Details of Lighthouse*

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

Lighthouse is an on-line interface for a Web-based information retrieval system. It integrates two known presentations of the retrieved results – the ranked list and clustering visualization – in a novel and effective way. We describe a working implementation of the system. It accepts queries from a user, collects the retrieved documents from the search engine, organizes and presents them to the user. It is relatively fast and efficient. We also describe some experiments showing that Lighthouse helps the user to locate relevant information much faster than it could be done with the ranked list and can significantly improve the retrieval effectiveness of a search engine.

1 Introduction

Locating interesting information on the World Wide Web is the main task of on-line search engines. Such an engine accepts a query from a user and responds with a list of documents or web pages that are considered to be relevant to the query. The pages are ranked by their likelihood of being relevant to the user's request: the highest ranked document is the most similar to the query, the second is slightly less similar, and so on. The majority of today's Web search engines (Google, Infoseek, etc.) follow this scenario, usually representing a document in the list as a title and a short paragraph description (snippet) extracted from the text of the page. The evaluation methods for this approach are well-developed and it has been well studied under multiple circumstances for several decades [11].

The ordering of documents in the ranked list is simple and intuitive. The user is expected to follow the list while examining the retrieved documents. In practice, browsing the ranked list is rather tedious and often unproductive. Anecdotal evidence suggest that users quite often stop and do not venture beyond the first screen of results or the top ten retrieved documents.

1.1 Alternatives to the Ranked List

Numerous studies suggest that document clustering (topic-based grouping of similar documents) is a better way of organizing the retrieval results. The use of clustering is based on the Cluster Hypothesis of Information Retrieval: "closely associated documents tend to be relevant to the same requests" [32, p.45]. It has been studied in the context of improving the search and browsing performance by pre-clustering the entire collection [33, 7, 6]. Croft [5] and more recently Hearst and Pedersen [12], showed that the Cluster Hypothesis holds in a retrieved set of documents. Their system breaks the retrieved results into a fixed number of document groups. Leuski and Croft [19] considered a similar approach, but instead of fixing the number of clusters, they set a threshold on the inter-document similarity. While these systems operate with full-text documents, MetaCrawler-STC [37] at the University of Washington places the web search

*This is an extended version of "Lighthouse: Showing the Way to Relevant Information" that is published in Proceedings of InfoVis 2000.

results into overlapping clusters based on the snippets returned by the engine. The Northern Light [22] and Dataware search systems [8] are two examples of on-line commercial systems that organize the search results into folders.

All these system generally leave open two questions: how to decide where to draw the border between clusters and how to express the relationship between objects in different clusters. So what we need is a system that does not require the hard decision, a system that visualizes the documents and leaves it to the user to isolate the clusters. Galaxies [34] computes word similarities and displays the documents as a universe of "docustars". In a space with a huge number of "docustars" it is not easy to select the object the user wants to explore.

The Vibe system [9] 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. The LyberWorld system [13] includes an implementation of Vibe, but presented in three dimensions.

The Bead system [4] uses a Multidimensional Scaling algorithm called spring-embedding for visualizing the document set. The system was designed to handle very small documents – bibliographic records represented by human-assigned keywords. The Bead research did not evaluate the system. The browsing system studied by Rodden [24] uses the same algorithm to organize small sets of images based on their similarity. They observed that such an organization is a more effective way to navigate the image collection when compared to a random arrangement of the same images.

Swan and Allan [31] considered a system with the ranked list and the spring-embedding visualization. Their system presented complete, full-sized documents and it was studied in the context of searching for multiple topics across several query runs. There was no exploration of how the ranked list and the visualization could be effectively used together. Leuski and Allan [16, 17] adopted a similar approach and applied it to locating the relevant information among the retrieved documents. They have attempted an off-line analysis simulating a user browsing the system.

Other approaches exist that attempt to visualize the document space based on the concept similarities using some form of neural network such as Kohonen's self organizing maps [21, 25, 34]. The Narcissus system [14] applied the spring-embedding algorithm to visualizing structures of pages and links on the World Wide Web. Song experimented with visualization of clusters for a bibliographic database [30].

The prevalence of Web search engines points at the importance of search and the value of the ranked list. Clustering efforts illustrate the value of using inter-documents relationships to group the collection for understanding. This study is an effort to combine both presentations – ranked list and clustering – but in a way that avoids the troublesome problems of hard decisions in clustering.

1.2 Lighthouse: Combining Ranked List and Visualization

We describe Lighthouse, a working interface system for a typical web search engine that tightly integrates the ranked list with a clustering visualization. The visualization presents the documents as spheres floating in space and positions them in proportion to their inter-document similarity. If two documents are very similar to each other, the corresponding spheres will be closely located and the spheres that are positioned far apart indicate a very different page content. Thus the visualization provides additional and very important information about the content of the retrieved set: while the ranked list shows how similar the documents are to the original query, the clustering visualization highlights how the documents relate to each other.

The example in Figure 1 clearly shows three separate groups of spheres. An untrained eye can easily draw the boundaries of these clusters. That picture is the result of visualizing the inter-document similarities without any threshold to make the groups explicit. At the same time we can see the significant distance between the red and green spheres and the non-relevant document represented by a red sphere in the right cluster is nevertheless more similar to one relevant document (a green sphere with a black border) than to the other (a simple green sphere).

