Reconstructing Information Retrieved from Multiple Websites*

Héctor Valero, Carlos Castillo and Josep Silva

Universitat Politècnica de València
Valencia, Spain
hecvalli,carcasg1@posgrado.upv.es, jsilva@dsic.upv.es

Abstract

This work presents a model for information retrieval from multiple webpages. This model does not need to pre-process, parser or label the webpages; and thus, it can work online and in real time. The model introduces two new techniques for visualization that allow us to automatically reconstruct a new webpage with the information retrieved. This page is structured taking into account the semantically related information. The technique has been implemented and the implementation is discussed focussing on the main problems that appear when the proposed algorithms are integrated into a commercial web browser.

Keywords: Information retrieval, HTML filtering, Webpages visualization.

1 Introduction

Currently, *information retrieval* is one of the hot topics in Internet; and indeed more in the semantic web. However, the lack of online and real time applications able to retrieve information automatically, is a sign of the difficulties of this task. Current techniques for information retrieval in Internet are mainly specialized in retrieving webpages that are related to a particular query [1]. In this context, search engines such as Google or Bing implement very accurate and precise algorithms for finding the webpages related to a given query. Nevertheless, in many cases, there is too much information contained in the webpage that is not related to the user's query. Thus, the granularity level of the answer is too big: a whole webpage.

In the context of the semantic web, it is frequent to produce more concrete results that consist of texts that answer a given question. However, these techniques need to pre-process the web pages that are used as the sources of information. A common approach is to build an ontological model that is constructed and queried with languages such as RDF [2] and OWL [3]. This imposes important restrictions on the webpages that are going to be processed; and for this reason, the tools implemented with this approaches are often offline. The tools that use *microformats* [4, 5, 6] have the same problem. In particular, webpages that use microformats could be automatically processed thanks to the use of special meta-labels that qualify the information and that are inserted in the (X)HTML code. But, unfortunately, the technology that supports microformats is not mature enough, and this is the reason why much of the webpages in Internet do not use microformats. Therefore, they cannot be processed in real time.

In a previous work [7], we proposed an online and real time technique for information retrieval that is able to automatically retrieve information related to a given query from a

*This work has been partially supported by the Spanish *Ministerio de Ciencia e Innovación* under grant TIN2008-06622-C03-02, by the *Generalitat Valenciana* under grant ACOMP/2009/017, and by the *Universidad Politécnica de Valencia* (Program PAID-06-08).

single webpage. This technique is based on the use of *syntax distances* as a measure of semantic relations. Roughly, the technique extracts from a webpage those elements that are syntactically close to the terms specified by the user. Therefore, the technique assumes that those terms that are syntactically close are more semantically related. The technique was implemented and integrated into the Firefox web browser after it was approved by the Firefox experts developers area. Its extensive use with hundreds of users has confirmed that the use of syntax distances is a simple but powerful idea that works in practice [8].

Recently, we have developed a new information retrieval technique that generalizes the idea of using syntax distances between elements of a same page to multiple webpages incorporating page distances, domain distances, etc. This new distance has been named *hyper-syntactic distance*. The main problem of the new technique is the difficulty of automatically reconstructing a new webpage with the final result. In particular, when working with different webpages, new visualization problems appear such as the overlapping of elements with absolute positions, the incompatibility of different layouts, the integration of CSS files; and others such as the security imposed by the browsers, the time needed to load several webpages, etc.

In this work we face these problems and explain different approaches to solve them. When retrieving information from multiple webpages, we find that showing the information to the user is not as simple as joining together all the blocks of information retrieved from the webpages. The information must be presented in a way that the interrelations between the information from different webpages is explicit. In order to do this, we propose two visualization models: the tabular model and the hierarchical model.

The main advantages of this new technique are that it does not need the use of proxies [9], it can work online and in real time—with any webpage—without any pre-compilation or pre-processing phase [10]; it can process multiple sources; and it can retrieve information with a low granularity level: one single word.

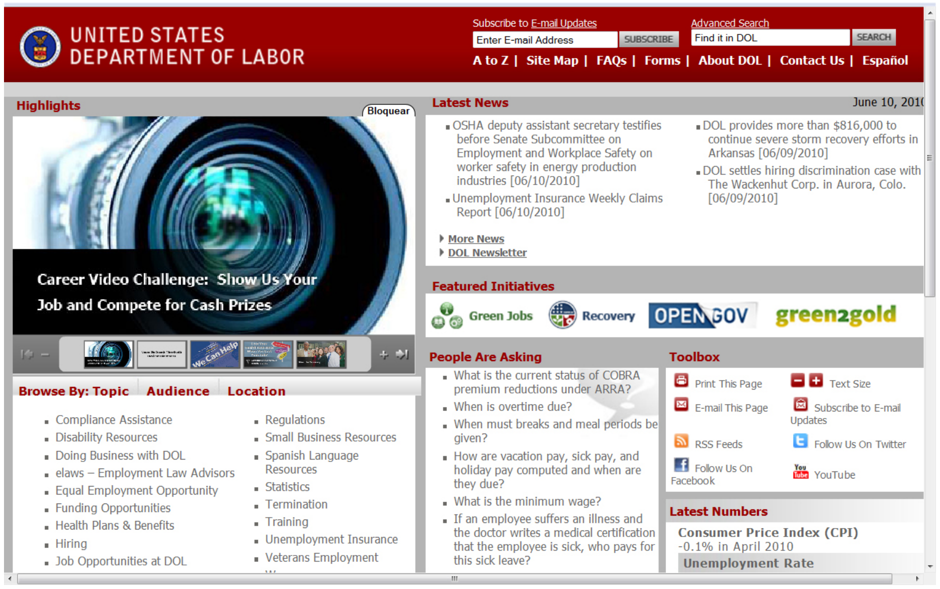
The rest of the paper has been organized as follows. Section 2 presents our technique by using a motivating example. Section 3 explains a novel technique for information retrieval from multiple webpages. This technique uses two visualization models that are introduced in Section 4. In Section 5, we describe our implementation. The main problems are discussed and their solutions explained. This section also presents the results obtained by a performance evaluation of our tool. Finally, Section 6 concludes.

2 Motivation

This section presents a real example of information retrieval using the technique presented in this work. Let us consider a user that is browsing in Internet and she loads the main webpage of the United States Department of Labor searching for ***work***.

In a normal scenario, the user reads the main webpage of the department (see Figure 1) and, using the standard browser's filter or the search box of the webpage (if it is provided), she looks for the word searched in the page, and she loads a new page using the corresponding hyperlink. If the information searched is found, the process finishes, but if the information is not in the loaded webpage, the user has to return to the previous page and explore another hyperlink. This is repeated successively until the desired information is found. During this process, (i) the user is forced to read much information not related to what she is looking for. This information appears in the webpages visited during the search, and (ii) she must explore several hyperlinks until the relevant hyperlinks are found. This process is a time-consuming task.

