Improving Interactive Retrieval by Combining Ranked Lists and Clustering

Anton Leuski and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
{leuski,allan}@cs.umass.edu

Abstract

We study the problem of organizing the documents returned by an information retrieval system in response to a natural language query. We consider two well-known approaches – the ranked list and clustering of the results – and we show how they can be integrated. This new procedure is designed to accept user feedback and direct the user toward the relevant material as effectively as the traditional relevance feedback approach. We show how our technique can be explained to the user by visualizing the process in two or three dimensions, providing him or her with complete control of the procedure. We show that increasing the dimensionality of the visualization generally improves its quality, albeit only a small amount. Additionally we present the result of a small user study designed to investigate how effective our visualization is in supporting the user navigating the retrieved results.

1 Introduction

Locating interesting information is one of the most important tasks in Information Retrieval (IR). An IR system accepts a query from a user and responds with a set of documents. Generally, the system returns both relevant and non-relevant material, and a document organization approach is applied to assist the user in finding the relevant information in the retrieved set.

The two most widely used document organization approaches are (1) the ranked list and (2) clustering of the retrieved documents. Both these techniques have their strengths and weaknesses. We begin the paper by putting our work in the context of the previous research done in the field of Information Retrieval. We then show how the ranked list and clustering can be combined in a novel and effective way. We define the evaluation methodology that we use to evaluate our approach. We then show how our approach can be explained to the user in a clear and intuitive fashion by presenting him or her with a clustering visualization. We describe the visualization technique and hypothesize that it is indeed an intuitive way to navigate the retrieved set, after which we present the result of a small user study that supports our hypothesis. We conclude with the discussion of the results and an outline of directions for future work.

1.1 Related Work

Multiple document organization approaches have been designed and studied in recent years. The most widely used method is the ranked list where the documents are ranked by their probability of being relevant: the highest ranked document is the most similar to the query, the second is slightly less similar, and so on. This ordering is simple, intuitive, and the user is expected to follow it while examining the retrieved documents. The evaluation methods for this approach are also well-developed and the ranked list has been shown to perform well under multiple circumstances (Harman & Voorhees, 1997; Harman & Voorhees, 1998).

The main disadvantage of the ranked list is that once the ordering is defined it cannot be changed. Thus, a small mistake in the query formulation may result in the relevant material appearing in the list far away from the top and it is impossible to recover from this situation short of changing the query and reordering the documents. The relevance feedback procedure does exactly this: it allows the user to mark the examined documents as relevant, adjusts the original query to resemble the relevant documents, and uses the modified query to rerank the documents. This approach has been shown to dramatically improve performance (Salton & Buckley, 1990).

The use of clustering in Information Retrieval is based on the Cluster Hypothesis: "closely associated documents tend to be relevant to the same requests" (van Rijsbergen, 1979, p.45). Croft (1978) and more recently Hearst and Pedersen (1996), showed that this hypothesis holds in a retrieved set of documents. A simple corollary of this hypothesis is that if we do a good job at clustering the retrieved documents, we are likely to separate the relevant and non-relevant documents into different groups. If we can direct the user to the right group of documents, we would increase his or her chances of finding the interesting information with minimal effort.

A major problem of course is to find the cluster (or clusters) containing relevant documents. For example, Hearst and Pedersen (1996) suggested having the users select the cluster of relevant documents based on the textual descriptions of the clusters created automatically by their system.

Multiple visualization approaches for document organization have been developed in recent years. Generally these visualizations are browsing interfaces designed to reveal patterns in the document set. The format of the presentation varies significantly from system to system.

It is common for the information organization to be presented graphically. The documents, paragraphs, and concepts are usually shown as points or objects in space with their relative position indicating how closely they are related. Allan (Allan, 1995; Allan, 1997) 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.

The Vibe system (Dubin, 1995) is a 2-D 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 those terms in that document. The user can shift the location of terms and adjust their weights to better understand the relationships between the documents.

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 (Hemmje, Kunkel, & Willet, 1994) 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.

The Bead system (Chalmers & Chitson, 1992) uses a graph drawing algorithm called spring-embedding for placing high-dimensional objects in a low-dimensional space. The system puts documents in 3-dimensional space, positioning them according to the inter-document similarity. The Bead research did not investigate the question of separating relevant and non-relevant documents. The system was designed to handle very small documents – bibliographic records represented by human-assigned keywords. Leuski and Allan (Leuski & Allan, 1998; Leuski & Allan, 1999) adopted a similar approach and applied it to complete, full-sized documents. We use spring-embedding in our study.

All of this work has focused on developing the power of a ranked list or of a clustering approach. Our work, in contrast, unites the two approaches in one system and shows how their respective powers combine to yield an even more effective system. Swan and Allan (1998) considered a system with both

components but there was no exploration of how the two could be effectively used together. Anick and Vaithyanathan (1997) also integrated clustering and ranking to facilitate browsing of the retrieval results, but did not explore the effectiveness. In a recent TREC experiment Evans *et al.*, (1999) investigated the effectiveness of clustering presentation against traditional ranked list approach for relevance feedback. Their result is somewhat inconclusive – the observed advantage of the clustering presentation was small and not statistically significant.

2 Search Strategy

Bookstein (1983) argues that information retrieval should be envisioned as a process, in which the user is examining the retrieved documents in sequence and the system can and should gather the feedback to adjust the retrieval. We adopt a similar notion while looking at organizing the retrieval results. We consider the organization as a process that introduces an order on the retrieved documents and the users are expected to follow that order while examining the results. That process may be "static" – the ordering happens once per retrieval, or it may be "dynamic" – the documents are reordered as the user's feedback is gathered.

For example, to build a ranked list the documents are ordered by probability of being relevant, the user is expected to start at the top of the ranked list and proceed down the list examining the documents one-by-one.

The interactive relevance feedback approach defines another document ordering: the documents are ordered by probability of being relevant, the user is supposed to start from the top and examine the documents until the first relevant document is found. That document is used to modify the query, the unexamined documents are reordered by probability of being relevant to the new query, and the process continues.

Ultimately the document organization defines an ordering of the documents or ranking. However, the word *ranking* is closely associated with the original ranked list. To avoid confusion we call this type of process and the resulting order in which documents are supposed to be examined the *search strategy*. So we have already defined the *ranked list search strategy* and the *interactive relevance feedback search strategy*. The search strategy for clustering could be something like: "Select the best cluster. Pick a document from that cluster and examine it. If the document is relevant, examine the rest of the cluster, otherwise pick another cluster."

