Int J Digit Libr (2000) 3: 170–184 © 2000 Springer-Verlag

Strategy-based interactive cluster visualization for information retrieval

Anton Leuski, James Allan

Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA;
E-mail: leuski@cs.umass.edu, allan@cs.umass.edu

Received: 21 December 1998/Revised: 30 May 1999

Abstract. In this paper we investigate a general purpose interactive information organization system. The system organizes documents by placing them into 1-, 2-, or 3-dimensional space based on their similarity and a spring-embedding algorithm. We begin by developing a method for estimating the quality of the organization when it is applied to a set of documents returned in response to a query. We show how the relevant documents tend to clump together in space. We proceed by presenting a method for measuring the amount of structure in the organization and explain how this knowledge can be used to refine the system. We also show that increasing the dimensionality of the organization generally improves its quality, albeit only a small amount. We introduce two methods for modifying the organization based on information obtained from the user and show how such feedback improves the organization. All the analysis is done offline without direct user intervention.

Key words: Information organization – User interface – Evaluation

1 Introduction

An important part of a digital library is the ability to access the stored information effectively. Due to recent achievements in the area of information retrieval, a digital library is usually equipped with an automatic search and retrieval system that users of the library may employ to find documents. Such a system accepts a free text ("natural language") query and responds with a list of documents in the order they are most likely to be relevant: the first document is the best match to the user's query, the second is the next most likely to be helpful, and so on. We are interested in situations where this simple model breaks down — where the user is unable to find enough relevant material in the first ten retrieved documents. In particular, we are interested in helping a searcher find all of the relevant material in the ranked list without forcing him or her to wade through all of the non-relevant material. We believe that in this case an information organization technique that arranges the retrieved data and reveals how individual documents relate to each other will help the user to isolate relevant material quickly.

The purpose of this paper is twofold. First, we define an evaluation approach that we believe can be used to analyze interactive information organization techniques. The main idea of the approach is to perform multiple offline simulations of a user interacting with the system. Second, we apply this evaluation framework to investigate an interactive visualization technique where retrieved documents are placed in space and positioned according to the similarity among them [5]. We study the visualization for the specific task of helping the user find the interesting material in the retrieved document set.

There are several different ways of evaluating interactive systems. A user study is probably the most widely employed method. Here a person is involved in applying the system to a number of tasks. The user's performance is measured and used as the quality estimator. An example of a user study is the interactive track included in TREC [15]. It is a good example of how the "overall" performance of both the user and the system working together is measured. User studies are also applied to evaluate some particular aspects of the system. For example, in their user study of the Scatter/Gather system, Hearst and Pedersen [16] showed that users seem able to choose the cluster with the largest number of relevant documents using the textual summaries the system creates. User studies usually are very expensive, time-consuming and difficult to execute. Designing a good and informative user study is almost work of art.

A different evaluation approach is the predictive evaluation method (e.g., see Card and Morgan [8]). This technique estimates how fast a particular task can be executed using the system. It requires the task to be defined precisely and the system is evaluated particularly for this task. This is achieved by subdividing the task into a number of unit actions such as key presses and mouse clicks, where the time necessary to perform the unit actions is known. A set of possible strategies that combine the unit actions together is generally assumed for the user. This is similar to our approach as we are interested in how fast the user can locate all relevant information and we also assume different strategies for the user. However, we are interested in measuring the amount of data the user is forced to analyze before finding all relevant documents and not the actual time that is required to complete the search.

1.1 Visualization approaches

Multiple visualization approaches have been developed in recent years. Generally these visualizations are designed to present some type of patterns in a document set and are considered to be browsing interfaces. To our knowledge there have been no studies on how such visualizations help the user locate relevant information.

The Scatter/Gather interface [16] presents the document clusters as text. It groups the documents into five (or any preselected number) clusters and displays them simultaneously as lists. On a large enough screen, the top several documents from each cluster are clearly visible. Another text-based visualization is presented by Leouski and Croft [18]. Their method is similar to the one used by Scatter/Gather, but the number of clusters is based on a similarity threshold. Their display looks more like a standard ranked list because they can have an arbitrarily large number of clusters (limited only by the size of the retrieved set).

It is very common for clusters to be presented graphically. The documents are usually presented as points or objects in space with their relative positions indicating how closely they are related. Links are often drawn between highly-related documents to make it clearer that there is a relationship.

1.1.1 2D Visualization

Allan [1, 2] developed a visualization for showing the relationship between documents and parts of documents. It arrayed the documents around an oval and connected them when their similarity was strong enough. Allan's immediate goal was not to find the groups of relevant documents, but to find unusual patterns of relationships between documents.

The Vibe system [12] is a 2D display that shows how documents relate to each other in terms of user-selected dimensions. The documents being browsed are placed in the center of a circle. The user can locate any number of terms inside the circle and along its edge, where they form "gravity wells" that attract documents depending on the significance of that term in that document. The user can shift the location of terms and adjust their weights to better understand the relationships between the documents.

1.1.2 3D Visualization

High-powered graphics workstations and the visual appeal of 3-dimensional graphics have encouraged efforts to present document relationships in 3-space. The LyberWorld system [17] includes an implementation of the Vibe system described above, but presented in 3-space. The user still must select terms, but now the terms are placed on the surface of a sphere rather than the edge of a circle. The additional dimension should allow the user to see separation more readily.

Our system is similar in approach to the Bead system [9] in that both use forms of spring embedding for placing high-dimensional objects in 3-space. The Bead research did not investigate the question of separating relevant and non-relevant documents. Figure 2 shows sample visuals of our system (they are explained in more detail in later sections).

In this study we show how the relevant documents tend to clump with each other in space. We present a method for measuring the amount of structure in the organization and explain how this knowledge can be used to refine the system. We also show that increasing the dimensionality of the organization generally improves its quality. We introduce two methods for modifying the organization based on the information obtained from the user and show how such feedback improves the organization.

2 Evaluation framework

Consider the interaction process that occurs between a user and an information organization system. Figure 1 presents a simple model of this process. We assume that when the user turns to the organization system she/he has a particular goal to achieve, a *task* to complete. In our study we will consider the task of identifying the relevant material in the document set returned by an information retrieval system. The information organization system analyzes the provided information (the retrieved documents in our example), builds a *data model*, and uses the model to organize the data. The organization is shown to the user and the information reaches the user in the form of *"clues"*. For example, in the system presented in this paper the main "clue" is the spatial proximity of the documents that reflects the inter-document similarity — if the documents are shown nearby, they are

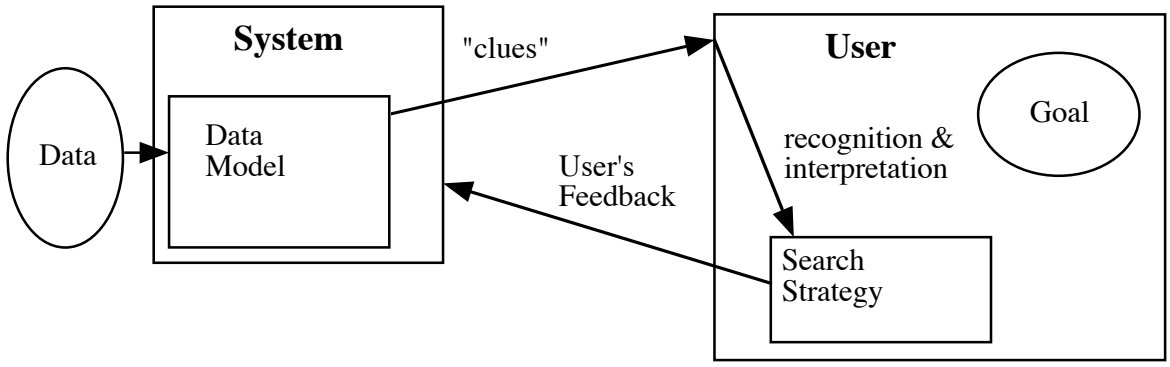


Fig. 1. The relationship between the system and the user in an interactive information organization setting

probably about the same topic. It is for the user to recognize and interpret the supplied "clues" as effectively as possible, apply this knowledge, and decide what documents to view and in what order. We call this decision process the *search strategy*. It might be as simple as selecting the next document randomly, or something more elaborate, e.g., "find a relevant document, find another relevant document in close proximity to the first one, keep looking at the documents by going away from the first document in the direction of the second document." After a document is selected, the user makes a relevance assessment and passes the judgments to the system. The system takes the *user's feedback* and adjusts the data model.

