2001

Evaluating combinations of ranked lists and visualizations of inter-document similarity

James Allan
University of Massachusetts Amherst

PERGAMON

Information Processing and Management 37 (2001) 435–458

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Evaluating combinations of ranked lists and visualizations of inter-document similarity

James Allan *, Anton Leuski, Russell Swan ☆, Donald Byrd

Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01003, USA

Accepted 19 September 2000

Abstract

We are interested in how ideas from document clustering can be used to improve the retrieval accuracy of ranked lists in interactive systems. In particular, we are interested in ways to evaluate the effectiveness of such systems to decide how they might best be constructed. In this study, we construct and evaluate systems that present the user with ranked lists and a visualization of inter-document similarities. We first carry out a user study to evaluate the clustering/ranked list combination on instance-oriented retrieval, the task of the TREC-6 Interactive Track. We find that although users generally prefer the combination, they are not able to use it to improve effectiveness. In the second half of this study, we develop and evaluate an approach that more directly combines the ranked list with information from inter-document similarities. Using the TREC collections and relevance judgments, we show that it is possible to realize substantial improvements in effectiveness by doing so, and that although users can use the combined information effectively, the system can provide hints that substantially improve on the user's solo effort. The resulting approach shares much in common with an interactive application of incremental relevance feedback. Throughout this study, we illustrate our work using two prototype systems constructed for these evaluations. The first, AspInQuery, is a classic information retrieval system augmented with a specialized tool for recording information about instances of relevance. The other system, Lighthouse, is a Web-based application that combines a ranked list with a portrayal of inter-document similarity. Lighthouse can work with collections such as TREC, as well as the results of Web search engines. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Information retrieval; Clustering; Document visualization; User study

* Corresponding author. Tel.: +1-413-545-0463; fax: +1-413-545-1789.

E-mail addresses: allan@cs.umass.edu (J. Allan), leuski@cs.umass.edu (A. Leuski), dbyrd@cs.umass.edu (D. Byrd).

☆ Deceased.

0306-4573/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.
PII: S0306-4573(00)00056-X

1. Introduction

Most information retrieval (IR) systems rank the documents in a collection according to how likely they are to satisfy the query. The documents are presented in that order, in what is called a *ranked list*. The higher in the ranked list a document is (i.e., the lower the rank number assigned to the document), the more the system expects that document is likely to be relevant to the query. The goal of much research in IR is to improve the quality of the ranked list, i.e., to modify the system such that relevant documents appear higher in the ranked list and non-relevant documents are "pushed" further down.

The problem with ranked lists is that the relevant documents are often not at the top of the ranked list. Even when a system is particularly accurate, there might be a few relevant documents highly ranked, with the rest substantially further down in the ranking. Finding the rest of the relevant documents requires wading through many that are non-relevant. In some situations this problem is not important, but it is particularly acute in cases where users want broad coverage of a topic.

Another technique that some IR systems use to help users find relevant documents is *clustering*. In this case, the system starts with a set of documents and every document is compared to every other document, resulting in pairwise similarities. Documents that are sufficiently similar to each other are grouped together into clusters. These clusters can be used in several ways: the clusters might be compared to the query and presented in a ranked list, or the cluster containing a high-ranked document might be highlighted in case the user is interested.

The advantage of clustering is that it brings highly similar documents together, and it is well known that those documents are likely to be relevant to similar queries. The disadvantage is that "highly similar" is difficult to define. Setting a threshold for what should and what should not be in a cluster is very difficult. As a result, clusters are likely to be either useful but very small, or else larger and too scattered to be useful.

In this study, we are interested in evaluating efforts to combine clustering with the traditional ranked list. We will do that by creating systems that present the user with a ranked list as well as a visualization of the way that documents cluster. To avoid the cluster threshold problem, we will place the documents in a visualization that illustrates their similarity to each other, without making explicit choices about where the cluster begins or ends. We will allow the users to make that choice.

Our interactive system evaluations will be carried out using TREC data. We show in this study how the judgments from TREC can be used in two different ways. The first and most obvious is the user study that carefully monitors several users attempting a task using variants on the same system. Unfortunately, much work of this nature has had inconclusive results (Veerasamy & Belkin, 1996; Lagergren & Over, 1998): it is very difficult to craft a meaningful user study that shows significant differences between two systems. [2] For that reason, we will also show how the TREC relevance judgments can be used in a batch-oriented way to find effective interactive search

[2] User studies can also be used to study the process by which searchers find information, and not just to decide whether a system is effective. When we disparage the results of interactive systems, we refer only to the issue of effectiveness and not to whether interesting data about user processes were discovered.

strategies that can be integrated into a system. We are using TREC data in different ways to achieve the same goal: improving the effectiveness of interactive systems.

More specifically, we make two efforts to address the question of system value. First, we construct a system (AspInQuery) designed to help a user with a specific retrieval task. We evaluate the effectiveness of combining a ranked list with clustering by monitoring users on two versions of the system: one with and one without the visualization. We will show that although there is some indication that the visualization helps the user, the results are not statistically significant, and no solid conclusion can be drawn.

In the second part of this study, we build a new system (Lighthouse) that adds an inter-document similarity visualization to the ranked list. In this portion, we use the TREC collections and relevance judgments to evaluate the ability of a system to use the information in the visualization to improve the ordering of documents in a ranked list. We show that this re-ranking operates much like interactive relevance feedback, and that it affords substantial improvement in retrieval effectiveness. Because this portion of the study is a laboratory evaluation and not a user study, it is possible to process substantially more queries and draw statistically significant conclusions.

This paper proceeds as follows. In the next section, we will discuss the component techniques used in our systems. In Section 3 we will discuss the user evaluation that was done with AspInQuery as part of TREC. Section 4 describes the use of TREC corpora to evaluate the Lighthouse system intended for interactive use. We discuss related and prior work in Section 5, future directions for this work in Section 6, and then draw overall conclusions in Section 7.

2. Retrieval components

The work in this study used two systems for its research. The first, called AspInQuery, was a system specifically designed to address instance-oriented retrieval (explained below). That system included a ranked list, a visualization of inter-document similarity, and a "smart piece of paper" called an aspect window for recording instance information. The second system, called Lighthouse, is targeted toward more general purpose IR searching. It includes the ranked list and visualization only. This section discusses the method of generating ranked lists, how inter-document similarity is visualized, and what an aspect window is.