In a second scenario, the user uses our tool for information retrieval. She loads the main


UNITED STATES DEPARTMENT OF LABOR

Subscribe to E-mail Updates
Enter E-mail Address SUBSCRIBE
Advanced Search
Find it in DOL SEARCH

A to Z | Site Map | FAQs | Forms | About DOL | Contact Us | Español

Highlights
Bloquear
Career Video Challenge: Show Us Your Job and Compete for Cash Prizes

Latest News June 10, 2010

- OSHA deputy assistant secretary testifies before Senate Subcommittee on Employment and Workplace Safety on worker safety in energy production industries [06/10/2010]
- Unemployment Insurance Weekly Claims Report [06/10/2010]
- DOL provides more than $816,000 to continue severe storm recovery efforts in Arkansas [06/09/2010]
- DOL settles hiring discrimination case with The Wackenhut Corp. in Aurora, Colo. [06/09/2010]

More News
DOL Newsletter

Featured Initiatives
Green Jobs Recovery OPEN GOV green2gold

Browse By: Topic Audience Location

- Compliance Assistance
- Disability Resources
- Doing Business with DOL
- elaws – Employment Law Advisors
- Equal Employment Opportunity
- Funding Opportunities
- Health Plans & Benefits
- Hiring
- Job Opportunities at DOL
- Regulations
- Small Business Resources
- Spanish Language Resources
- Statistics
- Termination
- Training
- Unemployment Insurance
- Veterans Employment

People Are Asking

- What is the current status of COBRA premium reductions under ARRA?
- When is overtime due?
- When must breaks and meal periods be given?
- How are vacation pay, sick pay, and holiday pay computed and when are they due?
- What is the minimum wage?
- If an employee suffers an illness and the doctor writes a medical certification that the employee is sick, who pays for this sick leave?

Toolbox
Print This Page
Text Size
E-mail This Page
Subscribe to E-mail Updates
RSS Feeds
Follow Us On Twitter
Follow Us On Facebook
YouTube

Latest Numbers
Consumer Price Index (CPI)
-0.1% in April 2010
Unemployment Rate

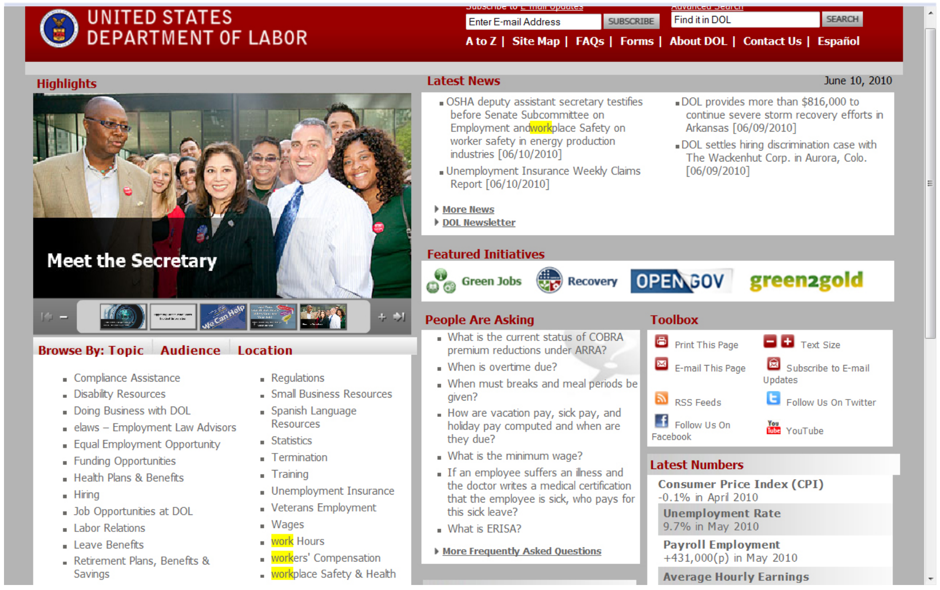
Figure 1: Main webpage of the United States Department of Labor

webpage of the United States Department of Labor, she types the filtering criterion ("work") and she clicks on the filtering button. With a simple click, the tool automatically constructs a new webpage with all the information related to work in the main webpage and in the webpages accessible from it. During the analysis of the webpages, the tool explores the relevant hyperlinks to extract the desired information from each of them. With all the information retrieved, a new webpage with the results is generated. This webpage, only contains information relevant for the user.

In Figure 2 all the ocurrences of the filtering criterion are highlighted. This is the result of using the standard search with the term "work". Unfortunately, the information provided by this search is poor; but it includes some useful hyperlinks such as "work hours" (at the left). Our algorithm is able to automatically filter this webpage and explore each relevant hyperlink to reach other pages in the same or in other domains in order to gather all the relevant information according to the filtering criterion. With this information, the algorithm produces a new webpage as the one shown in Figure 3.

In this case, the model used to reconstruct the final webpage is the ***hierarchical model***. The hierarchical model groups the retrieved information blocks according to their minimum syntactical distance; placing them as close as possible. For this representation, the information blocks are nested immediately after the hyperlink that points to the webpage to which they belong. This process is recursively repeated with the new hyperlinks that are found inside these retrieved blocks.

In the example, the first information shown is composed of the blocks retrieved from the main webpage of the Department of Labor. From these blocks the hyperlinks are extracted and ordered by relevance according to their syntactical distance. Then, these new webpages are analyzed producing new relevant blocks that are placed next to their corresponding hyperlinks in the already visualized webpage. Internally, a block with relevant information is a set of HTML nodes (i.e., a DOM subtree of an HTML webpage) grouped inside an HTML container


UNITED STATES DEPARTMENT OF LABOR

Subscribe to E-mail Updates
Enter E-mail Address SUBSCRIBE
Advanced Search
Find it in DOL SEARCH

A to Z | Site Map | FAQs | Forms | About DOL | Contact Us | Español

Highlights

Meet the Secretary

Latest News June 10, 2010

- OSHA deputy assistant secretary testifies before Senate Subcommittee on Employment andworkplace Safety on worker safety in energy production industries [06/10/2010]
- Unemployment Insurance Weekly Claims Report [06/10/2010]
- DOL provides more than $816,000 to continue severe storm recovery efforts in Arkansas [06/09/2010]
- DOL settles hiring discrimination case with The Wackenhut Corp. in Aurora, Colo. [06/09/2010]

More News
DOL Newsletter

Featured Initiatives

Green Jobs Recovery OPEN GOV green2gold

Browse By: Topic Audience Location

- Compliance Assistance
- Disability Resources
- Doing Business with DOL
- elaws – Employment Law Advisors
- Equal Employment Opportunity
- Funding Opportunities
- Health Plans & Benefits
- Hiring
- Job Opportunities at DOL
- Labor Relations
- Leave Benefits
- Retirement Plans, Benefits & Savings
- Regulations
- Small Business Resources
- Spanish Language Resources
- Statistics
- Termination
- Training
- Unemployment Insurance
- Veterans Employment
- Wages
- work Hours
- workers' Compensation
- workplace Safety & Health

People Are Asking

- What is the current status of COBRA premium reductions under ARRA?
- When is overtime due?
- When must breaks and meal periods be given?
- How are vacation pay, sick pay, and holiday pay computed and when are they due?
- What is the minimum wage?
- If an employee suffers an illness and the doctor writes a medical certification that the employee is sick, who pays for this sick leave?
- What is ERISA?

