Context Representation for Web Search Results

Jesús Vegas[1], Fabio Crestani[2], and Pablo de la Fuente[1]

[1] Dpto. Informática
Universidad de Valladolid
Valladolid, Spain
{jvegas,pfuente}@infor.uva.es
[2] Dept. Computer and Information Sciences
University of Strathclyde
Glasgow, Scotland, UK
F.Crestani@cis.strath.ac.uk

Abstract Context has long been considered very useful to help the user assess the actual relevance of a document. In Web searching, context can help assess the relevance of a Web page by showing how the page is related to other pages in the same Web site, for example. Such information is very difficult to convey and visualise in a user friendly way. In this paper we present the design, implementation and evaluation of a graphical visualisation tool aimed at helping users to determine the relevance of a Web page by displaying the structure of the Web site the page belongs to. The results of an initial evaluation shows that this visualisation technique helps the user navigate large Web sites and find useful information in an effective way, without increasing the cognitive load of the user.

1 Introduction and Motivations

Searching the Web is one of the most frequent information access tasks nowadays. Unfortunately, it is often also one of the most frustrating ones. When the user issues a query to a search engine she is trying to condense an information need in only a few words. This is a very hard task, but it is only part of the complex task of information searching on the Web. In fact, as a response to the user query, the web search engine very often shows a large number of Web pages, which the user is supposed to review to assess their actual relevance and find the sought information. The user gets very little help from a Web search engines for this task. In fact, while much work has been devoted to the design and development of algorithms to improve the search process, not a lot of work has been done to help the users to find the needle in this returned ”document haystack”. This issue is also present in traditional Information Retrieval (IR), although more work has been devoted to IR search results visualisation than to Web search results visualisation. This paper is mainly concerned with result presentation in Web searching, but some of the ideas presented here can also be applied to traditional IR. So, to generalise our arguments, in the remainder of this paper we will use the terms Web page and document interchangeably.

Results visualisation is not a very well developed area of Web searching [13]. Most search engines return in response to a user query only a set of links to web pages, sorted by their relevance to the terms in the query. Most search engines present the title, URL, and a short summary. But this is not very helpful when the user is dealing with thousands of pages in the result set [18]. Some search engines (e.g. Google) use a KWIC (Keyword in Context) approach to show query terms highlighted in a query-biased summary of a relevant page. Although this is useful, it fails to show how a relevant page is related to the content of the entire Web site it belongs to.

We believe that it is important to design interfaces that maximise the amount of useful information that users obtain as a result of a Web search, and that minimise the cognitive load needed to interpret it [14]. These interfaces have shown to increase considerably the user satisfaction with the system [12].

Another way to increase the user satisfaction in Web searching is by enhancing the expressive power of the querying process. Allowing the user to produce more well defined queries, can help the user find and recognise relevant documents. Enabling more powerful querying can also be done at the level of the user interface, so it is possible to build interfaces that allow the user more querying expressive power and, at the same time, enable the user to recognise in a easier way relevant documents returned by the system. In a Web searching context, this can be done at leats at two levels: the Web page (document) level, and the Web site level [8].

Web search engines usually present to the user only a set of retrieved pages. Each page is presented using different information items considered important, for example, the title, and the sentences containing query terms. It is difficult to find examples of more sophisticated results presentation techniques in Web searching. Notable exceptions are search engines that return in response to a query results organised in topically related clusters (e.g. Vivisimo, Kartoo). While these search engines might be useful for some tasks, their effectiveness depends not only on how well they estimate Web pages relevance, but also on how well they measure topical similarity between pages. The result is that few search engines adopt this way of visualising results given its complexity and relatively few users use these search engines.

So, at the pages level, we believe that the structure of the Web page could be of great interest for the user, because it can show the overall complexity of the page and where the relevant parts of the document are. This is especially important for complex pages, that use different structural elements like tables, graphics, columns, etc., since the user often finds it difficult to locate the relevant information in the page and often gets diverted from the search objectives.

Another level of Web search results presentation that should be considered is the Web site level. In this case, the Web pages returned as result of a query could be displayed in relation to the overall structure of the Web sites they belong to [9]. In addition, Web sites could be ordered by their relevance with respect to the query to indicate to the user where relevance information can be found. This poses the problem of evaluating the relevance of an entire Web site to a query. An intuitive solution is that a Web site is relevant to a query if many of its Web pages are relevant to the query. But this is only a very naïve solution, since it does not take into account the structure of the Web site, the differences between Web pages, and other relevant information [15]. Taking the structure into account is necessary, since it enables to consider additional important elements, like for example how deep a page is buried in the site, or if there are other relevant Web pages linked to it, etc. In addition, showing the user the result set at the Web sites level can be useful because it can give the user an overview of the context in which the potentially relevant page is placed and gives the user the ability to select directly which Web site to browse. Given the importance of browsing for finding information, especially when an information need is not well formulated, providing the user with good Web sites to browse, probably containing relevant information, is extremely important [6].

All these ideas seems sound in theory, but they need to be formalised and tested experimentally. This is what we are currently doing and is partially presented in this paper. So, in this paper we present an extension of the DocBall visualisation metaphor already described in [2,16], that enables the visualisation of both Web pages and Web sites in a clear and consistent way. This metaphor has been applied to the design and implementation of a proof-of-concept Web search engine called Bubble [17], which enables a user to search the Web and visualise results at two levels: the Web site level and the Web page level. An initial evaluation of this approach shows that it enables user to make faster and better judgements than a standard ranked list on the relevance of a Web page returned by a search engine.

The paper is structured as follows. In Section 2 we further explain the rationale of our proposal, relating it to past relevant work. In Section 3 we remind the reader of the DocBall model we proposed in [2]. A more formal presentation of the model is reported in Section 4. This formalisation enables to extend the applicability of the model to Web pages and Web sites, as presented in Section 5. Based on this extensions a prototype system has been developed, which is presented in Section 6. The first results of an evaluation of the proposed result visualisation and the prototype system are reported in Section 7. Finally, Section 8 concludes the paper and outlines future work.

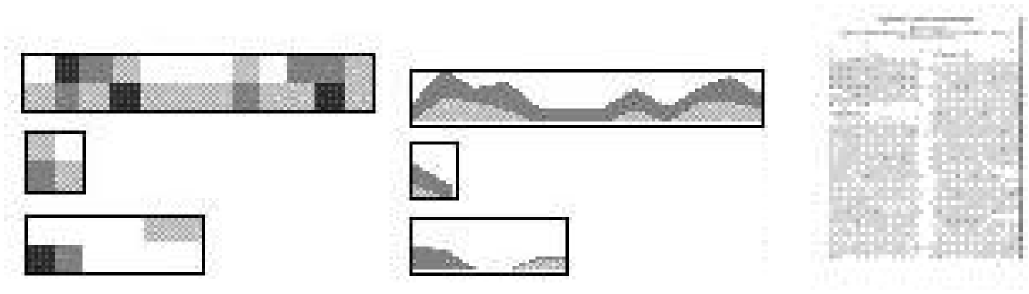
2 Putting Web Search Results in Web Site Context

One problem to deal with when the user is analysing the web search results is their granularity. Usually, web search engines returns the results at the document level, resulting, in almost all the cases, in a huge amount of results to be listed. This is related to the way the user interrogates the system. Usually, the user puts a few general words as query terms and obtains thousands of web pages as a response. There are very few things a user can do to reduce the cardinality of the result set. In almost all the cases, the user has to make a more specific query, either introducing additional or more specific query terms or using the allowed query syntax to express more accurately her information need. It is not infrequent that in this process the user ends up retrieving too few pages, or none, because the query was too specific.

