Increasing the speed of Information Access on the Mobile web using HTML feature extraction

Andreas Komninos
Glasgow Caledonian University
Cowcaddens Road
Glasgow G4 0BA, UK
+44 141 3313095
andreas.komninos@gcal.ac.uk

Chris Milligan
Glasgow Caledonian University
Cowcaddens Road
Glasgow G4 0BA, UK
+44 141 3313095
cmilli10@caledonian.ac.uk

ABSTRACT
Motivated by the cumbersome process of extracting information from webpages as rendered on mobile device web browsers, this paper focuses on describing an alternative and promising approach to facilitating the process for users. We present early work-in-progress on a system that attempts to extract information sections of a webpage and presents the extracted sections first, with the remaining page following. Early trials of a rudimentary prototype show promising results and we discuss further work to be carried out for the improvement of the system.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval – *Information Filtering, Retrieval Models*

H.5.2 [**Information Systems**]: Information Interfaces and Presentation - *User Interfaces*

H.5.4 [**Information Systems**]: Hypertext/Hypermedia - *Navigation*

General Terms
Algorithms, Design, Experimentation, Human Factors

Keywords
Mobile Information Access

1. INTRODUCTION
Browsing the web on mobile devices is well known as a problematic process, mostly from a usability point of view. Because of the general layout of normal web pages, which are designed for viewing on desktop computers, the inevitable vertical and horizontal scrolling required to access information contained therein, when using a mobile device browser, poses a serious impediment to the process of mobile information access. Due to the nature of mobile devices and their mode of use, which is drastically different to that of desktop computers, it is clear that alternative approaches are required to rendering webpage information on mobile device screens. The following section critically discusses work already undertaken in the field in a brief manner. The description of related work is non-exhaustive but indicative of the state of the art in this area. A description of our own approach is discussed thereafter, followed by recommendations on future work that we are planning on our early prototype.

2. RELATED WORK
We mentioned earlier the difference in the mode of use of mobile devices, compared to desktop computers. Typical usage patterns for mobile devices show that these are used for very short ("burst") periods of time during the day, when the user requires immediate access to some information or function of the device, followed by large periods of inactivity. The "always-on-standby" model for managing power on mobile devices such as PDAs or smartphones is a good example of functionality derived from the requirement arising from these usage patterns. Fujmoto [1] uses the term "nagara mobilism" (nagara = "while doing something else") to explain that this pattern of usage is central to the behaviour and adoption of devices by young users. This mode of use highlights the need for quick access to information that is relevant to the user's tasks, something that is currently not well supported in mobile web browsing.

To solve the problem of webpage rendering on mobile devices, two approach categories can be identified within existing literature and commercial systems: Client-based and Server-based processing of HTML documents. The first approach delegates the task of processing and rendering HTML documents in a more appropriate form to the mobile device itself. Documents are processed after having been downloaded in a variety of methods, most often in an attempt to eradicate horizontal scrolling, which imposes the largest interaction cost and impediment to mobile information access. Yin et al [2] devised the method of taking a normal webpage and rearranging the HTML in a way that eliminates horizontal scrolling. Their system examines the semantic relationships between HTML elements and sections to intelligently decide the order in which they will be "stacked" on top of each other. Other papers have also incorporated this method into their projects such as Liu et al[3], Dontcheva et al [4] etc. SmartView [5] divides the webpage into logical sections which can then be viewed independently of the rest of the document, although this requires explicit user instruction. This system benefits the users of PDAs whose touch-sensitive screens are easy to navigate, but usability problems would probably arise during use on a mobile device where the only navigation mechanism is the joypad. The Access NetFront browser [6] is a browser which implements this style of display with no horizontal scrolling, using a technology called SmartFit. SmartFit uses a process of restructuring the node arrangement so that there are no

nodes side by side. This is used to position nodes with a single breadth, one on top of the other. Another option for this browser is a process named JustFit, which allows for the page to be "squeezed" so that layout is very narrow, but all the sections are viewable without horizontal scrolling. As the nodes are squeezed, they appear long and narrow and are generally hard to read. Other commercial browsers like Opera [7] and Thunderhawk [8] address the same problem by "stacking" and providing a zoomable overview of sections respectively. A combination of both technologies is appearing in Microsoft's latest DeepFish browser [9].