The described interaction model defines a general process in which the system and the user are working together to achieve a predefined goal. Both serve as two components of a large "engine" that drives toward that goal. The quality of the system is estimated by applying the engine to a known data set and measuring how well the engine handles the task. The combined efforts of the system and the user are measured — i.e., the quality of how well the system and the user are able to perform the given task. Generally, real users participate in such experiments — a user study is performed.

We suggest an alternative where the user is "replaced" by a probabilistic model of some search strategy. This leads to a change in the research question: instead of investigating how well the system and the user perform a certain task, we study how well the system is able to support a random prototypical "user" in this task with that strategy. In other words, we compute the lower bound estimate of the system performance and isolate the system's effect on the overall quality. We call this approach the *Strategy-based Evaluation Method (SEM)*. The quality of the system becomes a function of both the task and the search strategy. By considering different search strategies we can investigate which one is the most suitable for the given organization system. We can recommend an effective way of using the system. More traditional user studies could be employed to validate the proposed strategies or to confirm their usability.

The following is a short summary of the Strategy-based Evaluation Method. We use this framework to present and analyze the information visualization system in our study and to organize the rest of the material.

- *Experimental task.* When a user turns to the information organization system he or she generally has a goal in mind. Thus, we always evaluate the system relative to a particular task. In this study we consider information organization as an information retrieval problem and the tasks we employ in the evaluation are retrieval tasks.
- *System design.* The system, including the data model and the algorithm for data organization, is the first main component. We specify what kind of feedback the system accepts and how this information affects the data model. The interface or the visualization part of the system is also very important. We describe what information about the data model is communicated to the user and how these "clues" are presented.
- *Search strategy.* A probabilistic *recognition value* is attached to each "clue". Then the user strategy is defined as a decision making process. The strategy is not necessarily deterministic: it could be random, as long as we have a probability for each part of the strategy determined a priori.
- *Performance measure.* The performance measure depends on the task in hand significantly. Several different statistics might be used to evaluate the same task.
- *Experiments.* The performance is computed analytically or by simulating the task multiple times. It is repeated for multiple data sets. If the user model contains a random factor, the result could be a probability distribution for the performance.

The following sections give more details on each part of the evaluation method.

3 Experimental task

Our evaluation approach is task-oriented — we assume that the user is working with the system to achieve a particular goal. In this study we consider information organi-

zation in the context of information retrieval. In particular, we study automatic organization of documents that were retrieved in response to a known query by an information retrieval system. Therefore, we assume that there exists a collection of documents and a topic of interest. On the basis of this topic a query is created and is used by the retrieval system to find a number of documents that are supposed to be relevant to the topic. The retrieved documents are usually organized in a ranked list according to their probability of being relevant. Generally, only a few of the retrieved documents are actually relevant. The user is faced with the task of locating the relevant documents among those retrieved. We believe that an organization system more sophisticated than a ranked list will make the process of locating the relevant material more effective.

The task of locating the relevant information is the process we analyze in this study. We assume that the user has located a few of the relevant documents — we believe this is a reasonable strategy and almost always could be done by looking at the titles in the ranked list. We investigate how the visualization helps to locate the rest of the interesting (relevant) documents. Thus, the experimental task is: given that some of the documents presented by the information organization system are marked as relevant or non-relevant, isolate the rest of the relevant material without encountering non-relevant documents.

4 System design

In this section we describe in detail how the visualization system works, how it represents the documents and what kind of user feedback it accepts. We define a technique that allows us to estimate the amount of spatial structure in the images generated by the system. We also show some examples of such images.

The system works by placing the retrieved documents in 1-, 2-, or 3-dimensional space according to the similarity among them [5]. The documents are represented as vectors of terms with vector size equal to the vocabulary size of the retrieved set. Each retrieved document's vector defines a point in a high-dimensional space. The distances between these points and their relative positions are strong indicators of the similarity among the corresponding documents. Unfortunately, it is difficult to visualize objects in more than three dimensions. To display the points properly and show the relationships to the user we need to reduce the number of dimensions to 1, 2, or 3. There are many different algorithms that do dimensionality reduction. We use spring-embedding in our system [13]. Our choice was motivated by our earlier work [5]; other techniques (e.g., Linear Programming) are entirely possible, though we have not investigated whether our results apply to them.

4.1 Spring-embedder

The idea of spring-embedder is as follows: consider a set of points in a high-dimensional space and a function that defines the distance between two points. We will call this high-dimensional space t-space, where t is the actual dimension of the space. Consider also a low-dimensional space where this point set is going to be visualized e.g., 1, 2, or 3 dimensions. We call this space v-space (as in visualization). The algorithm creates a point configuration in v-space space that "mimics" the configuration in t-space — it attempts to preserve the relative distances and positions of the points in t-space. Generally, it is impossible to reproduce the same configuration *exactly* in low dimensions.

Each object in t-space is modeled with a steel ring in v-space. The rings repel each other with a constant force: the rings are pushing away from each other and the system is striving to break apart. The "break-away" does not happen because the rings are inter-connected with springs. The force constant of a spring is proportional to the original distance between points in t-space. In this way a "mechanical" model is created. Left to itself the model oscillates and assumes an "optimal" final state. If two points were very close to each other in t-space, the corresponding rings are connected with a very strong spring, and they are very likely to end up close to each other in v-space. On the other hand, the rings that correspond to pairs of points that are far apart have a weak link and the general repulsive force among the rings will push them apart.

Although ring placements may vary widely across oscillations, the final configuration does not usually depend on the original ring locations and these locations are randomly selected. For N objects there are $(N^2 - N)/2$ springs. If all springs are present in the model, all rings are connected very strongly and the final configuration tends to resemble a tight "soccer-ball". Note that Bead [9] was originally designed for a collection of journal abstracts. Such documents are very small and rarely have words in common. Thus just a few of $(N^2 - N)/2$ possible springs were generally present. When Chalmers attempted to apply his system on a collection that contained complete (larger) documents, he observed the "soccer-ball" phenomenon and did not resolve that difficulty.

To prevent the "soccer-ball" from appearing and to reduce computational expense, we impose a limit on the inter-point distances in t-space. If a distance between two points in t-space exceeds a predefined threshold, such points are considered to be infinitely far apart and the corresponding rings are not connected with a spring. Indeed, this allows us to model a situation when two documents are known to be different, when at the same time they have some terms in common.

Unfortunately, selecting the right threshold is a difficult task. Changing the threshold value adds or removes

springs in the model and can have a dramatic effect on visualization.

4.2 Vector generation and embedding

For each document we created a vector V such that v_i was a $tf \cdot idf$ weight of the ith term in the vocabulary:

$$v_i = \frac{tf}{tf + 0.5 + 1.5\frac{doclen}{avgdoclen}} \cdot \frac{\log\left(\frac{N+0.5}{docf}\right)}{\log(N+1)} \quad (1)$$

where tf is the number of times the term occurs in the document, $docf$ is the number of documents the term occurs in, and N is the number of documents in the collection. For each query this resulted in a set of vectors in t-space, where t is the size of the vocabulary of the top retrieved documents (about 3000 words for 50 retrieved documents in most cases that we consider in this study).