We propose a way to reduce the cardinality of the result set without loosing recall in the process. We use information visualisation to achieve this. The basic idea is to add an additional logical level of results presentation before the Web page level: the Web site level. We propose a visualisation technique that presents, as a result of a query, first a list of Web sites found to be potentially relevant, and then, once a Web page in a Web site is selected, displays the structural elements that compose it. This enables a user to get first a view of a list of Web sites of potential interest, where each Web site is presented in a way that enables the user to get an idea of how many potentially relevant pages have been found in it. This provides a highly contextualised view of the search results at a site level. In addition, once a Web pages of one of the returned Web sites has been selected, the user can see which elements of the Web page have been found to be potentially relevant. Again, this provides a synthetic but highly contextualised view of the Web page that is particularly useful for long and/or complex Web pages.

Other document visualisation techniques have been proposed in IR with this objective in mind, but have not been applied to Web searching. The most important example are Tilebars, relevance curves and thumbnails.

Figure 1 shows and example of document representations using Tilebars, relevance curves and thumbnails. The first two representations show the relevance of each passage or segment of the text, using a row for each query term. The size of the documents is represented by the Tilebars or the relevance curve length. The passages can correspond to paragraphs or sections, or units of fixed arbitrary length. Both Tilebars and relevance curves show whether and where the document is relevant. However, the document structure is represented in a linear way, being it impossible to appreciate the hierarchical relations between the structural elements found in the document. The third document representation is the thumbnail. In this case no information about the size or the structure of the document is provided. The only information presented is related to the appear-



a) Tilebars b) Relevance Curves c) Thumbnails

Figure 1. Tilebars, Relevance Curves and Thumbnails.

ance of the document. Therefore, a thumbnail representation can complement the two first representations and it is useful when the user works frequently with the same document set and can recognise a document by its thumbnail view. The only information about the structure that the user can obtain from this representation is the information that can be inferred from the appearance of the different elements in the document, and this is not enough, since the user also needs information about the relevance of the different parts of the document.

We believe that providing as much information as possible, but with the lowest cognitive load, helps the user to assess in the best possible way the relevance of a document to her information need. The way we tried to accomplish this in the work presented here is by a user interface that extends to the Web the DocBall visual metaphor that we presented in [2]. The DocBall metaphor is used to represent in a consistent way both Web sites (WebSiteBalls) and Web pages (WebSiteBalls). In the next section we briefly remind the reader the main features of this metaphor.

3 The DocBall Metaphor

The DocBall is a visual metaphor that has proved useful to explain to the user the structure of an information item [3]. It can also be useful to represent in a graphical way the importance of each structural element of the document with respect to some criteria. In a DocBall, each ring represents a level of the hierarchical structure of the document, with the centre representing the entire document. In a ring, the regions represent the structural elements in the corresponding level. Inside a ring elements are presented in a clockwise order, following the order in which they appear in the document. Each region in the DocBall is coloured with regard to its estimated relevance: a hotter colour indicates more relevance, a colder colour represents less relevance. White regions have been estimated to be not relevant at all. Regions are ”clickable”, so that the DocBall can be used as a navigation tool for the document. See Figure 2 for a graphical representation of this process.

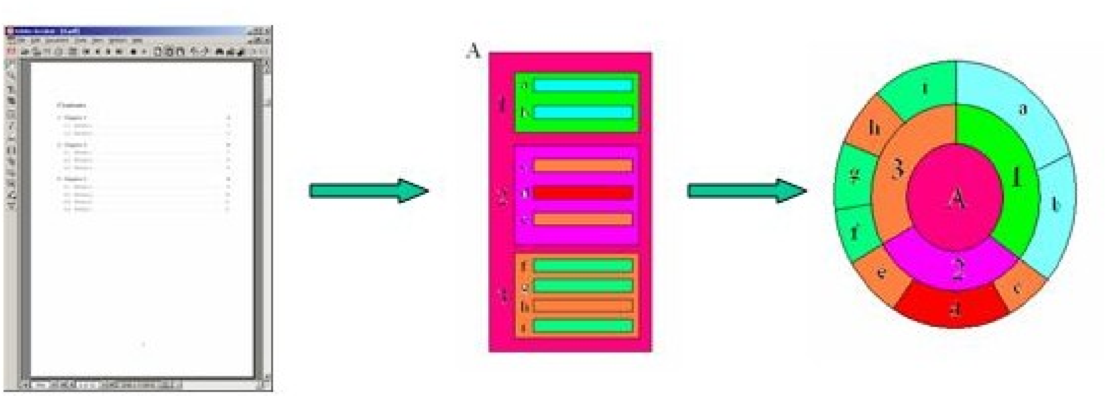


Figure 2. Visualisation of a structured document using a DocBall.

The DocBall representation can be considered as the representation of a tree structure, which depends on two parameters: the number of elements in the deeper structural level (the number of leaves) and the number of structural levels in the document (the height of the tree).

The DocBall metaphor has also been used to represent Web pages [17]. These are called WebDocBalls. For a Web page retrieved by a Web search engine, a DocBall representation can show which structural element has been estimated to be relevant to the user query. Figure 3 shows an example of WebDocBall representation of a Web page.

4 Formalisation of the DocBall Metaphor

The DocBall representation has been evaluated to be useful in complex and well structured environments, such as, for example, XML documents and HTML documents [2]. Before we describe how we can use the DocBall metaphor and its implementation (the DocBall widget) to represent Web sites, we should discuss the main characteristics of the context in which the DocBall is useful.

The main advantage of the DocBall is in its ability to show the structure of an information item. In fact, a DocBall can represent the inclusion relation of one or more elements in another one. In a DocBall, the containing element will be presented in a ring inner than the elements included in it. The lines going from the centre to the outside help to make this impression. The centre of the DocBall

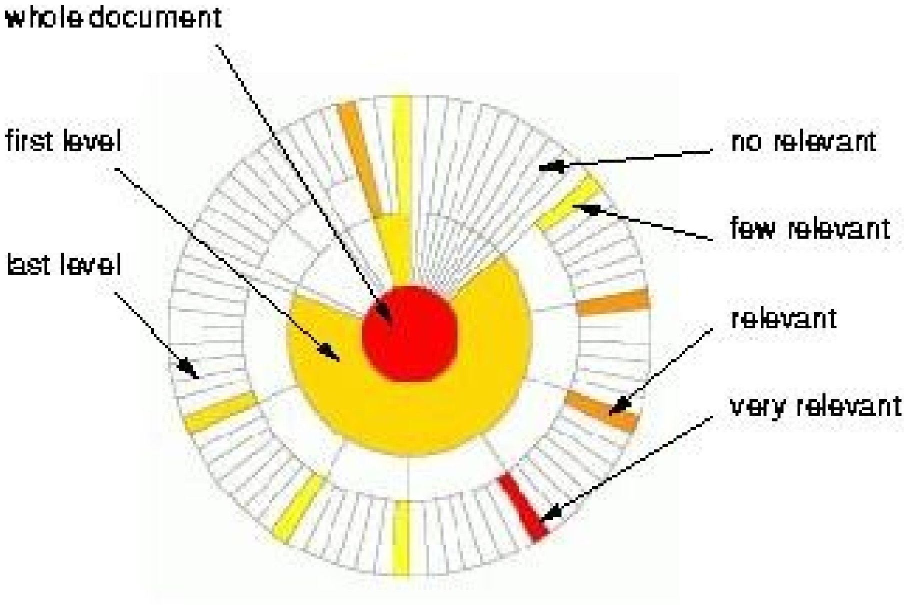


Figure 3. A WebDocBall.

represents the entire document, which includes the 360 degrees of an circle. Each ring (from inner to outer rings) is divided in sections, sharing the 360 degrees between them according to a metric function showing, for instance, the relative size of each structural element. These graphical properties can be formalised as follows.

