Visualization Support for Interactive Query Refinement

Orland Hoeber Xue-Dong Yang Yiyu Yao

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{hoeber, yang, yyao}@cs.uregina.ca

Abstract

It has been well documented that web searchers have difficulties crafting queries to fulfill their information needs. In this work, we use a concept knowledge base generated from the ACM Computing Classification System to generate a query space that represents the query terms in relation to the concepts they describe and the other terms that are related to these concepts. A visual representation of this query space allows the user to interpret the relationships between their query terms and the query space. Interactive query refinement within this visual representation takes advantage of the user's visual information processing abilities, and allows the user to choose terms that accurately represent their information need. A preview of the search results from Google provides the user with an indication of the current state of their query refinement process. This work allows the user to take an active role in the information retrieval process, supporting the fundamental shift from information retrieval systems to information retrieval support systems.

1. Introduction

Studies of web search behaviour have shown that a large portion of queries to web search engines are very short, often one to three terms in length [11, 24]. While the quality of the search results from short queries can vary, this short length indicates that users often have difficulties formulating queries that accurately represent their information needs. Additionally, there is evidence that users also have problems making subsequent modifications to their queries [22, 24]. In this paper we present the design of a system and a prototype implementation that supports the user's task of query refinement, allowing the user to visually and interactively generate a query that is a more accurate representation of their information need.

Even when presented with a list of potential terms to add to a query, research has shown that users may still have difficulties choosing good terms to add to their query [19, 14]. We suggest that one of the reasons users struggle with choosing new terms is because they are not provided with enough information about the relationship between their original query terms and the new terms that are made available to them. Rather than simply providing a list of additional terms, we use information visualization techniques to represent the query space generated by the initial query.

One of the challenges in creating a visualization of the query space is obtaining additional information regarding the query term relationships upon which to base the visual representation. The concept knowledge base developed as part of our previous research in automatic query expansion is used for this purpose [10]. We have extended this work and generated a new concept knowledge base using the ACM Computing Classification System [1] as the source of conceptual knowledge. The resulting knowledge base contains relationships between terms and the concepts these terms describe, within the computer science domain.

By visually representing the query space generated by the initial query, we provide a method that allows users to take advantage of their visual information processing capabilities and intelligence to determine the relationships between their query terms and the concepts they describe. Users may interactively refine their queries by choosing new terms that are also related to these concepts, or removing terms (either previously selected or part of the original query) that do not adequately describe their information need. The query space is interactively re-calculated based on these selections, and Google is accessed to provide users with a preview of the search results.

This work, which provides tools to support users in the task of visually refining their queries, is a step in the direction of developing an information retrieval support systems (IRSS). Yao describes IRSS as the next evolution of retrieval systems, where the emphasis is on the functionality that supports the users tasks of browsing, investigating, analyzing, understanding, organizing, and searching a collection [32]. As a first stage in the support of these tasks, we provide a visual representation of the query space that allows the users to refine their queries in a visual manner, and a preview of the results of their query that allows the users to determine the status of their query refinement process. The focus of our work is to allow the user to take an active role in the information retrieval process (rather than the passive role that is common in traditional information retrieval systems).

One of the principles that has guided our work has been to maintain a balance between computer automation and human control [21]. There is a critical human role in the crafting of a query that accurately represents a user's information need. To this end, we have developed a system that takes advantage of the computer's ability to store, retrieve, and process large amounts of data, and the human's ability for visual information processing, complex decision making, and judgement. One of the fundamental features is the method by which the human and the computer exchange information and initiate tasks. Information visualization and interaction techniques promote this exchange, allowing the users to interpret and make sense of the query space and easily refine their query to more accurately reflect their information needs.

This paper is organized as follows: A overview of the previous work on query expansion, information visualization, and visual web search systems is provided in Section 2. In Section 3, an overview of our prototype software framework and a description of how the concept knowledge base is used to generate a visual representation of the query space is given. The process for interactively refining a query using our system is explained in Section 4. The paper concludes with a discussion of the system in Section 5, and the conclusions and future work in Section 6.

2. Background

As a background to our research, we provide a brief overview of query expansion and an introduction to information visualization techniques applied to the domain of information retrieval.

2.1. Query Expansion

Query expansion, the process of adding additional terms to the user's original query, can be classified according to the work required by the user in order to generate the expansion[4]. *Manual query expansion techniques* are those which require the user to do the work of evaluating, selecting, and adding new terms to their query (i.e., without any additional computer support). *Automatic query expansion techniques* choose and add new terms to the user's query, without the involvement of the user other than to submit an initial query. *Interactive query expansion techniques* allow the user to interactively make choices which are then used to generate the query expansion.

