ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Automatic web content personalization through reinforcement learning

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Ferretti, S., Mirri, S., Prandi, C., Salomoni, P. (2016). Automatic web content personalization through reinforcement learning. THE JOURNAL OF SYSTEMS AND SOFTWARE, 121, 157-169 [10.1016/j.jss.2016.02.008].

Availability:
This version is available at: https://hdl.handle.net/11585/564849 since: 2022-02-03

Published:
DOI: http://doi.org/10.1016/j.jss.2016.02.008

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2024

This is the final peer-reviewed accepted manuscript of:

Ferretti, S., Mirri, S., Prandi, C., & Salomoni, P. (2016). Automatic web content personalization through reinforcement learning. Journal of Systems and Software, 121, 157-169.

The final published version is available online at: http://dx.doi.org/10.1016/j.jss.2016.02.008

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Automatic Web Content Personalization Through Reinforcement Learning

Stefano Ferretti, Silvia Mirri, Catia Prandi, Paola Salomoni

Department of Computer Science and Engineering
Università di Bologna
Bologna, Italy
{s.ferretti, silvia.mirri, catia.prandi2, paola.salomoni}@unibo.it

Abstract

This paper deals with the automatic adaptation of Web contents. It is recognized that quite often users need some personalized adaptations to access Web contents. This is more evident when we focus on people with some accessibility needs. Based on the user profile, it is possible to transcode or modify contents (e.g., adapt text fonts) so as to meet the user preferences. The problem is that applying such a kind of transformations to the whole content might significantly alter Web pages that might become unreadable, hence making matters worse. We present a system that employs Web intelligence to perform automatic adaptations on single elements composing a Web page. A reinforcement learning algorithm is utilized to manage user profiles. We evaluate our system through simulation and a real assessment where elderly users where asked to use for a time period our system prototype. Results confirm the feasibility of the proposal.

1. Introduction

Recent advances in Web technologies have revealed that adaptation of contents improves the access to the Web [1, 2, 3]. Among the different types of adaptation, self-adaptation is a way to organize contents based on the specific needs of a given user. Visualization preferences of a given user can be encoded and stored in a user profile. Such a profile can be employed to perform automatic content adaptation, thus realizing a content personalization. Commonly, these transformations adapt the layout of a Web page, as well as the formats and shapes of contents within the page.

The typical approach consists in applying such transformations to all the elements which compose the entire Web page. This might result in a complete alteration of the Web page. Without a wise manipulation, the Web page can become unreadable, thus making the efforts to help the user useless, or even worse, detrimental. In substance, a more user-centered approach should be exploited, able to customize only the shape of some specific Web page elements, instead of the whole page. In essence, customization is just limited to those elements that really affect users' reading.

Such an approach could have a strong impact, in particular for those users with some reading-related disabilities (i.e. people with dyslexia, users with low vision, users with color blindness, elderly people). Moreover, this approach can provide benefits even to those users

who are equipped with devices with different capabilities, such as tablets, smart-phones, and smart TVs. In this context, both readability and legibility are affected by different issues, such as Web text characteristics [4, 5, 6] and users' abilities [7, 8, 9].

This work presents a system, called ExTraS (EXperiential TRAnscoding System [1, 10, 11]), and thought to improve Web content legibility. The approach adapts some characteristics, in particular the text formatting ones (e.g. font size, font face, luminance contrasts), according to users' preferences and needs. ExTraS tracks users' behavior to learn and model their preferences and to automatically provide the best adaptation, tailored for each user, predicting his/her needs. Such an adaptation is based on a machine learning mechanism (Reinforcement Learning), exploiting the idea of reward/punishment. We have built a prototype system and we have evaluated it by means of a simulation assessment and tests with real users. Both of them confirm the viability of our proposal.

The remainder of the paper is organized as follows. Section 2 describes background and related work, while Section 3 presents the architecture of our system, detailing how it profiles users, how it applies the machine learning mechanism and how it adapts the content. Section 4 introduces the prototype we have developed, showing some screenshots. Section 5 reports the simulation assessment and the tests with users we have performed, detailed the obtained results. Finally, Section 6 concludes the paper illustrating main findings and further work.

2. Background and related work

Our work takes into account several issues and has been based on many related work. This section aims to briefly describe the most significant ones, which are related to improving Web pages legibility with font adaptation, adaptation and personalization of digital and Web contents and services, and the use of machine learning techniques and algorithms to track and understand users' experiences and behaviors.

According to the definitions in the literature, legibility is related to perceiving text by distinguishing each character from all other ones in the font, without any ambiguity. Hence the more text letters are distinguishable, the more such a text is legible [5, 6]. A different issue is the readability, which is related to reading and understanding textual information: the more a text is complex (with difficult words, long paragraphs, acronyms, abbreviation, technical terms, foreign sentences, etc.) and the less a text is readable [5, 6]. Summing up, legibility refers to text perception, while readability refers to its comprehension. In this context, the reading activity can be strongly affected by textual characteristics, such as font face, text size, background and foreground colors, alignments, paragraphs, words and letters spacing [4, 7]. Online reading abilities and textual characteristics, which better support users in such a kind of activities, are the topics of several studies, taking into account also the device in use (including mobile [12] and smart devices [13]), as well as specific users contexts ([14]) or condition (including reading disabilities [8, 15], visual impairments [9], ageing [7] and low-literacy [16]). Some of these studies [16, 17] have emphasized the importance of adapting only those parts of content that represent an effective barrier to users. This way, the whole content and its layout are not distorted, and customization is just limited to those elements that really affect users' reading. Outcomes from these previous studies have been taken into account in our work.

In context of supporting users' reading abilities, research results have pointed out that users need to be able to adapt text to meet their particular needs. Without such a personalization a user's need can conflict with general best practice, or can conflict with another user's need [15]. For instance, many people with dyslexia and other reading impairments need low contrast between text color and background color [17], while many people with low vision and many people with declining eyesight due to ageing need high contrast [15]. Moreover, there are differences not only between users, but even an individual user's need can change.

The personalization and adaptation of digital Web contents and services are at the basis of several studies. The necessity for personalization ranges from the need for the use of different formats and aspects (due to different devices, context of use, or specific preferences and conditions [3, 18]) to the need for different contents (recommending resources that can be of interest for the users, according to the preferences and the behaviors they have already shown somehow). Several works have been done in this field, such as [19], where users' preferences are taken into account User-centric personalization of Web sites, by means of personalized content recommendations is at the basis of [20]. In works like this latter one, user profiling is a fundamental activity, because it can drives the personalization and adaptation of content and services [21]. Users' profile and model can also be built according to their behaviors.

Users' behaviors as personalization-drivers have been exploited in [22]; in this case, the author proposes an approach to construct a dynamic user model, strongly based on user's behavior, which drives the personalization of media augmentation, accessed by using some mobile devices. Another adaptation approach, called "experiential transcoding" [23, 24], exploits the concept of experience-based transcoding. Compared with more traditional forms of Web transcoding and adaptation, the main advantage of experiential transcoding is that it is strongly user-centered. It applies techniques and mechanisms which adapt contents on the basis of users' experience. In particular, in [24], eye tracking has been used to understand users' experiences.

Analogously to these cited solutions, our work exploits the idea of adapting Web contents on the basis of users' experience. In particular, we track past users' adaptations so as to learn users' preferences and to provide (and/or suggest) adequate personalization. This approach has been inspired by recommendation systems used in e-commerce Web sites, social networks, search engines, etc. A similar approach has been used in [25], where the authors exploited a reinforcement learning algorithm in order to customize the rendering of advertisements in Web pages, on the basis of users' preferences. Learning and predicting users' preferences, so as to drive recommendation systems is also the main aim of [26]. The reward/punishment approach has been exploited also in [27], where the authors propose the use of the Q-learning algorithm for modeling the behavior of agents in simulations.

There are various proofs that Q-learning does converge to the optimal Q function, under very mild conditions [28, 29, 30, 31]. Moreover, several works compare Q-learning with other reinforcement learning algorithms [32, 33, 34]. The Q-learning algorithm has been chosen among all the other ones, because of its capability to continue learning while changing policies [33]. In fact, Q-learning tends to converge a little slower if compared with other reinforcement learning algorithms, in particular with the other temporal difference learning ones (e.g. SARSA), but it shows a more efficient reaction when policies change during time [33]. Such a

characteristic is strategic in our approach, because, as shown in the following sections of this paper, we take into account that users' preferences and needs can change, due to temporary conditions in the context of use (e.g. lighting in the room, or external light, or different devices with different display characteristics) or due to more permanent conditions (e.g. changes in the vision while getting older). Another reason which has driven our choice is that Q-Learning is a simple and widely understood formulation of reinforcement learning, this would make more feasible the transfer of the lesson learnt to any other reinforcement-based approach in future works, as foreseen in [35]. An interesting approach which would be taken into account in our future works is represented by a novel alternative, called Reinforcement Learning-based Decision Trees (RLDT), that combines Reinforcement Learning (RL) and Decision Trees and that provided interesting results, as shown in [36].