More Frequently Asked Questions

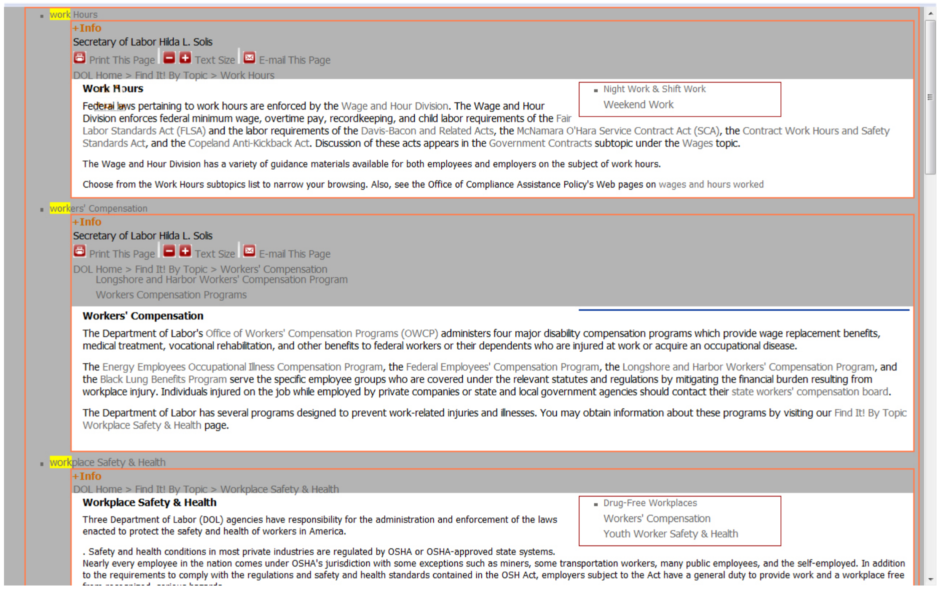
Toolbox

Print This Page
Text Size
E-mail This Page
Subscribe to E-mail Updates
RSS Feeds
Follow Us On Twitter
Follow Us On Facebook
YouTube

Latest Numbers

Consumer Price Index (CPI)
-0.1% in April 2010
Unemployment Rate
9.7% in May 2010
Payroll Employment
+431,000(p) in May 2010
Average Hourly Earnings

Figure 2: Ocurrences of word *work*


- work Hours

+Info
Secretary of Labor Hilda L. Solis
Print This Page Text Size E-mail This Page
DOL Home > Find It! By Topic > Work Hours

Work Hours

- Night Work & Shift Work
  Weekend Work

Federal laws pertaining to work hours are enforced by the Wage and Hour Division. The Wage and Hour Division enforces federal minimum wage, overtime pay, recordkeeping, and child labor requirements of the Fair Labor Standards Act (FLSA) and the labor requirements of the Davis-Bacon and Related Acts, the McNamara O'Hara Service Contract Act (SCA), the Contract Work Hours and Safety Standards Act, and the Copeland Anti-Kickback Act. Discussion of these acts appears in the Government Contracts subtopic under the Wages topic.

The Wage and Hour Division has a variety of guidance materials available for both employees and employers on the subject of work hours.

Choose from the Work Hours subtopics list to narrow your browsing. Also, see the Office of Compliance Assistance Policy's Web pages on wages and hours worked

- workers' Compensation

+Info
Secretary of Labor Hilda L. Solis
Print This Page Text Size E-mail This Page
DOL Home > Find It! By Topic > Workers' Compensation
Longshore and Harbor Workers' Compensation Program
Workers Compensation Programs

Workers' Compensation

The Department of Labor's Office of Workers' Compensation Programs (OWCP) administers four major disability compensation programs which provide wage replacement benefits, medical treatment, vocational rehabilitation, and other benefits to federal workers or their dependents who are injured at work or acquire an occupational disease.

The Energy Employees Occupational Illness Compensation Program, the Federal Employees' Compensation Program, the Longshore and Harbor Workers' Compensation Program, and the Black Lung Benefits Program serve the specific employee groups who are covered under the relevant statutes and regulations by mitigating the financial burden resulting from workplace injury. Individuals injured on the job while employed by private companies or state and local government agencies should contact their state workers' compensation board.

The Department of Labor has several programs designed to prevent work-related injuries and illnesses. You may obtain information about these programs by visiting our Find It! By Topic Workplace Safety & Health page.

- workplace Safety & Health

+Info
DOL Home > Find It! By Topic > Workplace Safety & Health

Workplace Safety & Health

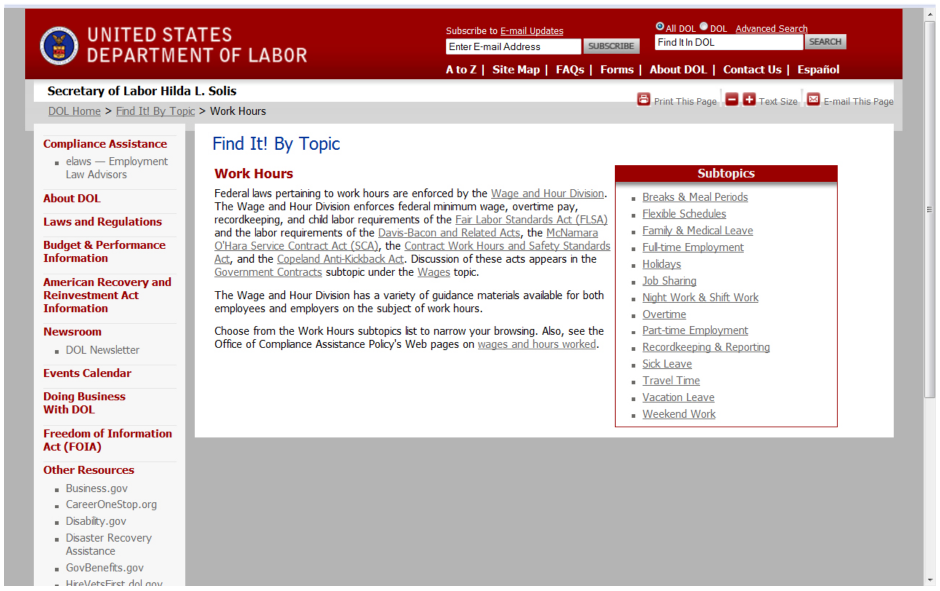
- Drug-Free Workplaces
  Workers' Compensation
  Youth Worker Safety & Health

Three Department of Labor (DOL) agencies have responsibility for the administration and enforcement of the laws enacted to protect the safety and health of workers in America.