Query expansion techniques may also be classified according to the type of information used in order to generate the expansion [4]. Query expansion techniques that are *based on an initial set of search results* evaluate the result set from the initial query to generate the query expansion. Techniques *based on collection-dependent knowledge bases* generate the query expansion by matching the original query to a knowledge base that was created based on some or all of the collection being queried. Techniques *based on collection-independent knowledge bases* match the original query to a knowledge base that was created using a source of information that is independent of the collection being queried.

Query expansion techniques that fit all the combinations of these classifications have been well studied [31, 18, 29, 20, 3, 8]. Of particular interest is the work by Harman [8], which used three different sources of information for new terms to add to the query, and provided lists of terms from which the user could choose. These term lists were generated via local analysis (relevance feedback), query analysis (variations of the query terms), and global analysis (term co-occurrence in the collection). While the results reported from this technique were good when users made the perfect choices from the lists of available terms, there is little evidence to show that users are able to make these perfect choices. On the contrary, studies have shown that users are often unable to make good choices when provided with a list of terms [19, 14]. In our work, we seek to provide additional information within a visual interface that will allow users to make theses choices.

2.2. Information Visualization

Information visualization is a technique for creating a graphical representation of data or concepts [30]. Moreover, information visualization promotes a cognitive activity in which users are able to gain understanding or insight into the data being graphically displayed by taking advantage of the human's visual information processing capabilities [23]. At the most basic level, we use information visualization techniques when we draw a graph to visually represent a data set. However, when these data sets are large, high-dimensional, or complex, information visualization becomes a challenging problem. In general, information visualization techniques allow the display of large data sets in a coherent manner, allowing us to compare and explore the data visually [28].

While the techniques for information visualization are as numerous as the types of data sets they graphically describe, we limit our discussion to information visualization techniques that have been applied to information retrieval systems, and in particular to those that support the visualization of queries.

The application of information visualization to queries has primarily been focused on the visualization of Boolean queries [13, 33, 2, 25]. This activity in supporting Boolean queries can be attributed to the great difficulties that many users have with crafting correctly specified Boolean queries [9]. In the context of visualizing free text queries, very little work has been done. Stojanovic [26] visually represented the current query within the context of its neighbour queries using a simple overlapping box layout. Joho [12] used a menu structure to visually represent the hierarchy of terms to add to a query in an interactive query expansion system.

Visualization of the query space is not common since there is often little or no information available upon which to generate a visual representation. For example, in the absence of additional information, how do we generate a visual representation of a set of keywords entered by a user? In our work, we employ a concept knowledge base as a means of deriving additional information about a query, in terms of the concepts it is related to and the other terms that have been used to describe those concepts. We call this additional information the *query space* generated for a given query.

3. Query Visualization

3.1. Framework

The framework for our visual query refinement software consists of three primary components. The *query space generation component* takes the query and the concept knowledge base as input, and produces the query space. The *query space visualization component* uses graph drawing algorithms and techniques for colour coding and data representation to create a visual display of the query space. The *Google preview component* sends the current query to the Google search engine [6] and displays summary information about the query results.

This framework is illustrated in Figure 1, the details of which are provided in the following sub-sections. The user interaction steps shown in this diagram are described in Section 4.

3.2. Generation of the Query Space

In order to generate a visual representation of a user's query, we must first generate the query space. That is, we must determine the context or setting in which the query terms exist. For this purpose, we use the concept knowledge base, which we developed as part of our previous work on automatic query expansion [10]. A concept knowledge base is a bipartite graph consisting of two sets of nodes: concepts and terms. In this graph, edges are assigned a weight and can only exist between concept nodes and term nodes.

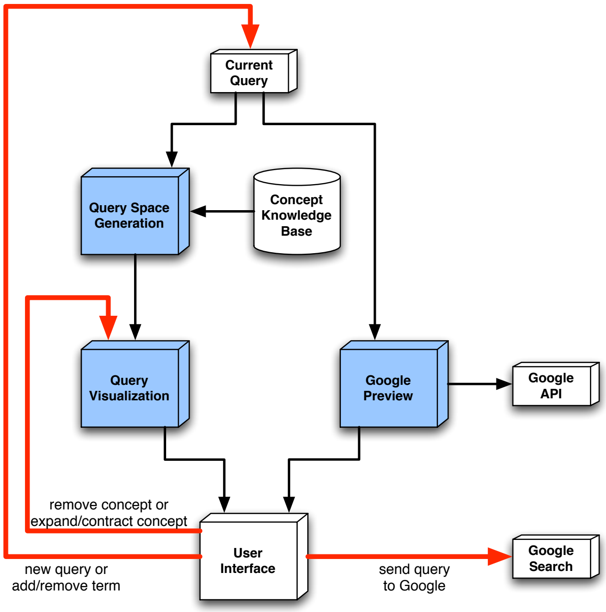


Figure 1. The framework for the visual query refinement software. The primary software components are highlighted in blue; the interaction paths are illustrated by the thick red arrows.