A simple corollary of the Cluster Hypothesis is that if we find one relevant document, the rest of the relevant documents should be similar to it. With our clustering visualization it literally means that the

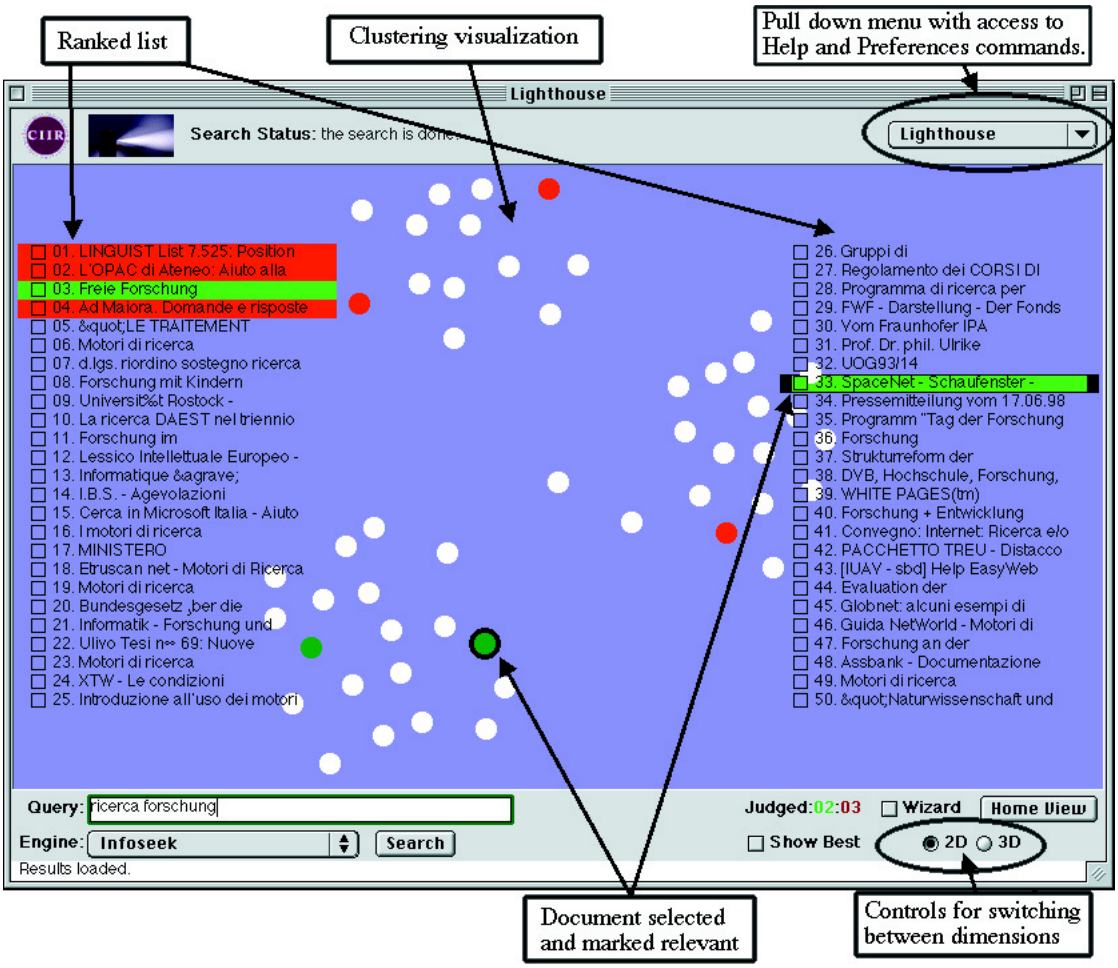

Figure 1: Screen shot of the Lighthouse system. The top fifty documents retrieved by the Infoseek search engine for a query. A two-dimensional picture is shown.

relevant documents tend to be in the neighborhood of the other relevant documents. Locating the interesting information should be as easy as examining the spheres that are close to the sphere of a known relevant document. We study how effective is the visualization in helping the user to locate the relevant documents.

We give a detailed overview of Lighthouse in the next section and follow it with a description of implementation aspects of the system. We also describe some experiments showing that the clustering visualization is indeed an effective way of locating the relevant information in the retrieved data. We conclude with our plans for future work.

2 System Overview

Lighthouse is an interface system for presenting results of a search session to the user. It combines a traditional ranked list with a clustering visualization. The clustering visualization presents each document as sphere and places the spheres in space, positioning them proportionally to the inter-document similarity. Figure 1 gives a screen shot of the system. Here the top fifty documents retrieved by the Infoseek search engine are presented as the ranked list of titles and fifty spheres corresponding to each page.

The ranked list is broken into two columns with 25 documents each on the left and on the right side of the screen with the clustering visualization in the middle. The list flows starting from top left corner down