. Safety and health conditions in most private industries are regulated by OSHA or OSHA-approved state systems. Nearly every employee in the nation comes under OSHA's jurisdiction with some exceptions such as miners, some transportation workers, many public employees, and the self-employed. In addition to the requirements to comply with the regulations and safety and health standards contained in the OSH Act, employers subject to the Act have a general duty to provide work and a workplace free

Figure 3: Results webpage after filtering the the United States Department of Labor's website

(e.g., a table, a layer, etc.). In Figure 3 we can observe that some blocks are placed after their corresponding hyperlink. For instance, at the top, we have a block after the "work hours" hyperlink. This block has been extracted from the webpage that contains information about work hours regulations in USA. The original webpage about work hours is shown in Figure 4. Note that the results webpage only shows a part of the information of this webpage. The blocks of information extracted contain the information that is related to the filtering criterion. The other information was discarded.


UNITED STATES DEPARTMENT OF LABOR
Subscribe to E-mail Updates
Enter E-mail Address
SUBSCRIBE
All DOL
DOL
Advanced Search
Find It In DOL
SEARCH
A to Z | Site Map | FAQs | Forms | About DOL | Contact Us | Español
Secretary of Labor Hilda L. Solis
Print This Page
Text Size
E-mail This Page
DOL Home > Find It! By Topic > Work Hours
Compliance Assistance
elaws — Employment Law Advisors
About DOL
Laws and Regulations
Budget & Performance Information
American Recovery and Reinvestment Act Information
Newsroom
DOL Newsletter
Events Calendar
Doing Business With DOL
Freedom of Information Act (FOIA)
Other Resources
Business.gov
CareerOneStop.org
Disability.gov
Disaster Recovery Assistance
GovBenefits.gov
HireVetsFirst.dol.gov

Find It! By Topic

Work Hours

Federal laws pertaining to work hours are enforced by the Wage and Hour Division. The Wage and Hour Division enforces federal minimum wage, overtime pay, recordkeeping, and child labor requirements of the Fair Labor Standards Act (FLSA) and the labor requirements of the Davis-Bacon and Related Acts, the McNamara O'Hara Service Contract Act (SCA), the Contract Work Hours and Safety Standards Act, and the Copeland Anti-Kickback Act. Discussion of these acts appears in the Government Contracts subtopic under the Wages topic.

The Wage and Hour Division has a variety of guidance materials available for both employees and employers on the subject of work hours.

Choose from the Work Hours subtopics list to narrow your browsing. Also, see the Office of Compliance Assistance Policy's Web pages on wages and hours worked.

Subtopics

- Breaks & Meal Periods
- Flexible Schedules
- Family & Medical Leave
- Full-time Employment
- Holidays
- Job Sharing
- Night Work & Shift Work
- Overtime
- Part-time Employment
- Recordkeeping & Reporting
- Sick Leave
- Travel Time
- Vacation Leave
- Weekend Work

Figure 4: Work hours webpage

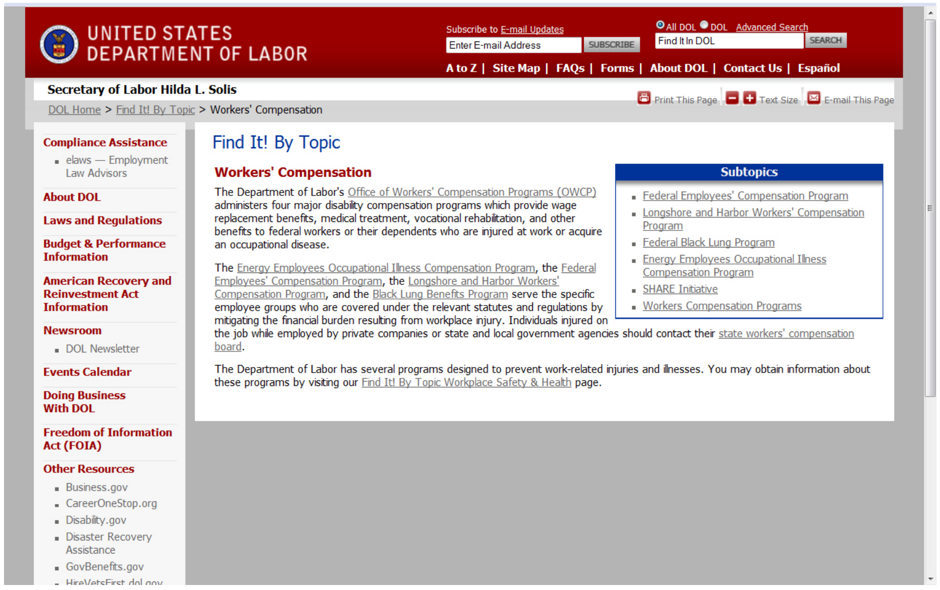
In Figure 3 the third block of information contains information related to worker's compensation. This block is placed after the block associated to work hours because it has been extracted from the page pointed by the hyperlink after the one that pointed to the work hours webpage. In particular, the webpage source of this information is shown in Figure 5. Some blocks are nested in the final webpage (see Figure 3) because they are associated to a webpage accessed from a hyperlink of the parent block. As a consequence of this scheme, in this model the webpages are decomposed and their relevant blocks are mixed in a hierarchical shape according to their navigational relations.

In addition to the hierarchical model, in this work we propose another model called ***tabular model***. In contrast to the hierarchical model, the tabular model provides a representation with a page granularity. That is, the information retrieval engine retrieves blocks with relevant information from each analyzed webpage; but these blocks are maintained in the final representation so that they keep their original structure. Both models will be presented in the next sections.

3 Information Retrieval from Multiple Webpages

This section presents an information retrieval algorithm able to extract information from multiple webpages. Our algorithm uses a previous algorithm to extract information from single webpages. The details of this algorithm can be found in [11]. In the following, we will assume the existence of a function *getSlice(p,q)* that implements this algorithm. Given a webpage p and a filtering criterion q, *getSlice* extracts from p all the information related to q.

We provide now a precise definition of webpage and filtering criterion. We will assume that a webpage is represented by a DOM tree [12].


UNITED STATES DEPARTMENT OF LABOR
Subscribe to E-mail Updates
Enter E-mail Address
SUBSCRIBE
All DOL
DOL
Advanced Search
Find It In DOL
SEARCH
A to Z | Site Map | FAQs | Forms | About DOL | Contact Us | Español
Print This Page
Text Size
E-mail This Page
DOL Home > Find It! By Topic > Workers' Compensation
Compliance Assistance
elaws — Employment Law Advisors
About DOL
Laws and Regulations
Budget & Performance Information
American Recovery and Reinvestment Act Information
Newsroom
DOL Newsletter
Events Calendar
Doing Business With DOL
Freedom of Information Act (FOIA)
Other Resources
Business.gov
CareerOneStop.org
Disability.gov
Disaster Recovery Assistance
GovBenefits.gov
Subtopics
Federal Employees' Compensation Program
Longshore and Harbor Workers' Compensation Program
Federal Black Lung Program
Energy Employees Occupational Illness Compensation Program
SHARE Initiative
Workers Compensation Programs

Secretary of Labor Hilda L. Solis

Find It! By Topic

Workers' Compensation

The Department of Labor's Office of Workers' Compensation Programs (OWCP) administers four major disability compensation programs which provide wage replacement benefits, medical treatment, vocational rehabilitation, and other benefits to federal workers or their dependents who are injured at work or acquire an occupational disease.