With mobile device processing power increasing, the requirement on resources for client-side adaptation is not as taxing. Another advantage of client-side adaptation is that the client knows its own properties, thus being able to guide the adaptation process more effectively than a server-side system which has to rely on standardized profiles. A client-side adaptation system can also more easily forward decision-making to the user when the right decision is not obvious. For example, it can ask the user whether he wants a shortened, high-usability or a full, low-usability version of the content. This information brings to light the fact that device properties may have an influence on the behavior of the adaptation system.

On the other hand, server-side adaptation, where HTML documents are processed by a proxy before being served to the device, has not been studied to the same extent. The process occurs entirely on the server end, which means that the server has to estimate the characteristics of each mobile device in order to give an accurate transformation of web content. Of course this is not always possible so the content is adapted for a stereotypical mobile device. An advantage of server-side adaptation however, is that (especially with the removal of irrelevant content) the amount of data that a device would have to download and store would be slightly less, thus reducing download times and system overhead. Finally, server-side conversions may result in wasted (expensive) bandwidth if the adaptation is not desirable or restructures a page in such as manner that it is rendered unreadable by the user. Thusfar, server-side adaptation is provided by a very small number of proxies. Google have their own technology for adapting pages for mobile devices by segmenting and presenting a single page as multiple pages. Their system will also attempt to take the user to the "section" sub page which is most likely to contain text relevant to the query. A server-side webpage conversion service is offered by Skweezer.net [10] using the Ask.com web search engine.

All the approaches mentioned above try to address the problem of rendering standard web sites on small screen devices. While the approaches are more or less successful, they contribute little to the problem of facilitating access to relevant pieces of information. In eradicating the problem of horizontal scrolling, they aggravate the amount of vertical scrolling that is required to navigate the page. The Google approach seems to be the only one trying to intelligently aid the user, however, because of its default behaviour that strips each page of all non-textual elements and the non-existent control that the user has over the process, it has been heavily criticized with some users and content developers treating it as a form of "censorship".

3. METHODOLOGY

Our system is based on the simple assumption that when looking for relevant information on a webpage, a user will most likely prefer to have immediate access to those sections of the webpage that are likely to contain the information required. We therefore hypothesized that if, through query analysis, we could display those sections before the entire webpage, users would be able to obtain the information required much more quickly and thus reduce the need for horizontal scrolling.

Work by Kamvar and Baluja [11] on searches conducted using Google through mobile devices highlights the type of query most likely to be sent as predominantly belonging to one of the following categories: Local services, Travel & Recreation, Technology and Entertainment. The existence of "technology" searches can probably be attributed to the early adopters of mobile web technology being generally interested in technology. However the other categories, especially local services, hint towards the type and, importantly, size of information required: Addresses, types of business, short reviews or directions. Such information is typically collated with several unwanted elements (e.g. websites will list all Italian restaurants in a particular city). It is clear that users require "snippets" of information that would allow them to carry out a very specific task, not the entirety of the website information.

Based on these observations and our assumption as stated above, we began building a prototype system that operates on the principles of

a) Identifying the logical sections of the webpage that contain information relevant to the query, as input by the user;

b) Reconstructing the webpage presented to the user in such a manner that these sections are presented first, stacked on top of each other, followed by the remaining web-page which is otherwise not manipulated.

To accomplish the above, our system uses currently a naïve approach that assumes logical sections on a webpage correspond to physical sections marked by , <td> and <p> HTML tags. The sections are again very naively weighted for relevance to the original query by determining the Term Frequency of each query keyword in those relevant sections. We limited the system to display only the top 5 sections in terms of their calculated weight, in an attempt to limit the (potentially) lengthy additions to the top of the resulting viewable document.

Futhermore, each section text is "wrapped" around some custom HTML to ensure that it is clearly presented as an extract of the original website and not part of its original structure. For this purpose the extracted sections are presented as part of a table with double border and gray background.

It is important to note here that we chose not to strip other HTML elements from the sections marked by our designated tags. For example, text following a <p> tag, which can contain other tags, such as images, formatting or link tags, is kept "as-is". The reason for that is that we do not want to extract from the context of the retrieved content (for example, an image might be part of the required information of a link might be of benefit to the user for following more information). Sample screenshots of our system can be seen in Figure 1 below.

