REFINING SEARCH QUERIES WITH SVG GRAPHICS IN THE OMNIPAPER PROJECT

SEBASTIEN THYS¹; BERT PAEPEN¹; JAN ENGELEN¹

¹Research Group on Document Architectures, Department of Electrical Engineering, ESAT, Katholieke Universiteit Leuven Kasteelpark Arenberg 10, 3001 Heverlee, Belgium e-mail: sebastien.thys@esat.kuleuven.ac.be; bert.paepen@esat.kuleuven.ac.be; jan.engelen@esat.kuleuven.ac.be

The IST-funded European project OmniPaper[1] is investigating techniques to obtain a novel online news experience. These include XML- and Artificial Intelligence related technologies. The OmniPaper architecture starts from distributed news archives, all within different operating environments, database formats and indexing mechanisms. SOAP (Simple Object Access Protocol)[2] is used to create a uniform access method to these archives. Rich indexing and meta-data structures, such as Topic Maps[3], make intelligent search possible. A cross-archive intelligent index (or ‘knowledge layer’) contains concepts, relationships between them and occurrences (articles) in different languages. The actual OmniPaper prototype gives the user a graphical relational view on concepts relevant to the current query. The relation view is a representation of a small piece of the web of concepts. Using this highly interactive view, the user can refocus his query using different word senses of the query keywords. This allows the user to redefine/refine his query if one of the query keywords has different possible meanings (e.g. bank as a financial institution or as a piece of furniture). The relational view allows the user to navigate between different concepts relevant to the query.

Keywords: OmniPaper; SVG; DOM; topic maps; EuroWordNet; visualization; graphics; newspaper

INTRODUCTION

Since the emerging boom of the Internet a lot of newspapers are being published electronically. In spite of this increasing availability of information, news items remain scattered throughout various archives, countries and languages. Furthermore, the distributed electronic information has different data structures, storage formats and access methods. Searching for news is still mostly done the “brute force”-way using full-text search robots and search result quality highly depends on the sophistication of the user’s search input. The net result is that finding news from various international newspapers still is easier in an airport news stand than on the Internet.

The IST-funded European project OmniPaper is investigating techniques to obtain a novel online news experience. These include XML- and Artificial Intelligence related technologies. The OmniPaper architecture starts from distributed news archives, all within different operating environments, database formats and indexing mechanisms. SOAP (Simple Object Access Protocol) is used to create a uniform access method to these archives. Rich indexing and meta-data structures, such as Topic Maps, make intelligent search possible. A cross-archive intelligent index (or ‘knowledge layer’) contains concepts, relationships between them and occurrences (articles) in different languages.

According to Steve Pepper[4], XTM Topic Maps will become the answer for organizing and navigating through large and continuously growing information pools. They provide a “bridge” between the domains of knowledge representation and information management. This standard defines both an abstract data model and a serialization syntax to represent knowledge structures.
Refining search queries with SVG graphics in the OMNIPAPER project

and to link them to information resources. The topic map’s basic building blocks are: Topics, Occurrences of topics and relationships (Associations) between topics. They are often referred to as the TAO of topic maps.

The OmniPaper information pool consists of two the different kinds of information: an EuroWordNet[5] ontology and a vast amount of news articles. EuroWordNet is a multilingual lexical database containing number of European languages. EuroWordNet is a European resources and development project supported by the Human Language Technology sector of the Telematics Applications Programme (LE-2 4003 & LE-4 8328).

In this database, keywords are semantically grouped into synsets (synonym sets) and a number of semantic relations between these synsets are established. A synset (or concept in OmniPaper terminology) in fact corresponds to the digital representation of a real-world entity, object, person, idea. It consists of a set of keywords in a specific sense, that all have similar meaning. In the OmniPaper project the EuroWordNet ontology is stored in a topic map. In this topic map the keywords and concepts are topics and the relations between the concepts are represented by associations in the topic map. This topic map represents what is called the web of concepts in the OmniPaper project. Finally the news articles are attached to one or more concepts that are considered relevant to the article. In Topic Map terminology the articles are occurrences of the topics. Of course an article can be attached to different concepts.