More formally, a concept knowledge base $CKB = \{C, T, E\}$ consists of a set of concept nodes C, a set of term nodes T and a set of edges $E = \{c_i, t_j, w_{ij}\}$, where $c_i \in C$, $t_j \in T$, and c_i and t_j are related with a weight w_{ij}. The weight w_{ij} of an edge in the concept knowledge base represents the degree to which the term represents the intension of the concept.

In our initial work on the concept knowledge base, we described a method for automatically constructing such a knowledge base using a concept hierarchy such as the Open Directory Project [15]. In this work, we chose to use a more specific and focused source of concept information: the ACM Computing Classification System [1]. This classification is used to categorize papers submitted to the ACM Press, and contains both a concept hierarchy representing the various disciplines within the computer science domain, and short descriptions of these disciplines. In generating the concept knowledge base, we use the nodes in this concept hierarchy (i.e., disciplines in the computer science domain) as concepts, and the descriptions of these nodes as the source of terms.

A bag-of-words approach is used to count the occurrences of terms within each document. To calculate the the edge weight values w_{ij}, we consider the set of documents (or descriptions) $D_i = \{d_{i1}, \ldots, d_{in}\}$ which are associated with a concept $c_i \in C$. For each document d_{ik}, the set of terms used in this document is $T_{ik} = \{t_{1,ik}, \ldots, t_{m,ik}\}$. We define the function $f(d_{ik}, t_j)$ as the occurrence count of term t_j in document d_{ik}. The value for the edge weight between concept c_i and term t_j is given by:

$$w_{ij} = \frac{\sum_{k=1}^{n} \frac{f(d_{ik},t_j)}{\sum_{l=1}^{m} f(d_{ik},t_{l,ik})}}{n}$$

After all the concepts have been analysed, we normalize the edge weights from each term in the concept network. For a term t_i that is connected to r concepts whose index is given by the relation $f(x)$, $x = 1 \ldots r$, the normalization is performed via the simple calculation:

$$w_{ij} = \frac{w_{ij}}{\sum_{k=1}^{r} w_{if(k)}}$$

In effect, the weights in the concept knowledge base represent the average term frequency, which is then normalized with respect to the occurrences of the term in other concepts. Terms that have a high frequency of occurrence in the descriptions of a concept will result in a high edge weight between these nodes in the concept knowledge base. Terms that have a low frequency of occurrence in the descriptions of a document will result in a low edge weight. Due to the normalization, terms that are used frequently to describe many different concepts will also have a low edge weight to these concepts.

A further extension of our previous work was to use Porter's stemming algorithm [17] to pre-compute the stems of all terms contained in the concept knowledge base. This allows terms to be matched based on the stems or roots of the words, rather than exact word matches.

Given a concept knowledge base $CKB = \{C, T, E\}$, and a query $Q = \{q_1, \ldots, q_n\}$ consisting of query phrases q_i, the process of generating the query space $QS = \{C^*, T^*, E^*\}$ is given by the following five steps:

1. Match the query terms q_i to the term set T to obtain a $T' \subseteq T$ using the stems of the terms.

2. Obtain the set of concepts $C^* \subseteq C$ which are connected to the terms in T'. We use two parameters to control this operation: a weight threshold w_e, and an term ratio TR. First, all the concepts that are connected to the terms in T' with a weight greater than w_e are chosen as candidate concepts in C^*. Each of these concepts are then evaluated to determine the ratio of the terms in T' to which they are connected with a weight greater then w_e. If this ratio is less than TR, the candidate concept is dropped from C^*.

3. Obtain the set of terms $T'' \subseteq T$ which are connected to the concepts C^*. We use a weight threshold parameter w_d to control this operation. All terms that are connected to the concepts in C^* with a weight greater than w_d are chosen as the phrases in T''.

4. Perform a union of the original query terms and the new set of terms to obtain the query space terms: $T^* = Q \cup T''$.

5. Obtain the set of edges $E^* \subset E$ such that $e_{ij} \in E^*$ iff $c_i \in C^*$ and $t_j \in T^*$.

We note that the query space generated above is a sub-graph of the concept knowledge base bipartite graph. Therefore, the query space is also a bipartite graph. We further note that the thresholds w_e and w_d, along with the term ratio TR control the size of the query space generated for a given query. Through initial experiments with the concept knowledge base generated from the ACM Computing Classification System, we set these values at $w_e = 0.05$, $w_d = 0.10$, and $TR = 0.51$.