Figure 1. The left screenshot shows the query interface (up to 3 keywords are allowed). On the right, the original page with sections added on top (gray boxes)

We proceeded in carrying out preliminary tests of this rudimentary prototype, although it is clear that it is quite far from being perfect. We will discuss our planned improvements in the following sections; for now we will focus on the findings of our initial investigation, which aimed to assess the users' perceptions of the utility of such a system and whether it would be possible, even with a naïve approach, to obtain results that might be encouraging towards further development.

4. INITIAL EXPERIMENTATION

We asked 4 subjects to try out our early prototype by asking them to find out certain pieces of information by browsing the mobile web. Three of the subjects were computing students, with one being a computing novice. One of the volunteers owns a PDA and has used it to browse the web, and another volunteer uses PDAs quite frequently (although they don't own one).

4.1 Search Comparison Test

The first test to be carried out was in order to assess the ease of information retrieval from the prototype. The test setup involved an initial browse of a web page on a desktop computer to pick a random piece of information. The testing participant was then told to find this piece of information using three keywords relating to the information, and to signal once found. We asked for 3 keyword queries to replicate the fact that the average query length for the top 5 search categories, as described in [11]. Timing started from when the user clicked on the Google search results link for the particular page. When the piece of text had been found, the timer was stopped and recorded. This process was achieved using Pocket Internet Explorer with Google.com, and then using the prototype browser. Each participant repeated the task three times with different items of text from different web pages to search. The following figure (figure 2) shows the average access time, which appears to be roughly 10% less when using our prototype.

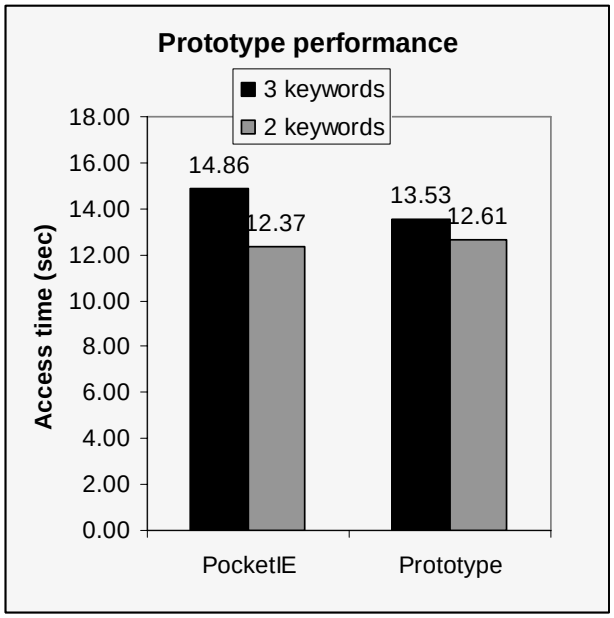

Figure 2. Performance of prototype vs. PocketIE

We repeated the experiment instructing the users to formulate 2 queries this time (again from [11] this figure is the average query length for all categories), using the same procedure and similar tasks. In this instance we found the performance to be comparable with PocketIE being marginally (~0.3 sec) better than our prototype.

Finally we asked the users to complete a post-experiment questionnaire, in which we asked them to rate their perceptions of the system on a scale of 10 (a larger score is more positive). Overall the prototype received average feedback on the look of the reconstructed page (5.4/10) and an average perceived effectiveness score of 5.0/10. While these results are perhaps not exactly splendid, we must consider them as a sign of indifference to the effect of the overall presentation of the viewed document, which in itself, is a positive finding as the users don't seem to mind the addition of extra elements on the document. The perceived effectiveness score is also rather average, but again we consider this to be an encouraging result given the naivety of the adopted approach for this early prototype.

Another section of the post-experiment questionnaire allowed subjects to leave general comments of their opinion of the prototype. This was probably the most encouraging section with all users commenting positively on the intuitiveness of the browser and its speed of use. Three users also commented positively on the simplicity of the design. Finally two users commented on the effectiveness of the section extraction methods as an area they would want to see improved.

5. FUTURE WORK

When testing the prototype we anticipated that the initial reaction to it would not have been enthusiastic, given the simplicity of our approach. However, based on the test subject comments, we were encouraged by the fact that it was immediately obvious to them that our approach would be a helpful aid in browsing the mobile web, if it could be perfected further. The users readily identified

the problem of horizontal scrolling and the lack of speed in finding information on the mobile web. Furthermore, we were encouraged by the findings that seem to indicate that our system, even in its simplest form, is not only non-disruptive to the users, but can also outperform the standard browser for a large proportion of mobile devices.