Definition 1. *A DocBall is composed of a set of sections arranged in concentric rings. The cental ring represents the whole document and it is composed of only one section. The sections of the outermost rings represent the basic structural document elements, that is those that are not composed of any other elements. The rest of sections represents regular document elements.*

So we can define a DocBall representation as an pair $DB = \{S, Q\}$ *where:*

- $S = \{s_0, s_1, \cdots, s_n\}$ *is the set of sections,*
- $Q = \{q_0, q_1, \cdots, q_n\}$ *is the set of qualification values.*

Each s_i*, can be defined as a 3 element tuple,* $s_i = \{l_i, \alpha_i, \beta_i\}$ *where:*

- l_i *is the level of the section in the DocBall, where 0 is the centre of the ball.*
- α_i *is the angular degree at which the section begins, where the starting angle is the origin of the DocBall (at 12 hours).*
- β *is the angular degree at which the section ends.*

Each qualification value q_i refers to the relative section s_i. The value can be obtained by applying different qualification functions, such as, for example:

- *$f_{relevance}(s_i)$, which evaluates the relevance of section s_i with respect a query.*
- *$f_{type}(s_i)$, which determines the type of section s_i, for example text, image, audio, or more specific types like PDF, PS, etc.*
- *$f_{user}(s_i)$, which is determined by specific user actions, like for example relevance feedback.*

The choice of qualification function to apply depends on the task the system is performing. Obviously, if the system is displaying to the user the results of a query evaluation, then $f_{relevance}$ is used. However, if the system is displaying to the user the type of information contained in ones section, instead, then f_{type} is used. New qualification functions can be defined in relation to different tasks so, for example, a function can be defined to highlight sections that have recently be added or modified or to indicate to the user sections with a large number of links.

The hierarchical relations among the sections can be obtained directly from the above definition. So, it is straightforward to determine both the inclusion relation and the order relation between structural elements. This can be useful to determine the hierarchical relation between the elements that is needed to draw the DocBall.

Definition 2. *A section s_i is said to be an ancestor of a section s_j, with $s_j \subset^* s_i$, iif the area determined by the angles of s_j is included in the area delimited by the angles of s_i.*

$$\forall si, s_j \in S, s_j \subset^* s_i \Leftrightarrow (\alpha_i \leq \alpha_j) \wedge (\beta_j \leq \beta_i) \qquad (1)$$

Definition 3. *A section s_i is said to be the parent of section s_j, with $s_j \subset s_i$, iif the area determined by the angles of s_j is included in the area delimited by the angles of s_i and the level of s_i is exactly the level of s_j minus* 1.

$$\forall si, s_j \in S, s_j \subset s_i \Leftrightarrow (\alpha_i \leq \alpha_j) \wedge (\beta_j \leq \beta_i) \wedge l_j = l_i + 1 \qquad (2)$$

It is also necessary to have an order relationship between two sections, to establish which section comes first in the sequential linear order of the document.

Definition 4. *A section s_i is said to be the previous section to another section s_j, with $s_i \ll s_j$, iff the ending angle of s_i if smaller than the starting angle of the section s_j, and both section belong to the same level.*

$$\forall si, s_j \in S, s_i \ll s_j \Leftrightarrow (\beta_i < \alpha_j) \wedge (l_i = l_j) \tag{3}$$

Similarly one can define the *following section*.

Definition 5. *A section s_i is said to be exactly preceding section s_j, with $s_i < s_j$, iff there is no other section s_k which is a following section of s_i and a previous section of s_j.*

$$\forall si, s_j, s_k \in S, s_i < s_j \Leftrightarrow \nexists s_k | (s_i \ll s_k) \wedge (s_k \ll s_j) \tag{4}$$

Similarly, it is possible to define the relation of *exactly following.*

All these relations can be viewed in the example of the Figure 4, where the section s_1 is the parent of sections s_4 and s_5. And the section s_4 precedes section s_5.

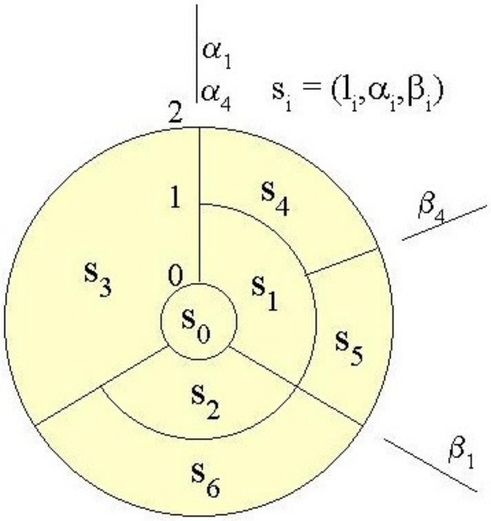


Figure 4. Example of a DocBall representation showing the relations among elements.

In addition, a DocBall not only enables to represent the structure of a document and the relationship between its sections, it also enables to qualify them. Qualification is related to the attribution of specific discrete or continuous properties, such as type or relevance. Qualification applies to each section of a DocBall. It is important to note that the qualification can either be done by applying the relative function to each section or it can be done in a way that takes into account the relationships between section, with a propagation function. This is made easier by the fact that sections are related in a hierarchical way. It is then possible to determine the quality of a section by applying the function to it and by considering the values of related sections, in particular the section ancestors. This can be expressed in the following way:

$$q_i = p(s_i, q_j, \ldots, q_m) | s_j, \ldots, s_m \subset^* s_i \qquad (5)$$

The choice of aggregation function p depends on how we decide to model and use the relationships between sections of the Web page. In [5], for example, we used a Bayesian Network to model the relationships between sections of hierarchically structured documents. This was then used to propagate relevance from the leaves of the structure to the root. In this case the aggregation operator p is defined as a Bayesian probability aggregation function. In [4], instead, we used two different aggregation functions, one to model the hierarchical relationships between sections (the same as in [5]), and one to model a utility function that is used to determine the best entry point for the Web page. This approach can also be used for any hierarchical structured document, like for example an XML document (see for example [11] for a different approach to the same problem). Section 5 reports on the aggregation function used in our first prototype system and in the evaluation reported in this paper.

The quality of a section could be graphically represented in several ways, such as for example with different colours, textures, labels or with a combination of them. This enables to represent in a DocBall a large amount of information about the document.

5 The DocBall Metaphor Applied to a Web Site

When we apply the DocBall representation to Web sites[1] the objective is to show to the user: 1) the structure of the Web site and, 2) which are the relevant pages in the site, and how relevant is the whole site. The first objective is very useful since it enables the user to have an idea of how the site is structured, its size in terms of number of pages, and where the relevant pages are located. This allows the user to have an overall view of the site but also to go directly to the most relevant pages in the site.

[1] The definition of Web site adopted in this work is pragmatic: a Web site is any URL that ends with the symbol "/". This is the same implicit definition used by several search engines, including Google.

A question that arises is the following. What could be more interesting for a user: a Web site with just one very highly relevant page, for example, or a site composed by many relevant pages with varying (but relatively low) degrees of relevance? In the context of Web searching, we think the user should/could be more interested in the last type of Web site because we expect relevant pages to be somehow grouped and not isolated among completely irrelevant pages. It is the nature of the Web to somehow cluster relevant pages together [1].

While the above hypothesis still needs to be proved, we believe it provides a strong motivation for the use of a WebSiteBall as a first step in the result presentation of the search process. In this way the user can choose which Web site to explore, while still leaving the user the possibility to go directly to the most relevant pages found by the search engine.

In order to apply the DocBall representation to a Web Site we need to gather the information needed to apply the DocBall model to the Web sites. So:

- S will be the set composed of the Web pages of the Web site, and the hierarchical relations among them will be determined by the links between the pages of a Web site. Initially, this is a directed connected graph, that is transformed into a tree using a depth-first algorithm.
- Q will have to be chosen in such a way that some characteristics of each page are presented to the user, in relation to the specific search objectives. If relevance is the characteristics to be presented than we need to be able to evaluate and present the relevance of each section, up to the relevance of the whole Web site, to qualify accordingly the section of the Web site.

In the following sections we will explain this is more detail.