3. System Architecture

Our system architecture is structured as shown in Figure 1. ExTraS is structured in three modules: the Profiling Module, the Learning Module and the Adaptation Module.

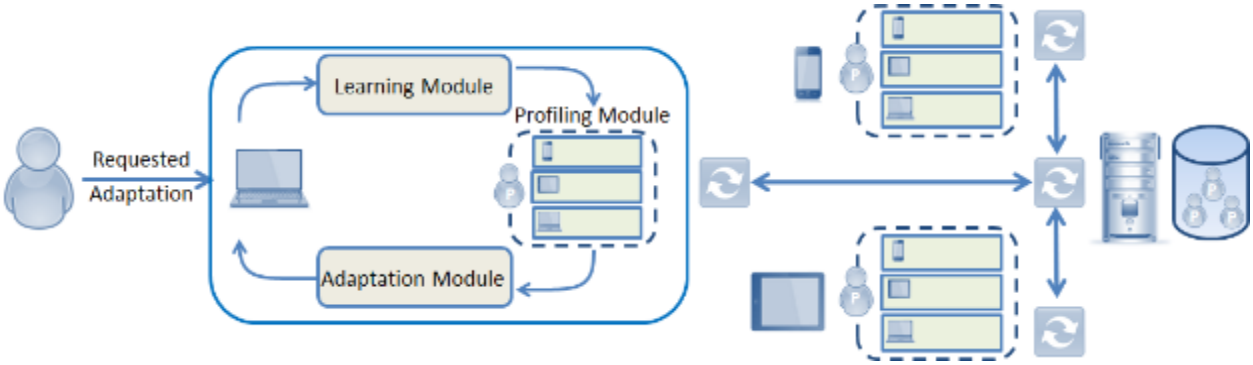

Fig. 1 - System Architecture

The aim of ExTraS is to set adaptations on different kinds of markup document, such as LaTeX, PDF, RTF documents, etc. Currently, our system works on explicit and descriptive markup documents, in particular HTML pages. While the user exploits a simple contextual menu to set his/her preferred adaptations (performed by means of the Adaptation Module), his/her behavior is tracked, with the aim of learning his/her preferences (thanks to the Learning Module, described in Section 3.2). All the user preferences are stored in a profile, managed by the Profiling Module, (presented in the following Section 3.1). This lets to apply or propose suitable adaptations automatically during future interactions. The profile is automatically created, while surfing the Web, and it is locally stored on the device. If a user exploits different devices, and the profile is shared among them, updates are synchronized, as in typical cloud-based approaches. In particular, specific sections of the profile are tailored according to the user's devices (smartphones, tablets, laptops, and so on). Whenever a new device comes into the picture, a default section of the profile is set up according it and then the system begins to track the user's behavior while he/she exploits such device.

The Adaptation Module (described in Section 3.3) is in charge of transcoding HTML tags, attributes and their related values and of personalizing CSS rules. In order to (automatically and/or on the basis of users' explicit requests) adapt contents based on the user's profile, the

system is capable to inject new tags or attributes into the content, and it is able as well to substitute original tags or attribute values with customized ones.

3.1 Profiling ExTraS Users

Generally, profiling users is an activity which can be useful in several contexts and applications, such as user interfaces and Web applications (i.e., recommendation system in e-commerce Web sites, personal data in social networks or in search engines filters, and so on). Data gathered in users' profiles can range from personal data to contextual conditions, from user's skills to his/her personal preferences [37, 38]. Such data can be collected in different ways, for instance the user can explicitly declare them or they can be learnt by the system as the user exploits it.

In our system, the user profile is a collection of typographic characteristics gathered by the system, on the basis of adaptations the user has requested so as to improve Web content legibility and readability. As the user exploits the system, asking for adaptations, our system computes automatic adaptations and related reward/punishment values (according to our Learning System, described in the following subsection), tracking the user's behavior. Periodically, the system updates the user's profile, adding new characteristics, updating reward/punishment values for the already existing ones or adding new characteristic values as requested by the user.

The system provides adaptations as state changes: from the original one to the adapted one, according to each user's request. Characteristics of the original state are the ones the user has discarded, while characteristics of the adapted state are the ones the user has chosen. Hence, the system learns all the Web page characteristics which affect user's reading ability (both in positive and in negative ways). Obviously, the user's profile will be the more accurate the more the user asks for adaptations. As described in the following subsection, the system punishes discarded characteristics, while it rewards characteristics the user has chosen through the requested adaptations.

We have designed an xml-based profile which is structured in different parts, according to the devices the user exploits (see Figure 2 as an example). In each of these parts, the profile describes all the preferences of the user in terms of typographic characteristics (e.g. font family, font size, colors, alignment, etc.). In order to describe this, each one of such typographic characteristic is set as an element (tag), which is equipped with a "v" value that identifies the specific value (i.e. "Verdana" for font family, or "16" for font size, or "black" as background color, and so on) and with a "**r**" value that tracks the preferences of the user with respect to "**v**". In the following, when we say "the user has discarded an element v", we mean that the user decided to not visualize some content using that typographic characteristic "v", considering it as a barrier to his/her reading activity, affecting the content legibility. On the contrary, when we say "the user has chosen" to exploit the "v" element, we mean that he/she prefers or accepts to visualize some content that have that specific textual characteristics.

The "r" attribute is related to the reward/punishment assigned by our system and can assume different values (as described in the following subsection). Summing up, we can face three different possible cases, according to user's behaviors:

- if “r”<0, the user has discarded the “v” value for a given element;
- if “r”>0, the user has chosen (or accepted) the “v” value for a given element;
- if “r”=0, the “v” value for a given element has been chosen and discarded the same amount of times.

The absence of a characteristic or a specific “**v**” value in the user’s profile means that the user has never asked for such characteristic adaptation or he/she has never discarded or chosen such a “**v**” value. Thus, the idea is that user’s preferences are identified by ExTraS on the basis of discarded and chosen text formatting characteristics, tracking even those elements that represent a barrier for that specific user. The profile is initially empty; the system learns preferences and updates the profile while the user surfs the Web. In particular, the requirement of adaptations and changes to some elements in a Web page corresponds to punishments to the values for these elements, since the user considers them as barriers. Further, our system can perform automatic adaptations (as described in subsection 3.3); when these proposed adaptations are accepted by the user, the values related to the adapted elements obtain a reward.

```
...
<device type="tablet" id="3" display_width="" display_height="">
      …
      <font_face family="sansserif" r="9" v="verdana"/>
      <font_face family="sansserif" r="5" v="arial"/>
      <font_face family="serif" r="-2" v="times new roman"/>
      <font_face family="monospace" r="-5" v="courier new"/>
      …
      <font_size r="8" v="18"/>
      <font_size r="3" v="16"/>
      <font_size r="-12" v="12"/>
      <font_size r="-5" v="9"/>
      …
      <line_height r="6" v="1.5"/>
      <line_height r="-2" v="1"/>
      …
      <text_alignment r="7" v="left"/>
      <text_alignment r="-1" v="center"/>
      <text_alignment r="-7" v="justify"/>
      …
</device>
...
```

Figure 2. A fragment of a user’s profile.

The profile groups the preferences by the devices the user exploits. For each device, the profile stores the type, an id value, the display width and height (in terms of pixel) and other data about the device in use (i.e. the ability of changing display luminosity on the basis of ambient lighting). The device in use is deduced by the HTTP request and its capabilities are collected from repositories, such as WURFL [39].

Figure 2 shows a fragment of a user's profile, as an example. Textual characteristics are grouped by a tablet device. In such an example, the user has preferred (among all the adaptations): Verdana as font face, 18 as font size, 1.5 as line height and left as text alignment. In the same example, the user has discarded: Times new roman and Courier New as font face, 9 and 12 as font size, 1 as line height and center and justify as text alignment, while he/she has used the tablet.

3.2 Learning System

A reinforcement learning algorithm, based on the idea of reward/punishment [40] is exploited as the machine learning system that manages the user profile [11]. Based on it, the user profile is updated, either by adding new preferences/barriers, or by updating reward/punishment values associated to existing elements of the profile.

