cases of the previous methods. The classic Principal Component Analysis (PCA) creates linear projections [Jol02], while the many variants of MDS produce non-linear mappings [BG05]. Recent techniques such as t-SNE [MH08] have been designed to capture cluster structure in situations where MDS typically fails. However, we conjecture that any method based on the minimization of distance-based error metrics will be ineffective for modisco data in the sense of being prone to the false negative problem.

The critical insight behind our claim of ineffectiveness is that the non-identical distances are overshadowed by the identical ones in a way that standard DR approaches are not equipped to handle. Most of their work is done on distances that matter the least; that is, their error functions are dominated by these $O(N^2)$ large distances.

Unfortunately, dimensionality reduction approaches that attempt to handle modisco data by discounting the error contributions of large distances fail to solve the problem. As Buja and Swayne point out [BS02], attempts to adapt MDS to operate from predominantly local information run into trouble because the computation often does not converge to meaningful global configurations, a problem faced by the proposal of Shepard and Carroll [SC66] to reduce in local neighborhoods based on distances.

Buja and Swayne also proposed *within-groups* MDS to solve only the local problem, where groups of items with homogenous distances would be separately laid out [BS02]. However, they did not suggest how to carry out that partition; DiscoTree is exactly such an algorithm.

3.4. Disconnected Component Tree

Because adapting MDS to handle structure at multiple scales is difficult for fundamental mathematical reasons, our proposal is to build another data structure that can be used in conjunction with standard MDS. The main idea behind the DiscoTree is to interpret distances greater than a specific threshold distance as indicating a lack of connection and decompose the dataset into connected components. Distance thresholding is a transformation that discards information for the purpose of clarity; while some global relationships might be lost, important local structure is revealed that is obscured when all distances are considered.

Varying this threshold distance from the maximum possible inter-point distance to the minimum distance produces an ordered set of component decompositions. Lowering the threshold can cause a previously connected component to split apart into multiple pieces. The relationship between all of these components forms a tree, representing the sequence of ever-finer decompositions of components into smaller ones; we call this the Disconnected Component Tree, or DiscoTree for short. It is an example of a hierarchical clustering. The single component at the root encompasses all items in the dataset, and the leaves are singleton nodes with one item each. Figure 2 shows a simple example. The DiscoTree explicitly shows the full space of possible disconnections and their resulting groupings directly; it represents neighborhood structure at multiple scales.

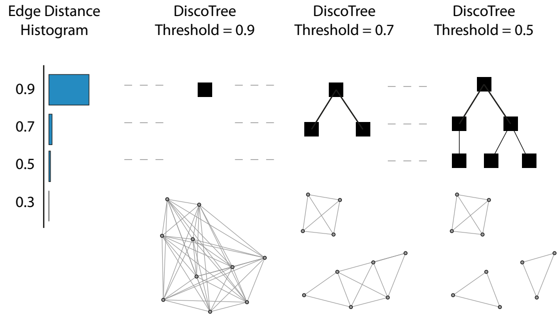

Edge Distance Histogram
DiscoTree Threshold = 0.9
DiscoTree Threshold = 0.7
DiscoTree Threshold = 0.5
0.9
0.7
0.5
0.3

Figure 2: *Constructing the DiscoTree.*

Considered as a clustering algorithm, DiscoTree is most similar the single-link clustering algorithm SLINK [Sib73]. The output of the two algorithms will be similar, and in certain cases identical. While the obvious implementation of SLINK is $O(N^3)$ and an optimized version runs in $O(N^2)$

time, DiscoTree exploits the characteristics of the mostly-disconnected data to achieve $O(N)$ performance, allowing a cluster analysis of much larger datasets.

Many approaches to clustering rely on the number of requested clusters as an input parameter, but that number is not known in advance in many tasks, including the exploration and annotation of large document collections. Thus, the popular k-means [Mac67] and subspace clusterings such as PROCLUS [AWY*99] would not be appropriate. The LDA topic modelling approach scales to massive feature spaces [Ble11], but it too requires the number of clusters as input. While a recent nonparametric approach claims to relax this requirement, no complexity or benchmark information is provided [BGJ10]. In contrast, DiscoTree does not require the number of clusters to be specified in advance and is scalable to large datasets with tens of thousands of input dimensions.