The actual OmniPaper prototype gives the user a relational view on concepts relevant to the current query. The relation view is a representation of a small piece of the web of concepts. Using this view, the user can refocus his query using different word senses of the query keywords. This allows the user to redefine/refine his query if one of the query keywords has different possible meanings (e.g. bank as a financial institution or as a piece of furniture). The relational view allows the user to navigate between different concepts relevant to the query.

The relational view can be viewed in textual format or in a more interactive graphical format. The OmniPaper consortium has chosen to implement a graphical view using SVG[6] technology. SVG is believed to be a good choice for this because it offers a lot of scripting and animation possibilities and it is scalable. Moreover, this scalability increases the accessibility: being able to zoom in and zoom out of an image means that someone with limited vision can blow up part of an image to get a better view. Another interesting feature of SVG is styling. One can define multiple stylesheets for the same SVG document. This way a partially sighted user may be able to see an image more clearly with a particular high-contrast colour scheme.

The most important aspect of the graphical SVG visualization is that it is highly interactive. The user can put interesting concepts in the foreground, he can delete irrelevant concepts from the graph and expand interesting ones. He can also select concepts and display their description combined with other information. Concepts, keywords and relations between them are symbolized by different shapes and styles according to their type.

METHODOLOGY

Each search request consist of different steps. First the search string entered by the user is processed. Stop words are filtered out and the remaining words are stemmed resulting in a list of keywords. Each keyword is looked up in the topic map. If the keyword is attached to only one concept, that concept is returned, otherwise a number of related concepts is returned to disambiguate the keyword. This collection of topics (both distinct concepts and keywords together with their related concepts) serves as a starting point for the SVG view. As explained the user is
able to expand every concept to discover its related concepts to refine/redefine the query. An expansion means an updating of the SVG view. Different possible approaches can be thought of. A first option is doing the processing server side and sending a new SVG document to the client. This approach has a serious drawback because users are allowed to drag and drop topics. The positions of the topics must be maintained and thus communicated to the server. Another drawback is the rather annoying reloading of the document on the client-side. A second approach is the opposite: keeping a in memory representation of the whole topic map on the client and updating the SVG view fully through scripting. Looking at the size of the EuroWordNet topic map this is definitely not a good choice. The third approach is the one that the OmniPaper consortium has adopted. It consists of a good mix of both previous solutions. The EuroWordNet topic map is stored on the server. For each search query, information needed for the view is extracted from the EuroWordNet topic map and sent back in XML format to the SVG viewer. The client reads in this response, parses the XML to obtain a DOM representation of the XML fragment, processes the DOM representation and finally stores the information in ECMAScript objects. The whole view is controlled through client side scripting but when some new information is needed a request is fired to the server which responds to the client to enrich his knowledge. An example of such a situation is the expansion of a topic to discover its related concepts.

The concepts and the keywords resulting directly from the request query string are considered “focus topics” and are centred in the view. Keywords are surrounded by their related concepts. To narrow the search, or in other words to decrease the number of articles returned, the user must disambiguate the remaining keyword(s) by choosing one of the surrounding related concepts. If done, that concept replaces the keyword en the other concepts attached to that keyword are removed from the view. In the ideal case the user select for each keyword (if any remains from the search result) a concept that matches his meaning of the keyword to narrow the search as much as possible. When every keyword has been disambiguated there are only concepts left in the view. These can be expanded to show relationships with other concepts. A non-“focus concept” can be selected to make it a “focus topic”. Changing the “focus topics” updates the result list of relevant articles. Concepts that are estimated non relevant by the user can be removed.

EuroWordNet contains a large amount of synsets (concepts), therefore it is essential to select only relevant information as it is impossible to display the whole data efficiently. Filtering techniques are needed in order to select and display only relevant information. As the problem still holds for the textual view, this operation is done server side, so we can assume that the amount of related concepts is limited in the SVG view.

EuroWordNet also contains a large amount of associations types that can exist between concepts. In this implementation only four relation types have been retained, whereas the original WordNet distinguished about 70 different types. For our search and navigation purposes however, we don’t need such fine-grained semantic differences, and therefore all relations have been reduced to one of these four higher level types: “is more specific than”, “is more general than”, “is equivalent with” and “is associated with”.