The t-dimensional vectors were embedded in 1-, 2-, and 3-dimensional space using the spring-embedder. Distance between vectors was measured by the sine of the angle between the vectors. The embedded structure depended on the number of springs among objects. This number is determined by a threshold: a maximum distance between documents at which the corresponding objects are connected with a spring. For a set of 50 objects there are $(50^2 - 50)/2 = 1225$ different spring configurations, and therefore, 1225 different embeddings.

4.3 Relevance feedback

We consider two methods for incorporating user feedback into the visualization: *Space Warping* and *Restraining Spheres*.

4.3.1 Space warping

Suppose the system received relevance judgments for a subset of the documents being used. The known relevant documents are averaged to create a representative relevant vector, V_R, i.e.,

$$V_R = \frac{1}{|Rel|} \sum_{\forall V \in Rel} V\,, \quad (2)$$

where Rel is the set of known relevant documents. Similarly, the remaining known non-relevant documents are averaged to create a representative non-relevant document, V_N, i.e.,

$$V_N = \frac{1}{|Non|} \sum_{\forall V \in Non} V\,, \quad (3)$$

where Non is the set of known non-relevant documents. With,

$$\Delta V = V_R - 0.25 \cdot V_N\,, \quad (4)$$

each known relevant vector is modified by adding ΔV to it and the known non-relevant vectors are modified by subtracting ΔV. Any resulting negative values are replaced by zero (the vector-space model generally uses only non-negative values).

$$\forall V \in Rel, \quad V = V + \Delta V$$
$$\forall V \in Non, \quad V = V - \Delta V\,.$$

This approach is very similar to relevance feedback methods traditionally applied in information retrieval, but rather than modifying the query, the relevant documents themselves are modified to be brought "closer" to each other.

The vectors are modified in t-dimensional space and the entire set is then embedded in 1-, 2-, and 3-dimensional space as described previously. We hypothesize that unjudged relevant documents will move closer to the known relevant, and unjudged non-relevant will shift towards the known non-relevant.

4.3.2 Restraining spheres

An advantage of a ranked list is the direction it implies: the user always knows where to start looking for relevant information (i.e., at the top of the list) and where to go to keep looking (i.e., down the list). We observed that space warping, however effective it is in bringing together relevant documents, tends to "crowd" the objects, making the whole structure more compact and not easily separable. We developed a small modification to the warping approach that enhances separation among documents, i.e., at the same time creating a general sense of direction on the object structure.

During spring-embedding, judged (i.e., known) relevant documents are placed inside a small sphere and forced to remain there during the oscillation of the embedding, even if tensions in the system would normally move them far from there. Similarly, judged (i.e., known) non-relevant documents are forced into another sphere positioned apart from the first one. The rest of the documents are allowed to assume any location *outside* of these spheres. Intuitively, we grab the spring-embedded structure by the judged documents and "pull it apart".

4.4 Visualization

Figure 2 shows several presentations of 50 documents retrieved in response to a representative query and embedded with the spring-embedder. We assume that the relative position of the documents serves as the main "clue" for the user and that he or she can distinguish even a tiny difference in inter-object distances. Another assumption was made when we designed our system: we assumed that the visualization algorithm will generate "interesting" spatial configurations — e.g., pictures that

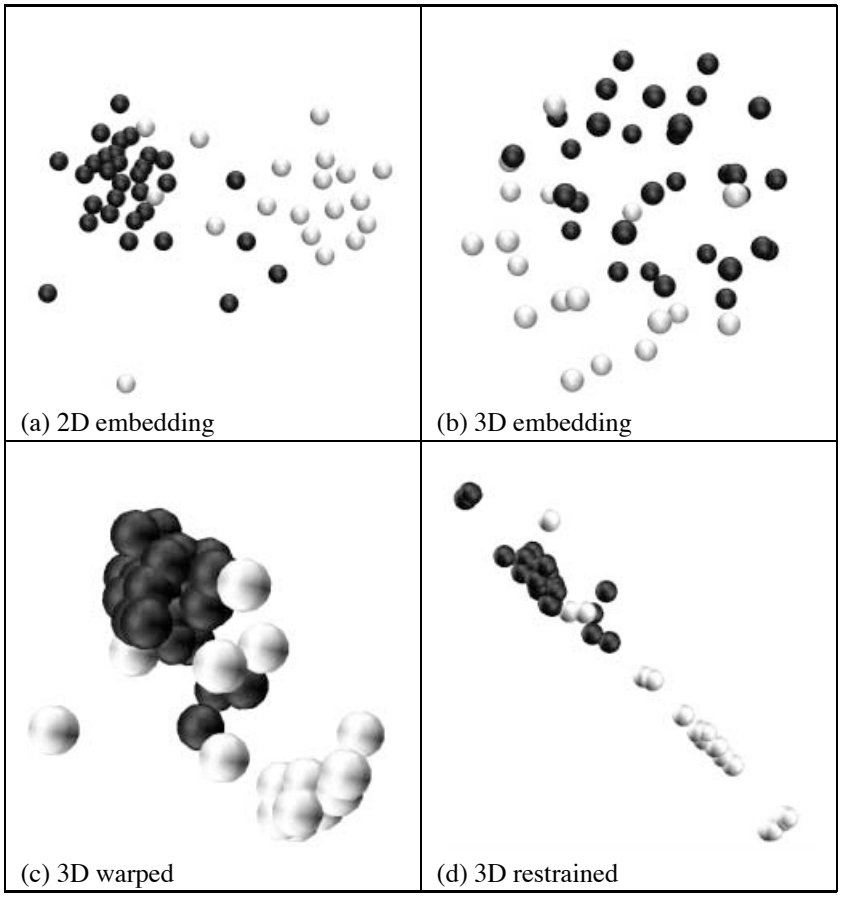


Fig. 2. Visualization of retrieved documents for one of the queries. Both 2- and 3-space embeddings are shown, plus two variations on the 3-space visualization. Relevant documents are shown as black spheres; non-relevant as white. In a real system, the user would initially not know the colors and all spheres would be gray

have some structure, pictures with "clumps" and "gaps". Such structure can serve as an additional navigational "clue" and would provide the user with an overview of the inter-document relationships in the set. The importance of such a structure is that it could not be easily described and explained by analyzing individual pairwise similarities between documents. As Fig. 2 shows, the layout generated by the spring-embedder can exhibit a significant amount of structure.

4.5 Estimating spatial structure

We require a way of measuring this spatial structure. Specifically, we desire answers for the following two questions:

- Are the spatial locations random, or are they clustered? A spatial point pattern that exhibits some structure provides potentially more information than a set of randomly scattered objects. We require a statistical test to determine if the spatial pattern shows any structure.
- If the spatial pattern shows any structure, what is the extent of the structure? We require a way to quantify the amount of "clumpiness" in the point pattern that does not require asking a person's opinion. Such a statistic is crucial for this study: different observers would disagree as to the amount of structure in the point pattern. Further, the process of obtaining such judgments would be enormously expensive.

The theory of point fields (i.e., point processes) [21] introduces a simple and efficient technique for measuring spatial dependencies between different regions of a point pattern.[1] Here we give an overview of key points. A full description is available in a book by Cressie [10]. Consider a set of points in a d-dimensional space and a distance function on this space. Suppose λ is the mean number of points in a unit volume of space, or the *intensity* of the point field. Let $N(h)$ be the number of extra points within a distance h of a *randomly* chosen point. Then Barlett [6] defines a function K as:

$$K(h) = \lambda^{-1} E(N(h)), \quad h \geq 0 \ , \tag{5}$$

where $E(\cdot)$ is the expectation operator on the point field. In other words, the K function is the average number of

[1] Random point fields are mathematical models for irregular "random" point patterns. We will use this terminology to describe the location pattern of objects corresponding to the retrieved documents.

points in the point field within distance h of any point in this field, normalized by the mean number of points in a unit volume of space. Practically, it measures a local concentration of points, or what part of the point field on average is within distance h of any point in the point field. Ripley [19, 20] shows that the K function has properties that make it an effective summary of spatial dependence in a point field over wide range of scales.