3.3. Visualization of the Query Space

The key to creating a visual representation of this bipartite sub-graph is to ensure the users can properly interpret the underlying features of the query space. That is, users must be able to identify the three classes of nodes (concepts, unselected terms, and selected terms), along with the relationships between the concept nodes and the term nodes. To this end, we require that our graph drawing algorithm represents the distance between two nodes as a function of the edge weight, and ensure that the edges are always visually present to reinforce the relationship between concept-term pairs.

We use a heuristic-based graph drawing algorithm that attempts to satisfy two criteria: node distance must represent the edge weights, and nodes must not be placed too close to one another. The algorithm attempts to place nodes that are connected to one another a distance that is a function of their edge weight. Therefore, a concept-term pair with a high edge weight will be placed near one another, and a concept-term pair with a low edge weight will be placed away from one another. In addition to satisfying this criteria, the system will also ensure that no two nodes are placed too close to one another, reducing the overlapping and occlusion of nodes. In order to increase the speed at which a steady-state solution is found, the concept nodes remain

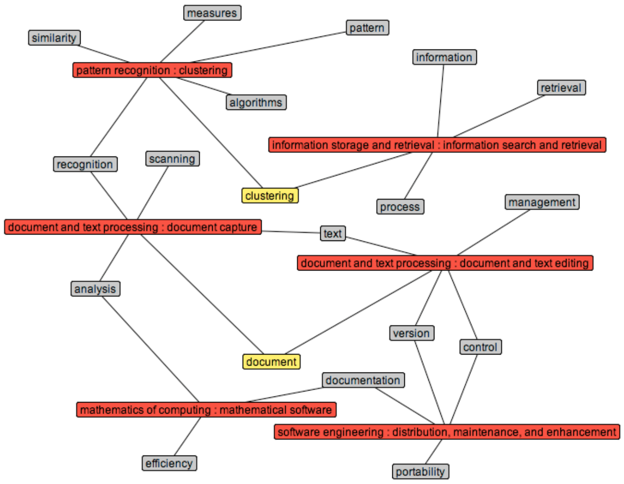

measures
similarity
pattern
pattern recognition : clustering
information
retrieval
algorithms
information storage and retrieval : information search and retrieval
recognition
scanning
clustering
process
management
document and text processing : document capture
text
document and text processing : document and text editing
analysis
version
control
document
documentation
mathematics of computing : mathematical software
software engineering : distribution, maintenance, and enhancement
efficiency
portability

Figure 2. The visual representation of the query space generated by the terms "document" and "clustering".

static and only the term nodes are moved to satisfy these criteria.

Animation techniques are used to move the nodes to their locations as the graph drawing algorithm iteratively attempts to satisfy the above criteria. The purpose of this animation is to illustrate to the user that they too can interact with the visual representation. For example, users can drag the nodes to visually organize the query space, after which, the system will continue to attempt to satisfy the above criteria.

In order to promote the perception of the different classes of nodes in the visual representation, we use colour coding techniques [27, 30]. Neutral colours are used to represent the underlying graph structure; contrasting colours are used to highlight nodes of interest. As illustrated in Figure 2, the red nodes represent the concepts, and the yellow nodes represent the selected terms. Since the terms from the original query are automatically selected, users can easily determine the meaning of the colour coding.

By default, the visual representation of the query space shows all the concepts and terms that are present. However, when the query space is large, a more compact representation is beneficial. To this end, we provide a technique for compacting and expanding the terms that are connected to a concept into the visual representation for that concept.

When a concept is to be displayed in an expanded representation, all the terms that are connected to that concept are displayed as described above (see Figure 2). A compacted representation is also available, wherein the terms connected to the concept are displayed as thin layers below the concept, providing the illusion that these terms are stacked behind the concept. However, when one of the terms connected to a compacted concept is also connected to an expanded concept, that term remains external to the compacted concepts it is connected to until all the concepts it is connected to are placed in compacted representation.

The user can choose between the expanded and compacted visual representation of a concept by double-clicking the concept node. This feature allows the user to interactively adjust the level of detail of the visual representation. In effect, we can consider this as a type of localized semantic zooming [23] where information is hidden or shown, depending on the level of detail the user has chosen for the concept. The benefit of this technique is that the user can still identify the presence of terms and relationships to other terms or concepts, and can choose to view this information in more detail easily.

The end result of these features is an animated visual representation of the query space that provides the user with the ability to gain insight and understanding of the features of the query space. From Norman's stages of action theory, this visual representation supports the user in crossing the gulf of evaluation [16]. In particular, the visual representation takes advantage of the user's visual information processing capabilities to quickly perceive the fundamental objects in the visual display (i.e., the three classes of nodes and their edges), interpret the meanings of these objects, and make sense of the information being displayed with respect to the initial query.