2.1. Ranked lists

Document ranking in both systems is done using InQuery (Allan et al., 1998). Although InQuery allows a query to be specified using a variety of combination of evidence operators, in this work queries are always a list of words, each of which contributes equally to the belief that a document is relevant to an information need. To calculate the belief that a *document* is relevant, a belief is determined for each *word* in the query and then those beliefs are averaged. The weight of the ith term in the vocabulary, w_i, is computed using the InQuery weighting formula, which uses

Okapi's *tf* score (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1995) and InQuery's normalized *idf* score:

$$w_i = 0.4 + 0.6 \cdot \frac{tf}{tf + 0.5 + 1.5(doclen/avgdoclen)} \cdot \frac{\log((N + 0.5)/docf)}{\log(N + 1)}. \tag{1}$$

Here, *tf* is the number of times the term occurs in the document, *docf* the number of documents the term occurs in, *doclen* the number of terms in the document, *avgdoclen* the average number of terms per document in the collection, and N is the number of documents in the collection.

Every document in the collection is assigned a score in this way. Documents are then presented in descending order of belief score.

2.2. Inter-document similarity

To determine inter-document similarity, we use a vector-space approach where each document is represented by a vector of term weights V. Here, the weight of an individual term, v_i, is the same as w_i without the 0.4 and 0.6 scaling (i.e., $w_i = 0.4 + 0.6 \cdot v_i$), making the weighting more consistent with other vector space systems. The *dis*similarity between a pair of documents is measured by the secant of the angle between the corresponding vectors (i.e., $1/\cos\theta$). That value is the inverted measure of similarity between documents that is widely used in the vector-space model (Salton, 1989).

Given the dissimilarity (or similarity) between documents, a clustering system would choose which documents to group. In our systems, however, we merely wish to portray the relations between the documents. Note that the document vectors exist in a high-dimensional vector space (there is one dimension for every unique word in the corpus). In general, we cannot accurately portray *n*-dimensional objects in two or three dimensions.

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 work we have selected an approach called spring-embedding. Our choice was motivated by the graph-drawing heritage of 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 document vectors as objects in two or three-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. Fig. 1 gives an example of 50 documents "spring-embedded" in two and three dimensions.

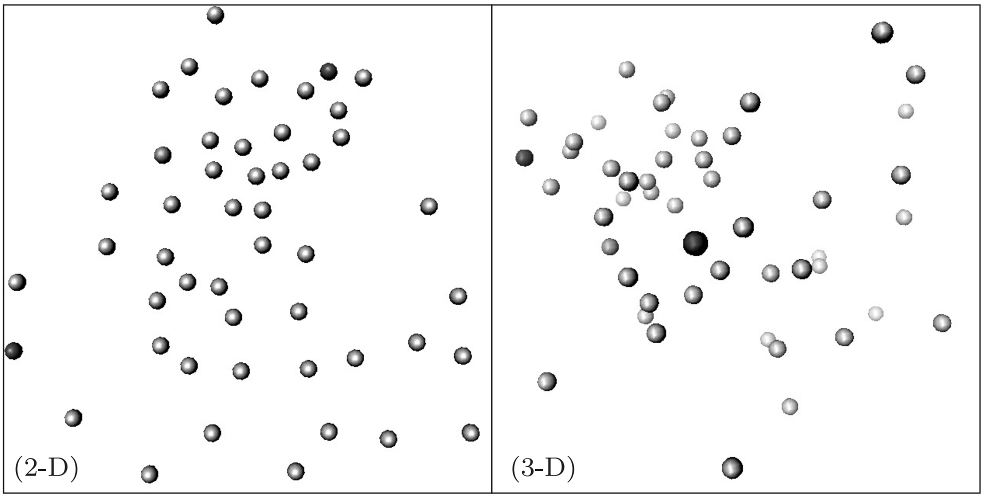


Fig. 1. A set of 50 documents visualized in both two and three dimensions. The visualization is created by spring-embedding the 50 documents based upon their similarities in a high-dimensional space.

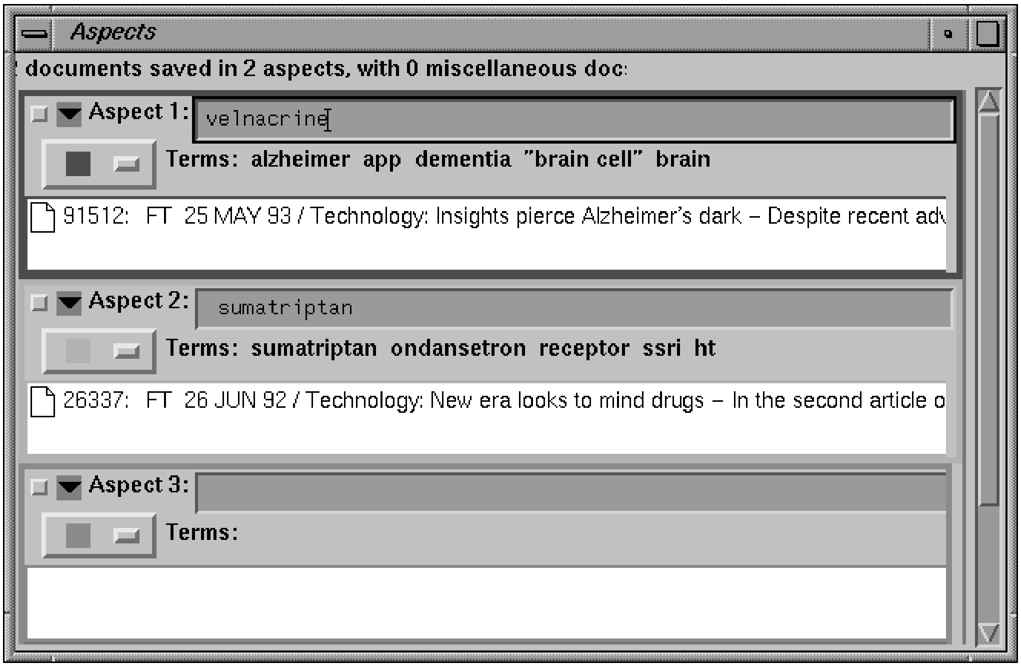
2.3. Aspect window

The aspect window portion of our system (used only in AspInQuery) addresses a task-specific problem: given a query, find documents that cover as many instances (aspects) of relevance as possible.[3] For example, in a query about ferry sinkings in the news, the task was to find a list of all ferries that sank, not to find all documents about ferry sinkings. (Section 3.1.1 details how this task was defined in our work.)