The main application of the K function is to test if a point field exhibits any structure [21, p. 224]. Indeed, $K(h)$ is proportional to the number of points at most h away from an arbitrary point. If this number is unusually high, we find many points in close proximity to any given point — i.e., we have clumps or clusters of points in the point field. If the number is low, we have few points in close proximity — i.e., we have gaps in the field. Because of the expectation operator in (5) these conclusions apply "on average" to the whole point field. Therefore, the K function should not be much affected by outliers if enough points are considered. The function does not explicitly depend on point locations, making it independent of the shape of the point field.

The K function is just a metric for comparing one point field to another. To decide if the the point field has clusters, we compare this field to some configuration that is known not to have clusters. Generally, a completely random arrangement of points with neither clumps nor gaps is selected. This configuration of points is called a "random point field".

It is customary [10] to model this random point field with a Poisson point field, a configuration where a point is equally likely to occupy any location in the space of the field. The only condition is that no two points can occupy the same location in space. The K function for a d-dimensional Poisson field is defined as:

$$K_{Poisson}(h) = \frac{\pi^{d/2} h^d}{\Gamma\left(1+\frac{d}{2}\right)} \tag{6}$$

To compute the values of the K function the expectation operator in (5) is replaced with an empirical average over the N given points:

$$\hat{K}(h) = \hat{\lambda}^{-1} \sum_{i=1}^{N} \sum_{j=1}^{N} I(\|s_i - s_j\| \le h)/N,\ i \ne j \tag{7}$$

Here $\hat{\lambda} = N/v$ is the estimator of the intensity, v is the volume that contains the point field, s_i is the location of the ith point, and $I(\cdot)$ is the indicator function:

$$I(x) = \begin{cases} 1, & \text{if } x \text{ is } true \\ 0, & \text{if } x \text{ is } false \end{cases} \tag{8}$$

It is also customary to use the following statistic $\hat{L}(h)$ instead of $\hat{K}(h)$:

$$\hat{L}(h) = \sqrt[d]{\hat{K}(h)\frac{\Gamma\left(1+\frac{d}{2}\right)}{\pi^{d/2}}} \tag{9}$$

When $\hat{L}(h)$ is greater than $L_{Poisson}(h) \equiv h$, there are clumps in the point field; $\hat{L}(h) < h$ implies gaps in the configuration.

The test variable τ is used to test the amount of structure in the point fields:

$$\tau = \max_{h \le h_0} |\hat{L}(h) - h|\,, \tag{10}$$

where h_0 is the upper bound on the interpoint distance. The outcome of the test is based on comparing τ with its table values [21, p. 225]. We will use the term *spatial structure* when referring to τ in the rest of the discussion.

5 Search strategy

The previous section presented the system design and we now continue with the Strategy-based Evaluation framework by describing the search strategy, the performance measure, and the experimental setup.

The evaluation analysis requires some assumptions about the search strategy — i.e., how we expect a user to look for the relevant material. It is impossible to define the degree of separation between the relevant documents and the non-relevant documents without assuming some search strategy first. In most studies, the strategy is rather intuitive and goes unspecified. For example, consider a linear separation test — two sets of points in 2-dimensions are considered well-separated if it is possible to draw a straight line between them. Here the assumed strategy is "draw the line; consider all the points that are on one side of the line".

The assumptions about a particular search strategy may also have a strong influence on the performance measure. We proceed by defining two different search strategies. Note that there exist many different strategies; the procedures proposed in this section are two reasonable examples. [2] Other alternatives are entirely possible.

- *Single document strategy.* Recall that we know the values of relevance judgments for some of the documents (see Sect. 3). The strategy starts at an arbitrary known relevant document and proceeds by analyzing the rest of the unknown documents in proximity order to the starting point. If there are several possible starting points (if we know more than one relevant document) the final performance has to be averaged across all starting points.
- *Relevant cluster strategy.* We assume that we know relevance judgments for some of the documents. The strategy begins by defining a cluster that contains all

[2] We confess that "reasonable" is defined by our intuition and experience. It would be preferable to employ user models or field studies to devise truly reasonable strategies. We view such an approach as important future work to explore the range of strategies possible. This work focuses on evaluating two strategies and showing the extent to which they are useful for this task.

the known relevant documents. It then proceeds by analyzing the rest of the unknown documents in the proximity order to the cluster. If the document is relevant, it is added to the cluster. Note that the distance between a document and the cluster can be defined in multiple ways by analogy with the traditional clustering algorithms [23]. Here d is an arbitrary unknown document, C is the cluster of relevant documents, and $\rho(\cdot,\cdot)$ is Euclidean distance in the visualization space.

single-link. The documents are ranked by the distance to the *closest* document in the cluster.

$$\rho(d, C) = \min_{\forall d_c \in C} \rho(d, d_c) \tag{11}$$

complete-link. The documents are ranked by the distance to the *furthest* document in the cluster.

$$\rho(d, C) = \max_{\forall d_c \in C} \rho(d, d_c) \tag{12}$$

average-link. The documents are ranked by the average of distances to all documents in the cluster.

$$\rho(d, C) = \frac{1}{|C|} \sum_{\forall d_c \in C} \rho(d, d_c) \tag{13}$$

centroid. A center of mass is computed for the cluster. The documents are ranked by the distance to the centroid.

$$\rho(d, C) = \rho\left(d, \frac{1}{|C|} \sum_{\forall d_c \in C} d_c\right) \tag{14}$$

Note that average link and centroid would be identical if we used the Manhattan metric instead of Euclidean to define the distance in the visualization space.

Thus, our simulated search strategies create a new ranking order for the unknown documents. Information Retrieval has a long legacy of dealing with document rankings. In the next section we adapt a well-known definition of precision to our "spatial" ranking.

6 Performance measure

A search strategy "walks" over the unknown documents creating a new ranking order. As soon as the search strategy finds a relevant document we record precision at that point — the proportion of relevant documents among those already considered by the search strategy. When the search strategy completes the ranking, we average the recorded precision numbers. This is the average non-interpolated precision [14] and it serves as the performance measure for the ranking. If the search strategy assumes multiple starting points (as Single Document Strategy does) we average the average precision across all possible starting points.

7 Experiments

In this section we describe the testbed that we used for running our experiments. We then proceed by describing seven experimental questions that we consider, presenting their results, and analyzing the implications of our results where appropriate. The questions we consider are:

1. Does the Cluster Hypothesis hold?
2. Can we choose a threshold for spring embedding?
3. Are more dimensions better for the embedding approach?
4. To what extent does relevance feedback help the visualization?
5. What is the benefit of different search strategies?
6. How does the embedding compare to the classical ranked list?
7. Are there indications that we might be able to do better?

7.1 Experimental testbed

For our experiments we used TREC [14] ad-hoc queries with their corresponding collections and relevance judgments. Specifically, TREC topics 251–300 were converted into queries and run against the documents in TREC volumes 2 and 4 (2.1GB) that include articles from Wall Street Journal, Financial Times, and Federal Register. For each TREC topic we considered four types of queries: (1) the title of the topic; (2) the description field of the topic; (3) a query constructed by extensive analysis and expansion [3]; and (4) a query constructed from the title by expanding it using Local Context Analysis (LCA) [24].