The specific algorithm in charge of managing the punishments/rewards described in the previous section is Q-learning [29]. In essence, this scheme allows learning the optimal policy to accomplish, based on the history of interactions of the system with the environment. The history is a sequence of state-action-rewards $<s_0,a_0,r_1, s_1,a_1,r_2, ...>$, meaning that when the system is in a state s_i, it takes the action a_i, obtaining a reward r_{i+1}. For each of these actions, the algorithm updates an estimation $Q[s,a]$ of the reward obtained by taking the action a when the system is in state s. Each time the system takes an action, given a state, and it receives a reward, such an estimation $Q[s,a]$ is updated based on the following equation:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r+ \gamma \max_{a'} Q[s',a'] - Q[s,a]) \qquad (1)$$

where α is a step rate; r is the observed reward; s' is the state where the system goes by taking the action; $\gamma < 1$ is a parameter, that works as a discount value, which serves to give a weight to the estimation of the maximum reward $\max_{a'} Q[s',a']$ that system can measure by taking some future action in the novel state s', based on its current information.

The general algorithm executed by our system is described in the following.

1. When a user opens up some few Web pages, then he/she can adapt some characteristics. At this initial stage, the system just tracks his/her behavior and starts to learn his/her preferences. Moreover, it starts to assign reward to the chosen characteristics, according to the related state, so as to start building the user's profile.
2. When a user opens up a new Web page, then the system parses the DOM and the related style rules, taking into account the user's profile.
3. All characteristics of elements in the DOM are matched with the user profile. In particular, all those characteristics that the user has discarded (with a low "r" value in the profile) are collected.
4. For each characteristic, the system maintains a negative threshold value "t" that is exploited as follows. For each element with a low "r" value, the adaptation that has associated the highest reward is considered. In particular:
    a. if $t < r < 0$: the system proposes such an adaptation to the user; if the user accepts it, then a reward is assigned; else, a punishment is associated;

b. otherwise (r < t), the system automatically adapts such characteristics, by substituting them with the ones the user prefers. If the user ignores the automatic adaptation (i.e. it is accepted), then a reward is assigned. Else, the user rejects the adaptation; thus a punishment is associated.

5. The user can in turn adapt some element by himself/herself. In this case, the system observes the user's behavior and assigns related reward/punishment. Unchanged characteristics receive no reward.
6. All the updated rewards and punishments are stored into the user profile.

The system repeats phases from 2 to 6 any time the user opens up a Web page.

3.3. Adapting Content with ExTraS

Content adaptation is the action of transcoding or transforming content so as to meet users' preferences and needs, even according to the device he/she is using [41]. In ExTraS, we provide content adaptations with the aim of improving Web pages readability and legibility. In our system, the adaptation process is in charge locally, on the client side. The system can locally decide and employ the most appropriate adaptations, according to user's profile, which is fed on the basis of the user's behavior.

As we have already claimed, the aim of our system is to adapt several kinds of markup document (e.g. LaTeX, PDF, RTF, etc.). But currently ExTraS works on explicit and descriptive markup documents, in particular HTML pages. Our system performs adaptations by changing tags, attributes and related values: the system injects new tags or attributes and/or it substitutes original tags or attributes value with the customized ones, changing markups.

When a user opens up a Web page, then the system parses the DOM and the related style rules, by considering those characteristics with a negative "**r**" value in the user's profile. If such characteristics are found in the page, then the system decides if automatically adapting them (if the "**r**" value is less than a threshold "**t**") or if just proposing to the user such an adaptation (if the "**r**" value is greater than "**t**"). The discarded characteristics are substituted by the "**v**" values with the highest "**r**": since the user has chosen them the most (to improve his/her reading ability), so we can assume these are his/her preferences in terms of such characteristics.

For instance, let us consider the user's profile depicted in Figure 2. If the user opens up a Web page and the system finds a paragraph written in Times New Roman (with "**r**" equal to -2), then the system proposes an adaptation from Times New Roman to Verdana as font face. The system chooses Verdana instead of Arial, because the Verdana "**r**" value is higher than the Arial one.

A list of adaptations the system can employ is the following one:

- Zooming font size: the system can increase font size by a specific percentage or unit.
- Changing font face: the system can substitute the original font face with another one.
- Changing font style: the system can set a specific style (i.e., italic, underline or normal) to the text.
- Changing spacing: the system can change spacing-related attributes (i.e., letter spacing, word spacing, line-height, margins, paddings, etc.).

- Changing text alignment: the system can set users' favorite text alignment (left, right, center or justify).
- Enhancing luminance ratio: the system can increase the luminance ratio. The system keeps the same background color and computes a new foreground color, so as to enhance the luminance contrast ratio.
- Changing background and/or foreground colors: the system can set different background and/or foreground colors, according to users' choice.
- Language translation: the system can substitute original words and/or sentences in translated ones.
- Acronyms expansion: the system can substitute acronyms with their related expansions.

While the most of the ExTraS adaptations are devoted to improve Web pages legibility, the aim of personalization related to Language translation and Acronyms expansion is to enhance the readability of Web textual content.

4. ExTraS Prototype

We have designed and developed a prototype of our ExTraS which adapts HTML documents. Such a prototype has been implemented as a Firefox extension. Users can activate a contextual menu to set the preferred adaptations on an HTML page. Then the system performs such adaptations by suitably changing the HTML and/or the CSS code of the page, on the client-side. In the meanwhile, the system tracks users' behaviors with the aim of learning their preferences and then automatically applying or proposing suitable adaptations.

This prototype has been implemented as a Firefox add-on by means of Mozilla SDK [42].

Currently, the prototype has been tested on laptops (equipped with different operating systems) and on Samsung Galaxy Tab 2 devices, equipped with Android 4.0.4 and Firefox 33.0.

Scripts have been created so as to:

- provide an ad-hoc contextual menu, letting users choose among a sub-set of available adaptations (as shown in the screenshot depicted in Figure 3): changing font face changing font size, and changing background and foreground colors.
- Adequately modify the DOM of the HTML page, injecting new attributes or changing values for the already existing ones. In particular, the prototype can add "**`style`**" attributes with CSS rules to the element tag or can change the existing CSS rules values, according to the adaptations the user has requested. This way, thanks to CSS cascading feature, customized values of inline rules override the same ones eventually declared in external or internal CSSs.
- Add scripts to create and activate pop-ups: when an adaptation is just proposed to users, this is triggered on the mousehover event, by means of AJAX scripts. This is the case described in the step "4.a" of our algorithm, as detailed in subsection 3.2.

Figure 3 shows a screenshot of a Web page with the contextual menu of the prototype we have developed: after activating such a menu, it is possible to zoom in, to zoom out, to change the font family, to change the background and foreground colors of the specific element the user has chosen (which is highlighted by means of a colored border).

Figure 4 shows a screenshot (of the same Web page in Figure 3), when the chosen adaptations are automatically performed. In particular, the font size of links to results of the Google search has been increased. The description of each result has been adapted by increasing the font size and by changing the background and the foreground color. Also links which follow the result descriptions have been increased in terms of font size. This is the result of automatic adaptations when a user has frequently chosen to increase the font size of links and of small textual elements (less than 14px) and the color contrast of textual element where the original background color was white and the original foreground color was gray (with a low color contrast).

Figure 5 compares two screenshots of the same Web page (Yahoo! news home page, http://news.yahoo.com/), which have been taken respectively before and after having applied some adaptations by means of ExTraS. In particular, the adaptions were done with the aim of increasing the font-size and the color contrast of the titles and the summaries of the news.

Finally, Figure 6 shows a screenshot of the options available for the ExTraS extension, the user can set. For instance, the user can choose to highlight borders of the textual elements in the HTML page, when he/she activates the contextual menu in order to choose a specific adaptation to the element. By means of these options, the user can decide if he/she wants ExTraS always applies automatic adaptations to any Web page (*On*), or if the user can choose automatic adaptations only for a specific Web page (*Off*), or if all these options are always applied to any Web page (*Default*).

Screenshots in Figures 3, 4, 5 and 6 have been taken from a PC equipped with Firefox 33.0.