Finally the new SVG 1.2 XML-based extension mechanism RCC seems very interesting for the representation of concepts and keywords. Rendering Custom Content (RCC) is a new technology featured in the latest drafts of SVG 1.2 and offers a new framework for allowing seamless integration of custom XML grammars documents as well-defined extensions. Using RCC, a foreign namespaced element can be specified how to behave in an SVG document.
RESULTS

THE VIEW

A result of the query “Iraq OR weapons” is depicted in Figure 1. The different keywords and concepts found in the query are shown. Keywords and concepts are represented by rounded rectangles with buttons for expansion, remove and selection which are explained later on. Keywords are green, unfocused concepts are yellow; concepts that are brought into focus are pink. Keywords and concepts use a word to distinguish them from other topics. The word displayed for a keyword is the keyword itself, in case of a concept the word used in the view is the preferred keyword of the concept.

FIGURE 1: SVG VIEW REPRESENTING THE QUERY “IRAQ OR WEAPONS”

Two possibilities exist:

- No word sense ambiguity exists for the query keyword: the query keyword is represented directly in the view by its corresponding concept. Because this concept represents a query keyword it is therefore considered a focus concept and coloured pink. An example for this situation is depicted in Figure 1: SVG view representing the query “Iraq OR weapons” for the query keyword “Iraq”.

- Word sense ambiguity exists for the query keyword and no preferred meaning of the word has been chosen: the keyword is shown in green. All possible meanings of the keyword represented by yellow concepts are also shown around the keyword and can be selected. An example for this situation is depicted in Figure 1: SVG view representing the query “Iraq OR weapons” for the query keyword “weapons”. If such a concept is selected, the green keyword is replaced by this yellow concept and the other related concepts are removed. This yellow concept is a new focus concept and therefore the colour is changed to pink.

In the latter case, the displayed search results will relate to all possible meanings of the keyword. Upon selecting a certain meaning represented by a concept, the result set is limited to articles that are only related to the selected meaning.
In order to help the user to refine the keyword into a certain concept, a semantic explanation of the meaning represented by the concept is displayed in a tool tip upon mouse over. This explanation consists of a short textual description and other keywords for this concept (see Figure 2: Tool tip for the concept “arm”) and must allow the user to decide if this concepts matches his meaning of the keyword.

FIGURE 2: TOOL TIP FOR THE CONCEPT “ARM”

The expand button (rectangle with a plus sign) at the right of the each concept name allows the user to expand that concept to its semantically related concepts (relations “more general”, “more specific” and “associated”). Related concepts are drawn on an imaginary circle around the concept (see Figure 3: Expanding the concept “Iraq”). With these related concepts the user can broaden or narrow his or her query. Expanding a topic does not change the current query. It offers a way to the user to explore semantically related concepts. Every concept can be expanded. A concept “created” by expanding a query keyword or query concept can in turn be expanded and so on. A grey “plus sign” button means that the concept has no more relations than the ones shown and therefore can not be expanded anymore.

To avoid as much as possible overlapping concepts when a topic is expanded, some measures are foreseen:

- The radius is calculated in function of the number of associations.
- The circle around the concept is divided into equal radial sections based on the number of relations. The sections occupied by related concepts already drawn for a previous expansion of another topic are marked busy. The new related concepts are drawn in the remaining free sections.
- The remove button (rectangle with a minus sign) at the right of the each concept name allows the user to remove that concept from the query. Topics expanded by this topic and having only relations with this topic are also removed.
FIGURE 3: EXPANDING THE CONCEPT “IRAQ”

Clicking on the select button (circle with an “S”) at the left of the each concept name causes the system to take that concept as the focus concept (see Figure 4: Result after selecting the concept “arm” to disambiguate the query keyword “weapons”). That concept moves to the location of the previous focus concept and is shown in pink. Concepts not expanded by that concept are removed.

FIGURE 4: RESULT AFTER SELECTING THE CONCEPT “ARM” TO DISAMBIGUATE THE QUERY KEYWORD “WEAPONS”

Concepts and topics can be dragged & dropped with the same ease as in classic Microsoft Windows applications. Just catch the rectangle of the chosen topic by pressing down the mouse button, hold the button down while moving and release the button on the desired position.