The search strategy defines an ordering of the documents, so given a set of documents and relevance judgments two search strategies can be compared using well-known evaluation measures for the ranked list such as recall and precision (Harman & Voorhees, 1998).

In this paper we define a search strategy that integrates both the ranked list and clustering and we evaluate it by comparing it to both the ranked list and interactive relevance feedback search strategies.

Our study assumes that the retrieved set of documents remains constant. One alternative would be to apply the relevance information obtained from the user to retrieve more potentially relevant documents. For example, during the interactive relevance feedback the modified query could be used to bring new documents from the collection into the retrieved set (Aalbersberg, 1992). We believe that the approach discussed in this paper can be extended to accommodate new documents filtering into the set, however, we leave this question for future work. Nevertheless, the assumption of having a fixed document set is reasonable when the collection access is expensive, or the whole collection is not available as in routing or filtering (Allan, 1996).

3 System Design

In our study we use two IR tools: the ranked list and clustering. The ranked list is supplied by INQUERY (Allan *et al.*, 1998). The INQUERY system is based on a probabilistic model of retrieval and it neither incorporates the notion of similarity between documents nor includes document representations. Therefore, to cluster the documents we use a vector-space approach where each document is represented by a vector term weights V. The weight of the ith term in the vocabulary, v_i is computed using the INQUERY weighting formula, which uses Okapi's *tf* score (Robertson *et al.*, 1995) and Inquery's normalized *idf* score:

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

where *tf* is the number of times the term occurs in the document, *docf* is the number of documents the term occurs in, *doclen* is the number of terms in the document, *avgdoclen* is the average number of terms per document in the collection, and *N* is the number of documents in the collection. The dissimilarity between a pair of documents is measured by one over the cosine of the angle between the corresponding vectors (i.e., $1/\cos\theta$). That is the inverted measure of similarity between documents that is widely used in the vector-space model (Salton, 1989).

3.1 Combining Ranked List and Clustering

We define the following search strategy. At the beginning, in the absence of any relevant information, it seems a reasonable choice to start with the document that is the most similar to the query. Thus, we order the documents by probability of being relevant and follow this ordering until we find the first relevant document. We use this document as a seed to initiate the clustering algorithm. We grow a centroid-based cluster from that document using the unexamined documents: given the cluster of examined relevant documents we select the unexamined document that is the closest to the center of the cluster. If the document is relevant, we add it to the cluster, otherwise we discard it. Figure 1 illustrates how the cluster-growing phase works. There the document vectors are shown as disks. The black disks indicate relevant documents, the white – non-relevant, and the gray – unexamined documents. The "seed" document is the disk with the cross in the center. The cross indicates the current center of the relevant document cluster and the big circle – its boundaries. We show three separate snapshots of the cluster-growing process starting from the leftmost picture. The closest document to the center of the cluster is non-relevant (a white disk), it is ignored and the next disk is considered. It corresponds to a relevant document (a black disk), it is included into the cluster and the cluster center is adjusted – the cross shifts in the second snapshot; the gray cross indicates the old position of the cluster center. The process continues until all documents are examined.

This approach resembles the interactive relevance feedback procedure. However, we use only the information that is available after the first retrieval session and we do not require any query modifications and repetitive usage of the retrieval system.

The first question of our study is how well our search strategy performs when compared to the search strategies for the ranked list and interactive relevance feedback.

3.2 Visualization

In the study conducted by Koenemann and Belkin (1996) the users routinely expressed their desire to "see and control" the feedback process. We believe that we can explain our algorithm and provide a user with a sense of control by visualizing the search strategy. We present the documents as points in 2- or 3-dimensional space and map the inter-document distances onto the Euclidean distances between

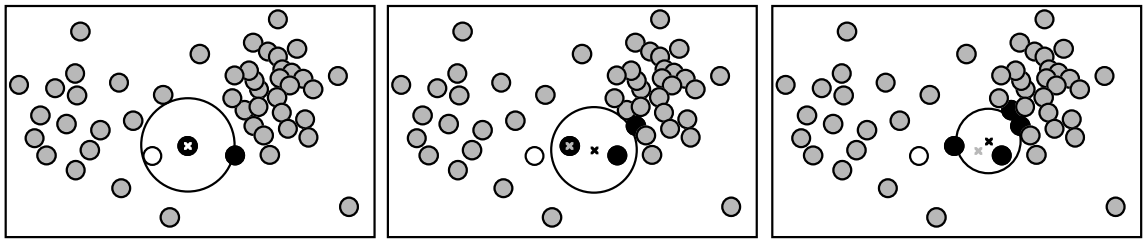

Figure 1: Shows three consecutive snapshots of the cluster-growing phase of our search strategy. We start from the document with an "X" inside and look for the rest of relevant documents. We show the state as the first, second, and third relevant documents are discovered. The white disks represent the known non-relevant documents, the black disks – the known relevant, and the gray disks are the unknown documents.

points. Then similar documents should be shown nearby and the choices of the clustering algorithm can be explained: "each time we select the object that is the closest to all relevant documents discovered so far." In fact, it should look like we are growing a spherical cluster from the first relevant document and shifting the center of the sphere as new documents are added to the cluster (see Figure 1).

A set of techniques under the generic name of Multidimensional Scaling (MDS) has been developed to present high-dimensional objects in just a few dimensions (Borg & Lingoes, 1987). An MDS algorithm accepts a matrix of inter-object dissimilarities and attempts to create a set of points in a Euclidean space such that the distances between the points as closely as possible correspond to the dissimilarities between original objects. A number of such algorithms exist; for our study we have selected an approach called spring-embedding. Our choice was motivated by the graph-drawing heritage of the spring-embedding (Fruchterman & Reingold, 1991; Swan & Allan, 1998) – it is supposed to generate eye-pleasing pictures – and the availability of the source code.