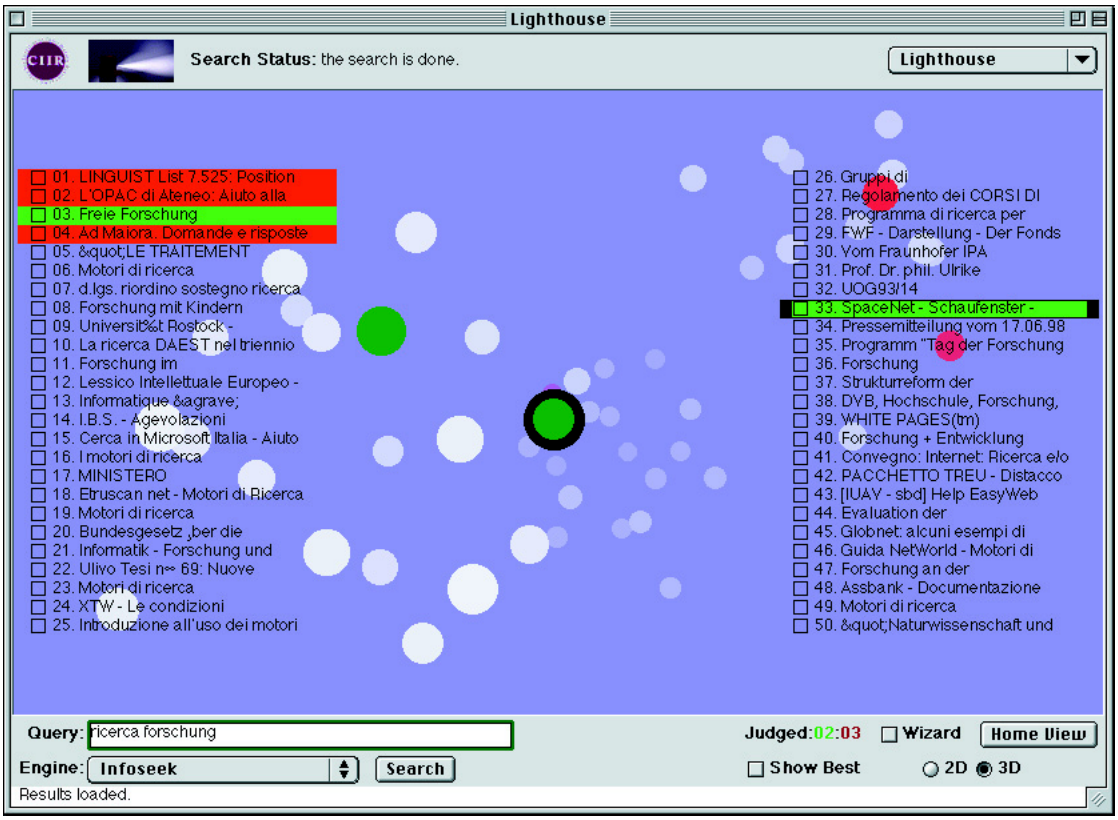

Figure 2: Screen shot of the Lighthouse system. The top fifty documents retrieved by the Infoseek search engine for a query. A three-dimensional picture is shown.

and again from the top right corner to the bottom of the window. The pages are ranked by the search engine in the order they are presumed to be relevant to the query. The rank number precedes each title in the list.

The clustering visualization, or the configuration of fifty spheres, is positioned between the two columns of titles. This organization makes the user focus on the visualization as the central part of the system. The spheres appear to be floating in space and the page titles are shown as placed on a transparent surface between the user's eye and the spheres. This is similar to how control data is projected on the windshield of a supersonic fighter plane or a race car. We believe that such an approach allows us to preserve some precious screen space and at the same time it stresses the integration of the ranked list and the visualization.

Each sphere in the visualization is linked to the corresponding document title in the ranked list so clicking on the sphere will select the title and vice versa. The user can examine the clustering structure and place it in the best viewing angle by rotating, zooming, and sliding the whole structure while dragging the mouse pointer. (Only the spheres can be manipulated in this fashion – the ranked list remains in place.)

The same set of spheres can appear as either a two dimensional (Figure 1) or three dimensional (Figure 2) structure. The user can switch the dimensionality on the fly by selecting the button in the bottom right corner of the screen (Figure 1). We achieve the effect of depth in the visualization by using perspective projection of the spheres – the remote spheres appear smaller than their front counterparts – together with the fog effect – the color of the remote spheres is closer to the background color than the color of the front spheres.

The similarity relationship among documents is rather complex and cannot be exactly reproduced by the clustering visualization. An additional dimension provides an extra degree of freedom, that in turn results in a more accurate representation of document relationships. Thus, a 3-dimensional picture should be more accurate and therefore more effective for the navigation than a 2-dimensional one. However, our observations

confirm a well-known fact that given a flat image, the users apply a significant cognitive effort to recreate a 3-dimensional structure in their minds [29]. The best results also require physical actions – it is much easier for the user to recognize and understand the proximity relationship among the spheres in the picture while slowly rotating the structure with the mouse pointer. We will show that these difficulties may eliminate all the advantages of the greater accuracy of the 3-dimensional visualization.

Because people differ in their ability to visualize the spatial structures, we give the user a freedom to chose the dimensionality of the presentation he or she is more comfortable with. From our own experience we found the ability to switch the dimensionality very rewarding: a 2-dimensional picture provides a great overview of the whole document set, but when a more precise analysis is required, e.g., when it is necessary to establish if two or more documents as close as they appear to be, the accuracy of the 3D picture can be more helpful. In this case we select the documents in question and switch the dimensionality to examine them. Sometimes this action reveals that three or four spheres separated in 2D appear clumped in 3D. For example, both Figure 1 and Figure 2 show the same configuration of documents. Consider two relevant documents, their green spheres appear closely placed in the central part of the 3-dimensional picture (Figure 2). The same two document spheres in 2 dimensions are separated by several other document spheres in lower left corner of the picture (Figure 1).

If the user points to a document title or a sphere with the mouse pointer while keeping a control key pressed, a small window similar to a comics balloon pops up showing the document description (Figure 3). The content of that window is the description paragraph (or snippet) returned by the search engine for the document. In addition a line connects the sphere and the title. This design preserves the screen space and keeps the snippet readily available to the user by a gesture with a mouse. The line literally links the two document representations – the title and the sphere – together. A double-click on the document title (or sphere) opens the document in the web browser.