The Energy Employees Occupational Illness Compensation Program, the Federal Employees' Compensation Program, the Longshore and Harbor Workers' Compensation Program, and the Black Lung Benefits Program serve the specific employee groups who are covered under the relevant statutes and regulations by mitigating the financial burden resulting from workplace injury. Individuals injured on the job while employed by private companies or state and local government agencies should contact their state workers' compensation board.

The Department of Labor has several programs designed to prevent work-related injuries and illnesses. You may obtain information about these programs by visiting our Find It! By Topic Workplace Safety & Health page.

Figure 5: Night and shift work webpage

Definition 3.1 (DOM tree)**.** *A DOM tree t=(V,E) is a tree whose nodes V are labeled with HTML labels, and they interconnected with a set of edges E according to the DOM specification [12].*

Definition 3.2 (webpage)**.** *A webpage is a tuple (u,t) where u is a URL and t is a DOM tree.*

A filtering criterion is composed of one or more words associated to the information that we want to retrieve; and one proximity measure that represents the syntax distance between what we are looking for and what we retrieve.

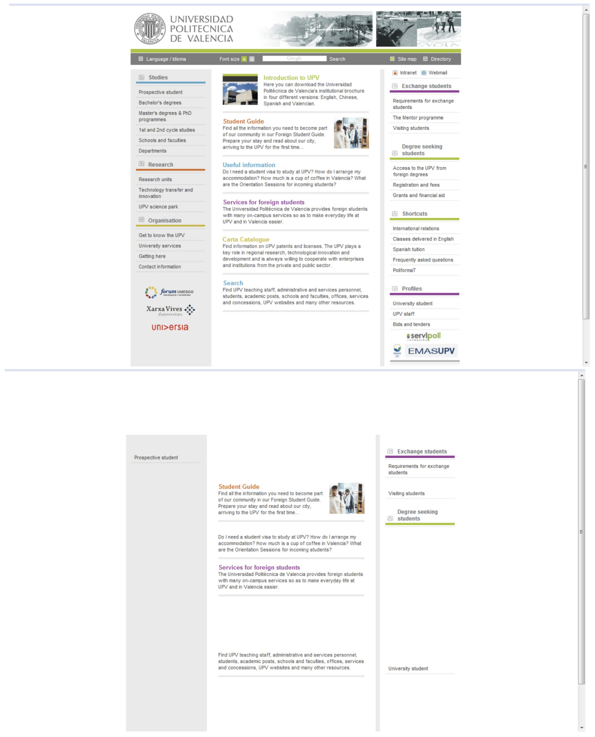
Definition 3.3 (Filtering criterion)**.** *A filtering criterion is a pair (w, d) where w is a string that represents the desired information; and d is an integer that represents the precision used in the search.*

Example 1. *The webpage in Figure 6 (left) is the main webpage of the Technical University of Valencia*[1]*. If we filter this webpage with the filtering criterion ("student", 0), we get the webpage in Figure 6 (right).*

The DOM tree of this webpage is huge. Hence, we focus on a part of the webpage. Concretely, if we observe the text in the gray column at the left, the part of the DOM tree that represents this text is shown in Figure 7.

In the figure, the black node contains the word "student". From this node, all its ancestors and successors are kept in the filtered webpage. Because the required precision is 0, no more nodes are retrieved, and thus, white nodes are discarded. The produced subtree corresponds to the text in the gray area of the filtered webpage.

[1]This is the English version, but note that, since the technique is based on syntax distances, it works equally with any language.


Language / Idioma Font size Search Site map Directory

Studies
Prospective student
Bachelor's degrees
Master's degrees & PhD programmes
1st and 2nd cycle studies
Schools and faculties
Departments

Research
Research units
Technology transfer and innovation
UPV science park

Organisation
Get to know the UPV
University services
Getting here
Contact information

Introduction to UPV
Here you can download the Universidad Politécnica de Valencia's institutional brochure in four different versions: English, Chinese, Spanish and Valencian.

Student Guide
Find all the information you need to become part of our community in our Foreign Student Guide. Prepare your stay and read about our city, arriving to the UPV for the first time...

Useful information
Do I need a student visa to study at UPV? How do I arrange my accommodation? How much is a cup of coffee in Valencia? What are the Orientation Sessions for incoming students?

Services for foreign students
The Universidad Politécnica de Valencia provides foreign students with many on-campus services so as to make everyday life at UPV and in Valencia easier.

Carta Catalogue
Find information on UPV patents and licenses. The UPV plays a key role in regional research, technological innovation and development and is always willing to cooperate with enterprises and institutions from the private and public sector.

Search
Find UPV teaching staff, administrative and services personnel, students, academic posts, schools and faculties, offices, services and concessions, UPV websites and many other resources.

Intranet Webmail

Exchange students
Requirements for exchange students
The Mentor programme
Visiting students

Degree seeking students
Access to the UPV from foreign degrees
Registration and fees
Grants and financial aid

Shortcuts
International relations
Classes delivered in English
Spanish tuition
Frequently asked questions
PoliformaT

Profiles
University student
UPV staff
Bids and tenders

Prospective student

Student Guide
Find all the information you need to become part of our community in our Foreign Student Guide. Prepare your stay and read about our city, arriving to the UPV for the first time...

Do I need a student visa to study at UPV? How do I arrange my accommodation? How much is a cup of coffee in Valencia? What are the Orientation Sessions for incoming students?

Services for foreign students
The Universidad Politécnica de Valencia provides foreign students with many on-campus services so as to make everyday life at UPV and in Valencia easier.

Find UPV teaching staff, administrative and services personnel, students, academic posts, schools and faculties, offices, services and concessions, UPV websites and many other resources.