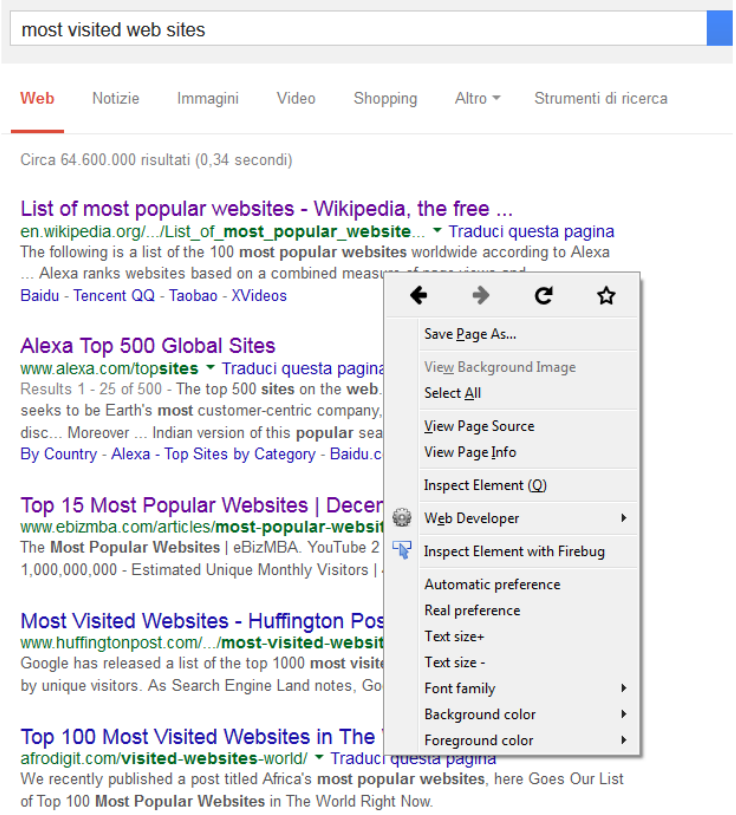

Figure 3. A screenshot showing our contextual menu.

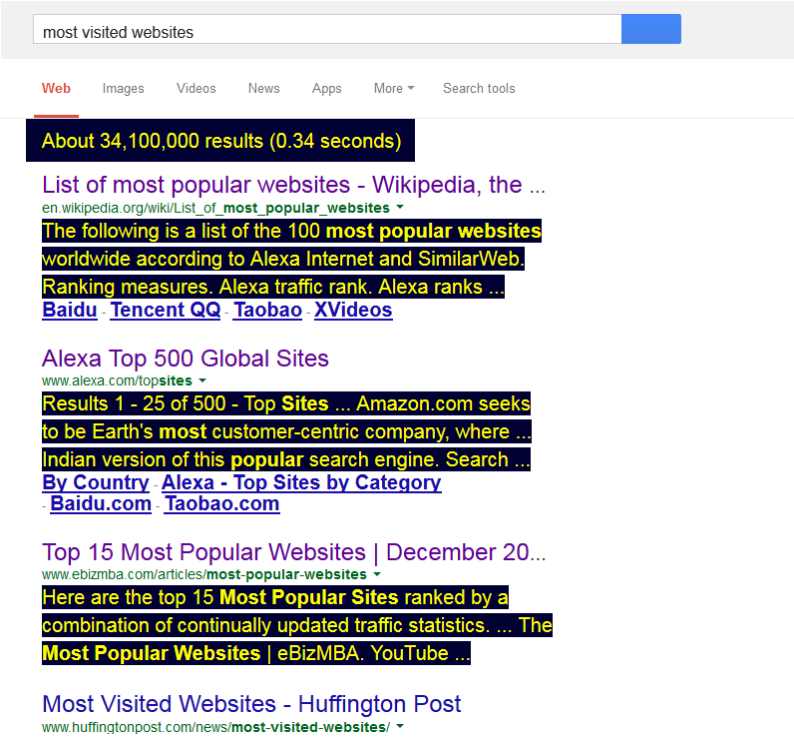

Figure 4. A screenshot showing adapted elements.

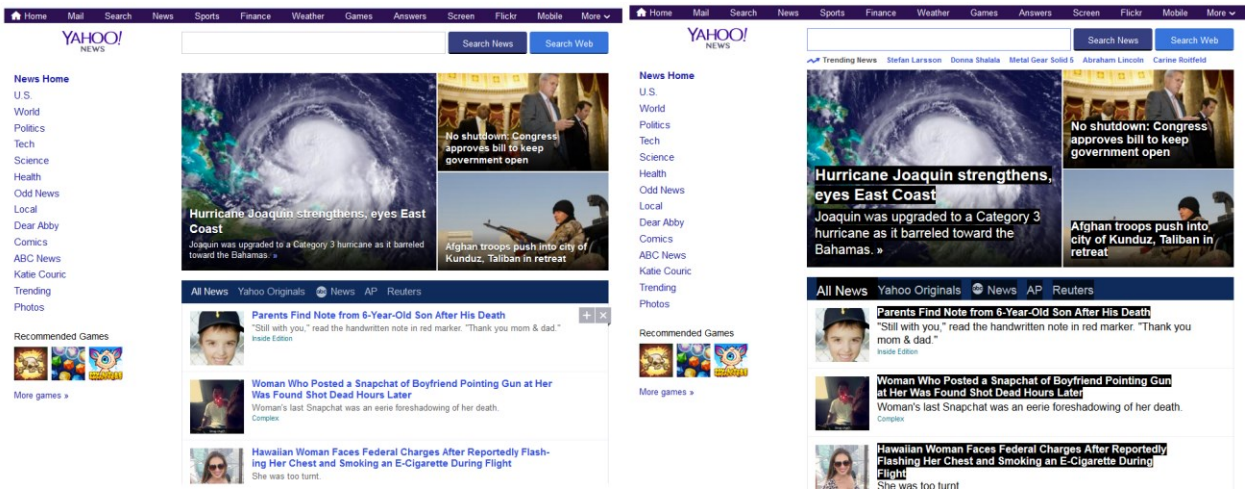

Figure 5. Screenshots taken from Yahoo! news home page (http://news.yahoo.com/) before and after adaptations on titles and summaries of news (increasing color contrast and font size).

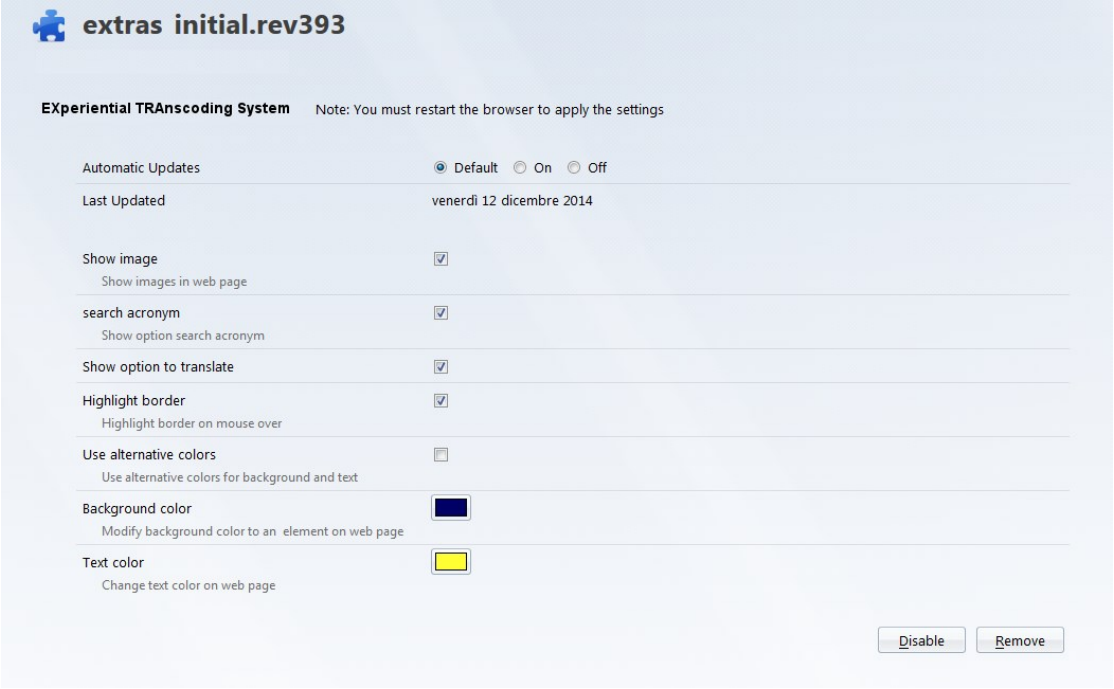

Figure 6. A screenshot of the ExTraS options page.

5. ExTraS Evaluations

In order to assess the viability of our proposal, we have developed a simulator which mimics the user behavior and we have integrated it with a framework that implements the reinforcement learning algorithm. Moreover, we have conducted tests with real users, equipping their laptops with the ExTraS prototype.