The user's interest or the relevance assessment of the document is expressed by clicking on the checkbox attached to each document title. One click marks the document as non-relevant, the corresponding title and sphere are highlighted in red. A second click marks the document as relevant and both the sphere and the title show up in green. Another click removes the mark from the document.[1]

Given the ranking information obtained from the search engine, and the relevance judgments collected from the user, Lighthouse estimates the expected relevance values for the unjudged web documents and provides two different tools to convey that information to the user. Both tools operate in suggestion mode – they point the user to the most likely relevant documents without forcing their choices on him. Both tools can be switched on and off using the controls in the bottom right corner of the window (Figure 1).

Shade Wizard The first tool, the *Shade Wizard* (controlled by the "Wizard" button on the Lighthouse screen shot Figure 1), indicates the estimated relevance for all unjudged documents by means of color and shape. Specifically, if the system estimates the document is relevant, it highlights the corresponding sphere and title using some shade of green. The intensity of the shading is proportional to the strength of the system's belief in its estimation – the more likely the document is relevant, the brighter the color is. The same is true for estimated non-relevant documents – the more likely the document is non-relevant, the brighter is the red shade of the corresponding object on the screen. The same shade of color is used to highlight the document title backgrounds. Additionally, the length of that highlighted background is proportional to the strength of the system's belief in its estimate. The highlighted backgrounds in the left column are aligned on the left side and the highlighted backgrounds in the right column are aligned on the right side. Note that a white sphere and a very short highlighting for the document title reflects that the system's estimate of that document relevance is almost exactly between "relevant" and "non-relevant" – i.e., even odds that the document is relevant.

Figure 3 shows an example of a retrieval session. We ran the query "Samuel Adams" on the Infoseek search engine. We judge relevant all the documents that mention the beer brand "Samuel Adams". The top ranked document is about Samuel Adams the Patriot and we marked it as non-relevant. The bright red sphere corresponding to that document is located at the bottom of the picture. The Wizard immediately pointed us to the document whose sphere is at the very top of the picture. The corresponding document

[1] The selection of colors reflects a common idea in the western world of green as equivalent to "go" and red as a synonym of "stop". The colors can be easily changed to reflect any other scheme using the preference commands.

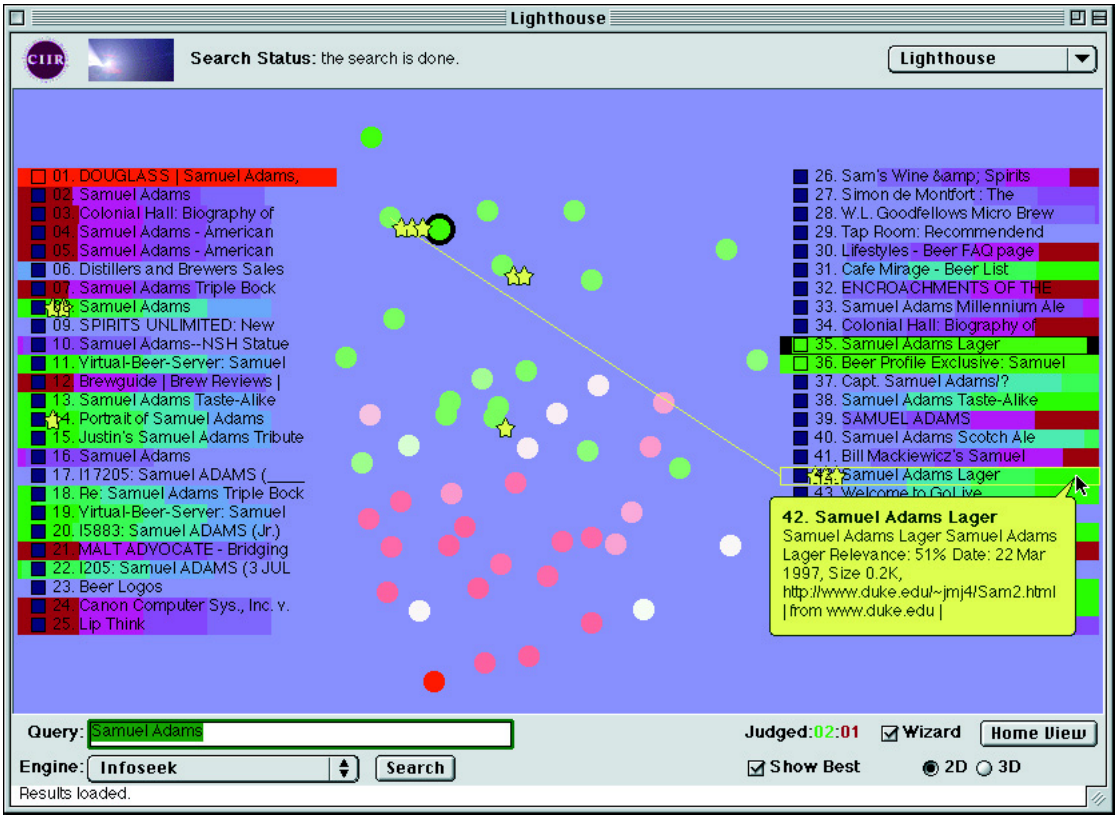

Figure 3: Screen shot of the Lighthouse system. The top fifty documents retrieved by the Infoseek search engine for a query. A two-dimensional picture is shown and both Shade and Star Wizard are switched on.

is ranked 35, it is about Samuel Adams Lager and we judged it relevant. The next document suggested by the Wizard is ranked 36 and the corresponding sphere is the green one with the black circle around it. Now one quick look tells us that the documents about the beer are probably occupy the top of the picture while the documents about the American patriot take the bottom part of the visualization. We can see how the colored shading propagates from the known relevant documents to the known non-relevant documents creating an impression of two lights – one green and one red – shining through the structure. This visual effect gave the name to the system.