Exchange students
Requirements for exchange students
Visiting students
Degree seeking students
University student

Figure 6: Webpage of the Technical University of Valencia (left) and its filtered version (right)

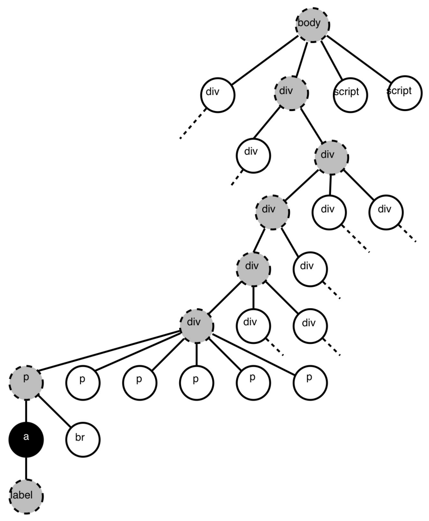


Figure 7: DOM subtree of the Technical University of Valencia's webpage

3.1 The Hyper-syntactic Distance

Usually, there exist in a webpage many hyperlinks to other related webpages. An information retrieval algorithm should be able to explore all these hyperlinks in order to reach the relevant information of other webpages and then, reconstruct a new webpage with all the retrieved information. However, we want our technique to work online and in real time. This means that loading all the hyperlinks is not possible because it would require too much time. Therefore, our technique must explore only some hyperlinks during the search and discard the others. In order to solve this problem, in a previous work [13] we proposed a unit of measurement called *hyper-sintactic distance*. It allows us to order the hyperlinks of a set of webpages by relevance with respect to a filtering criterion. In essence, the hyper-sintactic distance determines the relevance of a hyperlink H in a webpage p with respect to a DOM node n in a webpage p' considering three fundamental measures: the distance in the DOM tree from H to n if $p = p'$, or from H to the root of p if $p \neq p'$; the number of pages that must be traversed from p' to p and the number of domains that must be traversed from the domain of p' to the domain of p. The combination of these measures produces a value that is known as hyper-sintactic distance and that approximates the semantic relation la between n and H.

In the following, we will assume the existence of a function *getMostRelevantLink(links, q)* that given a set of hyperlinks *links*, and a filtering criterion q, it returns the hyperlink that is more relevant considering its hyper-sintactic distance with respect to q. The interested reader is referred to [13] where this unit of measurement is explained in detail.

For our purposes, in the rest of the article we can view an hyperlink as a directed arc between two webpages.

Definition 3.4 (Hyperlink)**.** *An hyperlink is a pair (u,v) where u is the URL of a source webpage and v is the URL of the target webpage.*

Thanks to the hyper-sintactic distance, we could define a simple information retrieval algorithm that repeats three fundamental steps:

1. Filter the current webpage (*getSlice*)
2. Find the most relevant hyperlink in the loaded webpages (*getMostRelevantLink*)
3. Load the webpage pointed by the most relevant hyperlink

Finally, when no more relevant hyperlinks exist, we could reconstruct a new webpage with the retrieved information. Nevertheless, this scheme is not appropriate for an online tool. The reason is that loading a webpage needs approximately one second. This means that the previous scheme would not show to the user any result until all the webpages had been analyzed and filtered, that implies too much time for a real time tool. Remember that web design guidelines establish 10 seconds as the maximum response time [14]. Therefore, we propose a more appropriate solution in which the information is reconstructed and shown to the user incrementally as it is being retrieved. This means that the user can see the retrieved information from the first second, and this information is increased as the analysis continues.

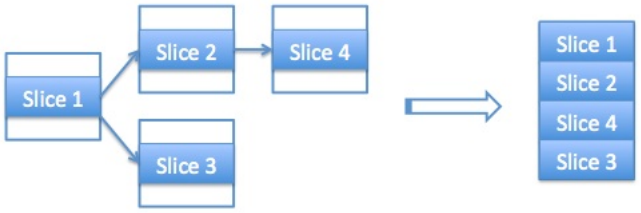
4 Visualization of the Retrieved Information

This section introduces a new technique to incrementally integrate and visualize the information recovered from multiple webpages. This technique uses two independent (but very related) algorithms for the visualization of the information. The first one presents the information tabularly, the second one uses a hierarchical representation.

Both algorithms are able to retrieve information from different webpages and show it incrementally while it is being recovered. The main difference between them is the way in which the information is visualized in the browser.

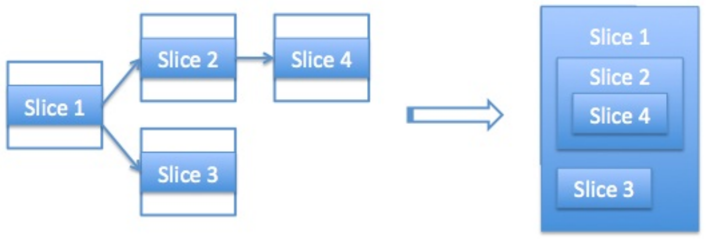
Tabular Visualization The lowest granularity level in this representation is a page. Basically, the final webpage is a linear succession of the filtered webpages. Each filtered webpage is considered as a whole, and thus, all the information that appeared together in the filtered webpage, is also together in the final webpage. The filtered webpages are ordered according to their navigational structure using a depth-first order.

Example 2. *The next figure shows a set of linked webpages where the dark part represents the relevant information. At the right, we see the tabular representation of this relevant information.*



Hierarchical Visualization The lowest granularity level in this representation is a word. In this representation, the final webpage is a tree where the filtered webpages are organized. In contrast to the tabular representation, the filtered webpages can be mixed because each filtered webpage is placed next to the hyperlink that references it.

Example 3. *The following figure complements Example 2 showing the hierarchical representation of the same set of webpages.*



4.1 Tabular Visualization

Algorithm 1 implements the tabular visualization model. This algorithm uses the following functions:

timeout() This function controls that the algorithm is not executed more time than the specified by the user in the configuration. When the specified time is reached it returns *True*.

getSlice(p, q) It returns the slice produced after filtering webpage p with the query q. This is done using the algorithm proposed in [11].

$createIframe(d)$ This function creates and returns one DOM node of type *iframe* whose content is the DOM tree d received as a parameter.

append(n, m) It appends the DOM node m as a child of the DOM node n.

getLinks$(nodes)$ It extracts all the hyperlinks of a set *nodes* of DOM nodes.

$getMostRelevantLink(links, q)$ It extracts from the set *links* the hyperlink with a lower hyper-syntactic distance with respect to the filtering criterion q. It is used to determine what is the next hyperlink that should be processed.

load$(page)$ It loads the webpage *page*.

$getNode(l)$ It returns the DOM node associated to hyperlink l.

Function $processWebPage$ is a recursive function that loads the most relevant pages from a set of hyperlinks that are potentially processable. Each time a new webpage is loaded, it is analyzed and new relevant hyperlinks are added to the set. Every loaded webpage is parsed, then filtered, and the result produced is shown with function *show*. This implies that the final webpage is shown incrementally, page after page. This process is repeated until a *timeout* is reached.

Function *show* has been specialized for each model. In the tabular model, it creates an iframe whose content is the webpage that has been just filtered, and this iframe is put next to the webpage that contains the hyperlink that pointed to this webpage.

4.2 Hierarchical Visualization

Algorithm 2 implements the hierarchical visualization model. This algorithm, shares most of the functions (including *processWebPage*) used in Algorithm 1 that were explained in the previous section. In addition, it uses the following functions:

getParent(n) It returns the first ancestor of node n that is of type container (table, div, frame, etc.).

createContainer() It creates and returns a DOM node of type *table*.

