# Web Clipping: Compression Heuristics for Displaying Text on a PDA

PEDRO GOMES, SÉRGIO TOSTÃO, DANIEL GONÇALVES, JOAQUIM JORGE

*Instituto Superior Técnico*
*Av. Rovisco Pais, 1049-001 Lisboa*

djvg@gia.ist.utl.pt , jorgej@acm.org
+ 351 917935280, + 351 937021798, +351 21841769

**Abstract.** While there is strong motivation to do so, reading Web pages on portable devices still leaves much to be desired. Most solutions need special versions of Web sites and do not cope adequately with resource limits of PDAs, in particular their small screen, which makes it difficult to display large amounts of information in a usable manner. We describe an approach that provides greater flexibility to users in selecting information they want displayed, while coping with display limitations by analyzing content and organizing it into abstract visualization levels. The user can zoom in and out successive levels of detail and navigate the content without being overwhelmed by clutter. We have developed heuristics for filtering text though task analysis and usability studies, whose results are described here. These studies provided meaningful insights to help us explore trade-offs between information filtering (and therefore text compression) and text comprehension.

*Keywords: Zoomable Interfaces, Web-Clipping, and Morphological Text Analysis*

## 1. Introduction

The use of mobile computing devices, such as *Personal Digital Assistants (PDAs)* is becoming widespread. These devices usually have a small screen and reduced storage and processing capacities, where the keyboard and mouse are replaced by a pen and direct manipulation of objects on-screen. Given the ever-increasing amount of information available on the World-Wide Web, there is an increased desire for reading web documents on a PDA. However, most web documents aren't designed to cope with the limitations of those devices. Hence, it is not usually possible to read them on portable devices without some kind of transformation. Some solutions have been tried to overcome those limitations. They usually require an alternate, trimmed-down, version of the documents to be prepared beforehand. Some of the most popular solutions, such as Web-Clipping, developed by Palm, Inc [1], or AvantGo (http://www.avantgo.com) do so. This is undesirable because it involves an increased effort in creating and maintaining alternate versions of a site, and because only prepared sites can be read. Also, it doesn't deal with the problem of having a small screen, where only a few lines of text can be shown at the same time. Long documents might become too cumbersome to read in such a fashion.

A questionnaire made to 30 Internet and PDA users about popular PDA Web-reading solutions confirms this. The most popular readers mentioned were AvantGo, iSilo and SmartDoc. Most persons mentioned that they allow "Web page off-line visualization, frequently updated", and are "Fast, Simple and Useful

(AvantGo)". However, they also complained that "AvantGo is not compatible with most of the web sites" there are problems of "operating system compatibility", and that they suffer from a "lack of useful information".
With this in mind we propose a solution to allow users to select pre-existing sites for visualization and access in a PDA, without changing the site contents. To achieve this and cope with the small screen, the system makes it possible for users to navigate the text using abstract levels of information, with a *zoomable interface* [2][3][4][5]. Also, users can customize the system by specifying which sections of a page they want to read on the PDA (thus getting rid of ads, navigation bars, and other content-poor items). Our system also has in common with existing solutions the fast clipping phase and simplicity of use. In what follows we concentrate on the user interface and heuristics to make it possible to display longer texts on PDA screens without sacrificing text comprehension. We present preliminary results on the performance of several heuristics for text compression and their impact on text comprehension.

# 2. Architectural Framework

The figure below illustrates our architectural framework, which is divided in two main components: (a) retrieval and conceptual analysis of the information inside the web page (**Clipping System**) and (b) visual manipulation on a PDA (**Visualization System**).

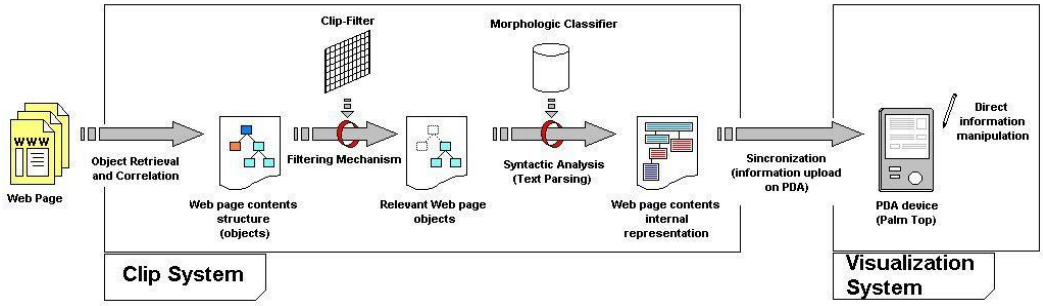


**Figure 1- Architectural framework of the web clipping application proposed**

The **Clipping System** involves a group of processes that require large memory and processing capacities. Therefore, it will be run on the user's PC and its performance is dependent of the capacities and limitations of that machine. The **Visualization System** requires few resources and will run on the PDA. In the following sections we'll discuss each component in greater detail.