With a basic IR system, an analyst may be able to find the documents containing various instances, but he or she has to use another window or a piece of paper to keep track of what has been found already. We implemented an "aspect window" tool to help with this task. The idea is to provide an area where documents on a particular instance can be stored. To help label the information, statistical analysis of word and phrase occurrences is used to decide what terms and phrases are most distinctive about a document or set of documents in an instance. In addition, we provided an area for the user to manually assign additional keywords or labels if needed.

Each area of the aspect window has a colored border, a text field at the top for entering a descriptive label, and an automatically generated list of the five noun phrases that most distinguish the group of documents assigned to this instance from the remainder of the collection. Fig. 2 shows an example of the aspect window. The system shows two groups of documents (two instances) already identified and a third area waiting for the next instance. The first instance contains one document that the user entered into the instance by dragging from the ranked list display

[3] In TREC-6, the task was referred to as "aspect retrieval". Since then, the terminology has shifted to refer to *instances* of relevance rather than *aspects*. That shift explains the name of the system – "AspInQuery" – as well as the name of its window.


Aspects

documents saved in 2 aspects, with 0 miscellaneous doc

Aspect 1: velnacrine

Terms: alzheimer app dementia "brain cell" brain

91512: FT 25 MAY 93 / Technology: Insights pierce Alzheimer's dark – Despite recent adv

Aspect 2: sumatriptan

Terms: sumatriptan ondansetron receptor ssri ht

26337: FT 26 JUN 92 / Technology: New era looks to mind drugs – In the second article o

Aspect 3:

Terms:

Fig. 2. The Aspect window.

into the instance's document list. The system then analyzed the selected document and found five phrases that describe the instance; the analyst manually added "velnacrine".

Although the aspect window is an integral part of AspInQuery, and is critical in the interactive evaluation discussed next, we did not evaluate its effectiveness directly. The focus of this work is on the other two components of our systems: ranked lists and visualizations of inter-document similarity.

3. Interactive evaluation

In this major part of the study we use the TREC relevance judgments and environment to evaluate the effectiveness of a three-dimensional (3-D) visualization of inter-document similarity. This work was driven by the TREC-6 Interactive Track (Voorhees & Harman, 1998; Swan & Allan, 1998), an evaluation of "instance oriented information retrieval", wherein users are tasked with identifying documents relevant to as many "instances" of a query as they can. The structure of our experiments was determined to a large extent by the TREC-6 guidelines.

In this task, the final output of the searches is not just a list of relevant documents, but is instead a list of relevant instances, where a document may contain information about multiple instances, and several documents may relate to the same instance – the output is an overlapping non-hierarchical clustering of relevant documents, and the interesting information is the clusters, not the documents. While the task design did not specifically call for the saved documents to be clustered in that way (it only required retrieving documents covering as many instances as possible), we felt that building a system that assisted in that clustering was valuable. For that purpose, we built the aspect window (discussed previously).

Because of our interests in task-targeted interactive systems, we chose to build and evaluate a system that was designed specifically to aid a user with instance retrieval. We constructed the AspInQuery system for the instance retrieval task without any visualization. A second version, called AspInQuery Plus below, includes the visualization. Our goal was to compare user performance with and without the visualization so that we could conclude whether or not it was helpful.

We were interested in the 3-D visualization because we hypothesized that it might help identify documents discussing the same instance – that is, documents about the same instance would be likely to be highly similar. If true, the 3-D display would provide the user with information about whether a document is worth investigating further, helping the user to sort through documents more quickly. In AspInQuery Plus, documents in the 3-D window are persistent between multiple queries issued for the same task: when new documents are retrieved they are colored light blue (light purple when read) and are placed in the 3-D window by the forces exerted from already placed documents. An analyst who is under time pressure could use the 3-D display to decide that the unjudged document near that instance is probably on the same instance and so not worth examining. A retrieved document that is far from all already-marked instances is more likely to be useful.

We also used a NIST-supplied control system called ZPRISE. The general design of the TREC-6 Interactive Track called for all systems to be compared via this common control, in the hope that it would be possible to compare all participating systems without doing pairwise comparisons. The ZPRISE system was a simple search engine without any visualization of inter-document similarity and without an aspect window.

3.1. User study evaluation

Our goal in the user study was to determine whether AspInQuery Plus provided any advantage over AspInQuery – that is, whether the 3-D visualization was helpful. We had a total of 24 participants in our user study. They were equally divided between librarians (12 experienced searchers, all with an MLS degree, 11 of them female) and a more general user population (12 inexperienced searchers, a range of education but mostly undergraduate students, 5 of them female).

The basic unit for the TREC-6 experimental design was a block, each block having four users. Each user ran six topics, three with the experimental system, and three with the control system. Two of the four users did the first three searches with the experimental system, and the other two users did the first three searches with the control system. Topic order was held constant. This Latin square design allows blocking on both topics and users, and the average of the diagonals gives an estimate of system-specific differences. All groups participating in the TREC-6 Interactive Track used this experimental design, which is described in greater detail by Lagergren and Over (1998). We ran three groups, each composed of two blocks, one block of general users and one block of librarians. This design allowed us to block on experienced/novice users in our assessment of the systems. Group (1) used AspInQuery and ZPRISE while Group (2) used AspInQuery Plus and ZPRISE. Those groups allowed comparison via the control system (ZPRISE) and represented our official participation in TREC-6. In theory, we could compare AspInQuery to AspInQuery Plus via the control. After TREC-6, we ran the study with an additional set of users: Group (3) used AspInQuery and AspInQuery Plus, allowing those systems to be compared *directly*. The breakdown into three groups allowed us to participate in the TREC-6 Interactive Track to evaluate system effectiveness, but also allowed us to examine the effect of AspInQuery Plus more accurately.

Before the searches, each participant filled out a questionnaire to determine age, education, gender and computer experience, and two psychometric tests (Ekstrom, French, Harman, & Dermen, 1976), a test of verbal fluency (Controlled Associations, test FA-1) and a test for structural visualization (Paper Folding, test VZ-2). We gave each participant a piece of scratch paper before each search, and a short questionnaire after each. Each search had a 20 min time limit, and the participant was instructed to stop the search if they had not finished in 20 min. After all the searches were finished the participant was given a final questionnaire, and then "debriefed". The study was conducted single blind: the participants were not told until the debriefing which system was the control and which was the experimental system.