The behavior of this algorithm and Algorithm 1 is very similar. The main difference is function *show*. In this case, *show* has been specialized to show the filtered webpage as a part of the webpage that referenced it. This is done by creating a new container of type table. The type table is a good selection because it allows us to establish relative sizes and because it has a Z axis equal to zero; and thus, it is never superposed to the elements already shown in the page. The new table is linked with the first ancestor node n that is a container. In this way, the new webpage is integrated into the webpage that referenced it, exactly in the point of the webpage where the hyperlink was (in the container of this hyperlink).

Algorithm 1 Tabular Visualization

```
Input: A webpage $P$, and a filtering criterion $q$
Output: A webpage $P'$
Initialization: $link = \emptyset$

function $show(Node\ n, DOM\ d)$
    $showTabular(n, d)$

function $showTabular(Node\ n, DOM\ d)$
    $iframe = createIframe(d)$
    $append(n, iframe)$

function $processWebPage(Link\ l, WebPage\ p)$
    $relevantNodes = getSlice(p, q)$
    if $(l \neq \emptyset)$
    then $nodeC = getNode(l)$
    else $nodeC = \ < BODY >$
    $show(nodeC, relevantNodes)$
    $links = links \cup getLinks(relevantNodes)$
    if $(timeout() \vee links = \emptyset)$
    then exit()
    else $link = getMostRelevantLink(links, q)$
         $links = links \backslash link$
         $newPage = load(URL2)$ where $link = (URL1, URL2)$
         $processWebPage(link, newPage)$

return
    $< HTML >$
    $< BODY >$
    $processWebPage(link, P)$
    $< \backslash HTML >$
    $< \backslash BODY >$
```

5 Implementation

In this section we describe the implementation of the algorithms proposed and we discuss the main problems that emerge when integrating them into a browser.

The implementation is much more complex than the algorithms presented here because it has to make some transformations of the filtered webpages in order to guarantee that they are correct. For instance, the size attributes of the retrieved webpages must be changed to ensure that they fit into the container where they are inserted. The CSS styles must be imported so that the retrieved information keeps the original format, etc.

All the source code of our tool is open. Therefore, for concrete details about the design decisions taken, we refer the interested reader to:

`http://www.dsic.upv.es/~jsilva/webfiltering`

The implementation has to perform some additional checks before it loads the webpages pointed by the relevant hyperlinks. In particular, the information retrieval engine only pro-

Algorithm 2 Hierarchical Visualization

```
Input: A webpage $P$, and a filtering criterion $q$
Output: A webpage $P'$
Initialization: $link = \emptyset$

function $show(Node\ n, DOM\ d)$
    $showHierarchical(n, d)$

function $showHierarchical(Node\ n, DOM\ d)$
    $container = createContainer()$
    if $(n \neq < BODY >)$
    then $append(getParent(n), container)$
    else $append(n, container)$
    $append(container, d)$

return
    $< HTML >$
    $< BODY >$
    $processWebPage(link, P)$
    $< \backslash HTML >$
    $< \backslash BODY >$
```

cesses (X)HTML pages; hence, it is a waste of time to load files of type PDF, MP3, etc. [2] In order to avoid the load of such files, the object *XMLHttpRequest* is used to inspect the headers of the page before loading it, and thus, only loading those that contain useful information. For that, we use the following functions:

$r = newXMLHttpRequest();$
$r.open(\text{“}HEAD\text{”}, url);$
$r.send(null);$
$r.getResponseHeader(\text{“}Content - Type\text{”});$

In this section we focus on the two main problems that appear when we implement the algorithms. These problems will appear in any platform or commercial browser where they are implemented:

- ***Layers.*** One of the most important visualization problems is caused by layers, because they use absolute positions. Concretely, during the filtering phase, it is frequent to find various webpages with layers whose position is defined with absolute values. When this happens, it is possible that, in the final webpage reconstructed from these webpages, the positions of different layers (extracted from different webpages) overlap. An example is shown in Figure 8.

  The solution is to transform all layers with absolute positions to tables of the same size that are placed in the same position than the layer in the original webpage, but they are

[2]We are currently implementing an algorithm which is able to extract information from text and PDF documents, but the algorithm which is distributed in Firefox only processes (X)HTML.

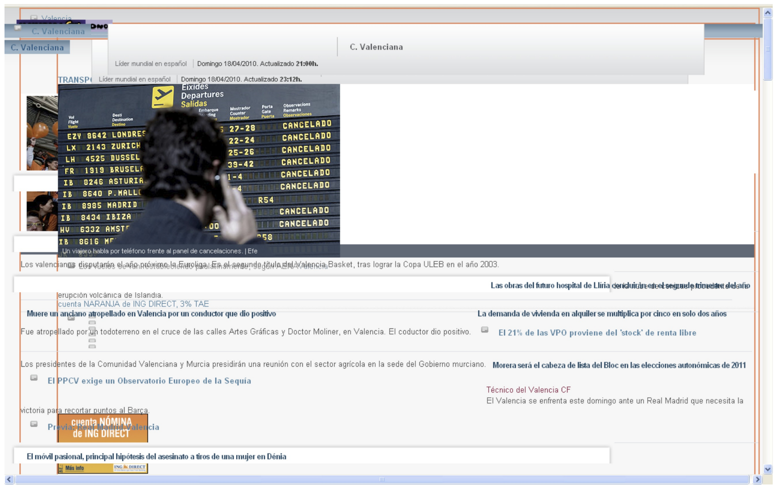


Figure 8: Problem caused by layers

relocated in the final webpage thanks to the use of relative positions.

- ***Security.*** Another of the main implementation problems is caused by the security systems of web browsers. In particular, there exists one kind of vulnerability of web browsers called *Cross Site Scripting* or XSS, where an attacker could execute scripts from a page or domain different from the loaded webpage. XSS has been avoided by current web browsers with a security system that blocks the execution of dangerous code.

  As it was explained in previous sections, our algorithms retrieve content from multiple webpages, they filter the content, and finally show a final webpage with the results. Therefore, the final webpage can contain scripts from multiple webpages and domains. When this happens, the security system anti-XSS is activated and it removes all the content that is potentially dangerous. This makes the final result to be incomplete, unstructured, or even completely empty.

  The security system anti-XSS works as follows: When a user loads a webpage, the browser explores the DOM tree to find ***script*** labels or nodes. If they are found, the script interpreter executes the scripts only if they belong to the loaded webpage. If, contrarily, the script belongs to another webpage, it is blocked. Concretely, the security system is activated when we filter the retrieved webpages; and it blocks all the scripts (and some other insecure elements), removing all the content that could be dangerous.

  Our implemented solution is the creation of an ***iframe*** object in which we load each filtered webpage. The *iframe* container does not activate anti-XSS because this container allows the load of URLs, thus embedding webpages inside other webpages. This solution works if we define the *iframe* object with some special properties:

  $frame = document.createElement(\text{“}iframe\text{”});$

  ...