We confess that the gradient filling of the title backgrounds was motivated mostly by design. Another alternative is to uniformly fill the corresponding title background, preserving the length of the highlighting as the indicator of the system's belief in its estimates. However, the uniform fill of the title backgrounds creates sharp boundaries of contrast color in the visualization that in our belief have a distracting effect on the user.

Star Wizard The second tool we call the *Star Wizard* It is controlled by the "Show Best" button on the Lighthouse screen shot Figure 1. It elaborates on the same information used by the Shade Wizard and indicates the three documents with the highest estimate of relevance. The highest ranked document is marked with three stars, the next one with two, and the third one is marked with one star. The stars are placed both by the corresponding document sphere and at start of document title.

3 Implementation

We have implemented the Lighthouse system following the client-server model. The client accepts the query and transmits it to the server. The server forwards the query to the search engine, collects the results as a list of URLs and descriptions in HTML format, parses these results, collects the corresponding web pages, parses and indexes the text of each page, computes the similarities between pages, generates the configurations for both 2- and 3-dimensional visualizations, and returns this data to the client. The server is written in Perl and C. It takes 0.5 sec to parse and index the documents, and another 0.5 sec to generate the spatial configuration on a computer with 600MHz Alpha CPU. The total time of a retrieval session is generally between 50 and 100 seconds, where most of the time is spend accessing the search engine and downloading the web pages. The efficiency of course depends on the current network congestion. The client side is written in Java (language version 1.1) and handles all the interaction between the system and the user including the necessary computations for the wizard tools. It can be installed and run locally as an application or it can be downloaded on the fly and run in the browser as an applet. The system is located at our web site [20]. Note that the server processes only one query at a time to avoid overloading the system.

3.1 Document Parsing and Indexing

The system removes HTML tags from the web pages and breaks the rest of the text into words. Then the words are stemmed and each word is replaced by a token – its root. For each document a vector of tokens is created V. The weight of the ith token in the vocabulary, v_i is computed using the INQUERY [2] weighting formula, which uses Okapi's *tf* score [23] and Inquery's normalized *idf* score:

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

where *tf* is the number of times the token occurs in the document, *docf* is the number of documents the token occurs in, *doclen* is the number of tokens in the document, *avgdoclen* is the average number of tokens per document in the collection, and *colsize* 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 [26].

The *docf*, *avgdoclen*, and *colsize* statistics have to be obtained from source collection. As the WWW is huge and it is impossible to exactly evaluate these numbers we use a 2.1GB collection of text created

by National Institute of Standards and Technology (NIST) for a large text retrieval and evaluation effort called Text REtrieval Conference (TREC) [11][2]. We believe this collection is a good source for contemporary English usage and provides an adequate values for the required statistics. This approach is common for IR systems operating with dynamic collections (e.g., routing and filtering TREC tasks) [1, 2, 3, 36].

3.2 Clustering Visualization

The document vectors exist in a very high-dimensional space where the number of dimensions is equal to the vocabulary size of the set. To present high-dimensional objects in just a few dimensions we use a Multidimensional Scaling approach called spring-embedding. Our choice was motivated by the graph-drawing heritage of the spring-embedding [10, 31] – it is supposed to generate eye-pleasing pictures.

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.

3.3 Wizard Implementation

For each unjudged document, Lighthouse computes its relevance estimation as a weighted sum of its similarity to all judged documents:

$$e(d) = \frac{\beta}{|R|} \sum_{\forall d_i \in R} sim(d_i, d) - \frac{\gamma}{|N|} \sum_{\forall d_i \in N} sim(d_i, d) \tag{2}$$

where *e(d)* is the estimated relevance value for the document d, R is the set of all judged relevant documents, N is the set of all judged non-relevant documents, and the $sim(d_i, d)$ is the cos similarity between the corresponding document vectors. In the Lighthouse system we set $\beta = 4$ and $\gamma = 1$. The estimated relevance values are also scaled to assume values between -1 and 1. Negative values result in the document sphere filled with the red color and the positive values lead to the green colored filling.

Note that this procedure is very similar to the traditional relevance feedback approach widely studied in Information Retrieval [27]. Generally during relevance feedback the marked documents are analyzed and their tokens are ranked using a similar weighting scheme. Out of these tokens a new query is created; it is run on the same collections resulting in a new set of documents. We however apply the formula to combine pairwise document similarities and we use only the information that is available after the first retrieval session. We do not require any query modifications and repetitive usage of the retrieval engine, an advantage particularly when the network access is costly or there are large network delays.

4 Evaluation

Our work extends the traditional ranked list with the clustering visualization. We claim that the combination is a more effective tool for locating relevant documents than the ranked list alone. We prove that starting from the highest ranked relevant document we can find the rest of the relevant documents much faster by using the visualization than by following the ranked list. Nevertheless the ranked list is still highly important for locating that highest ranked relevant document – we start from the top of the list and follow it until we find one relevant document.

The inter-document similarity information is reflected in the spatial distances between spheres in the clustering structure. Our evaluation focuses on how well the user can apply that information. We studied

[2] We use TREC volumes 2 and 4 that contain documents from Wall Street Journal, Financial Times, and Federal Register.

two questions: (1) how effective the users browse the visualization and (2) are they using it to its full effectiveness. In addition we have compared 2- and 3-dimensional versions of the presentation.