In subsection 5.1 we describe the simulations we have performed and the related results and in subsection 5.2 we present the tests we have conducted with the real users.

5.1 Simulations

A first evaluation was conducted via simulation [1]. The rationale was to understand how the presented approach behaves when different types of users (i.e. with different profiles) are considered. Since it is difficult to conduct a wide range of real tests with different users, we resorted to simulation. Users were simulated as agents having different needs, based on their profiles. We made real tests with a typology of users, which is described in the next subsection.

The aim was to perform a quantitative evaluation of the benefits provided by the intelligent adaptation system. Users were simulated to have some kinds of disabilities; this corresponds to set users having some specific preferences about characteristics of text styles, fonts and dimensions, associated to elements composing some Web pages.

We modeled three different kinds of users:

- users with dyslexia;
- users with low-vision;
- elderly users.

The typical profiles for these users are quite different and very specific. Thus, these contexts are particularly challenging for our system.

In substance, we have created three different user models, one for each type, with specific users' needs, grouped as a profile, corresponding to the most typical preferences that users might want when accessing a given content, avoiding those text characteristics which can represent a barrier, affecting legibility for them, according to the literature [7, 8, 9]. Each simulated user adopts its related model.

Going into the details of the user profiles, we assume that a user with dyslexia commonly prefers to read text with an Arial font, or a San Serif font family. He/she prefers that text elements are displayed with a font size equal or higher than 18 points. As concerns the text alignment, a left alignment is preferred while a justified alignment is highly deprecated [8].

A user with low vision, instead, benefits from San Serif type fonts; font size should be around 16-18 points. A left alignment is preferred, with a line height set to 1.5 [9].

Finally, a 14-point font size has a significant improvement in legibility for elderly people; they also prefer san serif fonts, in particular Arial and Verdana (as a second choice) [7]; moreover, left alignment is preferred, as well as space between lines, paragraphs and around clickable targets (such as links and buttons, so that each one is easy to target and hit separately) [6].

In order to assess whether (and how) the system is able to react to changes in user preferences, we also implemented a sort of inconsistent user, that from time to time gives some feedbacks to our system which are not compliant to his/her user model. Finally, we have also implemented a user who changes his/her preferences during the simulation. As an example, this can be the case of an elderly user.

Agents were set to browse the Web, by randomly selecting Web pages among a set of the 10 most visited Web sites, taken from the ranking provided by Alexa top Web sites list (November 2014) [43]. The set of Web pages was: Google home page, Google search page (http://www.google.com), Facebook news, Facebook user's wall (http://www.facebook.com), YouTube home page, YouTube video page (http://www.youtube.com), Yahoo! home page, Yahoo! result page (http://www.yahoo.com), Wikipedia home page, Wikipedia item page (http://wikipedia.org).

To implement the reinforcement algorithm, exploited by the Learning Module, the Piqle framework was utilized [44]. In order to do that, we have customized Piqle, so as to simulate our system and the users' behavior. Summing up, we have mainly implemented the following issues:

- Three main entities simulate the adaptation process: Environment, State and Action. The user directly interacts with the Environment. A State is the collection of the text characteristics of the Web page the user is surfing, while an Action is an adaptation of a specific text characteristic for a specific element of the Web page.
- The reward/punishment is assigned based on the couple *(State, Action). I*t is computed according to the user's feedback; this feedback was generated on the basis of the user model.

- The system starts learning during the initial observation phase. In this phase, it just observes and tracks user's behaviors, rewarding the chosen characteristics (according to the related state) and understanding the barriers that affect user's reading.
- After the initialization phase, it starts proposing and automatically performing adaptation.
- A punishment is assigned to the discarded text characteristics (identified as barriers), so as to enforce the need of adapting them.

In order to evaluate how our system learns users' preferences, we have conducted several simulations for each of the three user models we have already described. Users were set to surf the Web by browsing an amount of pages ranging from 1,000 to 500,000.

For each simulation, we have counted the adaptations the system has performed that were not compliant to the user profile. This way, we can observe the number of errors the system does, while it learns. Such a value represents the distance between the simulated user profile and the user model the system learns during the trials. Simulation steps are represented by browsed Web pages. When the errors are 0, then the system has learnt the users' preferences. We have observed that the average number of browsed Web pages after which the number of errors (in terms of adaptations the user rejected, because they are not compliant to his/her profile) tends to zero changes according to the initial phase of observation. This means that the longer is the observation phase and the less the number of wrong adaptations the system performs.

Figure 6 shows a single simulation of a user with dyslexia, which browses 1,000 Web pages. The user does not change his/her preferences during the trial and, in this case, our system proposes 10 errors after 14 browsed Web pages, after that the number of errors tends to zero.

In order to better test our system, we have also simulated users who change their minds during the trials. In particular, we have simulated some users with a certain probability (expressed in percentage, from 0.1% to 20%) of giving feedbacks which are not compliant to their profile. Figure 7 shows a single simulation of a user with low vision who gives a 1% of inconsistent feedback, in a trial with 1,000 Web pages. The peaks in the plot correspond to the situations when the user does not provide feedbacks compliant to his/her profile. In these cases, the system applies some sort of wrong adaptations contextually, due to those inconsistent user's behaviors. Such inconsistent behaviors could be due to some contextual situations that can affect user's reading, such as, for instance, a different ambient light while the user is exploiting a device which cannot automatically adjust the display luminosity according to the ambient lighting. In this case, the user needs adaptations which different from the ones that the system usually automatically performs, requiring a different color contrast. After those peaks, the system continues to perform adaptations compliant to the profile and then the user accepts them as usual, with no errors. No additional learning phase is required, since the profile has not been changed.

Moreover, we have simulated users who change their minds just once during the trials. Figure 8 show a single simulation of an elderly user who asks for bigger font size dimensions after 500 pages. The peak shown in the plot corresponds to that change of preferences. After that, an additional learning phase is needed. This means that the system applies some adaptations which are still compliant to the old preferences, but not to the new ones. Hence, the user has to reject such wrong adaptations, so as to let the system learn his/her new different needs. In this

case, since only one preference has been changed, then the number of steps needed, so that the number of errors results negligible, is less than the initial ones. The same is for the number of errors the system does at the very first step the user changes his/her mind about font size dimensions. In cases like this one, we have observed that the later the change of preferences happens and the higher is the number of errors that the system commits after this modification in the user's profile.