3.1.1. Data set and measures

The corpus used was newspaper articles from the Financial Times, 1991–1994, an approximately 200,000 article subset of the TREC collection. Six topics were selected by NIST from previous TREC experiments. The documents marked relevant by users were sent to NIST where they were combined with the saved documents from other sites participating in the Interactive Track. The assessors read the documents and developed a list of instances for each topic (e.g., the list of different ferries that sank) and a mapping between each saved document and the instance(s) covered, if any. From this, scores of instance precision and instance recall were obtained for each run. Instance precision is the proportion of the saved documents that contained at least one instance. Instance recall is the proportion of identified instances that are covered by the saved documents. Note that instance-oriented IR does not entail finding all the documents that mention a topic, as normal IR does, but is instead concerned with finding a set of documents that contains all the relevant information about the topic represented in the corpus.

The resulting TREC judgments allow us to evaluate the effectiveness of our interactive systems, so that we can compare them to see which is more effective. If users consistently do better with one system than another, we can conclude that one is more effective. Ideally, the instance judgments will allow repeated trials of numerous systems, so that we can converge upon the best possible result.

3.2. Results

Our participation in the TREC-6 Interactive Track (Swan & Allan, 1998) was intended to address several questions. In this study we only consider issues related to use of the 3-D window. One of those hypotheses was that better spatial reasoning ability would make it more likely that users would use the 3-D window. However, we found no significant evidence to support that idea. The best predictor of who will use an interface element was prior experience with similar elements, not spatial ability.

3.2.1. Indirect comparison

To determine the effectiveness of the two experimental systems we performed ANOVA on group 1 (AspInQuery and ZPRISE) and again on group 2 (AspInQuery Plus and ZPRISE). We treated topic, searcher, and system as factors and precision, recall, and time as dependent variables. We performed the ANOVA with all interactions and found no significant interactions, so we used a main effects model. Topic was the most significant predictor of recall, precision, and time taken. This is not surprising as it is well known that topic difficulty has a strong influence on IR results. Fortunately the topic effects were quite consistent, and the Latin squares design allowed it to be subtracted out. Without blocking on topics, topic effects would have hidden smaller effects.

User differences were the next most important factor after topic differences. Instance recall and elapsed time were both heavily influenced by the searcher. Once again blocking on individual differences is required in order to find system level differences.

System effects were smaller than either topic or user effects, affecting fewer dependent variables and showing far less significance for the variables affected. Notable system differences are reported in the first two columns of Table 1. Specifically, that ZPRISE outperformed AspInQuery whereas AspInQuery Plus outperformed ZPRISE.

The design of the TREC experiment was intended to allow comparisons between different systems by comparing those systems with a common control. We designed our two systems to be identical except for the presence of an additional window in AspInQuery Plus. We felt that if there were a strong difference in effectiveness between the two systems we would know that it was caused by the additional window. If use of a common control allows us to accurately measure system caused differences, we can combine the data for the two groups and perform ANOVA. Significance testing using Tukey's Studentized Range Test shows a difference between AspInQuery Plus and AspInQuery at the 0.03 confidence level, with a ranking of the systems AspInQuery Plus is better than ZPRISE which is better than AspInQuery, and with AspInQuery Plus outperforming AspInQuery in average recall by 0.15, equivalent to finding an additional three instances out of 20. Since the 3-D window was intended as a recall enhancing device we were encouraged by this result.

We appeared to have shown that the 3-D visualization is helpful. However, the comparison was made via a third control system. How reliable is that indirect comparison?

3.2.2. Direct comparison

In order to confirm our result, and to verify the assumption that different systems could be indirectly compared by comparing them with a common control, we compared the two systems

Table 1
Pairwise comparison of three systems, measured by instance recall[a]

	Ranking	Difference	Significance
AI − ZPRISE	ZP > AI	0.0867	$P < 0.04$
AI + −ZPRISE	AI+ > ZP	0.0616	$P = 0.06$
AI − AI+ (via control)	AI+ > AI	0.1500	$P < 0.03$

[a] The "Difference" column reflects the average difference in the instance recall scores for the better system over the worse. The comparison of AspInQuery and AspInQuery Plus is indirect via the control system, ZPRISE.

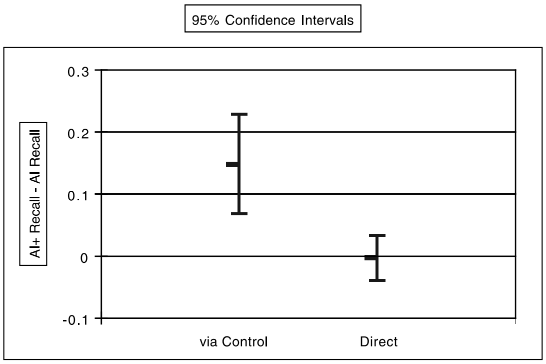


Fig. 3. 95% Confidence intervals for difference between AspInQuery Plus and AspInQuery recall.

directly in group 3. The ANOVA comparison showed no difference between the two systems in effectiveness. Fig. 3 shows the 95% confidence intervals for the difference in mean recall as determined by comparison through the control, and direct comparison. There is no overlap between the two confidence intervals. The indirect comparison (as conceived of for TREC-6) was not a reliable method for identifying which systems were more effective.

System level differences are small compared to differences caused by topics or by users. Ideally, to measure system effects we should hold both topic and user constant across tests. Due to learning effects this is not feasible, and users cannot run the same topic more than once. New users are required for each test, and differences between the sets of users can affect the results. The design of the Interactive Track experiment calls for the use of a common control, the same six topics presented in the same order, and a common Latin Square design to allow indirect comparisons of systems between sites (Lagergren & Over, 1998). However, the design only requires four users per system. Small sample sizes can affect experiments in several ways. The most obvious and expected is a reduction in the power of the test – large differences between systems are required in order to obtain statistically significant results. Another problem that can occur with small sample sizes, especially with human subjects, is the possibility of getting highly coherent samples of subjects that are not representative of the population as a whole.

3.2.3. Interference from traits

We recruited all our groups the same way, and balanced the distribution of experienced and novice users, but we made no attempt to balance the groups on other traits. We analyzed the characteristics recorded for the different groups of users to see if there were any traits where the groups differed radically. Fig. 4 shows the score for Spatial Ability (VZ-2) for the three groups. This distribution of VZ-2 scores for groups one and two has a t-value of 3.707 ($P < 0.01$).

Fig. 5 shows the difference in mean instance recall between the experimental systems and ZPRISE, plotted against VZ-2 (instance recall scores were normalized for each topic to have zero mean and unit variance to remove topic effects). Only two of the eight members of group 1 did better with the experimental system, and only one member of group 2 did worse. Also, only one member of group 1 scored above 11 on VZ-2, and only one member of group 2 scored below that.