The users' original query terms are highlighted in the visual representation, which results in the ability to visually identify the relationships between their query and the concepts, as well as the relationships between these concepts and new terms that can be added to the query. At the most basic level, this allows the user to identify whether the terms they entered are actually appropriate for their information need. That is, the resulting concepts should have some relevance to the information the users are seeking. In addition, users can easily identify whether their terms are very general (connected to many different concepts) or very specific (connected to only a single concept).

3.4. Generation of Google Preview

In order to provide the user with an indication of quality of their currently generated query, we provide a preview of the Google search results for this query. We use the Google Web API [7] to send the currently generated query to the Google Search Engine. The results (including the number of documents returned by the query, along with the title and URL of each of the first five documents) are displayed along side the visual representation of the query space. The keyword context is made available by hovering the mouse over the title of one of the search results, as illustrated in Figure 3.

This preview of the results returned by Google provides the users with an additional source of information about the quality of their query. That is, as users interactively refine their query (as described in the following section), they can determine the quality of the results of their actions. In effect, this feature provides the users with the ability to determine the outcome of their actions in an interactive manner, rather than the manual cycle of altering their query, submitting the query to Google, and viewing the results.

4. Interactive Query Refinement

There are five actions a user can perform in the pursuit of their task of query refinement. These actions are visually depicted by the thick red arrows in Figure 1, and are described below.

4.1. New Query

The user can initiate a new query by simply typing in the query text box, and either pressing the Enter key, or clicking the “Visualize” button. In doing so, the system will generate the query space, provide a visual representation of this query space, and provide a preview of the Google results.

4.2. Adding or Removing Terms

The users may interactively modify their query by double-clicking on term nodes they wish to add to the query. Newly added terms are highlighted in yellow, like the original query terms. As terms are added to the query, a new query space is generated, and the visual representation of this query space is updated (i.e., concepts and terms that are no longer in the query space are removed, and those that are new are added). In addition, a new Google preview is generated, and the refined query box at the bottom of the screen is updated to reflect this change.

To support the users in refining their queries (rather than just expanding them), the system also allows the user to remove terms from the query by double-clicking a highlighted term. Doing so will return this term to the non-highlighted (grey) colour, and will regenerate the query space, update the visual representation, and reproduce the Google preview.

We note that the user may choose to remove one of the original query terms. This illustrates one of the key features of a query refinement system over a query expansion system: in a query expansion system, additional terms may only be added to the user’s original query; in a query refinement system, the user may make modifications to the original query. This is a valuable feature in situations where the user identifies a term that is a better description of their information need than one of their original terms.

4.3. Remove a Concept

In reviewing the visual representation of the query space, the user may identify concepts which are not relevant to their current information need. These concepts may be present in the query space if the query terms are very general. In order to make it easier for the user to focus on the concepts that are relevant, the system allows a user to remove a concept from the query space. In all subsequent interaction steps in the current user session, these removed concepts will not be present.

4.4. Expand or Collapse a Concept

Users may choose to switch the representation of a concept between expanded mode and collapsed mode. Switching between these different modes (as described in Section 3.3) allows the user to reduce the visual clutter generated by concepts which they have finished considering.

This expansion and collapse of concepts also provides the system with a means for dealing with vague queries that generate large query spaces. Large query spaces will generally result in a visual representation that is difficult to interpret and make sense of (primarily due to edge crossings and node occlusion, as well as the inability of the graph drawing algorithm to satisfy the layout criteria). By collapsing the concepts, these large query spaces become easier to visually process, since the number of nodes are significantly reduced. The user may then interactively expand and collapse the concepts as needed to refine their query.