## 2.1 Retrieval and Correlation of Objects: Register Tree

This component identifies objects within the selected web page (headers, paragraphs, links, images...) and finds hierarchical relations between them to assist

the filtering process making it easier to both retrieve and discard correlated information, by selecting or eliminating certain kinds of objects such as images.

## 2.2. Clipping-Filter

The **clipping-filter** component allows users to specify what part of the contents of a web page that really matter to them, using constraints. This mechanism can reduce the amount of information processed and displayed on the PDA, eliminating content that users do not want to see. This prevents waste of processing resources in the Clipping and Visualization Systems (and precious screen real estate on the PDA). There are default rules to deal with advertising, navigation etc, and also to allow the user to select specific frames or table cells from pages. Also, we implemented other filtering criteria based on user studies:

- Selecting text blocks based on given keywords.
- Omit / retain links and images (specifying the desired dimension).
- Selecting Text based on font size and type (good for web pages that always present the same kind of contents in the same format).

We provide the user with a flexible tool that can be applied to different situations or web pages in order to obtain results as reliably as possible. Rule sets can be defined globally, or on a per-site basis, allowing tailorable behavior, without requiring extensive customization for new sites. A good example of use of this system is to read news pages, the activity preferred by most users (83%, according to the analysis of the first questionnaire).

## 2.3 Text Parsing

Text parsing uses an application named SMORPH [6] to perform text segmentation and morphologic analysis. This allows the system to classify words according to their morphologic classes (names, verbs...). This information determines on what visualization levels the word should be present (see **Defining Abstract Visualization Levels**). This process provides information about the page's text elements that will be used in the Visualization System, to define a set of abstract levels of detail for the text. Choosing an abstract level and then zooming to more detailed levels will allow navigating larger amounts of text than usually fit on the PDA's reduced screen. This visual manipulation (see **Visual Manipulation of Information)** depends of the correct classification of words made by the morphologic classifier. Once the text is analyzed, the system inserts extra markings, to present in a linear way, the entire object filtered down from the original web page(s). We've defined a simple Meta-Language, where a set of tags defines the text zone corresponding to each abstract level.

For example, consider the following sentence: "*Peter likes Maria very much but he is ashamed of telling her"*. After running the text through a morphologic analyzer using SMORPH, the program will produce something like this:

<1> Peter likes </1> <2> very much </2> <1> Maria </1> <3> but he </3> <1> is ashamed </1> <3> of </3> <1> telling </1> <2> her </2>

The user can then visualize the sentence at three levels of detail:

1: "Peter likes Maria is ashamed telling" (names and verbs)

2: "Peter likes Maria very much he is ashamed telling her" (L1+adjs.+ pronouns)

3: "Peter likes Maria very much but he is ashamed of telling her" (original)

## 2.4 Heuristics and Abbreviations

Another way to reduce the text is to identify special words whose abbreviations are commonly known and replace them by it. Some of the words that can be abbreviated in this fashion [6] are listed below:

| | | | | | |
|---|---|---|---|---|---|
| abbr. | abbreviation | esp. | especially | n. | noun |
| Brit. | British | illus. | illustration | pred. | predicative |
| compar. | comparative | masc. | masculine | possess. | possessive |

We have compiled these and other abbreviations derived from observing users composing SMS messages on cell-phones into a dictionary. It is also possible to apply a set of heuristics to shorten long words. Three such heuristics are:

- **adv_Heuristic**: Adverbs (commonly ended by "mente", in Portuguese, and "ly" in English) can be replaced by an abbreviated form ending with "m/". Examples: considerab/ (considerably), advisab/ (advisably)
- **qu_Heuristic**: Removes the vowels "u" and "e" after "q" turning "que" (a very common word similar to "that" or "which") into "q" (they are spelled almost alike)
- **vowels_Heuristic**: Removes all the internal vowels of the word, except if they are before and/or after any other vowel, or if they are the last vowels of the word. Examples: cmmnly (commonly), cmprssion (compression). This heuristic is language independent.

Our work is currently focused on the Portuguese language. However, we plan to develop our approach in a modular fashion such that new functionality can be added in a simple manner, e.g. adding more entries to the Abbreviation List or replacing heuristics in order to support new languages.