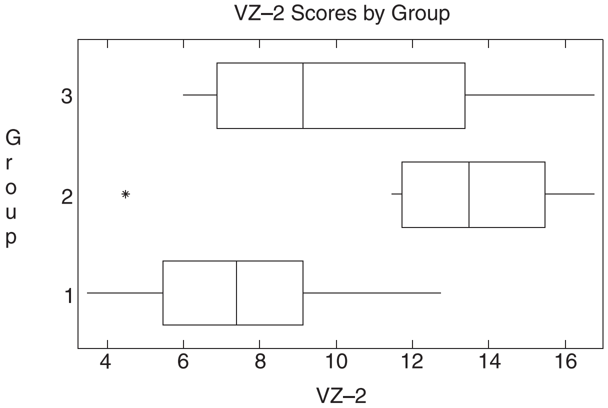


Fig. 4. Distribution of spatial ability across groups. Group 3 (at the top) compared AspInQuery to AspInQuery Plus; group 2 compared AspInQuery Plus to ZPRISE; and group 1 (bottom) compared AspInQuery to ZPRISE.

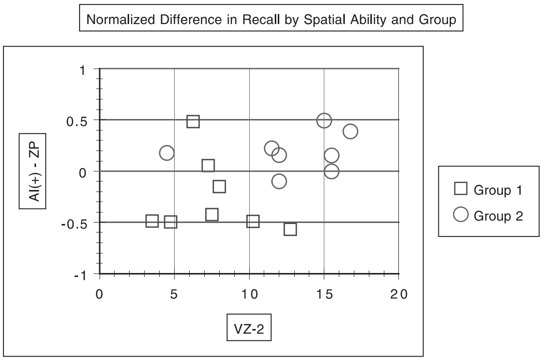


Fig. 5. Difference in experimental/control instance recall and spatial ability. Group 2 compared AspInQuery Plus to ZPRISE; and group 1 (bottom) compared AspInQuery to ZPRISE.

The difference between systems correlates with score on VZ-2, though not as strongly as it does with group number, with users scoring below 11 doing better with the experimental system 3 out of 8 times, as opposed to users scoring above 11 doing better 5 out of 7 times (with one user doing equally well with both systems).

Not only are there large differences *between* the two groups in VZ-2 score, there are also very small differences *within* each group. A likely explanation for the different results of the two comparisons (via control vs. direct) is that our two systems are essentially identical and *both* of our systems require high visual skills to be effective. The difference in response that we saw is caused by the large difference in average spatial skills between groups, but the differences within groups are too small for the interaction effects to be noticeable.

We had expected to find that users with high spatial skills would find the system with the 3-D window more usable, but we had not expected that result for the basic system. In hindsight, we note that the basic system required the use of “drag and drop” to save documents, and explicitly used a spatial metaphor, where relevant documents had to be dragged to a different window to be saved. This metaphor may be awkward or counterintuitive for users who do not have strong spatial skills. Alternatively, there may be another trait that happens to cluster with spatial ability that explains the difference.

3.3. Summary and implications

In our TREC-6 Interactive Track work, we explored the value of a 3-D visualization of document relatedness. We did that by comparing a system with and without the visualization. Because the evaluation was done in the text of the TREC Interactive Track, we did a comparison of the two systems indirectly via a control system (ZPRISE). This approach showed an advantage to using the visualization.

We also extended the TREC Interactive Track to include a direct comparison of our two systems. In that case, we found that there was no significant advantage to using the visualization. Instead, we found evidence to suggest that existing spatial reasoning ability accounted for the difference in user performance in the indirect comparison.

We were puzzled by these results because we had seen numerous examples during development where the visualization clearly provided useful information. Unfortunately, the user study model of the TREC-6 Interactive Track does not appear to be a good vehicle for exploring this question. The cross-system effects are small and difficult to tease out when combined with topic and searcher effects. For that reason, we decided to try more batch-oriented evaluation of a system that combines the ranked list and visualization. Our hope is that we can find a different way of using TREC data to evaluate interactive systems.

4. Off-line evaluation

Our goal is to find effective ways for searchers to use a visualization of inter-document similarities (such as the spring embedded portrayal). Our experience indicates such an interface has value, but we hypothesize that the interactive user study described above (like others Veerasamy & Belkin, 1996; Lagergren & Over, 1998) did not have sufficient power to detect the difference because the improvements are obscured by other variations between users and systems. That is, the huge number of possible differences between systems, compounded by variation between users and topics, makes it extremely difficult to isolate effectiveness improvements.

In this, the second major portion of this study, we show how TREC relevance judgments can be used in a batch setting to evaluate the effectiveness of combining two result presentation techniques. Specifically, we will describe an underlying algorithm used by the Lighthouse system and evaluate its effectiveness at finding relevant documents. Lighthouse accepts a user's query and presents the top ranked documents (usually 50) from a selected collection or the Web. In addition, Lighthouse calculates inter-document similarities and visualizes their relationships using the spring embedding algorithm described in Section 2.2. We will show that if the user uses the ranked list to find some relevant material, and then switches to the visualization, he or she will find the relevant documents faster than using either the list alone, with or without relevance feedback. The approach is described in more detail below.

An important aspect of this work is that the recommended algorithm that a user should follow was developed through batch experiments using TREC relevance judgments. By training and testing the system using the judgments as if they were entered in an interactive setting, we were able to build and tune a system that provides demonstrably increased retrieval effectiveness. This same process would have required hundreds of users in numerous expensive studies if it had been carried out more traditionally.

4.1. Merging ranks and relatedness

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 feedback to adjust the retrieval. We can adopt that notion while looking at ordering of retrieval results by combining the ranked list and visualization.

For example, when using a simple ranked list, the documents are ordered by probability of being relevant, the user starts at the top of the ranked list and proceeds down the list examining the documents one-by-one. In that case, any feedback information is ignored.

An interactive relevance feedback approach defines another document ordering, though it starts similarly: the documents are ordered by probability of being relevant, and the user begins at the top and examines the documents until the first relevant document is found. That document is used to modify the query, the unexamined documents are reordered by a new probability of being relevant to the new query, and the process continues. Here, the user feedback is combined with the ranked list to create an improved system.

Ultimately the way a system is used to find documents defines an ordering of the documents – a ranking. Because the word *ranking* is closely associated with the original ranked list, we will call this process and the resulting order in which documents are supposed to be examined a *search strategy*. 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”.

A 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 (Voorhees & Harman, 1998). In the remainder of this section, we describe and evaluate a strategy that combines a ranked list and a visualization. We will then evaluate its benefit over simpler approaches. Our goal is to determine whether there is value in the combination.