4. Efficiently Computing the DiscoTree

We propose a deterministic sampling approach to build a DiscoTree in $O(N)$ time. The main idea is that we build a graph G with a vertex for each input point, and with edges sampled from the set of distances. We discretize the set of possible distance thresholds into t equally spaced values, after normalizing all distances to the range 0..1. We then create a set of t subgraphs by filtering out the edges longer than the threshold value. For each subgraph, we find its connected components. We link components between subgraphs at neighboring levels if they have vertices in common.

The subgraph operations of filtering, connected component decomposition, and linking are all straightforward $O(N)$ operations that are performed t times, where t is a constant, so this part of the algorithm is $O(N)$. Building the list of points for G is a trivial copy operation, with $O(N)$. The potentially expensive part of this algorithm is building the list of edges for G, which would have $O(N^2)$ cost without sampling. We present an $O(N)$ time approximation where we deterministically sample a constant number of edges for each of the N vertices. The sampling produces short rather than long edges to exploit the modisco property that most of the distances are close to the maximum distance and should be ignored. Similarly, we rely on the fact that modisco data is sparse, having on average a constant number of nonzero dimensions per data point.

Our sampling strategy requires keeping two data structures, both arrays of sorted lists. These sorted lists allow us to quickly generate distances that are likely to be small; that is, to sample with a strong bias towards short edges. The algorithm we describe here is tailored for the cosine similarity metric and is suitable for any distance metric where shared nonzero dimensions imply similarity, such as the Jaccard metric [MSS83].

The array D is indexed by dimension, and each entry $D[i]$ is a list of (point,coordinate) pairs for all points that have a nonzero coordinate in dimension i. This list is sorted by coordinate values in descending order. The other array P is indexed by point, and each entry $P[k]$ is a priority queue of the nonzero dimensions for point k. Each item in the queue is a triple: the dimension index i, a counter j into the list stored in $D[i]$, and the priority value r that determines its order in the queue. A sampled edge is generated from a triple by computing the distance between the current point and the point stored in the jth element of $D[i]$. Then the counter j is incremented, the priority value r is recalculated, and then the triple is re-inserted in the queue. If j is greater than the length of the corresponding list $D[i]$, then the triple is not re-inserted in the queue.

The priority value r for each item in the queue $P[k]$ is the product of ith dimension values of two points: k and the jth point stored in $D[i]$'s list. Using this value as the priority order guarantees that a given point will visit the other data points in order of the largest factor in their dot product. A larger dot product translates into smaller cosine distance, so this scheme is strongly biased toward short edges.

We iterate through each point $p = P[i]$ in the point array P and generate m edge samples for it, which we store in a hash table. These sampled edges are then used for all threshold stages of the DiscoTree construction.

We empirically determined that 200 is a good value for m, yielding a fully connected sampled graph G. For example, 600K edges out of the 9M possible were sampled for a set of 3K points, yielding a sampled edge set of roughly 7% of the total edges. We expect that the absolute number of samples per point will not need to increase as the number of points increases because the connectivity properties of a random graph depend only on the average node degree [JKŁP93].

4.1. DiscoTree Validation

We validate the DiscoTree with runtime benchmarks of speed and approximation quality. In all validations we use the two datasets described further in Section 5: `Warlogs` with 3077 points and 4286 dimensions, and `Cables` with 6849 points and 65,777 dimensions.

The clock time to construct the DiscoTree on `Warlogs` and `Cables` is 5 and 6 seconds respectively. The test machine was a MacBook with an 2.4GHz Intel Core 2 Duo and 2GB RAM. The algorithm prototype was implemented in Java 6.

Figure 3 shows the sampling behavior as the algorithm runs, validating our claims that the short distances needed to compute the DiscoTree are well sampled by the algorithm while most of the maximum distance edges are not sampled.