The spring-embedding algorithm models each document vector as an object in 2- or 3-dimensional visualization space. It is assumed that the objects repel each other with a constant force. They are connected with springs and the strength of each spring is inversely proportional to the $1/cos$ dissimilarity between the corresponding document vectors. This "mechanical" model begins from a random arrangement of objects and due to existing tension forces in the springs, oscillates until it reaches a state with "minimum energy" – when the constraints imposed on the object placements by the springs are considered to be the most satisfied. The result of the algorithm is a set of points in space, where each point represents a document and the inter-point distances closely mimic the inter-document dissimilarity. Figure 2 gives an example of 50 documents "spring-embedded" in two and three dimensions.

The document vectors occupy a very high-dimensional space where the number of dimensions is equal to the vocabulary size of the retrieved set. When the documents are visualized with the spring-embedding algorithm some of the documents may be shown nearby when they are actually unrelated because of the constraints imposed by fewer number of dimensions.

Additionally, the document dissimilarity cannot be accurately mapped onto Euclidean distance because the triangle inequality is not always satisfied for the $1/cos$ metric:

$$\exists A, B, C \in D : \rho(A, B) + \rho(B, C) < \rho(A, C), \tag{2}$$

where $\rho(\cdot)$ is the dissimilarity function and D is the document set.

Thus, during the transition from the document vector space to a Euclidean space of a few dimensions the cluster of relevant documents might lose the intuitive spherical shape and appear distorted. The spherical

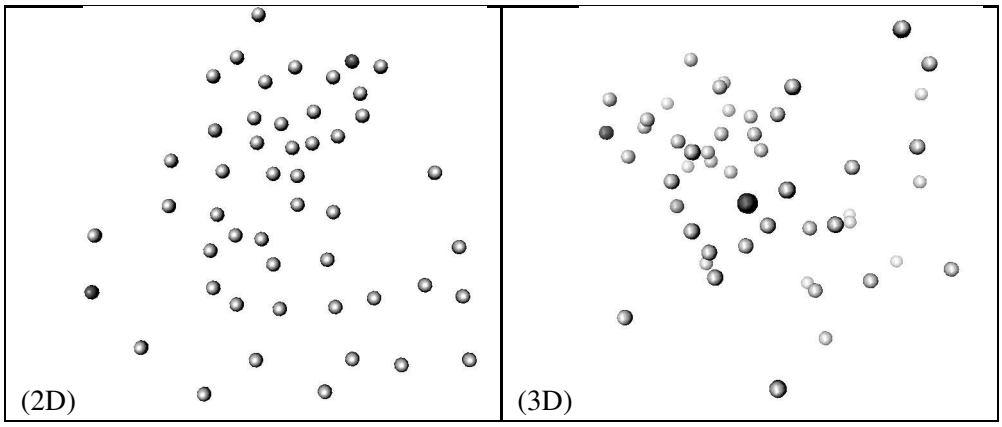

Figure 2: Shows a set of 50 documents visualized in 2 and 3 dimensions.

shape of the cluster is important as it is supported by the notion of spatial *closeness*. If the cluster is distorted e.g., it has an ellipsoidal shape we will have to explain why a particular spatial direction is preferred over another while the choice of the closest object is being made.

If we ignore the distortions and keep the cluster spherical, we preserve the visualization metaphor but we are likely to lose in the performance of the search strategy. To keep the cluster spherical while running the clustering algorithm we use the Euclidean distance between points in the visualization instead of $1/cos$ dissimilarity between the original document vectors.

Thus the second question we investigate in this study is how much of the search strategy quality we will sacrifice by preserving the consistency of the visualization. Our intuition is that a higher dimensional visualization will provide more degrees of freedom and therefore it has a better chance to represent the inter-document relationships accurately than a lower dimensional one. We expect that the search strategy in a 3-dimensional visualization will exhibit better performance than in 2 dimensions and the search strategy in the original document-vector space will be the best.

We now present the results of our experiments. In the section that follows we consider the question of how well the users navigate in the visualization.

3.3 Experimental Setup

For our experiments we used TREC ad-hoc queries with their corresponding collections and the relevance judgments supplied by NIST accessors (Harman & Voorhees, 1997). Specifically, TREC topics 251-300 were converted into queries and ran 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) a query constructed by extensive analysis and expansion (Allan *et al.*, 1997); (2) the description field of the topic; (3) the title of the topic; and (4) a query constructed from the title by expanding it using Local Context Analysis (LCA) (Xu & Croft, 1996). A query of the last type has size and complexity between the corresponding queries of the first and second types.

In addition, we used TREC topics 301-350 to create queries to be run against TREC volumes 4 and 5 (2.2GB) that include articles from the Congressional Records, Financial Times, and Los Angeles Times. Again, the same four different types of queries were constructed, except instead of just using the description field for the second query type, we used both the title and the description field of the topic (this was done to compensate for an error in the topic creation in TREC (Harman & Voorhees, 1998)).

Our assumption is that during a typical retrieval session a user does not generally look beyond the first

screen showing the retrieved material – that is approximately equivalent to ten retrieved documents. Thus, we are interested in analyzing just the top portion of the ranked list. For each query we selected the 50 highest ranked documents.

3.4 Results

The first question was to compare our search strategy with the search strategies for the ranked list and for interactive relevance feedback. The interactive relevance feedback is performed in the following way. We start from the top of the ranked list. Each time a new relevant document is discovered, we take all the examined documents and use them to modify the weights in the original query. Additionally, the query is expanded by adding the ten highest ranked terms from the examined documents (Allan, 1996). Note that this procedure takes into account both relevant and non-relevant documents. The unexamined documents in the set are reranked using the new query and we continue down the list.

We have measured performance of the ranked list, relevance feedback and the combination of the ranked list and clustering on the 50 highest ranked documents for 100 queries of 4 different types. We have adopted average precision as the main evaluation statistic. The order of documents in each search strategy is the same starting from the top of the ranked list to the first relevant document. As we are interested in examining the difference in our approaches, we compute the average precision for the part of the document ordering that is different for each search strategy – using only the documents that were originally ranked below the first relevant document. Note that a document set that have less than two relevant documents will have precision of zero. The first two columns in Table 1 show the average precision numbers for two hypothetical search strategies: one randomly distributes relevant documents among non-relevant ("random") and the other ranks all relevant documents above all non-relevant ("best"). The worst case performance – when all non-relevant documents are examined before any of the relevant ones – corresponds to the average precision of 14.7 and is always at least 40% worse than the random performance. These numbers provide a scale for the performance results in the next subsection.