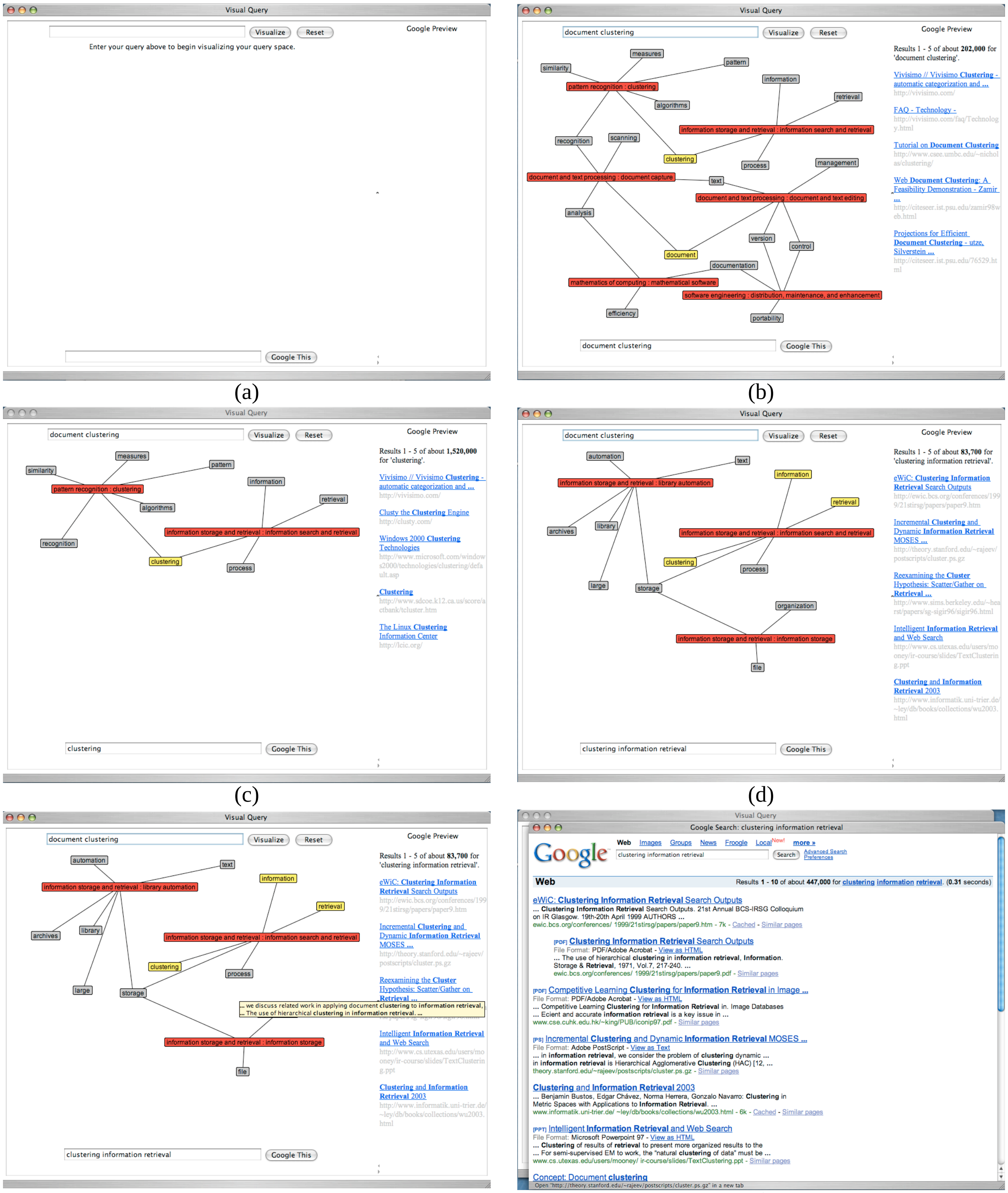
4.5. Send the Query to Google

In addition to providing a preview of the Google results as the user conducts their query refinement tasks, the system also allows the user to send the query they have created to the Google search engine [6] by clicking the “Google This” button. Doing so opens a new instance of the web browser and submits the query to Google automatically. This allows the user to access the complete set of results returned by Google, and to perform relevance judgement within the popular list-based search results interface.

5. Discussion

A complete query refinement process produced with our prototype software is illustrated in Figure 3. A video showing the use of this system for visualizing and interactively refining a query is provided on the author’s web site [1]. We

[1] http://www.cs.uregina.ca/˜hoeber/VisualQuery/


Visual Query
Visualize
Reset
Google Preview
Enter your query above to begin visualizing your query space.
Google This
document clustering
similarity
measures
pattern
pattern recognition : clustering
information
algorithms
retrieval
information storage and retrieval : information search and retrieval
recognition
scanning
clustering
process
management
document and text processing : document capture
text
document and text processing : document and text editing
analysis
version
control
document
documentation
mathematics of computing : mathematical software
software engineering : distribution, maintenance, and enhancement
efficiency
portability
Results 1 - 5 of about 202,000 for 'document clustering'.
Results 1 - 5 of about 1,520,000 for 'clustering'.
Results 1 - 5 of about 83,700 for 'clustering information retrieval'.
automation
information storage and retrieval : library automation
archives
library
large
storage
organization
information storage and retrieval : information storage
file
... we discuss related work in applying document clustering to information retrieval.
... The use of hierarchical clustering in information retrieval. ...
clustering information retrieval

(a)

(b)

(c)

(d)

(e)

(f)

Visual Query

Google Search: clustering information retrieval

Web Images Groups News Froogle Local New! more »

clustering information retrieval Search Advanced Search Preferences

Web Results **1** - **10** of about **447,000** for clustering information retrieval. (**0.31** seconds)

eWiC: **Clustering Information Retrieval** Search Outputs
... **Clustering Information Retrieval** Search Outputs. 21st Annual BCS-IRSG Colloquium on IR Glasgow. 19th-20th April 1999 AUTHORS ...
ewic.bcs.org/conferences/ 1999/21stirsg/papers/paper9.htm - 7k - Cached - Similar pages