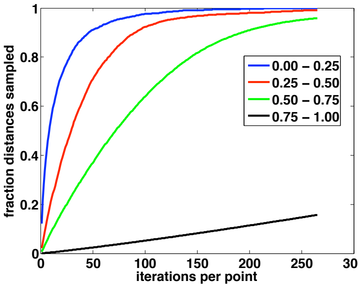


Figure 3: *Sampling performance on the* `Warlogs` *dataset, showing that the algorithm succeeds in producing the desired sample bias. The percentage of the complete edge set sampled is plotted against the number of edges sampled for each of the 3K points. The distances are binned into the categories of short, medium, long, and very long.*

5. MoDiscoTag Results

MoDiscoTag is a proof of concept application to address the task defined in Section 2, computer-assisted human classification through tagging items. It is based on combining clustering, tagging, and dimensionality reduction. The cluster view features an interactive representation of the DiscoTree, and the DR view shows a 2D scatterplot of a low-dimensional embedding created with the Glimmer MDS technique [IMO09]. The tag editor allows tags to be created or highlighted. Below these three windows is an active set view showing DiscoTree nodes on the left and items on the right, where the labels are the top terms of within a document, or of all documents within the cluster. The views are all linked to support cross-filtering [Wea10].

The most relevant previous work in the area is perhaps the Newdle system [YLL10]. It is also oriented around hierarchical clustering of topics and tag-based analysis. They do not incorporate linked dimensionality reduction, and they show only a particular cut through the hierarchy at once, whereas our interface based on DiscoTree allows the user to explore the entire multiscale structure. Chen et al. [CWDH09] also take a sampling approach to dimensionality reduction for large document collections to produce layouts that show clear clusters, but do not discuss any sort of interactive browsing or annotation. Österling et al. compute the contour tree of a density field, which has some conceptual similarities to the DiscoTree [OST*10]. We choose a point-based visual encoding rather than their landscape-based approach based on guidelines from previous empirical studies [TSW*07, TSD09].

We show results with two real-world journalism datasets from WikiLeaks, `Warlogs` and `Cables`. In both, documents were encoded as a vector using the TF-IDF term weighting scheme [SWY75] applied to all vocabulary words plus automatically detected common bigrams. The supplementary video showcases these case studies at more length.

5.1. Afghan War Logs

The `Warlogs` dataset is the subset of the WikiLeaks Afghan Warlogs dataset from July 2009. It has 3077 points and 4286 dimensions. These documents are military after-action reports with an extremely terse format and specialized jargon, so they are not trivial for non-experts to read.

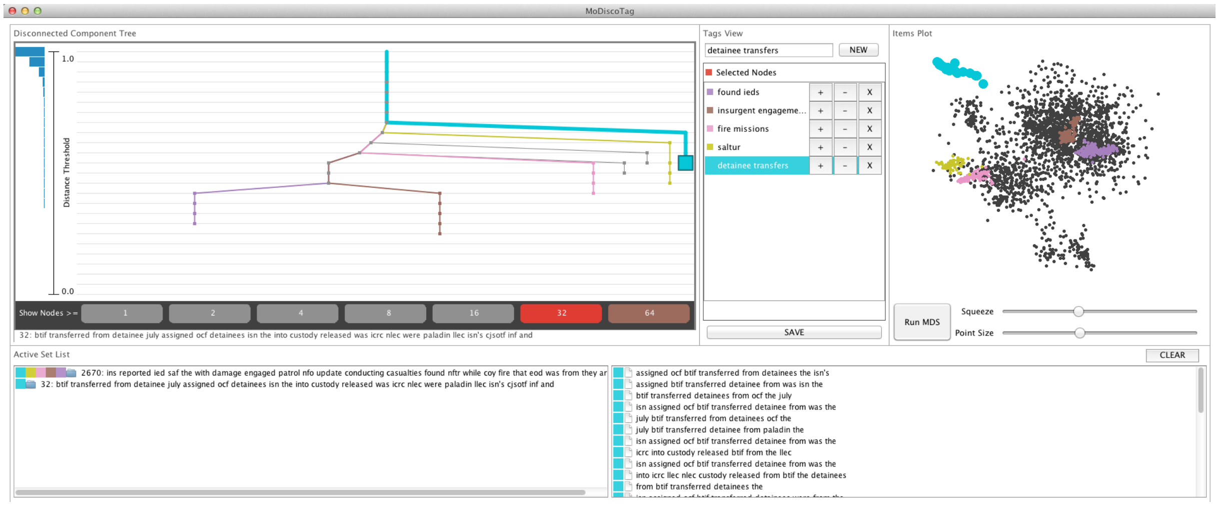
Figure 4 shows the results. We began with the most compact pruning level where only nodes of 64 or more items are visible in the DiscoTree, revealing seven main clusters. Exploring those with the combination of quickly reading labels in the `Active Set List` and using the `Item Viewer` to read the full text of documents led us to quickly confirm five these as semantically meaningful categories and tag them with the following names and colors: *found ied* (purple), *insurgent engagements* (brown), *fire mission* (pink), missions containing a *saltur* report (size, activity, location, time, unit, result) following an enemy contact (gold), and *detainee transfers* (teal).