  $frame.setAttribute(\text{“}type\text{”}, \text{“}content\text{”});$

  ...

$frame.webNavigation.allowJavascript = false;$

$frame.webNavigation.allowMetaRedirects = true;$

$frame.webNavigation.allowPlugins = false;$

It is interesting to highlight that the iframe must be of type **'content'**. When a container is defined of type ***content***, the browser only draws the information of the webpage but it does not execute any script. For this reason, we can retrieve the filtered information avoiding the anti-XSS security system. When a DOM tree is filtered, we put the new nodes inside a new container and we embed the container in the final webpage.

5.1 Performance Evaluation

The main bottleneck of this technique is the load of webpages; because it depends on the connection speed, the current state of the network, and many other external factors that affect the response time. In order to guarantee that the tool is able to work in real time, we must ensure that the results are shown in a bounded time. For this reason, the tool uses function *timeout*() (explained in Section 4.1). According to Jakob Nielsen's web usability design guidelines [14], in our implementation we established 10 seconds as the default value for the *timeout*. This value can be changed at any time, but our experiments demonstrate that it is often enough. In Table 1 we show the number of hyperlinks explored, with a timeout of 10 seconds, for several webpages analyzed. The average result is 13,5 webpages analyzed for each URL.

`URL`	`Filtering criterion`	`Links visited`
www.iee.org	student	12
www.upv.es	student	18
www.who.int	OMS	25
www.un.org	Haiti	6
www.esa.int	launch	13
www.nasa.org	space	16
www.mec.es	beca	18
www.edu.gva.es	universitat	20
www.ilo.org	projects	8
www.unicef.es	Haiti	10
www.mityc.es	turismo	9
www.mozilla.org	firefox	7

Table 1: Number of analyzed webpages with timeout=10 seconds

Note that a timeout of 10 seconds is the maximum time used to complete the final results webpage. But the visualization algorithms are incremental, thus, as an average, the first result is shown in less than a second (10/13,5 seconds).

All figures and examples of this paper are real examples produced with analyses made by our tool. More examples and information about the tool can be found at:

`http://www.dsic.upv.es/~jsilva/webfiltering`

6 Conclusions

This article introduces two new models of visualization for information retrieved from multiple webpages. The tabular model is specially good for complex webpages, because it keeps the original internal structure of the loaded webpages, and they are shown as unitary blocks. The hierarchical model is particularly interesting for pages with a lot of textual content, and it works very well, e.g., in forums, where it is able to find the relevant subjects and explore the threads with the answers joining together all the related information.

We are currently studding the possibility of combining both models. This combination would produce a hybrid model able to behave differently depending on the structure of the webpage that has been processed. In addition, we plan to change the hierarchical representation with a new tree-view that represents a personalized website map where the nodes of the tree are the filtered webpages, and where the user could collapse or expand the paths to the information. In this way, the generated web maps would be personalized with the user's filtering criterion.

References

[1] J.M. Gómez Hidalgo, F. Carrero García, E. Puertas Sanz. Named Entity Recognition for Web Content Filtering International Conference on Applications of Natural Language, NLDB2005, pages 286-297, 2005

[2] W3C Consortium, Resource Description Framework (RDF). www.w3.org/RDF

[3] W3C Consortium, Web Ontology Language (OWL). www.w3.oeg/2001/sw/wiki/OWL

[4] Microformats.org. The Official Microformats Site. http://microformats.org/, 2009.

[5] R. Khare, T. çelik Microformats: a Pragmatic Path to the Semantic Web. Proceedings of the 15h International Conference on World Wide Web. Poster Sessions pages 865-866, 2006

[6] R. Khare. Microformats: The Next (Small) Thing on the Semantic Web? IEEE Internet Computing, 10(1):68-75, 2006.

[7] J. Silva, Information Filtering and Information Retrieval with the Web Filtering Toolbar. Electronic Notes in Theoretical Computer Science, vol. 235, pages 125-136, 2008.

[8] Web Filtering Toolbar. Add-ons for Firefox. https://addons.mozilla.org/en-US/firefox/addon/5823, 2009.

[9] Suhit Gupta et al. Automating Content Extraction of HTML Documents. World Wide Archive vol.8 issue.2, 179-224, 2005.

[10] Po-Ching Li, Mind-Dao Liu, Ying-Dar Lin, Yuang-Cheng Lai. Accelerating Web Content Filtering by the Early Decision Algorithm. IEICE - Transactions on Information and Systems vol. E91-D, pages 251-257, 2008.

[11] S. López, J. Silva. A New Information Filtering Method for WebPages. Informe Técnico DSIC - Universidad Politécnica de Valencia. Available at: http://www.dsic.upv.es/ jsilva/papers/TechReport-IF.pdf

[12] W3C Consortium, Document Object Model (DOM). www.w3.org/DOM

[13] C.J. Castillo, H. Valero, J. Silva. Web Information Retrieval Based on Syntax Distances. Informe Técnico DSIC - Universidad Politécnica de Valencia. Available at: http://www.dsic.upv.es/~jsilva/-papers/ TechReport-IR.pdf

[14] Jakob Nielsen. Designing Web Usability: The Practice of Simplicity; New Riders Publishing, Indianapolis ISBN 1-56205-810-X; 2010.

[15] Filippo Ricca and Paolo Tonella. Web application slicing. In *Proc. of International Conference on Software Maintenance (ICSM'01)*, pages 148–157. IEEE Computer Society, 2001.

[16] M. Baggi and D. Ballis. PHIL: A Lazy Implementation of a Language for Approximate Filtering of XML Documents In *Proc. of 16th International Workshop on Functional and (Constraint) Logic Programming (WFLP'07)*, Electronic Notes in Theoretical Computer Science, vol. 216, pages 93-109, 2007.

[17] M. Hammami, Y. Chahir, and L. Chen. Webguard: A web filtering engine combining textual, structural, and visual content-based analysis. *IEEE Trans. Knowl. Data Eng*, 18(2):272–284, 2006.

[18] Finn, Kushmerick, and Smyth. Fact or Fiction: Content Classification for Digital Libraries. *DELOS Workshop*, 2001.

[19] Debnath, Mitra, and Giles. Automatic Extraction of Informative Blocks from Webpages. *20th ACM Symposium on Applied Computing (SAC'05)*, 1722-1726, 2005.

[20] Gottron. Evaluating Content Extraction on HTML Documents. *2nd International Conference on Internet Technologies and Applications. (ITA'07)*, 123-132, 2007.

[21] Gottron. Content Code Blurring: A New Approach to Content Extraction. *5th International Workshop on Text-based Information Retrieval. (TIR'08)*, 29-33, 2008.

[22] Weninger and Hsu. Text Extraction from the Web via Text-Tag-Ratio. *5th International Workshop on Text-based Information Retrieval. (TIR'08)*, 23-28, 2008.