[PDF] **Clustering Information Retrieval** Search Outputs
File Format: PDF/Adobe Acrobat - View as HTML
... The use of hierarchical **clustering** in **information retrieval**, **Information**. Storage & **Retrieval**, 1971, Vol.7, 217-240. ...
ewic.bcs.org/conferences/ 1999/21stirsg/papers/paper9.pdf - Similar pages

[PDF] Competitive Learning **Clustering** for **Information Retrieval** in Image ...
File Format: PDF/Adobe Acrobat - View as HTML
... Competitive Learning **Clustering** for **Information Retrieval** in. Image Databases
... Ecient and accurate **information retrieval** is a key issue in ...
www.cse.cuhk.edu.hk/~king/PUB/iconip97.pdf - Similar pages

[PS] Incremental **Clustering** and Dynamic **Information Retrieval** MOSES ...
File Format: Adobe PostScript - View as Text
... in **information retrieval**, we consider the problem of **clustering** dynamic ...
in **information retrieval** is Hierarchical Agglomerative **Clustering** (HAC) [12, ...
theory.stanford.edu/~rajeev/postscripts/cluster.ps.gz - Similar pages

Clustering and **Information Retrieval** 2003
... Benjamin Bustos, Edgar Chávez, Norma Herrera, Gonzalo Navarro: **Clustering** in Metric Spaces with Applications to **Information Retrieval**. ...
www.informatik.uni-trier.de/ ~ley/db/books/collections/wu2003.html - 6k - Cached - Similar pages

[PPT] Intelligent **Information Retrieval** and Web Search
File Format: Microsoft Powerpoint 97 - View as HTML
... **Clustering** of results of **retrieval** to present more organized results to the
... For semi-supervised EM to work, the "natural **clustering** of data" must be ...
www.cs.utexas.edu/users/mooney/ ir-course/slides/TextClustering.ppt - Similar pages

Concept: Document **clustering**

Open "http://theory.stanford.edu/~rajeev/postscripts/cluster.ps.gz" in a new tab

Figure 3. The process of visual query refinement begins with entering the initial query terms (a). A visual representation of the query space is generated (b), from which the users may choose to remove (c) or add (d) terms from the query. The users can determine the progress of their query refinement by consulting the Google preview (d). The refined query can be sent to the Google search engine with the click of a button (d).

note that the system currently only supports queries made up of a simple collection of terms.

Following the classification discussed in Section 2.1, our work is an interactive technique based on a collection-independent knowledge base. However, we describe our technique as a query refinement technique, since we allow the user to additionally remove some or all of their original query terms in the creation of their new query.

One of the critiques of the work by Harman [8] is that it assumes that users can make the perfect choices when presented with a list of potential query expansion terms. Subsequent studies have shown that users are often not able to make these good choices [19, 14]. We argue that this is a failure of the list representation used to display the potential query expansion terms. That is, since the list representation is not able to convey the relationships between the original query and the list of potential terms, nor the relationships between the terms themselves, users have difficulties in determining the value of adding a term to their query.

One of the goals of our work was to provide the users with additional information about the potential terms that can be added to their query. This information comes in the form of the relationship between the user's query terms and the concepts they are related to, and the relationship between these concepts and the other terms that have been used to describe them. Providing this information in a visual manner allows the user to interpret and make sense of these relationships as a result of their visual information processing capabilities. That is, it is easier for the user to understand these relationships by looking at a picture than by reading a list of terms.

Another important aspect of our work is that the query refinement process occurs completely in the query space. That is, the use of the concept knowledge base allows the query space to be generated completely independent of the initial search results or the collection of documents being searched. The value of this approach is best illustrated by the simple example below.

Consider the situation where a user enters a query term that is used to describe two different concepts. If one of these concepts is much more common in the corpus, or is given a higher value by the information retrieval system, then the top documents returned by the information retrieval system may all be related to this concept. In this case, the terms suggested for query expansion will be focused exclusively on this common or important concept. If the user was actually interested in the other concept, then this method of basing the query refinement in the document space will have failed.

In using a concept knowledge base and therefore keeping the query refinement in the query space, this situation is avoided. That is, the user will be able to easily identify the two different concepts the query term is related to, and can refine the query based on the concept that matches their information need (rather than the one the information retrieval system automatically selected).

It is clear that the quality of the query refinement process is directly impacted by the quality of the concept knowledge base. The concept knowledge base generated from the ACM Computing Classification System is of an adequate quality for the purposes of demonstrating the results of our research, although the query refinements are only possible within the computer science domain. We are currently investigating the further refinement of the concept knowledge base using the top documents returned by the ACM Digital Library for each of the disciplines, as well as the construction of a complete concept knowledge base using the Open Directory Project.