We can see that these groups do form neighborhoods in the MDS `Items Plot`, but would be difficult to identify without the coloring because the regions are not visually separated from the rest of the points by regions of low density. The exception is the well separated *detainee transfers*.

For clarity, the DiscoTree view is interactively prunable so that only nodes past a particular size are visible, using the logarithmic `Show Nodes >=` buttons along the bottom. Each node in the DiscoTree contains many document items, and due to the hierarchical clustering the same item will appear in every node along a path from the singleton leaf node at the bottom of the tree up to the root. When an item is selected through one of the other views, this entire branch is highlighted through an edge color.

The user can use the `Tags Editor` to create and assign an arbitrary set of named and colored tags. Clicking on a tag's name selects all items which have been assigned to that tag. While the user interface supports the useful shortcut of tagging all items in a node, in general tags may be arbitrarily distributed across items. Nodes are assigned a tag's color when all items within that node contain that tag. Due to the hierarchical nature of the tree, once a node is colored in, all of its child nodes will also be filled in. When the tree is pruned, then the lowest visible node along a branch is a filled in as proxy for what is hidden beneath it.

Each item in the MDS `Items Plot` is a single document, represented by a point. An individual item can have many tags attached to it, and a node has many items and thus many tags as well. We do not attempt to show all tags at once with any sort of glyph, since items and nodes cover small regions of the screen. Instead, the last selection color takes priority over the others.



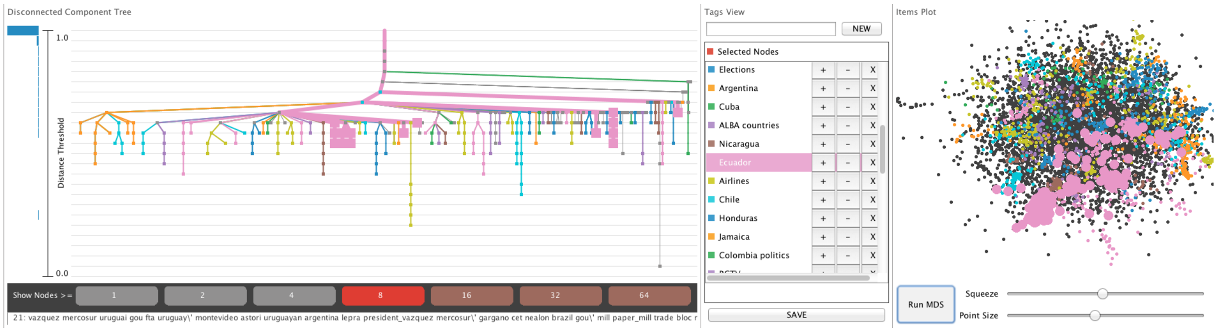
detainee transfers

32: btif transferred from detainee july assigned ocf detainees isn the into custody released was icrc nlec were paladin llec isn's cjsotf inf and

2670: ins reported ied saf the with damage engaged patrol nfo update conducting casualties found nftr while coy fire that eod was from they ar
32: btif transferred from detainee july assigned ocf detainees isn the into custody released was icrc nlec were paladin llec isn's cjsotf inf and

assigned ocf btif transferred from detainees the isn's
assigned btif transferred detainee from was isn the
btif transferred detainees from ocf the july
isn assigned ocf btif transferred detainee from was the
july btif transferred from detainees ocf the
july btif transferred detainee from paladin the
isn assigned ocf btif transferred detainee from was the
icrc into custody released btif from the llec
isn assigned ocf btif transferred detainee from was the
into icrc llec nlec custody released from btif the detainees
from btif transferred detainees the

Figure 4: *The* `Warlogs` *dataset after five nodes of the DiscoTree have been tagged. The colors reveal that the documents they contain fall into local neighborhoods in the MDS* `Items Plot`*, but most of this structure cannot be seen from proximity relationships alone because there is no visible separation from the other data points.*



21: vazquez mercosur uruguai gou fta uruguay\' montevideo astori uruguayan argentina lepra president_vazquez mercosur\' gargano cet nealon brazil gou\' mill paper_mill trade bloc r

Figure 5: *The* `Cables` *dataset, with the full set of categories from the AP Caracas Bureau Chief.*

5.2. Caracas Cables

The `Cables` dataset comes from a different WikiLeaks source, the diplomatic cables. We analyzed the subset consisting of cables sent to or from the US Embassy in Caracas, Venezuela, or containing the word “Caracas”. It has 6849 points and 65,777 dimensions.