The top 50 documents for each query were selected. Because each query behaved differently, there were four different ranked lists for each topic. We are interested in situations when there was not *enough* relevant material in the top ten documents, so ignored runs that contained too many relevant documents — they are successful already and the visualization is unnecessary. We also discarded complete failures, or runs that had just a few relevant documents. Finally, because we are interested in how the visualization changes when the user's feedback about both relevant and non-relevant documents is provided, a small amount of either relevant or non-relevant data renders such analysis uninteresting. Therefore, the lists with fewer than 6 relevant documents in the top 50 or with fewer than 3 or greater than 9 relevant documents in the top 10 were discarded. This resulted in 20 queries for the title-only version, 24 for the description queries, 26 for the full versions, and 17 for the expanded title version.

We also collected the same data using a different set of queries on a different collection. We used TREC topics 301–350 to create the queries and ran the queries against TREC volumes 4 and 5 (2.2GB) that include articles from Congressional Records, Financial Times, and Los Angeles Times. Again four different types of queries were constructed: (1) the title of the topic; (2) the title and the

description field of the topic; (3) the full version constructed by expansion [4]; and (4) the expanded version of title query. The same restrictions were imposed on the retrieved set. This resulted in 25, 27, 25, and 22 queries of each type, respectively.

7.2 Does Cluster Hypothesis hold?

The Cluster Hypothesis of Information Retrieval states that "closely associated documents tend to be relevant to the same requests" [22, p.45]. It has been shown that the hypothesis holds at least for the retrieved documents [11]. Each query has, on average, about 15 relevant documents in the top 50. If the documents were randomly scattered in space, then an automatic search strategy would produce a ranking where the relevant documents occupy arbitrary positions with equal probability. The expected average precision of such a ranking would be about 33.1% (for example, one may use bootstrapping to compute this number). Our search strategies build a ranking based on the spatial proximity to the known relevant documents. We have observed average precision values around 50% (Table 1), indicating substantial clustering among relevant documents. That is, we have found continued support for the Cluster Hypothesis' truth in retrieved documents: relevant documents tend to appear in close proximity to each other, often forming tight "clumps" that stand apart from the rest of the material.

7.3 Can thresholds be selected?

Recall from the system description in Sect. 4.1 that the embedding structure is affected by the threshold choice. For N objects there are $(N^2-N)/2$ springs, so there are $(N^2-N)/2$ different threshold values: each threshold allows an additional spring into the model. Nothing in the spring-embedding approach suggests a way of choosing one threshold value over another (i.e., one embedding over another), so in the absence of such information we must randomly select one of the $(N^2-N)/2$ structures to show to the user. We analyze system performance by averaging precision over all possible values of threshold.

We also determine the probability of randomly selecting a "good" threshold value. For all queries in question and for all possible spatial embeddings (i.e., for all threshold values) we count the number of times each average precision value occurs and normalize them over the total number of embeddings. This gives us a probability distribution for precision values. If we take this distribution, fix some precision value ($prec_0$), and add all the values in the distribution for each point that exceeds $prec_0$, we compute the probability for an arbitrary selected spatial configuration among all possible embeddings to exceed $prec_0$, or $P(prec > prec_0)$ (see Fig. 3).

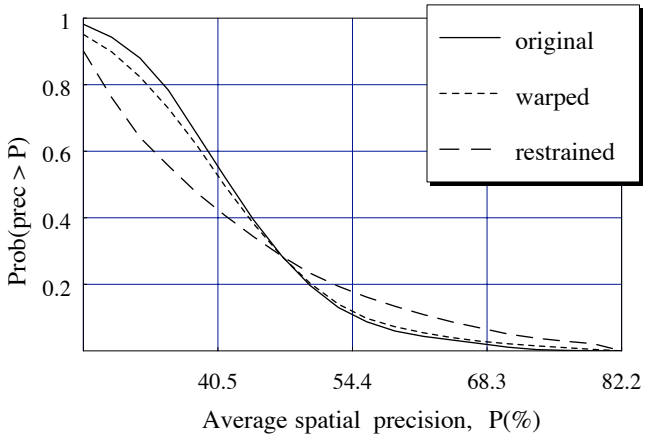
We begin by assuming that the user has identified two documents: one relevant and one non-relevant. (We believe this is a reasonable strategy and almost always could be done by looking at the titles in the ranked list.) For simplicity, let us assume the user identified the highest ranked relevant and the highest ranked non-relevant document. We evaluate how quickly the user would be able to find the rest of the relevant documents starting from the known relevant one using the spatial information. For this study we assume the Single Document search strategy of Sect. 5.

Our hypothesis is that embeddings with high spatial structure are more likely to have high precision scores. Here we rely on the Cluster Hypothesis (supported by the previous section): if the spatial structure has clusters, it is likely that these clusters are "pure" clusters of relevant documents. Clusters of non-relevant documents are also possible, but "mixed" clusters are less likely.

As indicated earlier, in the absence of any other information, a threshold value would have to be chosen randomly. However, limiting our choice to the embeddings

Table 1. Visualization quality evaluation of different query sets in different dimensions. Percent of average precision is shown. The first column is for the system's ranked list. The second column is for the original structure in t-dimensional space. The third column shows the result of spring-embedding. The last column is for embedding with threshold selection done by the τ measure. The relevance judgements for two documents are known to the system – the top ranked relevant and non-relevant documents

Queries		Rank List	t-D space	Embedding					
				w/o threshold selection			w/ threshold selection		
				1-D	2-D	3-D	1-D	2-D	3-D
TREC5	Title	63.0	43.8	38.0	41.8	41.8	42.5	58.2	59.1
	Desc.	54.7	42.1	39.2	42.1	42.1	41.0	51.3	52.2
	Full	58.4	53.1	45.3	46.3	46.7	47.0	49.9	50.9
	Exp. Title	66.6	60.0	46.6	48.5	48.5	49.0	57.3	59.7
TREC6	Title	58.8	52.1	44.7	47.5	47.8	46.9	57.6	59.9
	Desc.	57.7	48.2	39.8	44.0	44.6	41.7	54.8	55.3
	Full	68.6	53.9	42.5	48.8	49.5	43.4	57.9	59.4
	Exp. Title	64.3	52.0	42.3	45.5	45.9	44.0	55.9	59.0
average		61.5	50.7	42.3	45.6	45.9	44.4	55.4	56.9



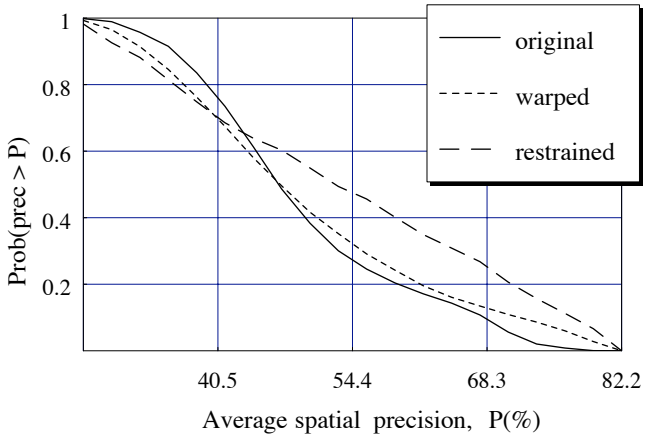


Fig. 3a,b. Probability of selecting an embedding at random with a given precision value or higher, for the full queries on the TREC-5 collection in two dimensions. These graphs illustrate the effect of different user feedback techniques: **a** No restrictions are imposed on the set of possible embeddings; **b** The set of embeddings is limited to high τ values by the threshold selection procedure

with spatial structure τ (see Sect. 4.5) in the top 20% of spatial structure values proved very effective. The average precision across all "eligible" threshold values was significantly increased by 17.2% relative to making no effort to limit the set considered ($p < 10^{-5}$ by the t-test). The numbers are shown in the last two columns of Table 1. The solid lines on Figs. 3a and 3b show how the threshold selection procedure increases the probability of randomly choosing a high quality spatial structure without any information supplied by the user. The effect is also consistent across relevance feedback methods. Note that there is almost no change in maximum and minimum values of precision. That means the method does not limit the possible choices of quality on the spatial structure: it just makes it more probable we will select a "good" one.