4.2. The Lighthouse strategy

Consider the following interactive search strategy. At the beginning, we 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 then switch to the inter-document similarity visualization. From that point forward, the searcher is presented with the unexamined document that is closest to the average known-relevant document. Fig. 6 illustrates how the algorithm moves through and selects from the set of documents. There the

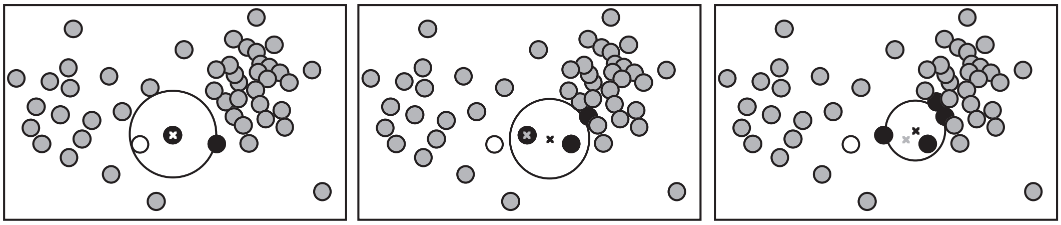


Fig. 6. Three consecutive snapshots of the cluster-growing phase of our search strategy. We start from the document with an "X" inside and look for the rest of relevant documents. We show the state as the first, second, and third relevant documents are discovered. The white disks represent the known non-relevant documents, the black disks are the known relevant, and the gray disks are the unknown documents.

document vectors are shown as disks: the black disks indicate relevant documents, the white are non-relevant, and the gray are unexamined documents. The first known relevant 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), so 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 an 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. Our system that uses this approach is called Lighthouse.

In the next section, we use the TREC relevance judgments to explore whether the described strategy is effective in comparison to other approaches. We also consider whether it helps to choose the next document based on proximity in 2-D, 3-D, or in the full number of dimensions needed to represent the vectors (D-D).

4.3. Experiments

For our experiments we used TREC ad-hoc queries with their corresponding collections and the relevance judgments supplied by NIST accessors (Voorhees & Harman, 1997). Specifically, TREC topics 251–300 and 301–350 were converted into queries and ran against the documents in TREC volumes 2 and 4 and against the documents in TREC volumes 4 and 5 correspondingly. For each TREC topic we have created four types of queries that varied in size and complexity: (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. That way we had eight different data sets, each containing the results of 50 queries.

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 10 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.

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 (by looking at the TREC judgments), 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 10 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 re-ranked using the new query and we continue down the list.

4.4. Results

We measured performance of the ranked list, relevance feedback and the combination of the ranked list and clustering (Lighthouse) on the 50 highest ranked documents for 100 queries of four different types. We 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 has zero or one relevant document will have precision of zero. The first two columns in Table 2 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 our performance results.

Table 2
Performance of the search strategies for the ranked list (RL), interactive relevance feedback (RF), and Lighthouse, the combination of the ranked and clustering (RL + CL)[a]

Total average	RL			RF	RL + CL (Lighthouse)		
	Random	Best	Original		D-D	2-D	3-D
	24.5	83.0	39.1	46.0	48.4	46.7	48.0
Improvement over RL				(17.6%)	(23.3%)	(19.0%)	(22.3%)
Improvement over RF					(4.8%)	(1.1%)	(4.0%)
Improvement over D-D						(−4.0%)	(−1.0%)
improvement over 2-D							(2.8%)

[a] 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 Lighthouse combination are considered: one in the original document vector space (D-D), another in 2-dimensional space (2-D), and another in 3-dimensional space (3-D).

4.4.1. Combination vs. ranked list alone

Table 2 shows that our Lighthouse search strategy outperforms the ranked list. We observed a 23.3% increase in precision across different query complexities and sizes. The difference is statistically significant by two-tailed t-test with $P < 0.05$. Our search strategy also outperforms the traditional relevance feedback approach by a small margin (4.8%). However, that difference is not generally statistically significant.

This result means that we have demonstrated there is value in the inter-document similarity information. In the sense that the visualization is derived from those similarities, and that it directly reflects the way we are using those similarities, we have also shown value in the visualization. It is possible to imagine Lighthouse being used without the visualization. For example, it might highlight the next document in a ranked list that should be considered, or it might reorder the list after every judgment (though Aalbersberg, 1992 claims that such abrupt changes in the document order are undesirable and should be avoided). However, users prefer systems that expose more of the reason for their actions (Koenemann & Belkin, 1996), perhaps because it gives them a greater sense of control (Shneiderman, Byrd, & Croft, 1997), suggesting value in visualization that guides the user to the next choice as well as making it visually apparent why that choice is reasonable.

4.4.2. Effect of fewer 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 it is not a metric: the triangle inequality is not always satisfied for the $1/\cos$ distance function.

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 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-D visualization will exhibit better performance than in 2-D and the search strategy in the original document-vector space (D-D) will be the best.

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 $P < 0.05$.

The modest loss in effectiveness because of spring-embedding into 2- or 3-D means that the visualization can be used to select the "next" document. Though we obtain slightly better performance by making the choice in D-D, we sacrifice some "explainability" because of the inaccurate mapping into fewer dimensions. There appears to be no difference between 2-D and 3-D, so that choice is primarily a user opinion.

4.4.3. Users' performance

We have argued that the reason for including the visualization with the ranked list (rather than just suggesting the next document) is that it provides more information to a searcher. We wondered whether choosing the document "closest to known relevant documents" was something that users could do from the visualization without system help.

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.

We carried out a user study with 20 people (Leuski & Allan, 2000b) and showed that spatial proximity is an intuitive and well-recognized (by the users) metaphor for similarity between objects. The users were presented with 50 spheres floating in space. Most of the spheres were painted in white, there always was one green sphere and possibly several red ones. The users were told that all spheres are either green or red, and that the white spheres are hiding their true color which can be revealed by double-clicking the sphere with the mouse. The users were also told that the spheres of similar color (e.g., green spheres) *tend* to clump together. We asked the users to locate all green spheres as quickly as possible and to try to avoid finding the red spheres. Each user had to solve 10 of these problems – five in 2 dimensions and five in 3 dimensions.

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 (see Table 3). We also observed that our algorithmic approach outperforms the users. The precision difference between the users and the algorithm arises from difficulty in precisely identifying the inter-object distances in the visualization. We also observed that the higher accuracy with which 3-D structure represents the document set is negated by the heavy cognitive effort that is required from the users to navigate the visualization (Sebrechts, Cugini, Laskowski, Vasilakis, & Miller, 1999).

4.5. Implications

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 (with some recent exceptions (Allan, 1996)) 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.