For our experiments we use the data available for TREC. Specifically, we took the titles of TREC topics 251-300 and 301-351 as queries and ran them against the documents in standard TREC collections[3]. The topic titles are short (2-3 words in length) and well suited to mimic web queries that generally are also short. The test collections contain documents from Congressional Records, Federal Register, Financial Times, Los Angeles Times, and Wall Street Journal. To run the queries we used the INQUERY [2] search engine. For each query we selected the 50 highest ranked documents. The relevance judgments for the documents are supplied by NIST accessors [11]. The documents were visualized in 2 and 3 dimensions with the spring-embedding algorithm. Thus we have created spatial configurations that result from the real documents and we know the relevance judgments for those documents.

4.1 User Study

We have conducted a user study to see how well the user can navigate the clustering visualization. We removed the ranked list and both wizards from Lighthouse and showed the participants only configurations of spheres floating in space. The participants were not told that the spheres represent the documents. Relevance judgments are known to differ across judges and for the same judge at different times [28]. We believe our design eliminates a high uncertainty that is generally connected with query formulation and making relevance judgments [15, 31] and allows us to isolate the navigation properties of the visualization which are the focus of our study.

We randomly selected ten document sets from our TREC-originated data. 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 document spheres were shown in white – i.e., as if starting after the first relevant document in the ranked list was found. Each document set formed a problem that we asked the users to solve. Specifically, we asked users to locate the rest of the green spheres while avoiding uncovering the red ones.

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.

The colored spheres – one green and possibly several red ones – were supposed to provide the users with the starting point in their exploration. A double-click on a white sphere opened up the sphere – revealed its true color. Once the color was shown, the sphere kept showing its color until the user was done with the problem. 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.

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 [35]. 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.

[3] We used volumes 2 and 4 (2.1GB) and TREC volumes 4 and 5 (2.2GB) correspondingly.

Question	2D	3D	significance
How easy was it to *understand how to use* the system?	4.4	3.4	$p < 0.002$
How easy was it to *learn to use* the system?	4.6	3.6	$p < 0.002$
How easy was it to *use* the system?	4.3	2.7	$p < 6 \cdot 10^{-5}$
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	$p < 0.006$
Are you satisfied with your performance in finding the green spheres?	3.6	3.1	$p < 0.03$

Table 1: 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"). The significance level is calculated by using paired two-tailed t-test.

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 1 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 also recorded the average precision of the users' search: each time a green sphere was revealed we calculated the ratio of green spheres to the total number of revealed spheres and averaged these values when the search was over:

$$prec = \frac{1}{|G| - 1} \sum_t \frac{N_G^t - 1}{N_G^t + N_R^t - N_R^0 - 1} \tag{3}$$

here $prec$ is the average precision, $|G|$ is the total number of green spheres (or number of relevant documents in the set), t is the moment when a new green sphere is revealed, N_G^t is the number of green spheres at time t, N_R^t is the number of red spheres at time t, N_R^0 is the number of red spheres shown originally. Note that 1 in this expression corresponds to the one green sphere shown in color in the beginning.

We average the precision across users and problems in each dimension and compare the precision of that search to the precision of the ranked list. Table 2 shows that the users are significantly more effective while navigating the visualization than they would be by traversing the ranked list. We observe 30% and 24% improvements for 2- and 3- dimensional visualizations. Surprisingly the 3-dimensional configurations were less effective than 2-dimensional ones. The two-factor analysis of variance (ANOVA) shows the statistical significance at $p < 0.01$.

Figure 4 illustrates this advantage of the visualization over the ranked list. Here are the results of the query "salsa" visualized in 3 dimensions. All the documents that mention the spicy sauce are marked as relevant. They are widely scattered in the ranked list and form a clump in the visualization. Given a starting point – on relevant document in that cluster – it much easier to locate the rest of them navigating the visualization than following the ranked list.

4.2 Off-line experiments

To test if the users exploited the visualization to its full potential, i.e., they used all the proximity information, we have defined a simple algorithm that is supposed to simulate a user looking for the green spheres in the visualization. Given a configuration of green, red, and white spheres in 2D or 3D, the algorithm assumes that the green spheres form a cluster and computes the location of the cluster centroid:

$$\vec{C} = \frac{1}{|G|} \sum_{\forall s \in G} \vec{s} \tag{4}$$

where $\vec{C}$ is the location of the centroid, G is the set of all green spheres, and $\vec{s}$ is the location of a sphere in the cluster. Note that all locations are computed in 2 or 3 dimensions.