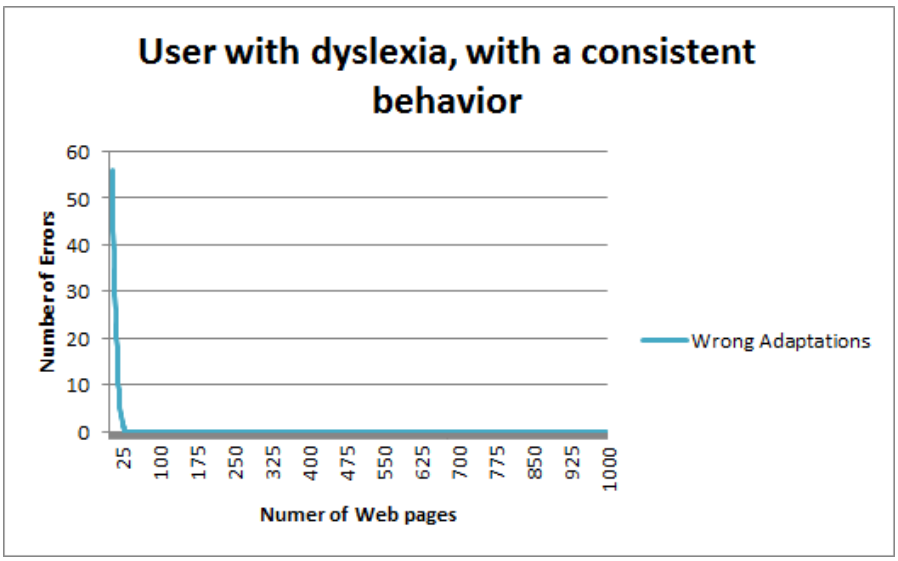

Figure 6 – User with dyslexia, with a consistent behavior

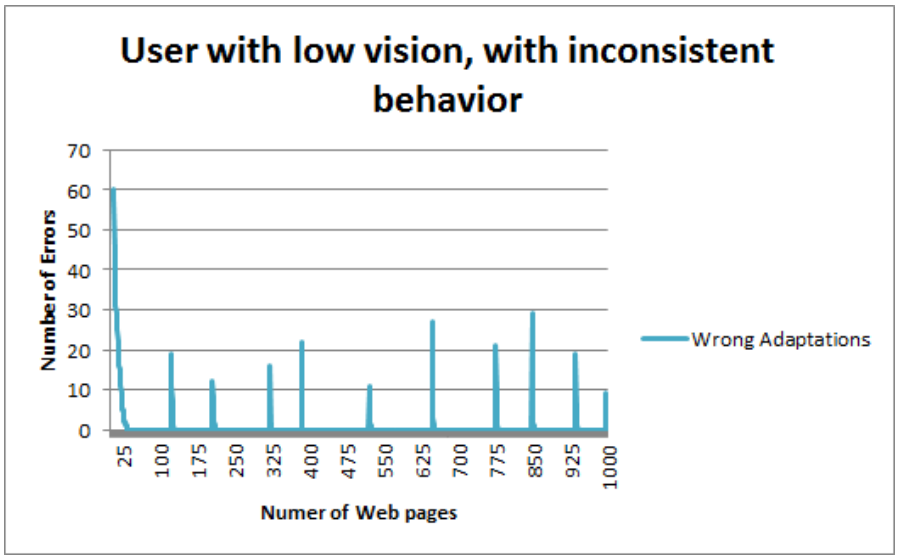

Figure 7 – User with low vision, with inconsistent behavior

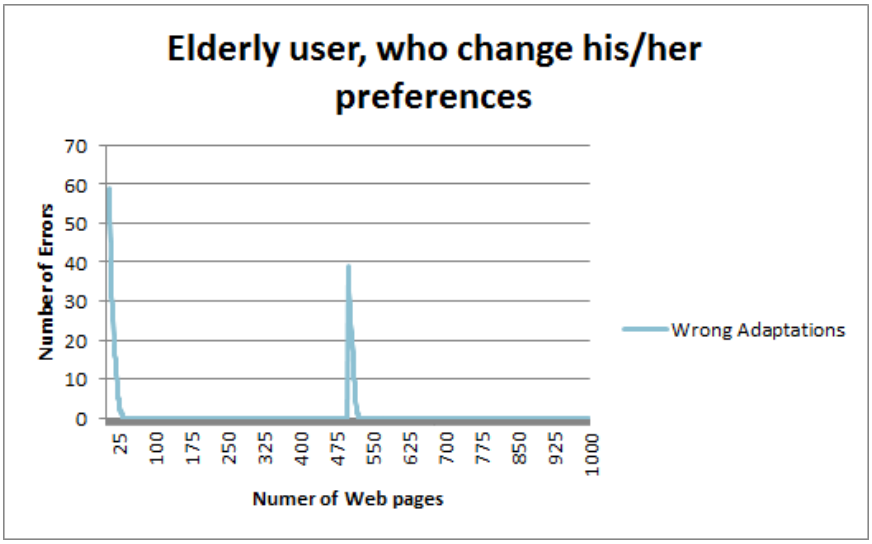

Figure 8 – Elderly user, who changes his/her preferences

The simulation campaign confirmed that the system learns the user's preferences, independently from the simulated user. But needless to say, having experimental results obtained from a real evaluation might provide useful indications on the performance and the goodness of the system. Results from this evaluation are reported in the next subsection.

5.2 Tests with users

We have conducted a test with users campaign involving 10 elderly users [45]. In this subsection we describe the users sample, how we conducted the tests and the related results.

The age of the involved users ranges from 65 to 69, with an average value of 67.1; 50% are female and 50% are male, all of them are Italian language mother tongue. Their use of the Web is vary various, some of them use to navigate 3 hours per day, while some other just few hours a week, with an average value of 10.4 hours per week, with a minimum value of 3 hours and a maximum of 22 hours per week. The Web site all of them navigate most frequently is Google, then they browse other Web-mail providers, Wikipedia, YouTube, news Web sites, Web sites about recipes, cooking tips, and how-to food, sport Web sites, etc. All these data have been collected by means of a pre-test questionnaire.

We have divided such users in two groups, according to their skills in using computers and other digital devices (i.e. smart phones, tablets) and according to the use of the Internet and the Web they usually do. In particular, we have considered the average time per week they spent navigating the Web, if they have got a smart phone or a tablet and if they usually also mobile devices to surf the Web, how many email, instant messaging and social network accounts they daily manage and their claims about the use of digital technologies they do and about how they feel comfortable in using them. The result is that we have identified 2 groups, each one is composed by 5 users, and we have called them the expert group and the non-expert one, according to the users' technological skills. The average amount hours spent while browsing the Web for the expert group is 16.4 hours per week, while for the non-expert group is 4.4.

The ExTraS users evaluation has been conducted in two phases: during the first phase we asked our users to complete some tasks by using ExTraS installed on their laptops, while during the second phase we let the users exploit ExTraS for an 8 weeks period. The users have filled a questionnaire related to such two phases, reporting about their whole experience with ExTraS.

The tasks we have asked our users to complete are the following ones:

1- Increasing the font-size of news in one of the most frequently navigated news Web site.
2- Changing the background and the foreground color of the results of a search on Google.
3- Changing the font-family of a Wikipedia term definition.
4- Changing some ExTraS options on the Firefox add-ons page.

All the users completed all the tasks, and answered our questionnaire, by declaring their appreciation of the feature exploited to complete each task (assigning a score ranging from 1 to 10) and the time (in minutes) they have spent to complete it. Results are reported in Table 1. The table groups the users in experts and non-experts. Generally, the assigned score and the time spent in completing the tasks show us that the users have appreciated the ExTraS features, found them easy to be used.

Table 2 reports the average values for the whole set of users and for each one of the two groups. This table shows that generally the expert group assigned highest scores to the task and spent less time to complete the task, compared with the non-expert group. In particular, the task with the less average score and which required more time to be accomplished was the one related to the option page on the Add-ons Firefox section, which is the task with the highest time needed to be accomplished. While the task with the highest score is the one related to increasing the font-size in a newspaper Web site and the task with the lowest time needed to be completed is the third one (changing the font-family of a Wikipedia term definition). This means that the most appreciated task is not the one which requires less time to be done (the third task). The first task was appreciated because each user experienced change in their vision ability due to their age and they really enjoyed the possibility of increasing the font-size of single paragraph and part of textual content in Web sites, while they generally found less interesting the idea of changing the font-family, as witnessed by User 5 who commented:

"*I use to navigate the Web every day and I often forget my glasses in my bag or in other rooms. Increasing the dimensions of single pieces of textual content in the Web pages could really help me while I browse. Obviously it is not the same, however it is useful*"

We assume that the users generally consumed less time to complete the third task because the operation on the ExTraS contextual menu is very similar to the one needed to complete the first task, which was already done.

Task 2 was particularly appreciated by User 9, who commented:

"*I like the idea of choosing background and foreground colors in Google results. I think this could be really useful when I read news on my favorite newspaper Web site in the morning."*

		Expert Group					Non-Expert Group				
		User 1	User 2	User 3	User 4	User 5	User 6	User 7	User8	User9	User10
Task 1	Score	9	9	9	9	9	8	8	8	8	8
	Time	<1	<1	1	<1	<1	2	1	1	1	2
Task 2	Score	7	7	7	8	7	6	8	7	8	7
	Time	<1	1	1	1	<1	2	1	1	1	2
Task 3	Score	8	7	7	6	8	7	8	6	6	8
	Time	<1	<1	<1	<1	<1	1	1	<1	1	2
Task 4	Score	8	9	7	6	7	6	6	6	7	6
	Time	2	2	3	2	2	5	3	2	4	5

Table 1. Phase 1 questionnaire results

		Expert group	Non-expert group	Average Values
Task 1	Score	9	8	8.5
	Time	<1	1:24	1:06
Task 2	Score	7.2	7.2	7.2
	Time	<1	1:24	1:09
Task 3	Score	7.2	7	7.1
	Time	<1	1:06	<1
Task 4	Score	7.4	6.2	6.8
	Time	2:12	4:24	3:18

Table 2. Phase 1 questionnaire results – Average values the two groups of users

After an 8 weeks trial period, we have asked our test users to fill a last questionnaire composed by 9 closed-ended questions and by an open-ended question. This latter one lets the users provide an open description of their experience with ExTraS, including weakness and strengths of the system, what they have really appreciated and what have affected their experience. The users are asked to respond to the formers, expressing how much they agree or disagree with the sentence in the question, fitting along an ordinal scale, ranging from 1 to 10 (where 1 indicates that the user totally disagrees with the sentence, while 10 means that the user totally agrees with it). These closed-ended questions aim to evaluate users' appreciation about: the ExTraS contextual menu (in terms of usefulness and easiness of use), the available

adaptations, the mechanism which automatically performs the adaptations, wrong adaptations automatically performed, additional Web pages loading time (due to the Adaptation Module phases). Moreover, we asked them about their willing of continuing using ExTraS on their browsers.