The Combination vs. Ranked List Table 1 shows that our search strategy outperforms the ranked list. Most of the differences indicated are statistically significant by two-tailed t-test with $p < 0.05$.

We observe a 23.3% increase in precision across different query complexities and sizes. It is also important to notice that for the "Title" queries – the two or three word queries that are more likely to be entered by hand than the "Full" queries that require a lot of processing – the performance increase is even greater: 31.4%.

Our search strategy also outperforms the traditional relevance feedback approach by a small margin (4.8%). However, that difference is not generally statistically significant.

Another interesting observation is that our search strategy performs well in two and three dimensions. In fact, even in two dimensions it is significantly better than the ranked list and is comparable with the interactive relevance feedback approach.

Effect of Fewer Dimensions The second question was to compare the performance of the search strategy using the dissimilarities between document vectors against the same strategy that used Euclidean distances in two and three dimensions. The document representations in 2- and 3-dimensional spaces were created using the spring-embedding algorithm. We used the same data set as in the previous experiment.

Table 2 shows a small drop in precision when the search strategy is moved from the high-dimensional document space into a fewer number of dimensions. The drop in precision is less when three dimensions are used instead of two and it is almost never statistically significant by two-tailed t-test with $p < 0.05$.

Data Set		RL			RF	RL + CL		
		Rand.	Best	Orig.	(v. RL%) DD (v. RF%)	2D (v. RL%) (v. RF%)	3D (v. RL%) (v. RF%)	
TREC-5	Full	24.7	82.0	36.2	39.9 (10.2 %)	42.4 (17.3*%) (6.5 %)	40.7 (12.5*%) (2.1 %)	41.6 (15.1*%) (4.4 %)
	Desc	21.3	88.0	34.1	45.1 (32.4*%)	48.7 (42.9*%) (8.0 %)	45.8 (34.4*%) (1.5 %)	47.2 (38.6*%) (4.7 %)
	Title	18.2	74.0	31.0	38.1 (23.0*%)	37.9 (22.6*%) (0.0 %)	39.0 (26.0*%) (2.4 %)	38.9 (25.5*%) (2.0 %)
	Title + Desc	20.2	64.0	31.0	34.8 (12.4 %)	37.4 (20.9*%) (7.6 %)	36.1 (16.5*%) (3.7 %)	37.2 (20.1*%) (6.8 %)
TREC-6	Full	31.1	90.0	51.0	53.6 (5.1 %)	53.6 (5.1 %) (0.0 %)	52.9 (3.6 %) (-1.0 %)	54.9 (7.5 %) (2.3 %)
	Desc	27.3	94.0	42.5	56.0 (31.9*%)	57.8 (36.0*%) (3.1 %)	52.5 (23.6*%) (-6.0 %)	55.2 (30.1*%) (-1.0 %)
	Title	24.8	84.0	40.1	52.0 (29.7*%)	55.4 (38.1*%) (6.5 %)	53.0 (32.1*%) (1.9 %)	54.4 (35.6*%) (4.6 %)
	Title + Desc	28.5	88.0	47.2	48.7 (3.1 %)	52.5 (11.2 %) (7.8 %)	52.4 (10.9*%) (7.6* %)	53.5 (13.4*%) (9.9* %)
total average		24.5	83.0	39.1	46.0 (17.6*%)	48.4 (23.3*%) (4.8* %)	46.7 (19.0*%) (1.1 %)	48.0 (22.3*%) (4.0* %)
titles only		21.5	79.0	35.6	45.0 (26.8*%)	46.7 (31.4*%) (3.6 %)	46.0 (29.5*%) (2.1 %)	46.6 (31.2*%) (3.5 %)

Table 1: Performance of the search strategies for the ranked list (RL), interactive relevance feedback (RF), and the combination of the ranked and clustering (RL + CL). Average precision numbers, percent improvement over the simple ranked list (the first line in each row), and percent improvement over the relevance feedback (the second line in each row) are shown. Besides the actual ranked list quality as generated by INQUERY ("original") we show two hypothetical cases: "random" – the relevant documents are equally distributed in the list and "best" – all the relevant are positioned before all non-relevant. Three different cases of the search strategy for the ranked list/clustering combination are considered: one in the original document vector space (DD), another in 2 dimensional space (2D), and another in 3 dimensional space (3D). Asterisks indicate statistical significance with $p < 0.05$.

Data Set		RL + CL		
		DD	2D (v. DD%)	3D (v. DD%) (v. 2D%)
TREC-5	Full	42.4	40.7 (-4.0 %)	41.6 (-2.0 %) (2.2 %)
	Desc	48.7	45.8 (-6.0 %)	47.2 (-3.0 %) (3.1 %)
	Title	38.0	39.0 (+2.8 %)	38.9 (+2.4 %) (0.0 %)
	Title + Desc	37.4	36.1 (-4.0 %)	37.2 (-1.0 %) (3.0 %)
TREC-6	Full	53.6	52.9 (-1.0 %)	54.9 (+2.3 %) (3.8* %)
	Desc	57.8	52.5 (-9.0* %)	55.2 (-4.0 %) (5.3* %)
	Title	55.4	53.0 (-4.0* %)	54.4 (-2.0 %) (2.6 %)
	Title + Desc	52.5	52.4 (-0.0 %)	53.5 (+1.9 %) (2.2 %)
total average		48.4	46.7 (-4.0* %)	48.0 (-1.0 %) (2.8* %)
titles only		46.7	46.0 (-1.0 %)	46.6 (-0.0 %) (1.3 %)

Table 2: Performance of the search strategies for the combination of the ranked and clustering (RL + CL). Three different cases of the search strategy for the ranked list/clustering combination are considered: one in the original document vector space (DD), another in 2 dimensional space (2D), and another in 3 dimensional space (3D). Average precision numbers, percent improvement over the same search strategy in the document vector space (the first line in each row), and percent improvement over the same search strategy in 2D (the second line in each row) are shown. Asterisks indicate statistical significance with $p < 0.05$.

3.5 Discussion

These results confirm, in the same way that relevance feedback experiments do, that user feedback can dramatically improve the effectiveness of a ranked list. Unlike most past efforts ((Allan, 1996) being a recent exception) we also show that it is also true when feedback is incremental – and even if no new documents are retrieved. Further, we have confirmed this result in a setting where we believe the user will be able to oversee and control the feedback process.