5.1 Structure

In order to obtain a WebSiteBall from a Web site we need to transform the Web site traversal graph into a tree. To do so we have two possibilities: breadth-first and depth-first. Both techniques allow to visit all the nodes (Web pages) of the graph (Web site), but they differ in the order of the visit. To choose between them, we have to find which of the two produces a better representation of the site. Although it is impossible to generalise the way a Web site is constructed, we can make some general hypothesis that we think hold for most Web sites:

- Web sites are relatively homogenous set of Web pages, that is they are about loosely related topics. So linked Web pages are somehow topically related.

– Pages of the Web sites are linked with a large number of redundant links, for the user convenience.

To see which algorithm will produce a better path, let us study a very simple, but rather general example of a Web site. This Web site has just four index pages, one is used to provide access to the site, while the others are used to group distinct topics in the site. The Web site of the example is depicted in the Figure 5. The paths obtained when we apply the two traversal algorithms to the example Web site are reported in the Figure 6.

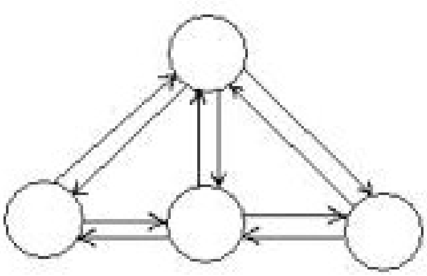
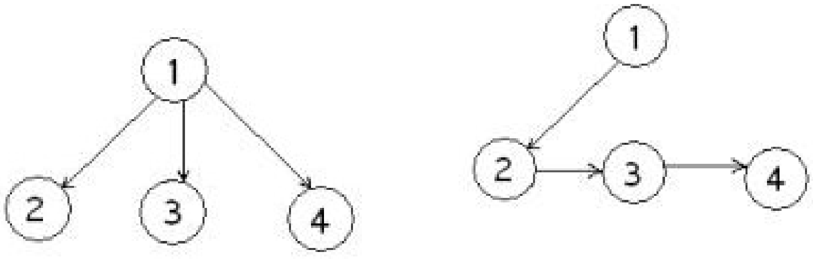


Figure 5. The graph of an web site.



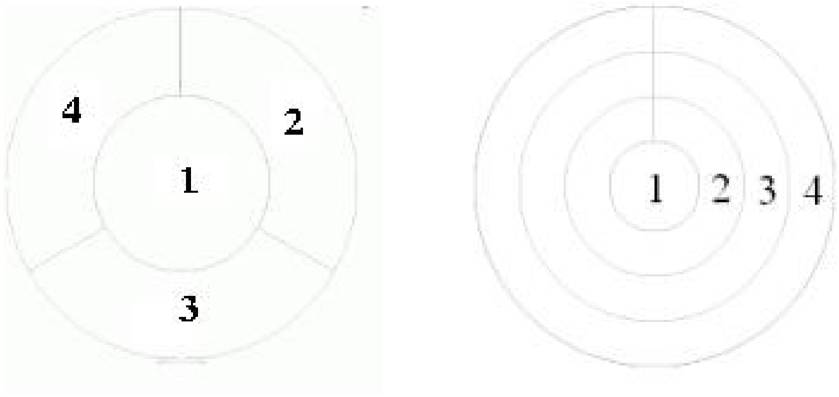
a) Breadth first tree.

b) Depth First tree.

Figure 6. The two different paths of the graph in the Figure 5.

Although both algorithms produce a tree structure, visiting all the nodes, there are important differences between the results. To help us understand which algorithm is more adequate to our case, let us draw the WebSiteBall representations of both and study them. The WebSiteBalls are reported in Figure 7. Each number relates the sections of the WebSiteBall with the pages in the example Web site. In Figure 7 we can see that both representations of the original Web site are valid, but the WebSiteBall on the left, obtained applying the breadth-first algorithm to the Web site graph, represents better the structure of the original Web site.

It is more intuitive and clearer to understand, as our experimentation reported in [2] has shown. We therefore decided to apply the breadth-first algorithm to transform the graph of a Web sites into a tree.



a) Breadth Fist WebSiteBall

b) Depth First WebSiteBall

Figure 7. Two different WebSiteBalls of the web site in Figure 5.

5.2 Qualification

While many different qualification of a section of Web site are possible, as explained in Section 3, the most important one in the context of a user Web search is relevance. Different ways of estimating relevance have been proposed in IR and different models can be implemented in our qualification function. However, at this stage, our study is not concerned with finding the best relevance estimate for a Web page or a Web site, but with the visualisation of the search results. So, in this section we explain the model we used for the evaluation reported later in this paper, without making any claim on its actual effectiveness.

In order to qualify in relation to relevance the sections representing a Web site we used a modified version of the Vector Space Model. The model evaluates the relevance only of pages at the outmost level of the WebSiteBall and then propagates the relevance up to the top of the hierarchy, to the entire site. This propagation could be more or less complex, as it can take into account several different aspects related to the hierarchical relations. For example, it could consider where is the origin of a link (in a title, in a paragraph, etc.) or the number of links to a page to fine tune the propagation function. We have only started to explore these issues. So, in this first prototype, we used a very simple propagation function that does not propagate the relevance over the entire hierarchical structure (each page's relevance is evaluated independently), but just computes it for the entire site using the average relevance value of all Web pages in a site. This can be expressed in the following equation:

$$q_i = \begin{cases} f_{relevance}(s_i) \text{ if } l_i \neq 0 \\ \frac{\sum_{i=1}^{n} q_i}{n} \qquad \text{otherwise} \end{cases} \tag{6}$$

As already mentioned, we are fully aware that this function is very simple and almost ”naïve”, but at this stage we are concentrating more on the representation power of the WebSiteBall and WebDocBall metaphor and far less on the best qualification function. More work will be devoted to this aspect in the future.

6 The Bubble System

In this section we introduce a Web search engine called *Bubble* that uses WebSiteBalls and WebDocBalls to represent the results of a search. Since this is just a tool for our exploration of the representation power of the DocBall metaphor, a detailed description of the system is outside the scope of this paper. Instead, we will only highlight some of the characteristics of this system that show how WebSiteBalls and WebDocBalls can be used for Web searching.

Bubble's architecture is that of a typical client-server system. On the server side, there is: 1) a module that carries out the crawling, from which the system obtains the Web pages of the sites to index, 2) a module for indexing, that indexes the pages found by the crawler, and 3) a module for searching, that searches the index to find pages relevant to a user query. The system is built in Java, and uses Servlets technology to integrate client and server sides.

The client uses graphics in SVG [7], which enables to create complex graphical representation without overloading the communication between client and server, which is based on XML. The visualisation of the graphics is delegated to a plug-in and the server only builds the structure of the DocBall using XML and sends it to the client. In this way we have separated the drawing of the WebSiteBall (the structure) from its colouring (the qualification). The drawing can be done at indexing time, leaving the colouring phase to the search time, so as to reflect the estimated relevance of each section, obtaining in this way better response times. The colouring phase is done using CSSs, that are written to answer a query and are applied to the DocBall at the client side. The process is depicted in Figure 8.

Now, we will give a brief example of user interaction with the Bubble system to show how the DocBall metaphor can be effectively used for Web searching and browsing.

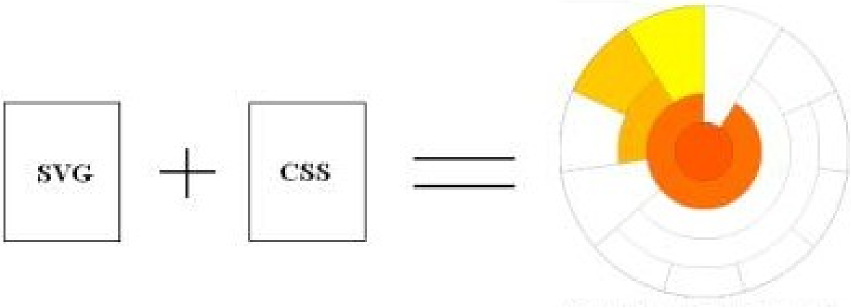


Figure 8. The process of applying a CSS to a WebSiteBall drawing with SVG.