Although our initial early trial is based on a small sample and cannot be presented as conclusive in any manner, based on its findings, we are ready to conduct much further work on the prototype. It is essential for us to improve the performance of the section identification algorithm, something that we anticipate to be a great challenge. Further from overcoming the problem of parsing loosely structured HTML documents that do not necessarily conform to development standards, the identification of what constitutes a "logical" section within a document will be a hard challenge. Context cues such as colours, blank space dividers, fonts or images, allow the human brain to immediately identify and separate content into logical sections such as the webpage designers meant it to be seen. While difficult, this process is not impossible as past research shows and semantic analysis in conjunction with DOM analysis of the webpage, filtered through layout heuristics, could greatly help.

More importantly, it is important for us to employ intelligent selection mechanisms for choosing the sections that will be displayed on top of the page. TF/IDF instead of simple TF is an obvious candidate for improvement, however we feel that we should be looking at a combination of various weighting heuristics to obtain a more accurate results. More specifically, a Bayesian probability model could possibly increase the selection process accuracy while maintaining the number of sections displayed at a minimum. Further to this, it would be extremely interesting to apply a Markov chain model trained on implicit relevance feedback indicators and user behaviour modeling, to try and accurately assess the order that sections should be presented to the user.

Having mentioned user behaviour modeling, we should also mention here our intention to augment the system by exploiting user models built and trained over time to perform tasks such as query disambiguation and expansion. We have previously explored such methods with good success in the past [12] and we feel that they would be highly appropriate here, given the low probability of long queries sent through a mobile browser. The process of query expansion and disambiguation can help locate sections in a given document that would have otherwise been given a low weighting if, for example, synonyms or closely related terms omitted from the query can be found in a text section.

Once a more advanced prototype has been completed, we would be interested in comparing its performance with a variety of existing systems and with a range of different algorithmic functionality options to determine conclusively what method works best.

6. CONCLUSIONS

In [5], the authors describe how a "section extraction" system could work in conjunction with a search engine by marking each section with a number of small boxes to indicate its potential relevance to a query. This might work quite well for a PDA but considering the navigation mechanisms of a typical phone (joypad), we believe that this will not necessarily result in any significant improvement as the interaction cost of navigating to the relevant section will probably remain just as high.

We are confident that our idea of intelligently "stacking" extracted sections can aid users to navigate the mobile web, thus helping towards the solution to a usability problem that has hindered the use of data services on mobile devices so far.

7. REFERENCES

[1] Fujimoto, K. "The Anti-Ubiquitous "Territory Machine"-- The Third Period Paradigm: From "Girls' Pager Revolution" to "Mobile Aesthetics", in Personal, Portable, Pedestrian: Mobile Phones in Japanese Life, edited by M. Ito, D.Okabe, and M. Matsuda. Cambridge: MIT Press, 2005.

[2] Yin, X. & Lee, W.S. "Using link analysis to improve layout on mobile devices", Proceedings of the ACM 13th international conference on the World Wide Web, New York, 2004

[3] Liu, Z., Ng, W.K., Lim, E.P., & Li, F., "Towards building logical views of websites". Data & Knowledge Engineering Volume 49, Issue 2, 2004

[4] Dontcheva, M., Drucker, S. M., Wade, G., Salesin, D., Cohen, M. F., "Summarizing Personal Web Browsing Sessions", ACM 19th annual Symposium on User Interface Software Technology (UIST), Montreux, 2006

[5] Milic-Frayling, N., Sommerer, R., "SmartView: Flexible Viewing of Web Page Contents". Proceedings of the ACM 11th World Wide Web Conference, Hawaii, 2002.

[6] Access NetFront: http://www.access-netfront.com

[7] Opera mobile browser: http://www.opera.com

[8] Thunderhawk browser: http://www.bitstream.com/wireless

[9] Microsoft DeepFish browser: http://labs.live.com/deepfish

[10] Skweezer mobile webside adaptation: http://www.skweezer.net

[11] Kamvar, M., Baluja, S., "A large scale scale study of wireless search behavior: Google Mobile Search", in Proceedings of the 24th ACM Conference on Human Factors in Computer Systems (CHI2006), Montreal, 2006.

[12] Komninos, A., Dunlop, M.D, "A calendar based Internet content pre-caching agent for small computing devices", Journal of Personal and Ubiquitous Computing (online First), Springer, 2007