We provided this dataset and a prototype of MoDiscoTag to the Associated Press Caracas bureau chief, who created several dozen tags over a session of a few hours. These tags, shown in Figure 5, cross-cut the data in different ways including geography, politics, and events; the supplemental video shows prototype on each of these three tags sets. Figure 7 shows a different subset of ten interesting tags. Tags are applied to documents, not nodes, because the automatically generated tree does not necessarily categorize the documents in a way that is meaningful to the user. For example, several different branches contain documents concerning *Ecuador* in Figure 5. Figure 6 shows hierarchical cluster structure, where the parent-child relationships between the branch tagged with *Finance* and its children about banking and oil are also visible as spatial nesting in the MDS view.

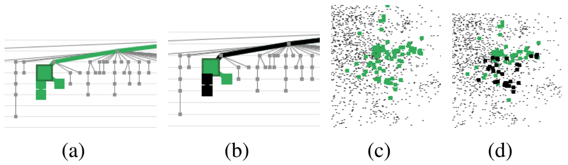
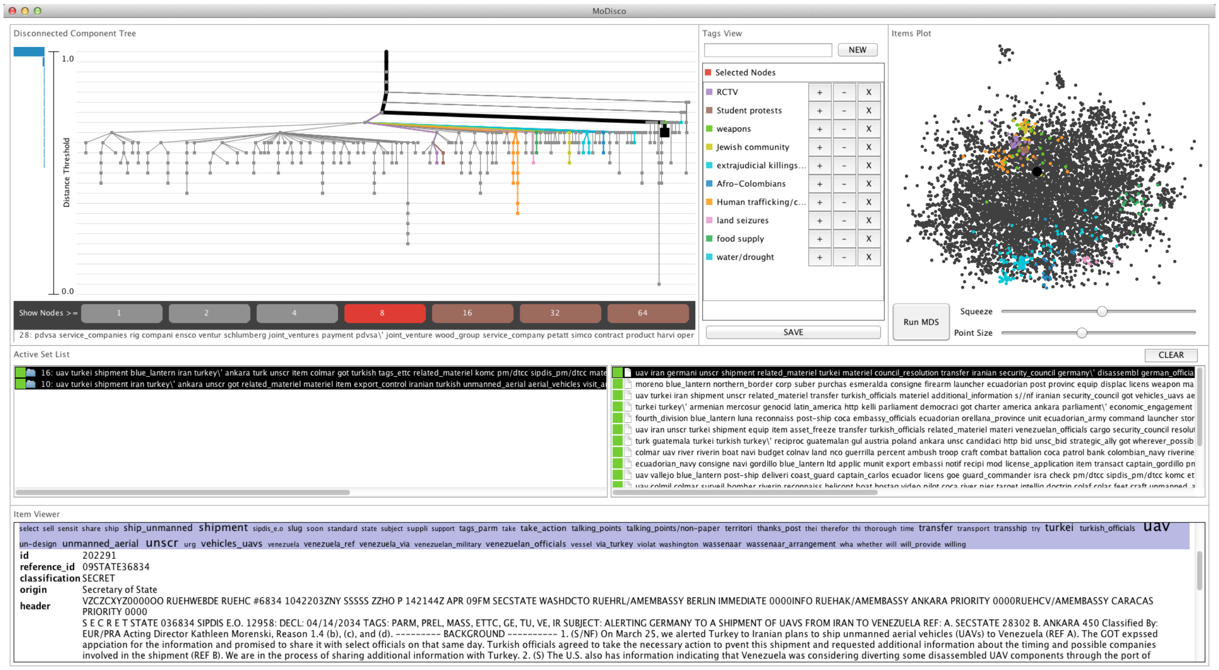