This interaction begins when a user writes a few words in the query form and submits the query to the server. The user can choose among three different query objectives: 1) Web sites, 2) Web pages, 3) specific file types in Web pages. Different colour codings are used to help the user differentiate between the three different kinds of search.

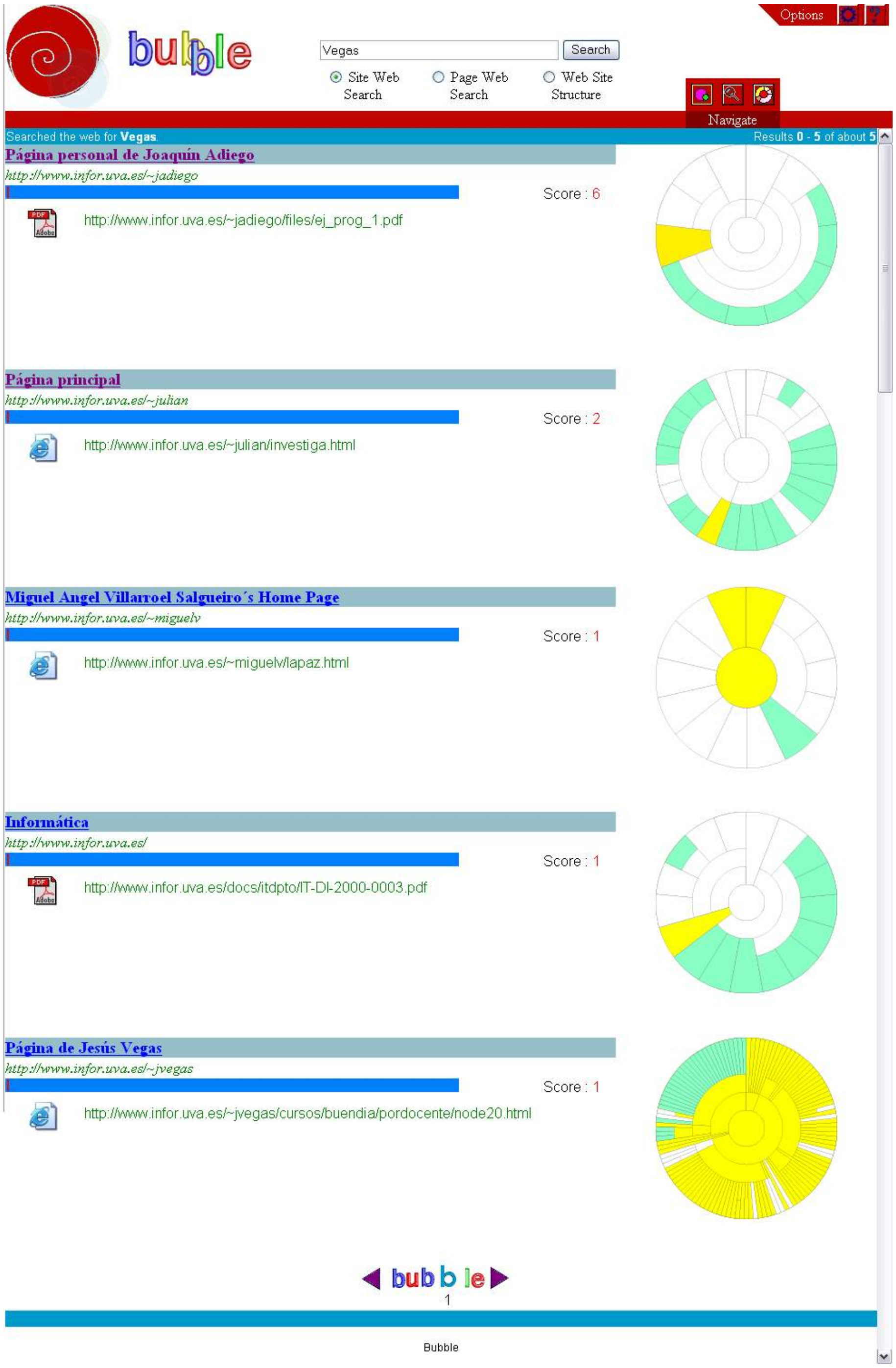
When a user chooses to search for Web sites, in response to the user query Bubble displays different WebSiteBall representations of the sites retrieved, ordered by the overall relevance of the sites (Figure 9). The user can then choose to go directly to explore a Web site, or go directly to explore a specific page of a site. Alternatively, the user could ask to see more closely the WebDocBall of a Web page (Figure 10), to see its structure and to check for the different file types included in it.

If the user chooses to search for Web pages relevant to a query, Bubble will produce a list of WebDocBalls representations of pages estimated to be relevant (Figure 11). The user could go directly to a Web page or visit any part of it by clicking the WebDocBall.

Finally, if the user chooses to search for specific file types (e.g. PDF, Postscript, jpg, wav, etc.) it can just ask for the WebSiteBall to display the different file types (Figure 12). This possibility could be very useful if the user is interested in specific media types. In fact, most multimedia files cannot be effectively indexed by search engines, so the fact that one of these files is close to a section that has been estimated to be relevant could be an indication of its possible relevance too.

There are also other interesting functionalities available in Bubble, such as, for example, the possibility to enlarge (zoom) a WebSiteBall or a WebDocBall. Also, it is possible to customise the presentation of the results in a grid of WebSiteBalls, a list of regular WebSiteBalls or list of small WebSiteBalls.

In summary, Bubble is a very powerful prototype tool for Web search result presentation and exploration. It enables the user to effectively and efficiently



Vegas

● Site Web Search ○ Page Web Search ○ Web Site Structure

Searched the web for **Vegas**. Results **0** - **5** of about **5**

Página personal de Joaquín Adiego
http://www.infor.uva.es/~jadiego
Score : 6
http://www.infor.uva.es/~jadiego/files/ej_prog_1.pdf

Página principal
http://www.infor.uva.es/~julian
Score : 2
http://www.infor.uva.es/~julian/investiga.html

Miguel Angel Villarroel Salgueiro´s Home Page
http://www.infor.uva.es/~miguelv
Score : 1
http://www.infor.uva.es/~miguelv/lapaz.html

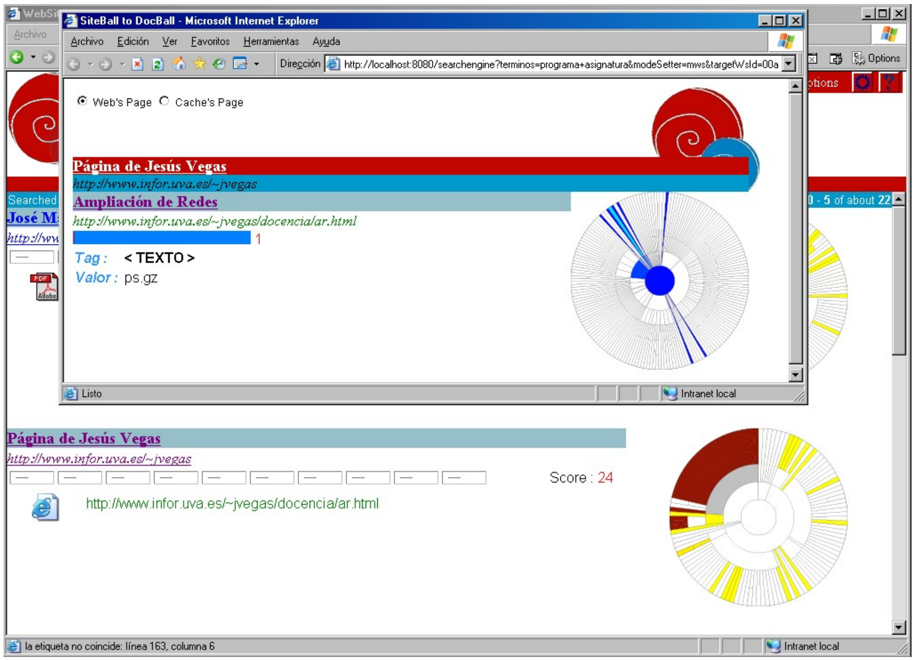
Informática
http://www.infor.uva.es/
Score : 1
http://www.infor.uva.es/docs/itdpto/IT-DI-2000-0003.pdf

Página de Jesús Vegas
http://www.infor.uva.es/~jvegas
Score : 1
http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html

1

Bubble

Figure 9. Searching for Web sites.