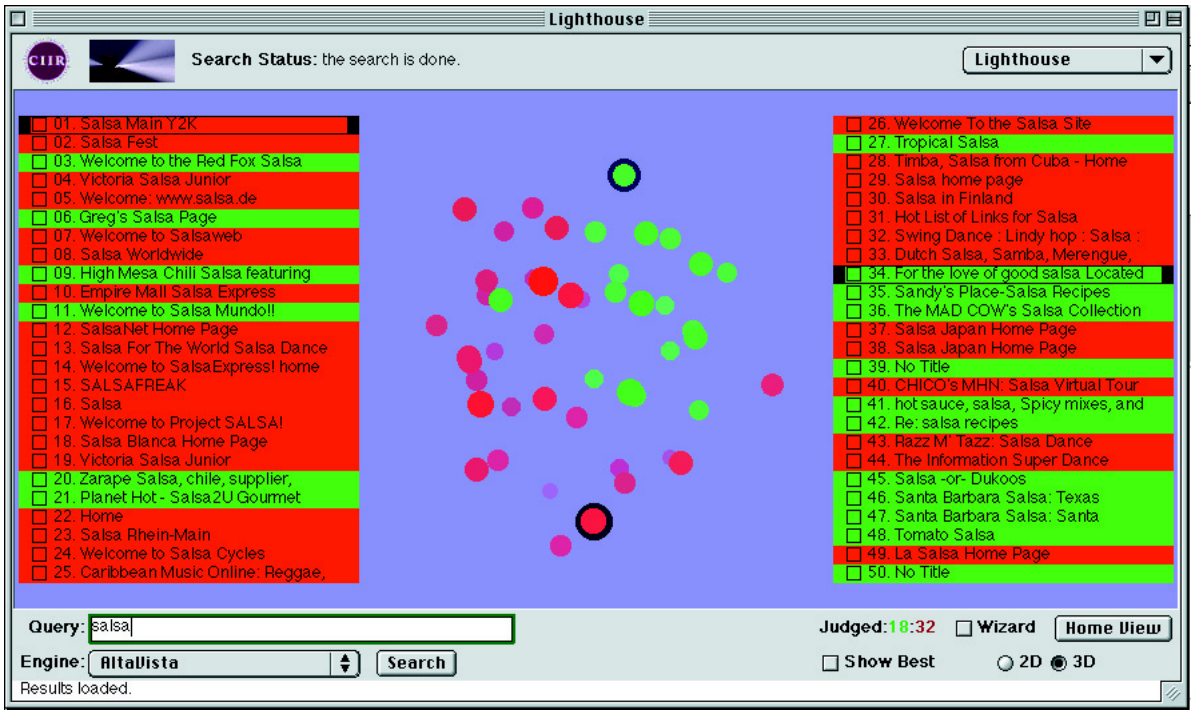

Figure 4: Screen shot of the Lighthouse system. The top fifty documents retrieved by the AltaVista search engine for the query "salsa". All the documents are marked. There an obvious clump of relevant (green) documents in the visualization. The same documents are widely scattered in the ranked list.

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 2: 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, percent improvement over the algorithm 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.

Table 2 shows that the algorithm outperforms the users by a small margin (5.7% in 2D and 13.3% in 3D). The differences are small but they are also statistically significant by the two-tailed t-test. Another difference between the users and the algorithm is that the algorithm is more effective in 3 dimensions than in 2. Assuming the truth of the Cluster Hypothesis, these results confirm that a 3-dimensional structure is more accurate in representing the inter-document similarities than a 2-dimensional one. For users, that higher accuracy of the representation is completely negated by the cognitive effort they have to apply recreating a 3-dimensional structure from the flat screen image.

A more detailed analysis [18] reveals that the foraging strategies employed by the users closely follow the described algorithm diverging from it when a good portion of spheres is revealed.

The higher effectiveness of the algorithm over users justifies the existence of both wizards. The Star Wizard points the user to the unjudged documents closest to the known relevant ones eliminating possible errors in determining the differences spatial distances. The Shade Wizard provides a general overview of the document set sketching the boundaries between the groups of relevant and non-relevant documents.

4.3 Wizard Performance

Forcing the document vector configurations into a small number of dimensions results in a loss of accuracy and loss of browsing effectiveness. Elsewhere [18] we have shown that the same foraging algorithm that selects the unjudged documents based on their proximity to the relevant information is more effective in the original vector space than in 2 or 3 dimensions. That research used only information about the relevant documents. Since then we have experimented with including the non-relevant information as well and observed additional modest improvement in the effectiveness. The Lighthouse wizards use the proximity information between document vectors in the original vector space. That accounts for possible inconsistencies in the visualization. For example, given that the relevance estimation is proportional to the spatial proximity, one would expect the colors to gradually fade from red to green while moving from marked non-relevant spheres to the marked relevant ones. However, Figure 3 shows a white sphere in the middle of a red cluster and placed very close to the the bright red sphere at the bottom of the page (the marked non-relevant document) – i.e., that unmarked document is actually far less similar to that non-relevant one than it appears from its position in the visualization. This is an artifact of the Multidimensional Scaling.

5 Conclusions

In this paper we have described Lighthouse, a working interface system for an on-line search engine that integrates the traditional ranked list with the spring-embedding clustering visualization. We showed that spatial proximity is an intuitive and well-recognized (by the users) metaphor for similarity between objects. The users were significantly more successful with the visualization than they would be by following the ranked list.

Our results also illustrated that – not surprisingly – the users are not able to recognize the proximity

relationship with complete accuracy and they have much more difficulties with 3-dimensional configurations than with 2-dimensional ones. We found out that a foraging algorithm that uses the exact inter-object distances is generally more successful. Lighthouse includes two wizard tools based on that algorithm. We expect these wizards will be very effective in supporting the users working with the system.

6 Future Work

The authors find the Lighthouse system very useful for their own web browsing. One interesting side effect of the system is its ability to notice and isolate the web pages that are unavailable (for whatever reason). Usually, these pages form an easily recognizable clump in the clustering visualization. For example, Figure 1 is the result of running a query "ricerca forschung" on the Infoseek engine. The query is the word "research" written in Italian and German. The German documents create a clump of spheres in the left bottom corner. The Italian documents form another clump on the right. A single document in the center is written in Spanish. The documents that group into a cluster at the top of the screen are the web pages that were inaccessible when we ran the query. In that case a page in English describing the error is returned to the user. This is a somewhat artificial example, but it illustrates an important property well.

We are considering extending the work on Lighthouse in the following three directions.