Figure 6: *Hierarchical structure can be seen in both views. The branch tagged with Finance has children concerning banking and oil.* `DiscoTree View` *detail when finance tag selected (a) and one child node selected (b);* `Item View` *detail for finance alone (c) and child (d).*



28: pdvsa service_companies rig compani ensco ventur schlumberg joint_ventures payment pdvsa\' joint_venture wood_group service_company petatt simco contract product harvi oper

16: uav turkei shipment blue_lantern iran turkey\' ankara turk unscr item colmar got turkish tags_ettc related_materiel komc pm/dtcc sipdis_pm/dtcc mate

10: uav turkei shipment iran turkey\' ankara unscr got related_materiel materiel item export_control iranian turkish unmanned_aerial aerial_vehicles visit_a

uav iran germani unscr shipment related_materiel turkei materiel council_resolution transfer iranian security_council germany\' disassembl german_officia

moreno blue_lantern northern_border corp suber purchas esmeralda consigne firearm launcher ecuadorian post provinc equip displac licens weapon ma

uav turkei iran shipment unscr related_materiel transfer turkish_officials materiel additional_information s//nf iranian security_council got vehicles_uavs ae

turkei turkey\' armenian mercosur genocid latin_america http kelli parliament democraci got charter america ankara parliament\' economic_engagement

fourth_division blue_lantern luna reconnaiss post-ship coca embassy_officials ecuadorian orellana_province unit ecuadorian_army command launcher stor

uav iran unscr turkei shipment equip item asset_freeze transfer turkish_officials related_materiel materi venezuelan_officials cargo security_council resolu

turk guatemala turkei turkish turkey\' reciproc guatemalan gul austria poland ankara unsc candidaci http bid unsc_bid strategic_ally got wherever_possib

colmar uav river riverin boat navi budget colnav land nco guerrilla percent ambush troop craft combat battalion coca patrol bank colombian_navy riverine

ecuadorian_navy consigne navi gordillo blue_lantern ltd applic munit export embassi notif recipi mod license_application item transact captain_gordillo pn

uav vallejo blue_lantern post-ship deliveri coast_guard captain_carlos ecuador licens goe guard_commander isra check pm/dtcc sipdis_pm/dtcc komc et

uav colmil colmar surveil bomber riverin reconnaiss helicopt boat hostag video pilot coca river pier target intellig doctrin colaf colar feet craft unmanned_a

select sell sensit share ship ship_unmanned shipment sipdis_e.o slug soon standard state subject suppli support tags_parm take take_action talking_points talking_points/non-paper territori thanks_post thei therefor thi thorough time transfer transport transship try turkei turkish_officials uav un-design unmanned_aerial unscr urg vehicles_uavs venezuela venezuela_ref venezuela_via venezuelan_military venezuelan_officials vessel via_turkey violat washington wassenaar wassenaar_arrangement wha whether will will_provide willing

id	202291
reference_id	09STATE36834
classification	SECRET
origin	Secretary of State
header	VZCZCXYZ0000OO RUEHWEBDE RUEHC #6834 1042203ZNY SSSSS ZZHO P 142144Z APR 09FM SECSTATE WASHDCTO RUEHRL/AMEMBASSY BERLIN IMMEDIATE 0000INFO RUEHAK/AMEMBASSY ANKARA PRIORITY 0000RUEHCV/AMEMBASSY CARACAS PRIORITY 0000
	S E C R E T STATE 036834 SIPDIS E.O. 12958: DECL: 04/14/2034 TAGS: PARM, PREL, MASS, ETTC, GE, TU, VE, IR SUBJECT: ALERTING GERMANY TO A SHIPMENT OF UAVS FROM IRAN TO VENEZUELA REF: A. SECSTATE 28302 B. ANKARA 450 Classified By: EUR/PRA Acting Director Kathleen Morenski, Reason 1.4 (b), (c), and (d). --------- BACKGROUND ---------- 1. (S/NF) On March 25, we alerted Turkey to Iranian plans to ship unmanned aerial vehicles (UAVs) to Venezuela (REF A). The GOT expssed appciation for the information and promised to share it with select officials on that same day. Turkish officials agreed to take the necessary action to pvent this shipment and requested additional information about the timing and possible companies involved in the shipment (REF B). We are in the process of sharing additional information with Turkey. 2. (S) The U.S. also has information indicating that Venezuela was considering diverting some disassembled UAV components through the port of