Table 3
Users' performance navigating the visualizations of 10 randomly selected document sets[a]

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

[a] 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 3-D over 2-D are shown. We also show the significance level for each difference by two-tailed *t*-test.

The visualization that we use to that effect is a 2- or 3-D 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, 3-D approximation is better than 2-D since it retains one extra degree of freedom to position the documents. On the other hand, our user experiments have suggested that the extra dimension is of no value to the users, because of additional cognitive overhead.

5. Prior and related work

Both of the studies described in this paper were possible because of the existing TREC relevance information (Voorhees & Harman, 1998). In this paper we have talked about how that information can be used in two different ways to evaluate interactive systems.

Portions of the work in this paper have been reported on elsewhere. Preliminary analysis of the TREC-6 interactive study was discussed in the TREC-6 proceedings (Allan et al., 1998). The same work has also been discussed much more broadly and with substantial focus on statistical significance (Swan & Allan, 1998). This paper has provided a different view of the work, considering only the portions related to 3-D interfaces, evaluating that interface with TREC judgments, and using it as a launching point for the Lighthouse research.

As our work on evaluating interactive information retrieval systems developed, we reported intermediate results in various venues. Our early efforts (Leuski & Allan, 1998, 2000a) explored the question of how TREC relevance judgments can be used to create meaningful evaluations of interactive systems. We showed that they could be combined with a simulated "search strategy" to find effective information browsing approaches. We presented an in-depth discussion of a browsing strategy (Leuski & Allan, 2000b) that motivated design choices for the Lighthouse system described in this paper. In this paper, we have continued and elaborated upon the analysis in that earlier work. In the next section, we will propose how the ideas can be extended to instance retrieval.

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 (Voorhees & Harman, 1997, 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 re-rank 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.

It is common for 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 (1995, 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-D 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 (the same algorithm is incorporated into the systems in this paper) for placing high-dimensional objects in a low-dimensional space. The system puts documents in 3-D 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. We adopted a similar approach and applied it to complete, full-sized documents in both of our systems. Additional details of our earlier work are reported elsewhere (Leuski & Allan, 1998, 2000a; Swan & Allan, 1998).

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. 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, Claritech investigated the effectiveness of clustering presentation against traditional ranked list approach for relevance feedback (Evans, Huettner, Tong, Jansen, & Bennett, 1999). Their result is somewhat inconclusive – the observed advantage of the clustering presentation was small and not statistically significant.

6. Future work

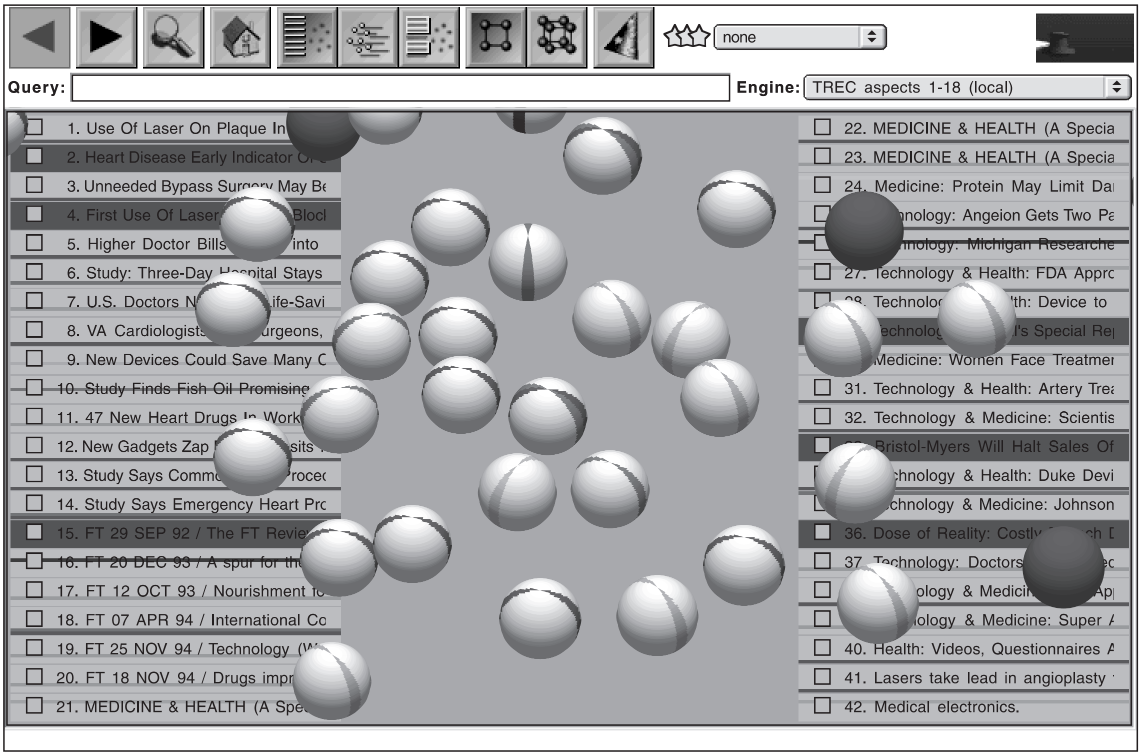
Our work on using 2- and 3-D visualizations of document similarity has proceeded through two stages so far. The first, AspInQuery, was only a marginal success, in that it was very task-specific and did not result in intuitive conclusions. The Lighthouse effort, however, is demonstrating results that match intuition and allowing us to better understand the way ranked lists and inter-document similarities relate. We are interested in moving Lighthouse forward in numerous ways.

For example, we 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 completely 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), so the query–document will appear as a sphere that is 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.

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, showing them as new objects "flying" into the visualization similar to the AspInQuery Plus. We believe that the batch-oriented evaluation 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).

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 does not 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 non-relevant 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.

A problem that is strongly related to the "multi-topic" notion of relevance is that of instance retrieval, the fundamental task of the user study discussed in Section 3. We have been exploring ways to enhance Lighthouse so that it includes instance information. For example, Fig. 7 shows a sample screen from Lighthouse where instances are represented as "pie slices" on the spheres. Lighthouse uses color and direction of the slice to discriminate between instances. The left side of the visualization is dominated by the spheres with a dark horizontal slice (slice in the position of "3 o'clock"). Note two "2 o'clock" sliced spheres in the middle of the cluster. The right side of the visualization is composed of two overlapping clusters of spheres with diagonal light-shaded slices: one with slices that go from top left to bottom right ("5 o'clock") and the other with slices from bottom left to top right ("1 o'clock"). The dark solid shaded spheres represent non-relevant documents that do not contain any instances. Note that the sphere in the center of the visualization has two slices: a horizontal one as its neighbor on the left ("3 o'clock") and the next clockwise slice as its neighbor on the right ("4 o'clock"). It remains an open problem how best to represent instances in this type of a system. Currently we are working on extending the search strategy evaluation to the task of instance-retrieval.