Table 2. Average precision computed starting from the first 5 relevant documents. The retrieved documents from TREC-5/full queries are embedded in 2 dimensions. The first column of numbers is for the case when no feedback has been yet given

Type of feedback	Number of pairs judged				
	0	1	2	3	5
warping	49.3	50.5	51.4	51.4	51.5
restraining	49.3	49.9	51.4	52.3	53.4

7.4 Are more dimensions better?

Is a high-dimensional visualization more useful than a low-dimensional one for the purpose of isolating the relevant documents? That is, is a 2D picture more helpful than its 1D counterpart; is 3D better than 2D? The documents exist in an extremely high-dimensional space (thousands of dimensions). When these configurations are forced down into 2 or 3 dimensions for the purpose of visualization, some documents are shown "nearby" when they are actually unrelated. We hypothesize that visualizing in extra dimensions will show the relationships among documents more accurately and the relevant documents will be better isolated from non-relevant ones.

Our results support this hypothesis only partially. Indeed, a step from 1 dimension to 2 leads to a statistically significant jump of 23.1% in precision ($p < 10^{-5}$). However, the difference between 2- and 3-dimensional embeddings is only 1.1%, a result that, although consistent, is not significant. (It is significant by the sign test, but not by the t-test. A cut-off value of $p = 0.05$ is used in both tests.)

Figure 4 shows how an increase in dimensionality of the embedding leads to a general growth in precision. It is difficult to see, but the maximum precision value for 1D is higher than for 2D or 3D. This seems to indicate that a better separation between relevant and non-relevant documents *could* be achieved in 1-dimension than in 2- or 3-dimensions. However, finding that high precision structure randomly is extremely difficult.

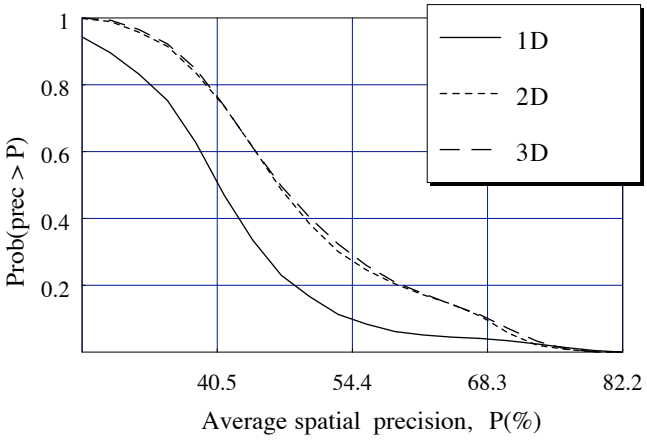


Fig. 4. Probability of selecting an embedding with a given precision value or higher, for the full queries on TREC-5 collection. The effect of different dimensions on the original embedding is illustrated. The set of embeddings is limited to high τ values by the threshold selection procedure. The values on the x-axis are averaged over the query set

7.5 Does user feedback help?

Feedback techniques enhance the separation between relevant and non-relevant documents and the visualization should be able to capitalize on that improvement. If a searcher expends the effort to mark some documents as relevant and others as non-relevant, can the separation between the two sets be enhanced — among both the marked documents and (more importantly) the unmarked part of the retrieved set?

Recall that the TREC queries used in this study come with relevance information for the documents in our retrieved sets. A small subset of the 50 documents being used was presumed known and marked as relevant (or not) using the TREC relevance judgments. Thus we simulate the user making relevance judgments. We experimented with subsets of 2, 4, 6, and 10 documents.

Figure 2c illustrates how the warping process can improve the separation between relevant and non-relevant documents. It shows the same documents as those in Fig. 2b, but with space warping added. The relevant and non-relevant documents are still grouped apart from each other, but the location of the groups is much more easily recognizable — particularly since 10 of the documents in the presentation have already been judged. Figure 2d shows the effect of restraining spheres on the same query. In this particular case, the simple warping would probably be useful, but the location of unjudged relevant documents is even more obvious since the documents have been "stretched".

We also studied how the quality of the visualization changes as the system is supplied with more and more relevance information. Given the first relevant/non-relevant pair of documents, we use it to warp the embedding space and apply the restraining spheres. Then we add information about the next relevant/non-relevant pair and so on until up to 5 pairs have been added. Table 2 and Fig. 5 illustrate how the average precision increases as more data become available to the system. We show the average precision of a ranking generated by the search strategy starting from an arbitrary document among five top ranked relevant documents. The warping does not have any effect after the first two steps. The restraining spheres keep pulling the documents apart; however, their influence is also diminishing.

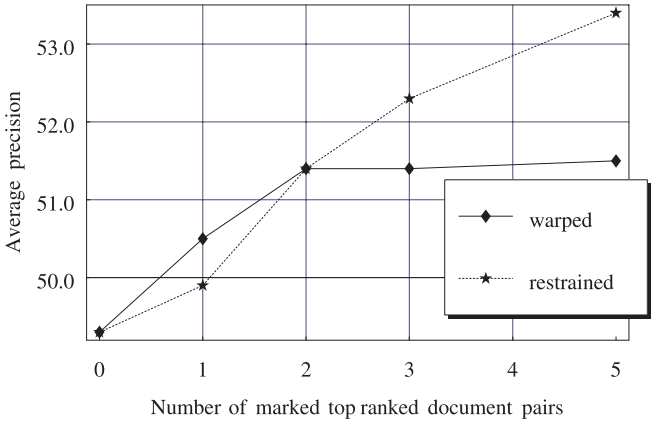


Fig. 5. Average precision computed starting from the first 5 relevant documents. The retrieved documents from TREC-5/full queries are embedded in 2 dimensions. The first column of numbers is for the case when no feedback has been yet given

The previous exploration of feedback used up to 10 documents, but always added them in relevant/non-relevant pairs. Our second strategy was to evaluate the effect of a user's feedback on the visualization using the top ranked 10 documents, regardless of whether there was a matching number of relevant and non-relevant documents. Recall from the corpus creation (Sect. 7.1), that we know there are from three to nine relevant documents in the top 10. We consider how quickly one can identify the rest of the relevant material starting from the known relevant documents (i.e., those in the top ranked 10). We compare the effects that warping and restraining have on this task.

From Table 3 we conclude that warping did not do as well as we had expected. It increased average precision by 1.1% consistently, but not significantly ($p < 0.02$ by the sign test and $p < 0.37$ by the t-test). It actually *hurt* precision in 3D. The effect of warping together with restraining was more profound and nearly always beneficial. The procedure significantly increased precision by 7.4% ($p < 0.001$ by the sign test and $p < 0.037$ by the t-test).

Figures 3a and 3b show that feedback techniques increase the probability of selecting an embedded structure with high precision value. The growth is observed both with and without threshold selection, but with threshold selection the difference between the restrained and original cases is more prominent.

We also observed a strong effect that poorly formulated and ambiguous queries have on feedback's benefit. The restraining spheres largely decreased the precision of the embeddings generated for documents retrieved by the title queries on TREC-5 collection. Expanding the "bad" queries (see the "TREC-5/Exp. Title" row in Table 3) to eliminate the possible ambiguity seems to alleviate the problem. The TREC-6 title queries were created to be of higher quality and ranked better.