Figure 7: *The journalist found a cable alleging Iranian plans to ship unmanned aerial vehicles to Venezuela.*

The journalist found several topics that he had not previously found through unassisted inspection of the dataset, including arms shipments to Ecuador. He noted that the application allowed him to quickly spot subject areas that could be of greater news interest, such as information on Colombian rebels. The tool also helped him find several interesting individual documents, for example claims that Chavez was giving millions to a particular Jamaican politician's election campaign, and the cable alleging Iranian plans to ship unmanned aerial vehicles to Venezuela shown in Figure 7.

6. Conclusions and Future Work

In this paper we focus on evoking an intuitive sense of the modisco property and how it relates to real-world datasets of interest, rather than formal mathematical definitions. We show empirically that the vector space representation of document sets has the modisco property, and argue that all very high dimensional datasets do; we conjecture that some lower dimensional ones do as well. While the modisco property is a problem for multidimensional scaling algorithms, interpreting these distances as disconnection allows us to produce a an efficient $O(N)$ clustering algorithm, the DiscoTree. The algorithm remains $O(N)$ if run on non-modisco datasets but would yield a poor quality clustering.

One next step would be to further define the boundaries of what datasets have modisco characteristics both empirically and theoretically. We show results specifically for natural language documents, but it would be interesting to test our ideas on non-linguistic corpora such as image, video, or audio collections. It may also be possible to modify existing DR algorithms to work better with this class of data. Conversely, extending the DiscoTree sampling algorithm to work on arbitrary distance metrics would also be useful.

It remains to explore the relationship between the DiscoTree, which clusters documents directly in the vector space representation, and the topic modeling algorithms widely employed in document set analysis. Such algorithms amount to dimensionality reduction in that they represent each document as a convex mixture of extracted topics [Ble11], often used as a space of intermediate dimensionality before projecting to 2D, as in Österling et al. [OST*10].

Because categorization is semantically complex and application specific, we define the document set exploration task as computer-assisted human categorization. We present the MoDiscoTag application for this task that combines a DiscoTree view and a traditional MDS view. We validate with two complex, real-world datasets from the journalism domain: subsets of the diplomatic cables and Afghan war logs from WikiLeaks. It could be interesting to extend the MoDiscoTag application to incorporate alternate hierarchical clustering approaches, to compare them directly with the DiscoTree. Also, while the DiscoTree algorithm is $O(N)$ time and should scale to very large data sets, the proof-of-concept MoDiscoTag user interface will not scale to more than a few tens of thousands of items due to screen space limitations. More generally, the MoDiscoTag application allows direct comparison of different item encoding, clustering, and layout algorithms with respect to each other and a fixed set of human-assigned tags, opening up a line of research into the semantic relationships between these different algorithmic stages in the visual exploration pipeline.

References

[AWY*99] Aggarwal C. C., Wolf J. L., Yu P. S., Procopiuc C., Park J. S.: Fast algorithms for projected clustering. *ACM SIGMOD Record 28*, 2 (1999), 61–72. 6

[Bai94] Bailey K. D.: *Typologies and taxonomies: an introduction to classification techniques*. Sage, Beverly Hills, 1994. 1

[Bel61] Bellman R.: *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961. 4

[Ber06] Berkhin P.: A survey of clustering data mining techniques. *Grouping Multidimensional Data* (2006), 25–71. 1

[BG05] Borg I., Groenen P.: *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005. 5

[BGJ10] Blei D. M., Griffiths T. L., Jordan M. I.: The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM 57* (February 2010), 7:1–7:30. 6

[Ble11] Blei D. M.: Introduction to probabilistic topic models. *Communications of the ACM* (2011). to appear. 3, 6, 9

[BS02] Buja A., Swayne D. F.: Visualization methodology for multidimensional scaling. *Journ. Classification 19*, 1 (2002), 7–43. 2, 3, 4, 5

[CLT*11] Cui W., Liu S., Tan L., Shi C., Song Y., Gao Z., Tong X., Qu H.: TextFlow: Towards better understanding of evolving topics in text. *IEEE Trans. Visualization Computer Graphics (Proc. InfoVis 2011) 17*, 12 (2011). 3