Query:

Engine: TREC aspects 1-18 (local)

□ 1. Use Of Laser On Plaque In
□ 2. Heart Disease Early Indicator
□ 3. Unneeded Bypass Surgery May Be
□ 4. First Use Of Laser Block
□ 5. Higher Doctor Bills into
□ 6. Study: Three-Day Hospital Stays
□ 7. U.S. Doctors N Life-Savi
□ 8. VA Cardiologists urgeons,
□ 9. New Devices Could Save Many C
□ 10. Study Finds Fish Oil Promising
□ 11. 47 New Heart Drugs In Work
□ 12. New Gadgets Zap sits
□ 13. Study Says Comm Proce
□ 14. Study Says Emergency Heart Prc
□ 15. FT 29 SEP 92 / The FT Revie
□ 16. FT 20 DEC 93 / A spur for th
□ 17. FT 12 OCT 93 / Nourishment fo
□ 18. FT 07 APR 94 / International Co
□ 19. FT 25 NOV 94 / Technology (W
□ 20. FT 18 NOV 94 / Drugs impr
□ 21. MEDICINE & HEALTH (A Spe
□ 22. MEDICINE & HEALTH (A Specia
□ 23. MEDICINE & HEALTH (A Specia
□ 24. Medicine: Protein May Limit Da
□ nology: Angeion Gets Two Pa
□ nology: Michigan Researche
□ 27. Technology & Health: FDA Appro
□ 28. Technolog lth: Device to
echnolog l's Special Re
Medicine: Women Face Treatme
□ 31. Technology & Health: Artery Tre
□ 32. Technology & Medicine: Scientis
□ Bristol-Myers Will Halt Sales Of
chnology & Health: Duke Devi
chnology & Medicine: Johnson
□ 36. Dose of Reality: Costly ch D
□ 37. Technology: Doctors ec
□ ology & Medici Ap
□ ology & Medicine: Super A
□ 40. Health: Videos, Questionnaires A
□ 41. Lasers take lead in angioplasty
□ 42. Medical electronics.

Fig. 7. Lighthouse visualization that includes instances of relevance as "pie slices" on the spheres that correspond to 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 spheres. 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.

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.

7. Conclusions

We have evaluated the improvements in effectiveness that can be gained through the use of visualization (in 2- or 3-D) of document relatedness. We used two types of evaluation:

1. We carried out a user study as part of the TREC-6 Interactive Track, and compared a task-specific system with and without a 3-D document visualization. We found suggestions that the visualization might help, but nothing significant. Although larger studies might find an effect, the cross-user and cross-topic differences were so large that it is difficult to tease out the effects of the visualization.
2. We developed an approach that rearranges the order of presentation of documents, but does it without actually shifting their position on the display. This provides the user with the power of immediate updates to the ranked list without abrupt and potentially confusing changes in the display. We showed that it is possible to combine the ranked list and the visualization to achieve something as powerful as relevance feedback, in a way that may be easier for searchers to understand.

Visualizations of clustering and inter-document similarity have appeared numerous times in the literature, however none of them has ever been widely adopted. Our user study experience raised the possibility that no one uses the techniques because they do not help. Our batch studies, on the other hand, suggest that the problem is not whether the combination is useful, but how to use it.

The TREC corpora, judgments, workshops, and style of evaluation were all instrumental in our being able to explore the value of combining two techniques. Continued progress in this area is likely to require a balance between user studies and batch-oriented experiments. That is, batch studies might be used to discover search and browsing strategies that are more effective than those already known. Armed with that information, a system can be built to guide users toward those effective strategies. Where necessary, user studies could confirm that benefit. We hope that TREC will continue to provide the material necessary to move in that direction.

Acknowledgements

This material is based on work supported in part by the National Science Foundation under grant number IRI-9619117, and in part by the National Science Foundation, Library of Congress, and Department of Commerce under cooperative agreement 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, and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the opinions of the authors and do not necessarily reflect those of the sponsor. Co-author Russell Swan passed away between the time this paper was submitted and the time it was accepted. His work on the TREC interactive track is a major part of this article. Russell's insight, creativity, and openness taught his co-authors a great deal, and we are keenly aware of his absence.

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

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

Anick, P. G., & Vaithyanathan, S. (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., & Lingoes, J. (1987). *Multidimensional similarity structure analysis*. Berlin: Springer.

Chalmers, M., & Chitson, P. (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).

Ekstrom, R. B., French, J. W., Harman, H. H., & Dermen, D. (1976). *Manual for kit of factor-referenced cognitive tests*. Educational Testing Service, Princeton, NJ (tests used by permission of ETS).

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

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

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

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

Koenemann, J., & Belkin, N. J. (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).

Lagergren, E., & Over, P. (1998). Comparing interactive information retrieval systems across sites: The TREC-6 interactive track matrix experiment. In *Proceedings of ACM SIGIR*, Melbourne, Australia, ACM.

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

Leuski, A., & Allan, J. (2000a). Evaluating a visual navigation system for a digital library. *International Journal on Digital Libraries*, *3*(2), 170–184.

Leuski, A., & Allan, J. (2000b). Improving interactive retrieval by combining ranked lists and clustering. In *Proceedings of RIAO'2000* pp. 665–681.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. In D. K. Harman (Ed.), *Third text retrieval conference (TREC-3)*. NIST.

Salton, G. (1989). *Automatic text processing*. Reading, MA: Addison-Wesley.

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

Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., & Miller, M. S. (1999). Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of ACM SIGIR* (pp. 3–10).

Shneiderman, B., Byrd, D., & Croft, W. B. (1997). Clarifying search: A user-interface framework for text searches. *D-Lib Magazine*.

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

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Veerasamy, A., & Belkin, N. J. (1996). Evaluation of a tool for visualization of information retrieval results. In *Proceedings of ACM SIGIR* (pp. 85–92).

Voorhees, E. M., & Harman, D. K. (Eds.). (1997). *The fifth text retrieval conference (TREC-5)*. NIST.

Voorhees, E. M., & Harman, D. K. (Eds.). (1998). *The sixth text retrieval conference (TREC-6)*. NIST.

Xu, J., & Croft, W. B. (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).