The visualization that we use to that effect is 2- or 3-dimensional approximation of relationships in a high-dimensional space. Those results show that the dimensionality reduction does not substantially degrade the inter-document similarity information. As one might expect, three dimensional approximation is better than two dimensional since it retains one extra degree of freedom to position the documents. However, past experience has suggested that the extra dimension is of no value to the users (Swan & Allan, 1998), perhaps because of additional cognitive overhead. We next investigate that question, along with other user issues.

4 User Study

The idea of searching for relevant information by examining the document that is the closest to already discovered relevant material seems simple. We assume that given an accurate visual representation of inter-document similarities the user can effectively locate the relevant documents without any aid from the system.

Thus, the third question of our study is how effective in locating the relevant information will the user be when given the spring-embedding visualization of the retrieved set? We believe that the notion of spatial similarity in the spring-embedding visualization is an intuitive and accurate metaphor for representing inter-document relationships. We hypothesize that the user's search strategy will be similar to ours in both procedure and effectiveness.

To test these hypotheses we have implemented a short computer-based user study. It was designed to simulate a user looking for relevant information in the visualization. Each participant in this study had to solve a number of information foraging problems. Every problem consisted of a set of white spheres floating in space. The participants were told that the spheres are of two colors: red and green. Initially only one sphere was shown in green and there could also be some red spheres. The true color of a white sphere could be discovered – the sphere could be "opened up" – by double-clicking on the sphere with the mouse pointer. The participants were asked to find all the green spheres as quickly as possible, trying to avoid opening red spheres. The participants received a small time penalty for opening a sphere – the sphere was animated for several seconds before showing its true color. They were also prohibited from double-clicking on a sphere while another was opening. This was done to discourage the users from mindlessly clicking the spheres in random order. At the same time it crudely simulated the delay that would have been experienced by a person while reading and judging the document.

The participants were told that spheres of the same color (e.g., green spheres) tend to appear in close proximity to each other (similar spheres generally group together) but not necessarily so. The last hint was a direct corollary from the Cluster Hypothesis as the spheres represented the documents, and the color, the document relevance value. A green sphere indicated a relevant document and a red one indicated a non-relevant document. However, the participants were not told the meaning of the spheres. We believe this design eliminates a high uncertainty that is generally connected with query formulation and passing relevance judgments (Koenemann & Belkin, 1996; Swan & Allan, 1998) and allows us to isolate the navigation properties of the visualization which are the focus of our study.

The problems were presented in two and three dimensions. The three-dimensional effect was created by

RL	Algorithm		User					
	2D	3D	2D (v. RL %) (v. Alg 2D%)	significance		3D (v. RL %) (v. Alg 3D%) (v. Usr 2D%)	significance	
42.9	59.1	61.4	55.8 (30.1* %) (-5.7* %)	$p < 5 \cdot 10^{-8}$	$p < 5 \cdot 10^{-4}$	53.2 (24.1* %) (-13.3* %) (-4.6* %)	$p < 5 \cdot 10^{-6}$	$p < 5 \cdot 10^{-12}$ $p < 0.01$

Table 3: Users' performance navigating the visualizations of ten randomly selected document sets. The numbers are averaged across all selected document sets. Average precision numbers, percent improvement over the ranked list search strategy, percent improvement over the algorithmic search strategy in the corresponding dimension, and percent improvement of using 3D over 2D are shown. We also show the significance level for each difference by two-tailed t-test.

using a 3D-rendering engine. To improve the depth perception, a simulated fog effect was added to the picture. The participants were able to rotate, slide, and zoom the set of spheres. The application interface of a two-dimensional presentation was equivalent to that of a three-dimensional one except that the user saw a flat structure on the screen (i.e., he or she could still rotate and zoom the 2-dimensional structure).

Each participant was presented with ten problems. We divided the problems into two equal groups. The problems in one group were shown in two dimensions, the problems in the other – in three dimensions. The dimensions in which each group of problems was shown alternated between users. We also varied the order in which the groups were presented and the order in which the problems inside each group were presented. This was done to account for a possible learning effect. Before each group of problems was shown to a participant he or she was given two training problems to familiarize herself with the application interface. The participants were also asked to fill out questionaries before and after the study.

The study was designed to be completely supervision-free. The software was written in Java and it is available via World Wide Web (User Study, 1999). We have advertised the study in local newsgroups and in information retrieval mailing lists on the Internet. At the time of this report 40 people have expressed their interest in the study by accessing the software; 20 of them have completed it, spending on average one hour and thirty minutes with the system.

4.1 Results

To create the problems we randomly selected ten topics from TREC topics 251-350 (see Section 3.3). The following topics were selected: 257, 259, 273, 277, 294, 298, 304, 327, 330, 342. We used the "Title" versions of the corresponding queries to retrieve the 50 top ranked documents for each topic. These documents were visualized and presented to the users in 2 and 3 dimensions. At the beginning of each problem the spheres corresponding to the highest ranked relevant document and the non-relevant documents that precede it in the ranked list were shown in color. The rest of the documents were shown in white – i.e., as if starting after the first relevant document in the ranked list was found. This was supposed to provide the users with the starting point in their exploration

The users examined the white spheres in sequence. The order in which each user double-clicked the spheres defined the search strategy for that user. To distinguish it from the search strategy discussed in previous experiments we call the latter the *algorithmic* search strategy. We calculated the average precision for the user's strategy and averaged it across all users and all problems. We observed a significant drop in average precision while comparing the algorithmic search strategy with the users' average performance (Table 3). Note, though, that it also indicates that the users do significantly better by using the visualization than by blindly following the ranked list.