1. By default Lighthouse retrieves the top 50 documents from the search engine. Although it is possible to specify any other arbitrary large number of documents, our analysis was limited to the retrieved sets of that size. We plan to extend our experiments to accommodate a large number of documents hoping that the clustering visualization might find 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" of the picture getting replaced with fresh unknown documents as the search progresses. The rate of document disappearance will depend on their relevance and the user preferences.

2. Our wizard tool should perform best when there is only one relevant topic in the retrieved set. In that case all the relevant documents usually form one cluster that is easily detected as soon as the first relevant document is located. The system 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 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 system wizard is not able to do that, though it might be possible to provide such an effect.

3. 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. This would be similar in spirit to our earlier work on "space warping" [17].

Acknowledgments

The authors thank Victor Lavrenko for the help in implementing the document parsing and indexing parts of the Lighthouse server.

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

[1] James Allan, Jamie Callan, Bruce Croft, Lisa Ballesteros, John Broglio, Jinxi Xu, and Hongmin Shu. Inquery at TREC-5. In *Fifth Text REtrieval Conference (TREC-5)*, pages 119–132, 1997.

[2] James Allan, Jamie Callan, W. Bruce Croft, Lisa Ballesteros, Donald Byrd, Russell Swan, and Jinxi Xu. Inquery does battle with TREC-6. In *Sixth Text REtrieval Conference (TREC-6)*, pages 169–206, 1998.

[3] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of ACM SIGIR*, pages 37 – 45, 1998.

[4] Matthew Chalmers and Paul Chitson. Bead: Explorations in information visualization. In *Proceedings of ACM SIGIR*, pages 330–337, June 1992.

[5] W. Bruce Croft. *Organising and Searching Large Files of Documents*. PhD thesis, University of Cambridge, October 1978.

[6] D. R. Cutting, D. R. Karger, and J. O. Pedersen. Constant interaction time scatter/gather browsing of very large document collections. In *Proceedings of ACM SIGIR*, pages 126–134, 1993.

[7] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of ACM SIGIR*, pages 318–329, 1992.

[8] Dataware search engine. `http://www.dataware.com/find/default.html/`.

[9] David Dubin. Document analysis for visualization. In *Proceedings of ACM SIGIR*, pages 199–204, July 1995.

[10] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software–Practice and Experience*, 21(11):1129–1164, 1991.

[11] Donna Harman and Ellen Voorhees, editors. *The Fifth Text REtrieval Conference (TREC-5)*. NIST, 1997.

[12] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of ACM SIGIR*, pages 76–84, August 1996.

[13] M. Hemmje, C. Kunkel, and A. Willet. LyberWorld - a visualization user interface supporting fulltext retrieval. In *Proceedings of ACM SIGIR*, pages 254–259, July 1994.

[14] R. J. Hendley, N. S. Drew, A. M. Wood, and R. Beale. Narcissus: Visualising information. In *Proceedings of IEEE Information Visualization*, pages 90–96, 1995.

[15] Jurgen Koenemann and Nicholas J. Belkin. A case for interaction: A study of interactive information retrieval behavior and effectivness. In *Proceedings of Conference on Human Factors in Computing Systems*, pages 205–212, 1996.

[16] Anton Leuski and James Allan. Interactive cluster visualization for information retrieval. In *Proceedings of ECDL'98*, pages 535–554, September 1998.

[17] Anton Leuski and James Allan. Evaluating a visual navigation system for a digital library. *International Journal on Digital Libraries*, 2000. Forthcoming.

[18] Anton Leuski and James Allan. Improving interactive retrieval by combining ranked lists and clustering. In *Proceedings of RIAO'2000*, pages 665–681, April 2000.

[19] Anton Leuski and W. Bruce Croft. An evaluation of techniques for clustering search results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996.

[20] Lighthouse. `http://toowoomba.cs.umass.edu/~leouski/lighthouse/`.

[21] Xia Lin, Dagobert Soergel, and Gary Marchionini. A self-organizing semantic map for information retrieval. In *Proceedings of ACM SIGIR*, pages 262–269, 1991.

[22] Northern light. `http://www.northernlight.com/`.

[23] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In Donna Harman and Ellen Voorhees, editors, *Third Text REtrieval Conference (TREC-3)*. NIST, 1995.

[24] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kennth Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *Proceedings of IEEE Information Visualization*, pages 36–43, October 1999.

[25] D. Rushall and M. D. Ilgen. DEPICT: Documents evaluated as PICTures: Visualizing information using context vectors and self organizing maps. In *Proceedings of IEEE Information Visualization*, pages 100–107, 1996.

[26] Gerard Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[27] Gerard Salton and Cris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[28] Linda Scahmber. Relevance and information behavior. *Annual Review of Information Science and Technology*, 29:3–48, 1994.

[29] Marc M. Sebrechts, John V. Cugini, Sharon J. Laskowski, Joanna Vasilakis, and Michael S. Miller. Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of ACM SIGIR*, pages 3–10, 1999.

[30] Min Song. Bibliomapper: A cluster-based information visualization technique. In *Proceedings of IEEE Information Visualization*, pages 130–136, 1998.

[31] Russell Swan and James Allan. Aspect windows, 3-d visualizations, and indirect comparisons of information retrieval systems. In *Proceedings of ACM SIGIR*, pages 173–181, 1998.

[32] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979. Second edition.

[33] P. Willett. Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management*, 24(5):577–597, 1988.

[34] James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, and Anne Schur. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of IEEE Information Visualization*, pages 51–58, 1995.

[35] User study URL. `http://toowoomba.cs.umass.edu/~leouski/SE/`.

[36] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of ACM SIGIR*, pages 28–36, 1998.

[37] Oren Zamir and Oren Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of ACM SIGIR*, pages 46–54, 1998.