SiteBall to DocBall - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Dirección http://localhost:8080/searchengine?terminos=programa+asignatura&modeSetter=mws&targetWsId=00a
Options
○ Web's Page ○ Cache's Page
Listo
Intranet local
la etiqueta no coincide: línea 163, columna 6

Página de Jesús Vegas
http://www.infor.uva.es/~jvegas
Ampliación de Redes
http://www.infor.uva.es/~jvegas/docencia/ar.html
1
Tag : < TEXTO >
Valor : ps.gz

Página de Jesús Vegas
http://www.infor.uva.es/~jvegas
Score : 24
http://www.infor.uva.es/~jvegas/docencia/ar.html

Figure 10. Viewing a WebDocBall from a WebSiteBall.

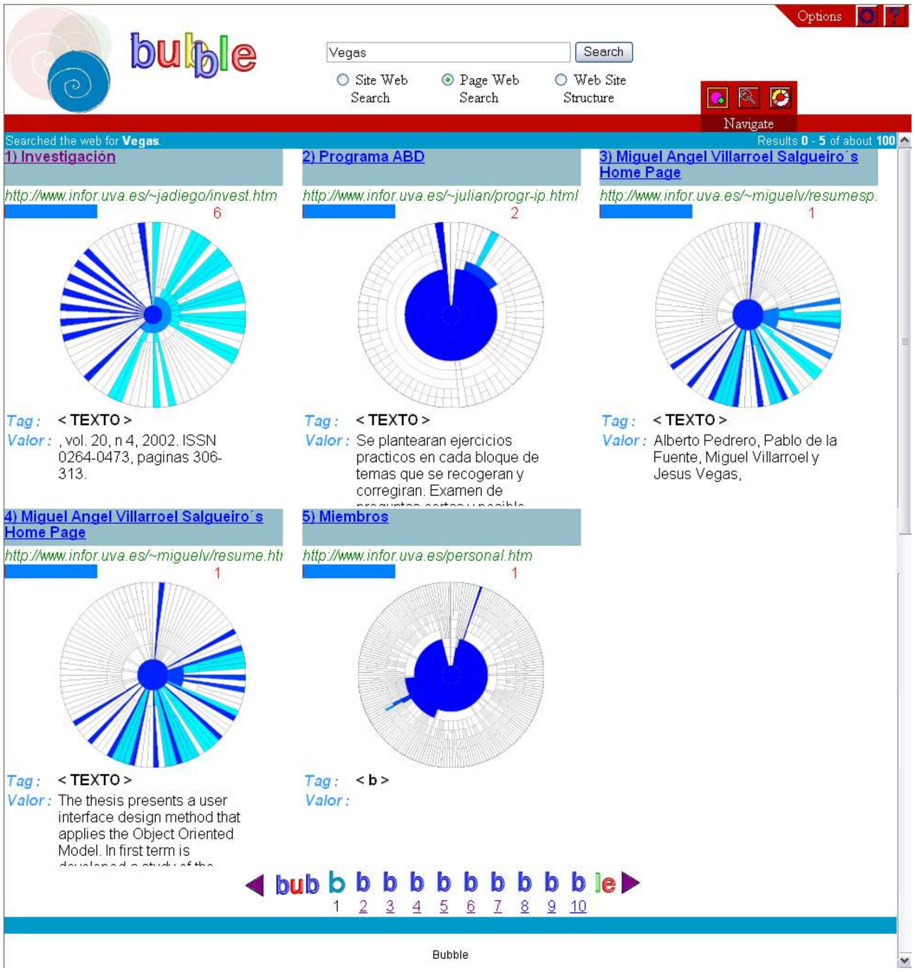
explore the results of a search performed using a standard search engine (so that Bubble could be used to present the result of any search engine), maximising the use of the user search time.

In order to prove the usefulness of the WebDocBall and WebSiteBall representations implemented in Bubble we carried out an evaluation. Next section reports on a small evaluation we carried out which enabled us to conclude that such results presentation for Web searching is indeed effective.

7 Evaluation

We evaluated the DocBall metaphor and the Bubble system using a user and task oriented approach [10]. We believe a user and task oriented evaluation to be the only suitable way to evaluate our prototype system, since no bench style evaluation (i.e. à la TREC) would enable to address issues related to expressiveness, clarity, complexity and usefulness of the proposed Web page and Web site results representations.

We have indexed a large Web site, the full site of the University of Strathclyde, and asked 10 users to locate pages relevant to some user defined task. Users were postgraduate students and members of staff of the University of Strathclyde and



Searched the web for **Vegas**.

Results **0** - **5** of about **100**

1) Investigacion

http://www.infor.uva.es/~jadiego/invest.htm

6

Tag : < TEXTO >

Valor : , vol. 20, n 4, 2002. ISSN 0264-0473, paginas 306-313.

2) Programa ABD

http://www.infor.uva.es/~julian/progr-ip.html

2

Tag : < TEXTO >

Valor : Se plantearan ejercicios practicos en cada bloque de temas que se recogeran y corregiran. Examen de

3) Miguel Angel Villarroel Salgueiro´s Home Page

http://www.infor.uva.es/~miguelv/resumesp.

1

Tag : < TEXTO >

Valor : Alberto Pedrero, Pablo de la Fuente, Miguel Villarroel y Jesus Vegas,

4) Miguel Angel Villarroel Salgueiro´s Home Page

http://www.infor.uva.es/~miguelv/resume.htr

1

Tag : < TEXTO >

Valor : The thesis presents a user interface design method that applies the Object Oriented Model. In first term is

5) Miembros

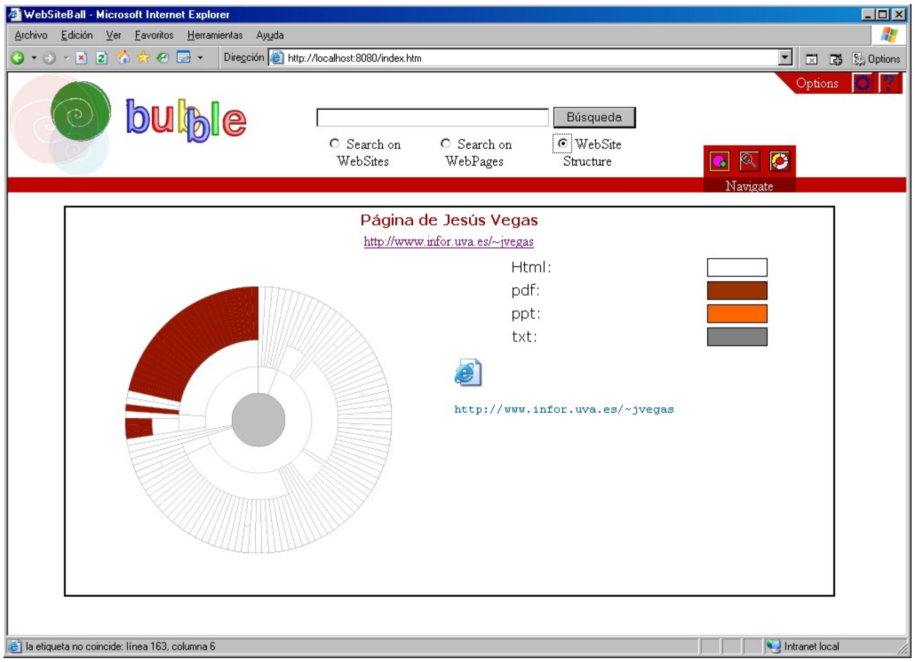
http://www.infor.uva.es/personal.htm

1

Tag : < b >

Valor :

Figure 11. Searching web pages with the Bubble system.