6. Conclusion & Future Work

In this paper, we have presented our work in the development of an information retrieval support system that focuses on assisting the user in the creation of a query that accurately reflects their information need. The driving force behind this work is the acknowledgement that the crafting of a query is a task that requires significant human involvement. The goal of our work was to develop a system that supports this human task through visualization and interaction with the query space.

The visual representation of the query space allows the user to visually process the information display, resulting in an effective interpretation of the relationship between the query terms and the additional terms that can be added to the query. Our system supports both the addition of new terms, and the removal of terms from the query. Doing so automatically generates a new query space and the corresponding visual representation. Animation techniques are used to generate a smooth transition from one query space to another, supporting the user in making sense of the results of adding or removing terms.

In addition, we provide a preview of the query results from Google, which allows the user to ascertain how their query refinement process is affecting the information retrieval process. The tools made available in this system allow the user to easily interact with the query space to generate a query that is an accurate representation of their information need.

The primary contributions of this work are the generation and visual representation of the query space, and the techniques by which the users are supported in their query refinement tasks. Much of the previous work has focused on the visual representation of an information retrieval task within the document space. We argue that the query space is a more appropriate context in which to conduct query expansion. The interaction techniques we provide within the query space allow the user easily add or remove terms from the query, as well as manipulate the visual representation to further assist in making sense of the query space. By interactively updating the query space and the visual representation as these changes are made, our system supports the user in understand the ramifications of their choices and manipulations.

In addition to the development of more complete and general concept knowledge bases, our future work includes adding support for complex queries, conducting an empirical evaluation our visual query refinement techniques with respect to the usability as well as the accuracy of the query generated, conducting a comparative evaluation of our work to other techniques for query refinement, and the development of visualization and interaction tools to support the user in the relevance evaluation of the documents returned by the information retrieval system. Our overall goal in this research is to develop an information retrieval support system that uses visualization to allow the user to take an active role in the information retrieval process.

References

[1] ACM. Acm computing classification system. http://www.acm.org/class/.

[2] P. Anick, J. Brennan, R. Flynn, D. Hanssen, B. Alvey, and J. Robbins. A direct manipulation interface for boolean information retrieval via natural language query. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1990.

[3] C.-H. Chang and C.-C. Hsu. Enabling concept-based relevance feedback for information retrieval on the www. *IEEE Transactions on Knowledge and Data Engineering*, 11(4), 1999.

[4] E. N. Efthimiadis. Query expansion. *Annual Review of Information Systems and Technology (ARIST)*, 31, 1996.

[5] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11), 1987.

[6] Google. Google search engine. http://www.google.com/.

[7] Google. Google web api. http://www.google.com/apis/.

[8] D. Harman. Towards interactive query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1988.

[9] M. Hearst. User interfaces and visualization. In R. Baeza-Yates and B. Ribeiro-Neto, editors, *Modern Information Retrieval*. Addison-Wesley, 1999.

[10] O. Hoeber, X.-D. Yang, and Y. Yao. Conceptual query expansion. In *Proceedings of the Atlantic Web Intelligence Conference*, 2005.

[11] B. J. Jansen and U. Pooch. A review of web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, 52(3), 2001.

[12] H. Joho, C. Coverson, M. Sanderson, and M. Beaulieu. Hierarchical presentation of expansion terms. In *Proceedings of the ACM Symposium on Applied Computing*, 2002.

[13] S. Jones. Graphical query specification and dynamic results previews for a digital library. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1998.

[14] M. Magennis and C. J. van Rijsbergen. The potential and actual effectiveness of interactive query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997.

[15] Netscape. Open directory project. http://www.dmoz.org/.

[16] D. Norman. Cognitive engineering. In D. A. Norman and S. D. Draper, editors, *User Centred Systems Design*. Lawrence Erlbaum Associates, 1986.

[17] M. Porter. An algorithm for suffix stripping. *Program*, 14(3), 1980.

[18] Y. Qiu and H. P. Frei. Concept based query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[19] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

[20] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4), 1990.

[21] B. Shneiderman. *Designing the User Interface*. Addison-Wesley, 1998.

[22] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1), 1999.

[23] R. Spence. *Information Visualization*. ACM Press, 2001.

[24] A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: the public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3), 2001.

[25] A. Spoerri. Infocrystal: A visualization tool for information retrieval. In *Proceedings of IEEE Visualization*, 1993.

[26] N. Stojanovic. Information-need driven query refinement. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, 2003.

[27] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[28] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.

[29] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[30] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.

[31] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1), 2000.

[32] Y. Yao. Information retrieval support systems. In *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, 2002.

[33] D. Young and B. Shneiderman. A graphical filter/flow model for boolean queries: An implementation and experiment. *Journal of the American Society for Information Science*, 44(6), 1993.