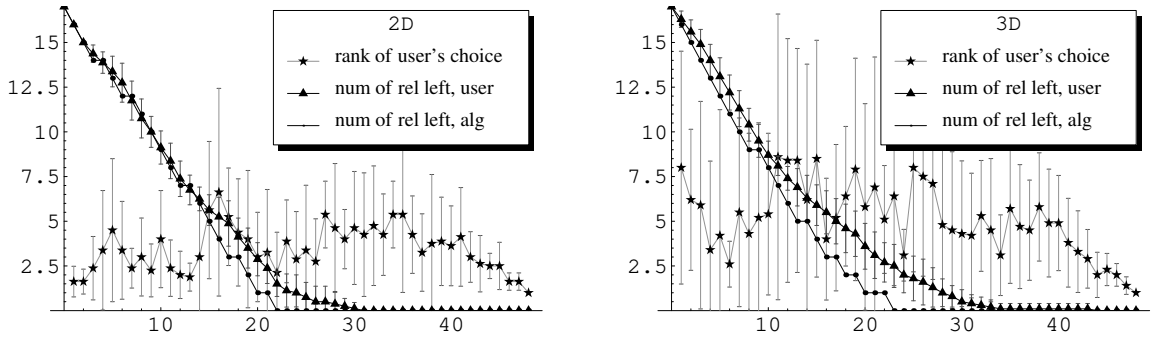

Figure 3: Comparing the users' average search strategy to the algorithmic search strategy in 2 and 3 dimensions for one document set. The X-axis is the number of the examined documents. We show the number of green spheres remaining unexamined by the algorithm ("num of rel left, alg"), the same number for the users ("num of rel left, user"), and the rank of the user choice ("rank of user's choice").

All the differences between the users' performance in both dimensions and the ranked list, between the users' performance and the algorithmic search strategy in both dimensions and between users' performance in different dimensions are statistically significant by two-tailed t-test with at least $p < 0.01$.

The algorithmic search strategy has a higher performance when working with a 3-dimensional representation of the document set than with a 2-dimensional representation. The users on the other hand show much better results in 2 dimensions than in 3 dimensions. From this observation and also from the comments we have collected during the study we conclude that the users have a much harder time establishing proximity relationships and navigating the 3 dimensional visualization.

We were interested in comparing the users' search strategy with the algorithmic search strategy. Each time the user selected a sphere to examine, the algorithmic search strategy ranks the unexamined (white) spheres by the spatial proximity to the current cluster of examined green (relevant) spheres and assigns a rank number to the user's choice. Note that in this situation the algorithm will select the highest ranked sphere. If both the algorithm and the user select the same sphere, the user's choice is ranked as one. Figure 3 shows the rank of the users' choice as each successive sphere is selected. The X-axis is the index of the examined sphere. We show both the average rank and the error bar indicating the standard deviation. We also show the average and standard deviation for the number of green spheres remaining unexamined by the user at each step, and the same number for the algorithm. For example, at the beginning there are 18 unexamined spheres ($x = 0$). In two dimensions both the users and the algorithm succeed in locating a green sphere on first pick – the number of unexamined green spheres drops to 17 for $x = 1$. The users always select a green sphere – the standard deviation is zero. We also see that the sphere selected by the users on that step has an average rank of 1.5. It means that the users almost always picked up either the first or the second closest white sphere to the first green sphere.

Figure 3 presents the algorithm and users' behavior for one of the document sets. We have observed similar effects on the rest of the data. Comparing the plots in 2 and 3 dimensions we see that

1. the algorithm is more successful in 3 dimensions than it is in 2D – the plot line for the number of green spheres left after the algorithm's pick descends faster in the right figure (3D).

2. the users are more successful in 2 dimensions than they are in 3D – the plot line for the number of green spheres left after the users' pick descends faster and trails the same line for the algorithm closer in the left figure (2D).

3. the users' choices have less variance in 2 dimensions than in 3 – the error bars are generally shorter

Question	2D	3D
How easy was it to **understand how to use** the system?	4.4	3.4
How easy was it to **learn to use** the system?	4.6	3.6
How easy was it to **use** the system?	4.3	2.7
Are you satisfied with the system's organization of data? Does the system's placement of the objects makes it easier to find the green spheres?	3.4	2.7
Are you satisfied with your performance in finding the green spheres?	3.6	3.1

Table 4: Users' responses to a number of questions comparing 2D with 3D visualizations. The answers were given as "grades" between 1 and 5. The average grade is shown for each question. The higher numbers are better ("easier", "more satisfied").

in the left figure (2D).

4. the users' search strategy is more similar to the algorithmic search strategy in 2 dimensions – the average rank of the users' choice is smaller.

5. the users' search strategy is not significantly different from the algorithmic search strategy at least at the beginning of the search – the average users' choice is always less than one standard deviation apart from the algorithm's first choice for the first 14 examined documents.

6. the users' "bad" choices in 2 dimensions – the line for the number of unexamined green spheres left for the users separates upward from the same line for the algorithm in the left figure – correlate with the users' search strategy strongly deviating from the algorithmic search strategy – shown as "picks" on the graph representing the rank of the users' choice at $x = 5$ and $x = 15$.

We asked the users to fill out short questionnaires about their experience with the system comparing 2D with 3D. Specifically, we asked the users to assign a "grade" between 1 and 5 measuring how easy it was to use each presentation and if, in their opinion, the system did a good job at organizing the objects. The average grades in Table 4 show that the users preferred the 2D presentation over the 3D one. They overwhelmingly found 2D visualization easier to use and they were generally satisfied with system's arrangement of the green and red spheres.

We conclude that both our hypotheses are supported. The users have no difficulty grasping the idea of spatial proximity as the metaphor for inter-object similarity. Their search strategy is very similar to the notion of selecting the spheres that are close to the known green spheres. In fact, one of the participants explicitly stated he was trying to pick up the "white ball that is the closest to the average of the green set." Generally the users were very successful in following this tactic. However, the more spheres were examined, the more difficult it became to correctly identify the similarity between the cluster of green spheres and the remaining white spheres. This task was even more difficult in three dimensions. The users constantly pointed out that correct identification of the inter-sphere distances in 3D required frequent rotations of the structure, thus making the visualization task more difficult. We believe these effects account for the observed differences in precision between the algorithmic and user search strategies.

4.2 Efficiency

The MDS algorithm used in this study requires $O(kN^2)$ time to position N documents in 2 or 3 dimensions. Given a matrix of pairwise document similarities the current algorithm implementation takes 0.4 sec to arrange 50 objects and 1.4 sec to arrange 100 objects on a computer with 600MHz Alpha CPU. To build the similarity matrix for 50 document vectors requires another 0.5 sec on the same machine. These

performance numbers indicate that the clustering algorithm can be applied in real-world interactive settings to visualize small document sets such as the top retrieved documents from a Web-based search engine.