Dirección: http://localhost:8080/index.htm

○ Search on WebSites ○ Search on WebPages ● WebSite Structure

la etiqueta no coincide: línea 163, columna 6

Figure 12. Viewing the structure of a web site.

tasks were typical search tasks that these users were used to routinely carry out on almost a daily basis. Here are some examples of these tasks:

T2: find the pages of the department of chemistry that describe spin off companies from that department;

T6: find how many information science journals the university library subscribes to for online access;

T8: find out if it would be possible and how much it would cost for somebody not affiliated with the university to use the university sport facilities.

Ten tasks were selected from a set of more than forty proposed by the users themselves and each of the 10 users carried out each of the 10 tasks. Users were asked to carry out the search tasks with Bubble and with an alternative system that presented the results as a flat list of Web pages and were asked to locate pages containing information relevant to their information need. Both systems used the same model to estimate relevance, the one reported in Section 5, so that the same documents were presented to the user for the same query. Each user carried out 5 tasks using Bubble and 5 using the alternative system. The experimental design ensured that each task was performed a total of 5 times with Bubble and 5 times with the alternative system. Half of the users were instructed to use Bubble first, while the other half started by using the alternative system. The search tasks were carried out one after the other in the same session. An instructor was present during all search sessions, observing the user search behaviour and taking notes. Users were asked to fill an evaluation form at the end of each task. In the form users were asked to grade on a Likert scale of 4 values the following questions:

Q1: Rate how much you were able to accomplish the task;
Q2: Rate the complexity of the task completion;
Q3: Rate the complexity of the use of the system for this task;
Q4: Rate the complexity of the interpretation of the search results for this task;
Q5: Rate how the result presentation helped you find the information sought for this task.

A detailed analysis of the results of the evaluation is beyond the scope of this paper, so we only report here a summary of these results.

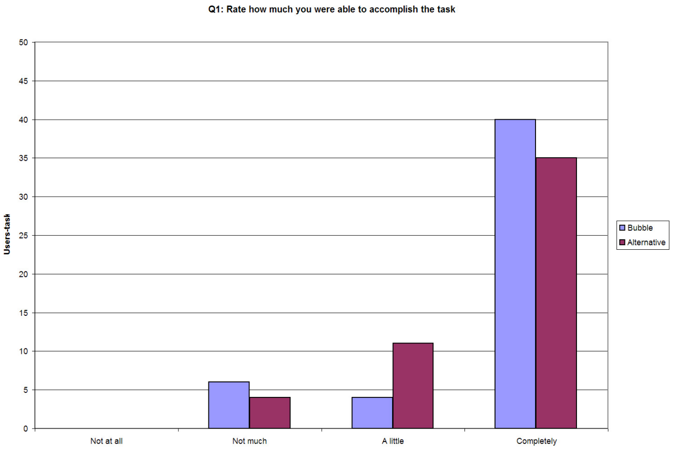


Figure 13. Evaluation question 1: Rate how much you were able to accomplish the task.

Figure 13 reports the results for Q1. This figure, like all the other that follow, shows the distribution of user-task Likert scores across all the 50 pairs user-task sessions (the number of users multiplied by the number of tasks) recorded respectively for Bubble and for the alternative system. As it can be seen, almost all users were able to accomplish in a complete or almost complete way all tasks. Only in a few cases users perceived that they did not fully accomplish the tasks. The number of user-task cases that users stated to have completely accomplished the task is significantly higher among those using Bubble that those using the alternative system. The number of user-tasks ”not much” accomplished is higher for Bubble than for the alternative system but the difference is not statistically significant.

Figure 14 reports the perceived complexity of the task completion (Q2). Some of the tasks were indeed rather complex and finding the sought information was

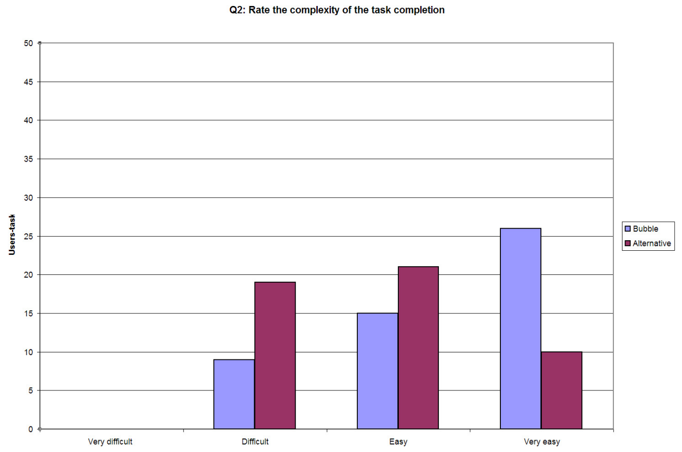


Figure 14. Evaluation question 2: Rate the complexity of the task completion.

difficult. An example of these tasks is the following: ”find out who has been teaching the applied physiology class last year”. For some of these tasks the alternative system, with its ranked list style presentation of results resulted in a lot more browsing than for Bubble, making the user perceive that it was difficult. Nevertheless, if we group together the user-task sessions indicated as easy or very easy, the difference between Bubble and the alternative system is much less significant.

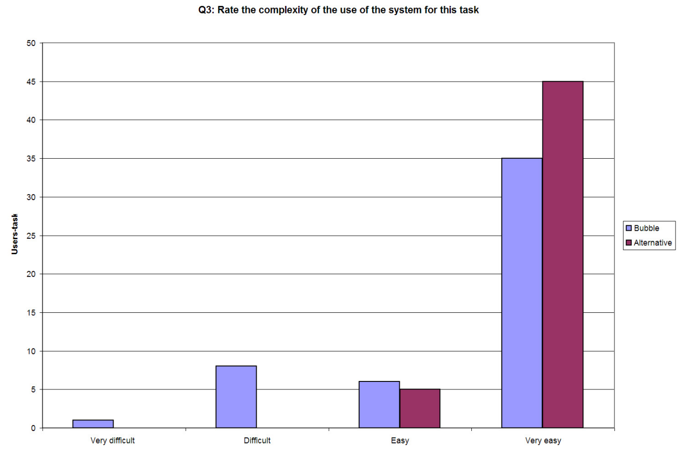


Figure 15. Evaluation question 3: Rate the complexity of the use of the system.

Figure 15 shows how users rated the complexity of the use of the system for each user-task session (Q3). Here Bubble did not score very well for some users. In fact, for some user-tasks Bubble was perceived as being very difficult (one case) or difficult (8 cases). The alternative system was perceived in almost every user-task case to be very easy. This is certainly due to the user familiarity with Web search systems like Google or Yahoo!, that have the same kind of user interaction.

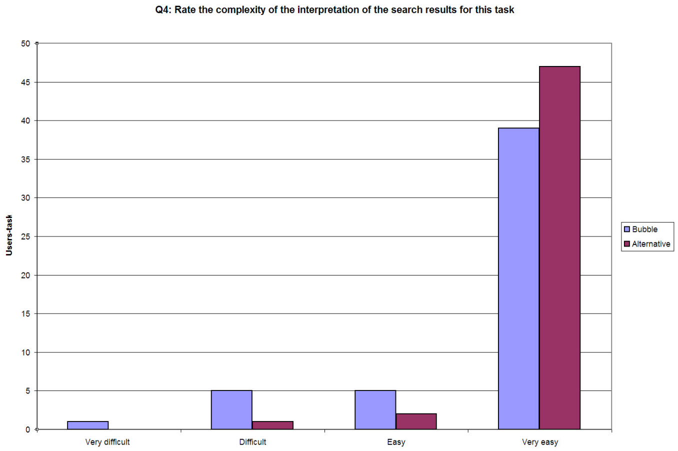


Figure 16. Evaluation question 4: Rate the complexity of the interpretation of the search results.