[CW06] Crestani F., Wu S.: Testing the cluster hypothesis in distributed information retrieval. *Information Processing & Management 42* (2006), 1137–1150. 3

[CWDH09] Chen Y., Wang L., Dong M., Hua J.: Exemplar-based visualization of large document corpus. *IEEE Trans. Visualization and Computer Graphics (TVCG) 15*, 6 (2009), 1161–1168. 1, 7

[FA10] Frank A., Asuncion A.: University of California Irvine (UCI) Machine Learning Repository, Bag of Words, 2010. `http://archive.ics.uci.edu/ml/datasets/Bag+of+Words`, last accessed Dec 2011. 5

[GK10] Grimmer J., King G.: General purpose computer-assisted clustering and conceptualization. *Proc. Natl. Acad. Sciences (PNAS)* (2010). 3

[IMO09] Ingram S., Munzner T., Olano M.: Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Visualization and Computer Graphics (TVCG) 15*, 2 (2009), 249–261. 4, 7

[JKŁP93] Janson S., Knuth D. E., Łuczak T., Pittel B.: The birth of the giant component. *Random Structures & Algorithms 4*, 3 (1993), 233–358. 6

[Jol02] Jolliffe I. T.: *Principal Component Analysis, 2nd ed.* Springer, 2002. 5

[JvR71] Jardin N., van Rijsbergen C. J.: The use of hierarchical clustering in information retrieval. *Information Storage & Retrieval 7*, 5 (1971), 217–240. 3

[Luh57] Luhn H. P.: A statistical approach to the mechanized encoding and searching of literary information. *IBM Journal of Research and Development 1*, 4 (1957), 309–317. 2

[Mac67] MacQueen J. B.: Some methods for classification and analysis of multivariate observations. In *Proc. Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, University of California Press, pp. 281–297. 6

[MH08] Maaten L. V. D., Hinton G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 5

[MRS08] Manning C. D., Raghavan P., Schütze H.: *Introduction to Information Retrieval*. Cambridge University Press, 2008. 3

[MSS83] Milligan G. W., Soon S. C., Sokol L. M.: The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 5*, 1 (1983). 6

[OST*10] Oesterling P., Scheuermann G., Teresniak S., Heyer G., Koch S., Ertl T., Weber G.: Two-stage framework for a topology-based projection and visualization of classified document collections. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)* (2010), pp. 91–98. 5, 7, 9

[RU11] Rajarama A., Ullman J. D.: *Mining of Massive Data Sets*. Cambridge University Press, 2011. 2, 4

[SB88] Salton G., Buckley C.: Term weighting approaches in automatic text retrieval. *Information Processing & Management 24*, 5 (1988), 513–523. 2

[SC66] Shepard R. N., Carroll J. D.: Parametric representation of nonlinear data structures. In *Multivariate Analysis*, Krishnaiah P. R., (Ed.). Academic Press, 1966, pp. 561–592. 5

[Sib73] Sibson R.: SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal 16*, 1 (1973), 30–34. 5

[SWY75] Salton G., Wong A., Yang C. S.: A vector space model for automatic indexing. *Information Retrieval & Language Processing 18*, 11 (1975), 613–620. 1, 2, 5, 7

[TSD09] Tory M., Swindells C., Dreezer R.: Comparing dot and landscape spatializations for visual memory differences. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009) 16*, 6 (2009), 1033–1040. 7

[TSW*07] Tory M., Sprague D. W., Wu F., So W. Y., Munzner T.: Spatialization design: Comparing points and landscapes. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 07) 13*, 6 (2007), 1262–1269. 7

[Wea10] Weaver C.: Cross-filtered views for multidimensional visual analysis. *IEEE Trans. Visualization and Computer Graphics 16*, 2 (2010), 192–204. 7

[WTP*95] Wise J., Thomas J., Pennock K., Lantrip D., Pottier M., Schur A., Crow V.: Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proc. IEEE Symp. Information Visualization (InfoVis)* (1995), pp. 51–58. 1

[YLL10] Yang J., Luo D., Liu Y.: Newdle: Interactive visual exploration of large online news collections. *IEEE Computer Graphics & Applications 30*, 5 (2010), 32–41. 7

[Zha02] Zha H.: Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proc. ACM Conf. Research & Devel. Information Retrieval (SIGIR)* (2002), pp. 113–120. 3