5 Conclusion

The Cluster Hypothesis of Information Retrieval has been shown true on multiple occasions. Both the ranked list and clustering have proven very successful in organizing the retrieved documents. In this paper we have presented a novel approach to combining clustering with the ranked list to interactively direct the user's search for relevant information in the top ranked portion of the retrieved set. We have experimentally shown that this approach significantly exceeds the initial performance of the ranked list and rivals in its effectiveness the traditional relevance feedback methods.

We argue that an approach such as this one is easier to understand because it does not involve rearranging documents nor changing the visualization, giving the user a sense of control and reducing potential confusion. We have shown that the retrieved set can be visualized in 2- and 3-dimensions and the same procedure of selecting the documents based on their proximity to the relevant material can be repeated in the fewer dimensions with just an insignificant loss in precision.

Visualizing the data in fewer dimensions decreases the accuracy of the presentation. We observed that a three dimensional visualization is more accurate in representing the inter-document relationships than a two dimensional one. Consequently, the document selection algorithm is more effective in three dimensions than in two.

We also showed that spatial proximity is an intuitive and well-recognized (by the users) metaphor for similarity between objects. We observed that the users' search strategy tends to follow the model incorporated into our algorithmic approach. The users' were significantly more successful with the visualization than they would be by following the ranked list. The precision difference between the users and the algorithm arises from difficulty to precisely identify the inter-object distances in the visualization. We also observed that the higher accuracy with which 3-dimensional structure represents the document set is negated by the heavy cognitive effort that is required from the users to navigate the visualization.

6 Discussion and Future Work

The combination of the ranked list and clustering suggested in this paper does not mandate the spring-embedding visualization. We see several possible alternatives to the visualization. For example, as soon as a new relevant document is discovered we can reorder the rest of the list by the similarity to the examined relevant documents. This is similar to reordering the list during the interactive relevance feedback method. Aalbersberg (1992) claims that such abrupt changes in the document order are undesirable and should be avoided. We believe they can distract the user and force him or her to lose the search context.

Another alternative is to highlight the next suggested document in the existing ranked list instead of reordering the list. Thus, there will be no sudden and dramatic changes in the document order. However, this procedure fails to explain why that particular document is selected. There seems to be no simple way to represent inter-document similarity in the ranked list that does not destroy the original ranking.

We also can consider the query as a document, place it in the visualization and initiate the search strategy from that "query-document" instead of the first relevant document in the ranked list. It will allow us to remove dependency on the ranked list and consider only the clustering visualization. However, a document-query similarity is almost always much lower than a document-document similarity (small queries have few words in common with documents). The query-document will appear as a disc that is

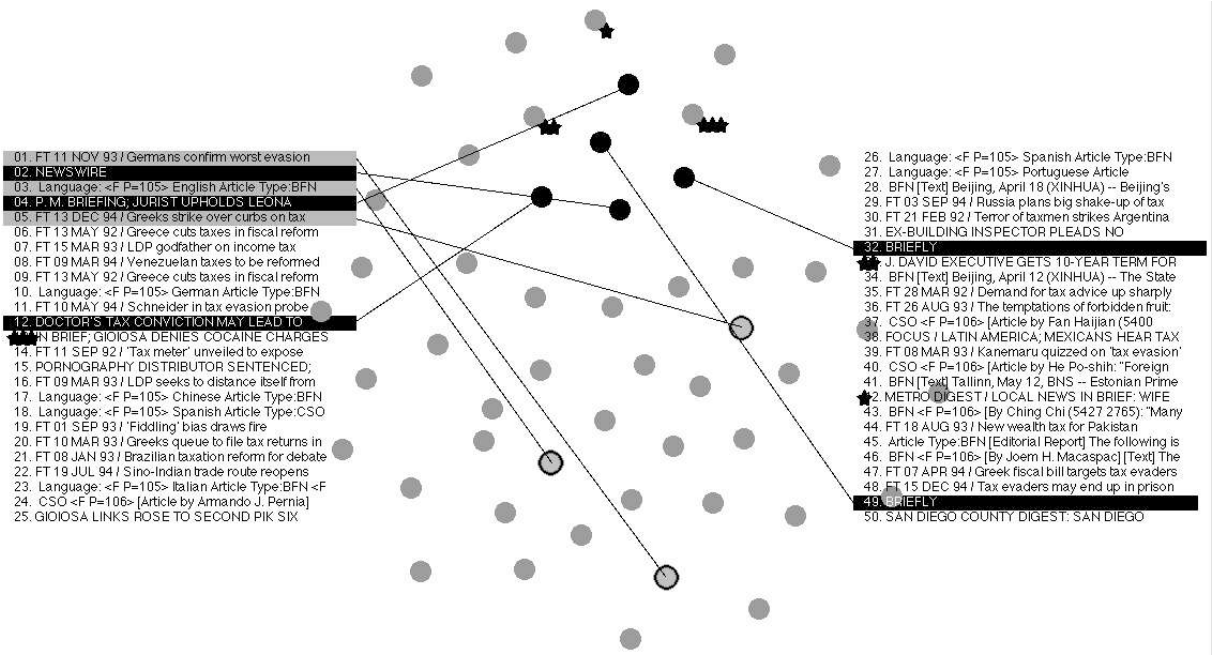

Figure 4: Shows a system that integrates the ranked list and spring-embedding visualization. The ranked list flows starting from the top left corner downward and then over to the top right corner and the visualization is in the middle. Document titles are shown in the list; missing titles are replaced with the first 20 terms from the document. The relevant documents are shown in black and non-relevant documents are shown as black circles filled with gray color. The lines connect the document title with the corresponding disc. The black stars indicate the three closest documents to the center of the relevant cluster (3 star document is the closest, 2 – the next closest, etc.).

far away from the rest of the documents. The positioning error created by the MDS algorithm will be far greater for the query than for the documents and this error might very well result in a bad selection for the seed document.

Figure 4 shows the system that we have created as the result of this study. It puts both the ranked list and spring-embedded visualization on the screen together. Here the visualization is positioned between two parts of the ranked list. If a document is selected in the ranked list, the corresponding sphere is highlighted, and vice versa. We show the first document as non-relevant; it is highlighted using light-gray in the ranked list. The corresponding sphere is in the lower part of the visualization. The second document is relevant and the highlighted black sphere is at the middle of the visualization.