7.6 What is effect of search strategy?

Because of concerns that the Single Document Strategy (used in all previous sections) was unintuitive and a poor match to a strategy that users were likely to use, we also considered the effect that other strategies have on effectiveness. We implemented four different versions of the Relevant Cluster strategy (see Sect. 5): single-link, complete-link, average-link, and centroid. We also considered two different starting conditions for each case: (1) the highest ranked relevant and non-relevant documents

Table 3. Relevance feedback effect on different queries in different dimensions. We show percent of average precision of a ranking that is generated starting from a random relevant document in the top 10 ranked documents. The threshold selection procedure (for high τ values) was applied

Queries		Rank List	Embedded in	Original	Warping		Restraining	
			1-D	35.7	36.2	(+1.3%)	31.5	(−11.7%)
TREC5	Title	46.8	2-D	47.3	48.9	(+3.3%)	40.7	(−14.0%)
			3-D	48.4	50.3	(+3.9%)	40.2	(−17.0%)
			1-D	38.6	39.8	(+3.3%)	37.1	(−3.1%)
	Desc.	40.8	2-D	48.5	48.8	(+0.6%)	49.6	(+2.2%)
			3-D	49.6	48.3	(−2.5%)	47.3	(−4.5%)
			1-D	41.9	42.4	(+1.2%)	47.3	(+12.8%)
	Full	43.1	2-D	45.9	47.1	(+2.8%)	52.0	(+13.4%)
			3-D	46.1	47.0	(+2.0%)	47.5	(+3.0%)
			1-D	42.4	42.7	(+0.6%)	51.7	(+22.1%)
	Exp. Title	42.5	2-D	46.2	46.8	(+1.4%)	54.4	(+17.8%)
			3-D	46.6	46.2	(−0.8%)	52.4	(+12.5%)
			1-D	42.9	45.0	(+4.8%)	45.7	(+6.6%)
TREC6	Title	50.6	2-D	53.6	53.9	(+0.7%)	57.4	(+7.2%)
			3-D	55.9	55.4	(−0.9%)	58.9	(+5.5%)
			1-D	37.6	38.8	(+3.2%)	43.8	(+16.6%)
	Desc.+Title	45.7	2-D	49.8	51.0	(+2.4%)	56.2	(+13.0%)
			3-D	51.3	50.9	(−0.8%)	56.4	(+9.9%)
			1-D	36.3	37.4	(+2.9%)	44.5	(+22.5%)
	Full	53.1	2-D	46.6	47.0	(+0.7%)	55.3	(+18.5%)
			3-D	48.9	47.5	(−2.8%)	54.0	(+10.5%)
			1-D	39.1	38.7	(−0.9%)	42.5	(+8.9%)
	Exp. Title	53.7	2-D	48.4	49.7	(+2.8%)	56.0	(+15.7%)
			3-D	50.6	50.0	(−1.1%)	56.5	(+11.7%)
			1-D	39.3	40.1	(+2.1%)	43.0	(+7.2%)
	average	47.0	2-D	48.3	49.2	(+1.8%)	52.7	(+7.2%)
			3-D	49.7	49.5	(−0.1%)	51.7	(+4.4%)

are known and (2) complete judgments for the top-ranked 10 documents are known.

Surprisingly, we did not find any significant difference in effectiveness between any of those search strategies, in either of the two starting conditions. However, we did observe a significant ($p < 10^{-5}$) improvement using the Relevant Cluster strategies over the Single Document Strategy. This result is not entirely surprising since the new strategies make more use of the relevant information than did the earlier strategy which just blindly moved out from a randomly chosen relevant document rather than adjusting to new relevant documents as they are found.

This result suggests that the significant results of earlier sections might be even better if improved strategies (e.g., Relevant Cluster) are employed. We leave this verification for future work.

7.7 Embedding compared to ranked list?

Table 4 shows average precision values for different query sets in different dimensions. The ranked list is treated as an embedding in 1-dimension where each document is positioned on a line according to its rank value.

It is evident that when only the highest ranked pair of documents (the highest ranked relevant and the highest ranked non-relevant) is known and the threshold selection procedure is applied, both 2- and 3-dimensional visualizations exhibit the same quality on average as the ranked list.

However, if all the judgments for the top ten documents are known and the threshold selection procedure is applied, the quality of the visualization exceeds the quality of the ranked list. These numbers are in the last columns of Table 4. We have observed that for the long queries ("TREC-6/Full") the quality of the ranked list is superior. However, users almost never use this type of query [7].

We have observed consistently across multiple search strategies that the quality of the ranked list degrades much faster as more documents are analyzed. For example, if only two documents are known then performance of the ranked list is 61.5% and for 3-dimensional visualization we get 61.4%. When the judgments for the top ten documents become available the quality of the ranked list is 47.0% and the quality of the visualization is 53.8%. This seems to indicate that if more documents are known, it is much faster to find the rest of the rele-

Table 4. Ranked list vs. spatial embeddings. Visualization quality evaluation of different query sets in different dimensions. The Relevant Cluster single-link search strategy is used and the threshold selection procedure (high τ values) is applied. Percent of average precision is shown. The first column is when the highest relevant/non-relevant pair of document is known. The second column is when the top ranked ten documents are known

Queries		Two documents are known				Two documents are known			
		Rank List	Embedding			Rank List	Embedding		
			1-D	2-D	3-D		1-D	2-D	3-D
TREC5	Title	63.0	43.8	61.9	63.5	46.8	37.4	53.1	54.3
	Desc.	54.7	42.4	59.9	56.9	40.8	40.4	51.6	50.8
	Full	58.4	48.4	55.1	55.0	43.1	41.5	52.7	52.5
	Exp. Title	66.6	50.3	62.8	65.4	42.5	43.6	53.3	54.2
TREC6	Title	58.8	48.0	62.4	64.3	50.6	44.7	58.7	59.8
	Desc.	57.7	42.9	59.1	60.8	45.7	39.1	52.5	52.5
	Full	68.6	44.6	63.0	62.5	53.1	37.7	48.0	49.6
	Exp. Title	64.3	45.4	60.1	63.2	53.7	41.5	53.6	56.3
average		61.5	45.7	60.5	61.4	47.0	40.8	52.9	53.8

vant material with the visualization than with the ranked list. Thus a strategy for a user might be suggested: "Start with the ranked list, find a few relevant documents, then switch to the visualization and look for the rest of the interesting data in close proximity to the ones you have found".

7.8 Can we do better?

We have also done some "best case" analysis, when instead of averaging precision over the set of possible embeddings we considered the structure with highest precision. In this case the values are about 15–20 points higher than in the average case and the system beats the ranked list "hands down".

Note that this type of analysis is of questionable value since it could be the result of random effects and smacks of testing on the training data. However, we feel it is interesting because it suggests that the visualization *might* be able to do substantially better *if* we can find the right threshold values. There are good embeddings out there; it is just difficult to find them.