Keeping a history of the SVG view is not an easy task because the user can drag and drop concepts. Status changes of the view caused by expansion or removing or concepts are purely consequences of scripting and not of refreshing pages. In other words the classical history buttons of traditional web browsers cannot be used. Therefore the OmniPaper SVG view uses another approach that makes use of an interesting property of SVG graphics: opacity. A limited history is kept in the view itself by decreasing the opacity of older concepts. When a concept is expanded the opacity of all topics belonging to the same focus topic is decreased. After a limit amount (two or three) of opacity decreases the topic is removed. This way we obtain a history for the latest two or three expansions of concepts, which is normally sufficient.
Despite the history mechanism and server side filtering techniques, the amount of concepts and keywords displayed on the view can rapidly increase after some concept expansions. To deal partially with this problem some additional features are provided. The user can make use of three scrollbars: one for zooming in or out and the two others to scroll vertically and horizontally to adapt the view like in most desktop applications.

**BUILDING UP OF THE VIEW**

The information that is needed to build up the graphical view of concepts is delivered to the SVG viewer in XML format. This happens in two situations: *query initialisation* and *expansion* of a topic. These are the only actions where additional data is required from the server. These two situations are clarified in the following paragraphs.

*Query initialisation* is needed for every search action, i.e. for each click on the “Go” button of the search form. Figure 5 shows the XML response document sent back to the SVG viewer for the query “Iraq OR weapons”.

![XML response document](image)

**FIGURE 5: XML RESPONSE DOCUMENT SENT BACK TO THE SVG VIEWER FOR THE QUERY “IRAQ OR WEAPONS”**
Refining search queries with SVG graphics in the OMNIPAPER project

All this information is processed with ECMAScript in de SVG Viewer and stored in ECMAScript objects. These objects are instances of one of the four main ECMAScript classes: Topic, Keyword, Concept and Association. Both the Keyword and the Concept classes are subclasses of the more general Topic class.

The <querystring> element contains the query string “Iraq OR weapons” based on the query keywords “Iraq” and “weapons” both represented by a <querykeyword> element. The query keyword “weapons” has a number of meanings represented by the <sense> child elements. In fact each meaning or <sense> element represents an associated concept. That meaning is described in the <description> element together with other keywords for the concept provided in the <conceptkeyword> elements. Each <sense> element has just one preferred keyword which is used in the view to represent that concept. Because the query keyword “iraq” has only one meaning no disambiguation is needed. For the query keyword “weapons” disambiguation is needed because 3 different meanings exist. The <conceptview> root element has a child element <languageproperties> that contains translations of some general phrases or words used in the view.

The second case where a response XML document is sent back to the SVG viewer is concept expansion because additional information is needed to update the viewer. Figure 6 shows the response XML document that contains information about associated concepts of the expanded concept “107325601” represented in the view by its preferred keyword “Iraq”. The associated concepts are organized according their relation type. Each <concept> element contains all information needed for interaction in the view (the <description> and <conceptKeyword> elements are needed for the tool tips of these concepts). The “numberOfAssociations” attribute gives necessary information to know if the concept can still be expanded. If not the expansion button is grey instead of green and cannot be clicked. For each related concept the system checks if the concept is already present in the view or not. If this is the case the related concept stays at its location and a relation is drawn. If not the topic is drawn together with its association. Relations of different types are drawn in a different colours and a tool tip on mouse over reveals the nature of the association.

Some actions are coupled with animation effects. In the case of expansion the related concepts move from their expander concept to their final position on the imaginary circle around it. Removing a topic is animated through gradual decrease of opacity.

Unfortunately the current implementations of the SVG Adobe Viewer do not provide sufficient support for RCC to be used in the OmniPaper project. The current version 3.0 does not provide any support for RCC and in the latest pre-release version 6.0 animation is not yet supported.
FIGURE 6: XML RESPONSE DOCUMENT SENT BACK TO THE SVG VIEWER FOR EXPANSION OF THE CONCEPT WITH ID = “107325601” REPRESENTED BY ITS PREFERRED KEYWORD “IRAQ”
REFERENCES
[3] Steve Pepper, Graham Moore. XML Topic Maps (XTM) 1.0 [online], August 2001; Available from: <http://www.topicmaps.org/xtm/1.0/>