Figure 16 shows how users rated the complexity of the interpretation of the search results using Bubble or the alternative system (Q4). The data show that for some user-task sessions Bubble was perceived as very difficult (one case) or difficult (5 cases). Although it is true that the alternative system was perceived as easy or very easy in almost all cases (except one), it should be noted that in 44 cases out of 50 Bubble was perceived easy or very easy too. While this result is lower than that scored by the alternative system, we believe that it shows that for the vast majority of user-task cases users found it easy to interpret Bubble result visualisation.

Finally, Figure 17 reports the sum of the Likert scores relative to the user rating of how the result presentation helped in finding the information sought for the task (Q5). It shows that in a large number of user-task cases (35 over 50) the users thought that Bubble helped them a lot in finding the information sought. This is a significantly larger number than that recorded for the alternative system (18 cases). Even summing together the cases were ”a little” and ”a lot” of help was perceived as provided by the result presentation metaphor, Bubble still outperform the alternative system. Most users that found Bubble superior to the alternative system commented very positively on the fact that Bubble made it easier to locate and assess relevant pages because the pages were shown in context. For example, knowing that a page reporting information about regulation for student final year project submission was located in the University administration part of the Web site made it easier to assess it as relevant than if found under a student Web site. Also, if the page was located inside the Web pages of

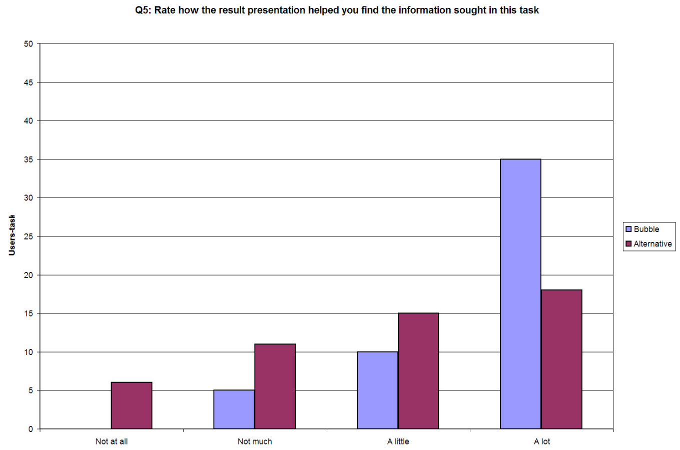


Figure 17. Evaluation question 5: Rate how the result presentation helped you find the information sought.

a specific department that easily qualified that page as containing specific and trustworthy information for that department. This information was available at a glance in Bubble and not so apparent in the alternative system.

While small in the number of users involved and tasks considered, this first evaluation enables us to conclude that the WebDocBall and WebSiteBall metaphors, as implemented in Bubble, are quite effective and powerful ways of results presentation that enable the user, by showing the context in which relevant information lies, to browse results and assess their relevance in a fast and easy way.

8 Conclusions and Future Work

This paper reported on an extension of the DocBall metaphor to represent Web pages and Web sites. We have formalised the DocBall metaphor enabling to adapt it to Web pages and Web sites after their graph structure has been reduced to a tree structure. A prototype system, Bubble, has been developed to facilitate the exploration and evaluation of the usefulness of WebDocBalls and WebSiteBalls metaphors. An initial evaluation of the result presentation metaphor using Bubble shows that it is effective and easy to use.

We believe it is very important to design and develop information visualisation techniques and tools that enable to represent the context in which a document estimated to be relevant lies. This is particularly important in Web searching, given the complexity of the task and the complexity and ambiguity of the relations existing between Web pages in a Web site and between elements of a Web page. The work presented in this paper is a step in this direction.

The next step of our work will be a more thorough analysis of the evaluation results and a redesign of Bubble, following a formative design approach. We believe useful insights into the usefulness of the DocBall metaphor can still be drawn from the evaluation. This will enable us to develop a new improved version of the Bubble system for further experimentation.

A second but not less important step will be the design of a more powerful relevance propagation algorithm that should enable a more precise and fast estimate of the relevance of a Web site from its pages. Work in this direction has only started, but interesting results have already been achieved using an approach based on Bayesian Networks [4,5].

Acknowledgements

This work was partially supported by the projects VA098/03 from Junta de Castilla y Len and TIC2003-09268 from MCyT (Spain).

References

1. A. Barabasi. *Linked: The New Science of Networks.* Plume, New York, 2003
2. F. Crestani, P. de la Fuente, and J. Vegas. Design of a graphical user interface for structured documents retrieval. In *Proceedings of SPIRE 2001*, pages 246–249, 2001.
3. F. Crestani, J. Vegas, and P. de la Fuente. A graphical user interface for the retrieval of hierarchically structured document. *Information Processing & Management*, 40(2):269–289, 2004.
4. F. Crestani, L.M. de Campos, J.M. Fernandez-Luna, and J.F. Huete. A Multi-layered Bayesian Network Model for Structured Document Retrieval. In *Proceedings of ECSQARU Conference 2003*, pages 74-86, 2003.
5. F. Crestani, L.M. de Campos, J.M. Fernandez-Luna, and J.F. Huete. Ranking Structured Documents Using Utility Theory in the Bayesian Network Retrieval Model. In *Proceedings of the SPIRE 2003*, pages 168-182, 2003.
6. B. Diebold and M/ Kaufmann. Usage-based visualization of web localities. In *Proceedings of Australian Symposium on Information Visualization.* Australian Computer Society, 2001.
7. SVG Working Group. Scalable vector graphics (svg) 1.1 specification. candidate recommendation. Web, 2002. http://www.w3.org/TR/2002/CR-SVG11-20020430/.
8. T.M. Mann. Visualization of WWW-search results. In *DEXA Workshop*, pages 264–268, 1999.
9. S. Mukherjea. Organizing topic-specific web information. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 133–141. ACM Press, 2000.
10. J. Reid and M.D. Dunlop. Evaluation of a prototype interface for structured document retrieval. In *Proceedings of HCI 2003*, 2003.
11. J. Reid, M. Lalmas, K. Finesilver and M. Hertzum. Best Entry Points for Structured Document Retrieval. Part I - Characteristics. *Information Processing and Management*, 42(1): 74–88, 2005.
12. H. Reiterer, G. Mu*β*ler, T.M. Mann, and S. Handschuh. Insyder - an information assistant for business intelligence. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–119. ACM Press, 2000.
13. J.C. Roberts and E. Suvanaphen. Visual bracketing for web search result visualization. In *Seventh International Conference on Information Visualization (IV'03)*, pages 264–268, 2003.
14. B. Shneiderman. *Designing the User Interface. Strategies for Effective Human-Computer Interaction.* Addison-Wesley, 1992.

15. E.G. Toms and R. Taves. Measuring user perceptions of web site reputation. *Information Processing & Management*, 40(2):291–317, 2004.
16. J. Vegas, P. de la Fuente, and F. Crestani. A graphical user interface for structured document retrieval. In *Proceedings of ECIR 2002, BCS-IRSG European Colloquium in Information Retrieval Research*, pages 268–283, Glasgow, UK, March 2002.
17. J. Vegas, P. de la Fuente, and F. Crestani. WebDocBall: A Graphical Visualisation Tool for Web Search Results. In *Proceedings of ECIR 2003, BCS-IRSG European Conference in Information Retrieval Research*, pages 351–362, Pisa, Italy, April 2003.
18. Ryen W. White, Joemon M. Jose, and Ian Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Information Processing & Management*, 39(5):707–733, 2003.