Table 3 reports the questions and the related scores assigned by the users, while Table 4 shows the average values assigned by the whole set of users and by each group. The assigned scores indicate an overall satisfaction among all the users, which affirm that the contextual menu is useful and easy to be used and that the system worked in a fine way, usually proposing them adaptations which improved the legibility of the Web page they were navigating, according to their preferences. Non-expert users assigned highest scores to the usefulness of ExTraS, to the adequateness of the available adaptations and to the effectiveness of the automatically performed adaptation in increasing the legibility of the Web pages. Expert users scored highest values about the conformance of automatically performed adaptation to their needs.

		Expert Group					**Non-expert Group**				
		User 1	**User 2**	**User 3**	**User 4**	**User 5**	**User 6**	**User 7**	**User 8**	**User 9**	**User 10**
1	The contextual menu (showing the available adaptations) is useful	9	7	7	9	9	8	8	7	10	10
2	The contextual menu (showing the available adaptations) is easy to be used	9	8	8	10	9	7	9	9	10	9
3	Available adaptations are adequate and sufficient	8	7	9	9	8	8	9	8	9	9
4	During the evaluation period the system automatically performed the adaptations I needed	7	9	9	10	8	8	9	8	9	8
5	I prefer to explicitly choose adaptations on my own instead of automatic ones	10	3	3	1	4	2	2	2	4	4
6	Automatically performed adaptations have always improved the legibility of the Web page I was navigating	7	8	9	10	8	8	8	9	9	9
7	Automatically performed adaptations have often decrease the level of legibility of the Web page I was navigating	5	3	3	2	5	4	4	4	2	3
8	Additional waiting time to load a new Web page is not acceptable	7	6	6	5	7	5	7	6	5	4
9	I will continue to use ExTras after this evaluation period	7	7	8	9	6	2	8	7	8	8

Table 3. Phase 2 Questionnaire results

Only User 1 has claimed that he prefers to explicitly choose the adaptions, instead of exploiting the automatic adaptation system.

"*I really would like to control what happens. I like the idea of changing textual characteristics, but I prefer to choose on myself when and what*."

The same user has indicated that he will continue to use ExTraS, by disabling the automatic adaptations.

Only one user claimed she will not continue to use ExTraS (User 6), while all the other ones expressed their willing in doing it. In particular, she is the user with the lowest average time spent browsing the Web, she is not particularly skilled (she was the only one with just an email account and no mobile devices) and she felt awkward in using what she perceived as an additional tool.

Additional computational time has been indicated as an important issue all the users have to face with, hence we have added the option described in section 4, with the aim of letting users enable or disable the application of the automatic adaptions to any Web page or to just choose whenever applying it or not (page by page). In particular, expert users are more subject to additional computational times than the non-expert users.

		Expert Group	**Non-expert Group**	**Average Values**
1	The contextual menu (showing the available adaptations) is useful	8.2	8.6	8.4
2	The contextual menu (showing the available adaptations) is easy to be used	8.8	8.8	8.8
3	Available adaptations are adequate and sufficient	8.2	8.6	8.4
4	During the evaluation period the system automatically performed the adaptations I needed	8.6	8.4	8.5
5	I prefer to explicitly choose adaptations on my own instead of automatic ones	4.2	2.8	3.5
6	Automatically performed adaptations have always improved the legibility of the Web page I was navigating	8.4	8.6	8.5
7	Automatically performed adaptations have often decrease the level of legibility of the Web page I was navigating	3.6	3.4	3.5
8	Additional waiting time to load a new Web page is not acceptable	6.2	5.4	5.8
9	I will continue to use ExTras after this evaluation period	7.4	6.6	7

Table 4. Phase 2 Questionnaire results Average values the two groups of users

By comparing the Phase 1 and Phase 2 results, we can observe that the group of non-expert users generally have more difficulties in accomplishing the assigned tasks (in terms of time spent to reach the goal of the tasks and of assigned scores), but they enjoyed the ExTraS system as well as the expert ones, scoring its usefulness, its automatically performed adaptations and the resulting legibility of Web contents with higher values than the expert group.

Thanks to log files, we have evaluated the performances of ExTraS, during the evaluation periods, in terms of manual adaptations (explicitly chosen by the users thanks to the contextual menu), wrong automatic adaptations and correct automatic adaptations. Figure 9 shows such values as average daily adaptations grouped by weeks for User 5, the user who mostly exploited ExTraS and with the highest average time spent on the Internet. It is worth noting that during the first weeks the system requests many manual adaptations and commits more errors in terms of wrong automatic adaptations, while, after such an initial period, these values tend to zero. On the contrary, the number of correct automatic adaptations increases during the evaluations period. Obviously, the more the user exploits ExTraS and the more ExTraS learns and improves correct automatic adaptation.

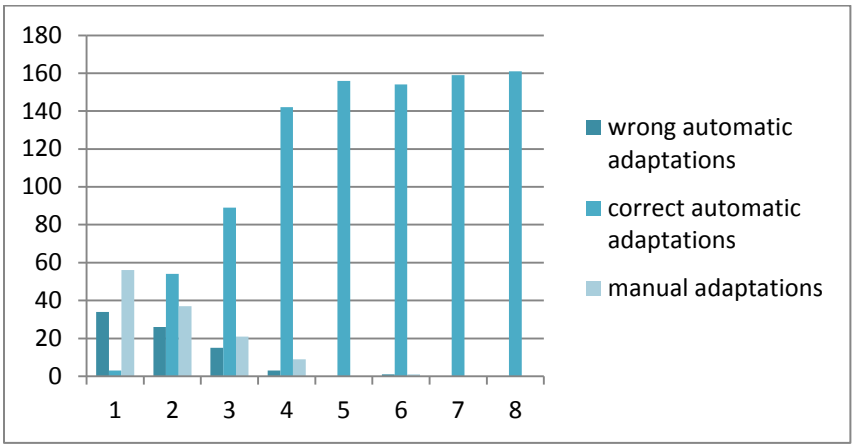

Figure 9 – User 1 test period

6. Conclusions

This paper discussed on the design choices, the implementation and evaluation of a Web page personalization system, based on reinforcement learning. Personalization is realized based on users' preferences and needs. The main aim is to improve Web pages legibility; this is accomplished by adapting those textual elements that represent a barrier to users reading. The approach exploits the Q-learning algorithm to understand which might be the best adaptations to perform on a Web page, tailored for each specific user.

The system has been evaluated through simulation and real tests. Simulation was employed to test different users' behaviors, with different accessibility characteristics. By tracking the adaptations requested during an initialization phase, the system is able to understand the preferences of the users and to automatically propose adaptations.

Real tests were devoted to understand if the real prototype we built can be of help for elderly users. We asked a set of elderly users (divided in two groups, according to their technological skills) to employ for a given period a Web browser equipped with our plugin. Then we asked them to rank it. Results are encouraging and confirm that the proposed methodology to dynamically adapt and customize contents is viable and effective. Therefore, we are able to conclude that a reinforcement learning approach to Web personalization is a strategy that could be adopted, especially during the design of software devoted to help people with specific needs.

Further studies would be conducted by implementing different reinforcement learning algorithms, so as to evaluate the most effective one.

Acknowledgement

The authors wish to thank Alberto Fariselli and Michele Lambertini for their precious support.