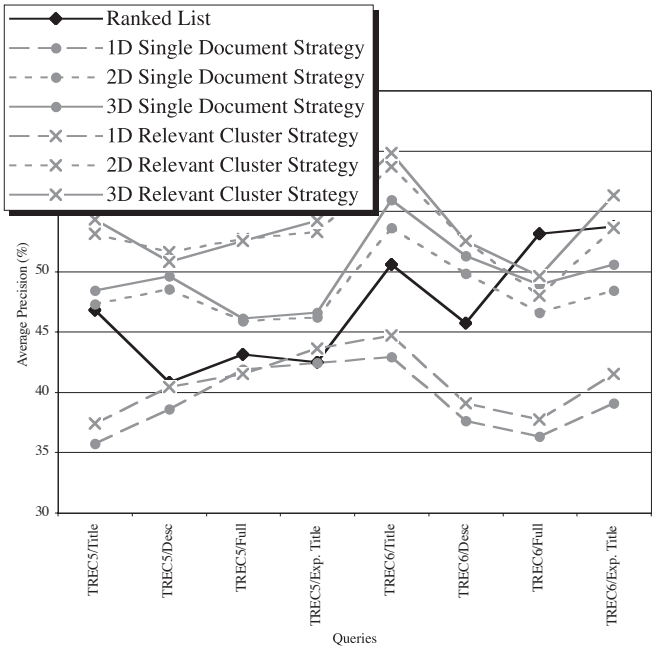


Fig. 6. Comparison of two different strategies and the ranked list over multiple query types. Both the Relevant Cluster (single-link) and the Single Document search strategies are shown. The threshold selection procedure was applied (for high τ values) and the relevance judgments for the top ten documents in the ranked list are known

8 Discussion and conclusions

We presented a Strategy-based Evaluation Method for analyzing interactive information organization techniques. This approach allowed us to discover important properties about the system in our study. We believe our approach can be used as a laboratory method for preliminary investigations of interactive information organization systems. Such a framework has several potential advantages:

- It helps to explain the performance. By breaking the interaction model into components (i.e., the user and the system), we can estimate the effect each component has on the overall performance.
- It is relatively inexpensive. Even a small user study is generally costly and time-consuming. Our analysis is done offline.
- It can produce statistically sound conclusions. By doing the analysis offline we can potentially process much large data sets than could be done in real user experiments. The results would have much stronger statistical power.
- It should provide more experimental control for the researcher. People are very different in their abilities and skills. When conducting a user study it is impossible

to control all the variables included. Our evaluation gives the researcher complete control over the experimental setup.
- It could help in designing user studies. By defining all the components of the interaction model we will have to state clearly all the hypotheses and assumptions we make about the system and the user strategy. A real user study could then concentrate on testing these assumptions. For example, in a user study the system might perform worse than expected if the search strategy of a naive user is different from that assumed in our evaluation. Then the system's interface would have to be redesigned to include a tutoring mechanism that recommends that better strategy to the users.
- It helps to determine the optimal strategy for the user. By considering several different strategies we could select the one that is the most suitable for each particular system and task. Not only we can give the user a system, but we can also describe an effective way of using it.

We have applied the evaluation framework to analyze an information organization system that visualizes the documents by placing them into 1-, 2-, and 3-dimensional space and positioning them according to the inter-document similarity.

- It has been known for at least two decades that the Cluster Hypothesis is true within the top-ranked retrieved documents. Although the system used in this study does not explicitly generate clusters, we show that the objects that represent relevant documents tend to group together.
- The Cluster Hypothesis also helped us to select good embedding structures. As a result we show that embeddings with a high spatial structure value (τ) tend to have higher precision.
- We have hypothesized that an extra dimension is always helpful for visualization. Our results support this hypothesis only partially. There is a clear advantage in using higher dimensions over 1D. However, there is almost no improvement in adding an extra dimension to a 2D visualization.
- In the context of our visualization, we confirmed the hypothesis that relevance feedback methods can improve separation between relevant and non-relevant documents. Figure 2 shows an example of how these methods can have a significant influence on the embedding structure.
- The suggested visualization method in its current state (no robust threshold selection procedure) works — on average — as well as or better than a ranked list for finding relevant documents. In another study [4] most of the users loved this visualization: they found it intuitive and fun to use. That study also found no difference in precision between ranked list and 3D visualization. We provide additional support that suggests visualization is no worse than a ranked list.
- The "best case" analysis *suggests* that the visualization has a very high potential. It seems worthwhile to attempt a deeper investigation in how to make the threshold selection process more robust.

8.1 Future work

In this study we considered only two classes of documents: relevant and non-relevant. This was caused by the lack of data of any other kind. We are looking into extending our approach into situations when the user places the relevant documents into multiple classes. That task is modeled after the interactive TREC task of "aspect retrieval".

We assumed that the user has already found some of the relevant documents (e.g., by means of the ranked list). We plan to look into the problem of helping the user to establish these first relevant documents. One way is to check the "clumpiest" areas of the visualization.

The document distance metric that we employed in this study (sine of the angle between the documents vectors, Sect. 4.2) causes the spring-embedding to generate very "tight" visualization structures that are highly sensitive to the threshold parameter. We are looking at adopting an alternative distance function that will not have this problem.

We plan on a user study to analyze if people take advantage of the spatial clues (such as proximity) that are used by the search strategy in our simulations. We are also interested in what other kind of information besides simple proximity people receive from the visualization.

Acknowledgements. We would like to thank Russell Swan for the preliminary work on the 3D spring embedder evaluated in this study.This study is based on work supported in part by the National Science Foundation under grant number IRI-9619117. This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. This material is also based on work supported in part by Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

1. Allan, J.: Automatic Hypertext Construction. PhD thesis, Cornell University, January 1995. Also technical report TR95-1484
2. Allan, J.: Building hypertext using information retrieval. Information Processing and Management 33(2):145–159, 1997
3. Allan, J., Callan, J., Croft, B., Ballesteros, L., Broglio, J., Xu, J., Shu, H.: Inquery at TREC-5. In: Fifth Text REtrieval Conference (TREC-5), pp. 119–132, 1997
4. Allan, J., Callan, J., Croft, W.B., Ballesteros, L., Byrd, D., Swan, R., Xu, J.: Inquery does battle with TREC-6. In: Sixth Text REtrieval Conference (TREC-6), 1998. Forthcoming
5. Allan, J., Leouski, A., Swan, R.: Interactive cluster visualization for information retrieval. Technical Report IR-116, CIIR, Department of Computer Science, University of Massachusetts, Amherst, 1996

6. Barlett, S.M.: The spectral analysis of two-dimensional point processes. Biometrika 51:299–311, 1964
7. Croft, W.B., Cook, R., Wilder, D.: Providing government information on the internet: Experiences with thomas. In: Proc. Digital Libraries Conference DL 95, pp. 19–24, 1995
8. Card, S., Moran, T.: User technology: from pointing to pondering. In: Baecker, Grudin, an Greenberg, B. (ed.): Readings in Human-Computer Interaction: towards the year 2000. Morgan Kaufmann, 1995
9. Chalmers M., Chitson, P.: Bead: Explorations in information visualization. In: Proc. ACM SIGIR, pp. 330–337, June 1992
10. Cressie, N.A.C.: Statistics for Spatial Data. John Willey & Sons, 1993
11. Croft, W.B.: Organising and Searching Large Files of Documents. PhD thesis, University of Cambridge, October 1978
12. Dubin, D.: Document analysis for visualization. In: Proc. ACM SIGIR, pp. 199–204, July 1995
13. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software–Practice and Experience 21(11):1129–1164, 1991
14. Harman, D., Voorhees, E. (ed.): The Fifth Text REtrieval Conference (TREC-5). NIST, 1997
15. Harman, D., Voorhees, E. (ed.): The Sixth Text REtrieval Conference (TREC-6). NIST, 1998
16. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In: Proc. ACM SIGIR, pp. 76–84, Aug. 1996
17. Hemmje, M., Kunkel, C., Willet, A.: LyberWorld – a visualization user interface supporting fulltext retrieval. In: Proc. ACM SIGIR, pp. 254–259, July 1994
18. Leouski, A.V., Croft, W.B.: An evaluation of techniques for clustering search results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996
19. Ripley, B.D.: The second-order analysis of stationary point processes. Journal of Applied Probability 13:255–266, 1976
20. Ripley, B.D.: Modeling spatial patterns. Journal of the Royal Statistical Society 39:172–192, 1977
21. Stoyan, D., Stoyan, H,: Fractals, Random Shapes and Point Fields. John Willey & Sons, 1994
22. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London, 1979. Second edition
23. Willett, P.: Recent trends in hierarchic document clustering: a critical review. Information Processing and Management 24(5):577–597, 1988
24. Xu, J., Croft, W.B.: Querying expansion using local and global document analysis. In: Proc. the 19th International Conference on Research and Development in Information Retrieval, pp. 4–11, 1996