Our study showed that the user might require some aid in pinpointing the neighborhood spheres. The system marks the three documents closest to the relevant cluster with stars: three stars indicate the closest document, two stars indicate the next closest, one star indicates the third closest document. Figure 4 shows an example of the top 50 documents retrieved in response "Income Tax Evasion" (TREC topic 332) from our test collection. The documents are embedded in two dimensions. The lines connecting the document titles with the corresponding discs are provided for illustration purposes and not present in the real system.

The system also utilizes color intensity to indicate the document proximity in the visualization: the document discs are filled with the color that varies in intensity from highly saturated red through white to highly saturated green in proportion to the document's spatial proximity to the known relevant and non-relevant documents.

Our analysis in this study was limited to the top 50 retrieved documents for each query. We plan to extend our experiments to accommodate a large number of documents hoping that the clustering visualization might pull in more relevant documents. However, we designed the system as a browsing tool to locate individual documents and not to study the topic distribution in the collection. For the interactive setting it will be difficult to accommodate more than 100 documents on the screen at one time. Even with 100 objects the visualization becomes "overcrowded" with discs. It will be more interesting to consider approaches when the known documents slowly "drop out" from the picture getting replaced with the fresh unknown documents as the search progresses. The rate of the documents disappearance will depend on their relevance and the user preferences.

One problem with any relevance feedback approach is the requirement for the user to provide relevance judgments. The user has to explicitly mark the examined documents, otherwise the feedback procedure fails. We plan to consider a more subtle approach for eliciting the judgments. Every time the user selects a document to examine, we will highlight two different documents that could be examined next: one if the user likes the current document and another if the user does not like it. We can then attempt to deduce the relevance value of that document by taking into account the user's next choice.

Our search strategy performs best when there is only one relevant topic in the retrieved set. In that case all the relevant documents form one cluster that is easily detected as soon as the first relevant document is located. The strategy doesn't take into account cases when there are several different but relevant topics and clumps of relevant documents appear to be scattered in the visualization. The users were observed to perform better than the algorithm in such a situation: getting annoyed by discovering many red spheres in one part of the visualization, the users jumped away and explored a different area. Our search strategy is not able to do that, though it might be possible to provide such an effect.

When the user discovers a new relevant document the relevance feedback procedure analyses the document and modifies the query. We are currently considering similar methods where instead the document representations are modified by moving the relevant documents closer together and away from the non-relevant documents.

Acknowledgments

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,and also supported in part by United States Patent and Trademark Office and 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

Aalbersberg, I. J. (1992). Incremental relevance feedback. In *Proceedings of ACM SIGIR*, pp. 11–22.

Allan, J. (1995). *Automatic Hypertext Construction*. Ph. D. thesis, Cornell University.

Allan, J. (1996). Incremental relevance feedback for information filtering. In *Proceedings of ACM SIGIR*, pp. 270–278.

Allan, J. (1997). Building hypertext using information retrieval. *Information Processing and Management 33*(2), 145–159.

Allan, J., J. Callan, B. Croft, L. Ballesteros, J. Broglio, J. Xu, & H. Shu (1997). Inquery at TREC-5. In *Fifth Text REtrieval Conference (TREC-5)*, pp. 119–132.

Allan, J., J. Callan, W. B. Croft, L. Ballesteros, D. Byrd, R. Swan, & J. Xu (1998). Inquery does battle with TREC-6. In *Sixth Text REtrieval Conference (TREC-6)*, pp. 169–206.

Anick, P. G. & S. Vaithyanathan (1997). Exploiting clustering and phrases for context-based information retrieval. In *Proceedings of ACM SIGIR*, pp. 314–323.

Bookstein, A. (1983). Information retrieval: A sequential learning process. *Journal of the American Society for Information Science 34*(5), 331–342.

Borg, I. & J. Lingoes (1987). *Multidimensional similarity structure analysis*. Springer-Verlag.

Chalmers, M. & P. Chitson (1992). Bead: Explorations in information visualization. In *Proceedings of ACM SIGIR*, pp. 330–337.

Croft, W. B. (1978). *Organising and Searching Large Files of Documents*. Ph. D. thesis, University of Cambridge.

Dubin, D. (1995). Document analysis for visualization. In *Proceedings of ACM SIGIR*, pp. 199–204.

Evans, D. A., A. Huettner, X. Tong, P. Jansen, & J. Bennett (1999). Effectiveness of clustering in ad-hoc retrieval. In D. Harman & E. Voorhees (Eds.), *Seventh Text REtrieval Conference (TREC-6)*.

Fruchterman, T. M. J. & E. M. Reingold (1991). Graph drawing by force-directed placement. *Software–Practice and Experience 21*(11), 1129–1164.

Harman, D. & E. Voorhees (Eds.) (1997). *The Fifth Text REtrieval Conference (TREC-5)*. NIST.

Harman, D. & E. Voorhees (Eds.) (1998). *The Sixth Text REtrieval Conference (TREC-6)*. NIST.

Hearst, M. A. & J. O. Pedersen (1996). Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of ACM SIGIR*, pp. 76–84.

Hemmje, M., C. Kunkel, & A. Willet (1994). LyberWorld - a visualization user interface supporting fulltext retrieval. In *Proceedings of ACM SIGIR*, pp. 254–259.

Koenemann, J. & N. J. Belkin (1996). A case for interaction: A study of interactive information retrieval behavior and effectivness. In *Proceedings of Conference on Human Factors in Computing Systems*, pp. 205–212.

Leuski, A. & J. Allan (1998). Interactive cluster visualization for information retrieval. In *Proceedings of ECDL'98*, pp. 535–554.

Leuski, A. & J. Allan (1999). Evaluating a visual navigation system for a digital library. *International Journal on Digital Libraries*. Forthcoming.

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, & M. Gatford (1995). Okapi at TREC-3. In D. Harman & E. Voorhees (Eds.), *Third Text REtrieval Conference (TREC-3)*. NIST.

Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.

Salton, G. & C. Buckley (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41*, 288–297.

Swan, R. & J. Allan (1998). Aspect windows, 3-d visualizations, and indirect comparisons of information retrieval systems. In *Proceedings of ACM SIGIR*, pp. 173–181.

User Study (1999). http://toowoomba.cs.umass.edu/~leouski/SE.

van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworths. Second edition.

Xu, J. & W. B. Croft (1996). Querying expansion using local and global document analysis. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pp. 4–11.