References

1. S. Ferretti, S. Mirri, C. Prandi, and P. Salomoni, "Exploiting Reinforcement Learning to Profile Users and Personalize Web Pages", in the Proceedings of the 1st IEEE International Workshop on User Centered Design and Adaptive Systems (UCDAS 2014) - The 38th Annual International Computers, Software & Applications Conference (COMPSAC 2014), Vasteras (Sweden), 21st July 2014. IEEE Communications Society.
2. K. Wecel, T. Kaczmarek, and A. Filipowska, “Scalable adaptation of Web applications to users’ behavior,” Proc. 4th International Conference on Computational Collective Intelligence: technologies and applications (ICCCI'12), Springer-Verlag Berlin, Heidelberg, Nov. 2012, pp. 79-88.
3. D. Wolff, M. Schaaf, S. Grivas, and U. Leimstoll, “Context-aware website personalization,” in Knowlege-Based and Intelligent Information and Engineering Systems, ser. Lecture Notes in Computer Science.Springer Berlin Heidelberg, vol. 6882, 2011, pp. 51–62.
4. D. Beymer, D. M. Russell, and P. Z. Orton, “An eye tracking study of how font size, font type, and pictures influence online reading,” Proc. 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction (BCS-HCI’08), September 2008, pp. 15−18.
5. G. Legge, “Psychophysics of reading,” Lawrence Erlbaum Associates, Mahwah, New Jersey, 2006.
6. M.A. Tinker, “The legibility of print,” Iowa State University Press, 1963.
7. M. Bernard, C.H. Liao, and M. Mills, “The effects of font type and size on the legibility and reading time of online text by older adults,” Proc. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI2001), ACM Press, March-April 2001, pp. 175-176.
8. L. Rello, G. Kanvinde, and R. Baeza-Yates, “Layout guidelines for web text and a web Service to improve accessibility for dyslexics,” Proc. 9th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’12), ACM Press, April 2012.
9. E. Russell-Minda, J.W. Jutai, J. Graham Strong, Kent A. Campbell, Deborah Gold, Lisa Pretty, and Lesley Wilmot, “The legibility of typefaces for readers with low vision: a research review,” Journal of Visual Impairment & Blindness, Volume 101, Number 7, July 2007, pp. 402-415.
10. S. Mirri, C. Prandi, and P. Salomoni, “Experiential adaptation to provide user-centered web content personalization,” Proc. IARIA Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC2013), October 2013, pp. 31-36.
11. S. Ferretti, S. Mirri, C. Prandi, and P. Salomoni, “User centered and context dependent personalization through experiential transcoding,” Proc. IEEE Consumer Communications and Networking (CCNC 2014), Workshop on Networking Issues in Multimedia Entertainment (NIME'14), IEEE Computer Society, January 2014.
12. B. Schildbach and E. Rukzio, “Investigating selection and reading performance on a mobile phone while walking,” Proc. 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI’10), ACM Press, September 2010, pp. 93-102.
13. P. Biswas and P. Langdon, “Investigating the accessibility of program selection menus of a digital TV interface,” Proc. International Conference on Human Computer Interaction (HCII2011), July 2011, pp. 425-434.

14. M. Pous, C. Serra-Vallmitjana, R. Giménez, M. Torrent-Moreno, and D. Boix, “Enhancing accessibility: mobile to ATM case study,” Proc. IEEE Consumer Communications and Networking Conference (CCNC2012), IEEE Computer Society Press, January 2012, pp. 404-408.
15. S. L. Henry, “Developing text customisation functionality requirements of PDF reader and other user agents,” Proc. 13th International Conference on Computers Helping People with Special Needs (ICCHP 2012), Springer-Verlag, July 2012, pp. 602-609.
16. W. M. Watanabe, A. Candido, M. A. Amâncio, M. de Oliveira, T. A. S. Pardo, R. P. M. Fortes, and S. M. Aluísio, “Adapting web content for low-literacy readers by using lexical elaboration and named entities labeling,” Proc. 7th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’10), ACM Press, May 2010.
17. L. Rello, R. Baeza-Yates, H. Saggion, S. Bott, R. Carlini, C. Bayarri, A. Gorriz, G. Gupta, G. Kanvinde, and V. Topac “Dyswebxia 2.0! Accessible text for people with dyslexia,” Proc. 10th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’13), ACM Press, May 2013.
18. K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, “Automatically generating personalized user interfaces with supple,” in Artificial Intelligence, vol. 174, no. 1213, pp. 910 – 950, 2010.
19. G. R. Santhanam, S. Basu, and V. Honavar, “Preference based service adaptation using service substitution,” Proc. IEEE Web Intelligence (WI 2013), IEEE Computer Society, November 2013, pp. 487-493.
20. K. Koidl, O. Conlan, and V. Wade, “Towards cross site personalisation,” Proc. IEEE Web Intelligence (WI 2013), IEEE Computer Society, November 2013, pp. 542-548.
21. O. Bursa, Ö. Can, and M. O. Ünalir, “User Profiling for Policy Management in Social Communities,” Proc. IEEE Signature Conference on Computers, Software, and Applications (COMPSAC 2012), IEEE Computer Society, July 2012, pp. 348-349.
22. S. Loeb, “Fluid user models for personalized mobile apps,” Proc. IEEE Consumer Communications and Networking Conference (CCNC 2013), IEEE Computer Society, January 2013, pp. 705-708.
23. A. Brown, C. Jay, and S. Harper, “Audio access to calendars,” Proc. 7th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’10), ACM Press, April 2010.
24. Y. Yesilada, S. Harper, and S. Eraslan, “Experiential transcoding: an eye tracking approach,” Proc. 10th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’13), ACM Press, May 2013.
25. P. Swapna Raj and B. Ravindran, “Personalized web-page rendering system,” Proc. International Conference on Management of Data (COMAD 2008), December 2008, pp. 30-39.
26. Y. Ren, G. Li, and W. Zhou, “Learning user preference patterns for Top-N recommendations,” Proc. IEEE Web Intelligence (WI 2012), IEEE Computer Society, December 2012, pp. 137-144.
27. R. Junges and F. Klügl, “Programming agent behavior by learning in simulation models,” in Applied Artificial Intelligence, Vol. 26, no. 4, pp. 349-375, 2012.
28. E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," in The Journal of Machine Learning Research Vol. 5, pp. 1-25, 2004.
29. C.J.C.H. Watkins and P. Dayan, “Q-learning”, Machine Learning, Vol. 8, no. 3, pp. 279-292, 1992.
30. D.P. Bertsekas and J.N. Tsitsiklis, “Neuro-Dynamic Programming,” Athena Scientific, Belmont, MA, 1996.
31. V.S. Borkar and S.P. Meyn, “The O.D.E method for convergence of stochastic approximation and reinforcement learning,” Siam J. Control, Vol. 38 (2), pp. 447–69, 2000.
32. Atsushi Wada, Keiki Takadama, Katsunori Shimohara, Osamu Katai, “Learning Classifier System with Convergence and Generalization,” in Foundations of Learning Classifier Systems (series “Studies in Fuzziness and Soft Computing”), Vol. 183, pp. 285-304, 2005.
33. R.S. Sutton and A.G. Barto, “Reinforcement learning: An introduction,” Vol. 1. No. 1. Cambridge: MIT press, 1998.

34. S. Manju and M. Punithavalli, "An analysis of Q-learning algorithms with strategies of reward function," in International Journal on Computer Science and Engineering, Vol. 3.2, pp. 814-820, 2011.
35. A.L. Thomaz, and C. Breazeal, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in AAAI. Vol. 6. 2006.
36. A. Garlapati, A. Raghunathan, V. Nagarajan, and B. Ravindran, “A Reinforcement Learning Approach to Online Learning of Decision Trees,” Proc. 14th European Workshop on Reinforcement Learning (EWRL 2015), July 2015, pp. 1-9.
37. S. Ferretti, M. Roccetti, P. Salomoni, and S. Mirri, “Custom e−learning experiences: working with profiles for multiple content sources access and adaptation,” Journal of Access Services, Taylor & Francis, vol. 6, issues 1-2, February 2009, pp.174-192.
38. P. Salomoni, S. Mirri, S. Ferretti, and M. Roccetti, “Profiling learners with special needs for custom e-learning experiences, a closed case?,” Proc. 4th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A’07), ACM Press, May 2007, pp. 84-92.
39. WURFL. Wireless Universal Resource File Library. Available from: http://wurfl.sourceforge.net, [last retrieved: September 2015].
40. C.J.C.H. Watkins, “Learning from Delayed Rewards,” Ph.D. thesis, Cambridge University, 1989.
41. C. Asakawa and H. Takagi, “Transcoding,” in Simon Harper and Yeliz Yesilada, editors, Web Accessibility: A Foundation for Research, Human-Computer Interaction Series, Springer, 2008, pp. 231-260.
42. Mozilla Firefox Add-on Builder and SDK, 2014. Available from: https://addons.mozilla.org/en-US/developers/ [last retrieved: September 2015].
43. Alexa Internet, Alexa Top 500 Global Sites. Alexa Internet. http://www.alexa.com/topsites [last retrieved: September 2015].
44. Piqle, “Platform for implementing Q-Learning experiments”, http://piqle.sourceforge.net/, [last retrieved: September 2015].
45. J. Nielsen and T.K. Landauer, “A mathematical model of the finding of usability problems,” in Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI’93), ACM Press, New York, NY, USA, pp. 206-213.