## 2.5 Defining Abstract Visualization Levels

Each level of detail is defined by the classes of words (in morphological sense) that appear in the text showed by the **Visualization System**, and by the use (or not) of the heuristics. We performed a questionnaire to identify the level of comprehension in texts synthesized in the several levels. The order in which questions and levels of detail were presented varied from person to person, to prevent bias due to fatigue or a priori knowledge of the texts. The results of that questionnaire are summarized on the following figure:

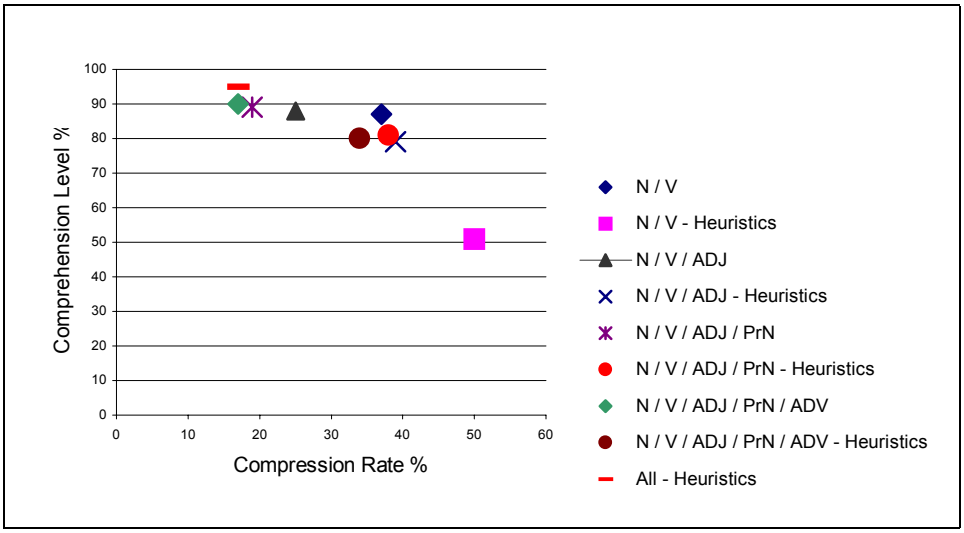


**Figure 2- Comprehension of different detail levels**

The levels shown on the figure are labeled with the morphological classes of words included in each level and by the use of abbreviations. For instance, in the level labeled as "`N/V/ADJ-Heuristics`", the text includes only nouns, verbs and adjectives and abbreviation heuristics are used.

It's easy to see that the comprehension level increases with the number of morphologic classes retained in the final text and if heuristics are not applied. The compression rate increases when the number of morphologic classes decreases or when heuristics are applied. These conclusions helped us design the following sequence of visualization levels of detail, looking at both comprehension level and compression rate:

| | |
|---|---|
| Level 0 | Title only (complete or incomplete depending of the number of words) |
| Level 1 | Title complete + First ***N*** words of the text block (First_words) |
| Level 2 | Level 1 + Names + Verbs + Abbreviation List + Heuristics – First_words |
| Level 3 | Level 2 + Adjectives + Adverbs |
| Level 4 | Level 3 + Classes remaining (pronouns, articles...) |
| Level 5 | Level 4 – AbreviationList – Heuristics + vowels_Heuristic |

| | |
|---|---|
| Level 6 | Level 5 – vowels_Heuristic |
| Level 7 | Original Text |

## 2.6 Visual Manipulation of Information: Visualization System

The user will be able to read the results of the **Clipping System** on the PDA with the help of the Visualization System. This system will behave like a zoomable interface. The user will be able to choose and change the detail level seen in each text block (not necessarily the whole text at once), zooming in or out through direct manipulation of the text. The visualization system is also responsible to implement the abbreviation heuristics, in order to reduce the size of the file loaded into the PDA (otherwise, extra tags would be necessary).

# 3. Conclusions

While the system we have described is still under development, the results so far are encouraging. Users appreciate being able to download any site to the PDA not just a special few. Also, tests and questionnaires show that it is possible to compress texts significantly (40% less characters) without major impact on reading comprehension. This approach seems therefore promising in handling the problems posed by small screens. We plan to explore the trade-offs involved (text compression vs. reading comprehension) further to see if these findings can be generalized for other languages and to assess the impact on reading speed.

## *References*


[1] Palm Web Clipping resources: http://www.palmos.com/dev/tech/webclipping/ resources.html

[2] Benderson, B.B. and Hollan, J.D.. "Pad++: A Zoomable Graphical Interface System", Demonstration, SIGCHI'95 Companion, 1995, 23-24

[3] Pad++: Zoomable User Interfaces, http://www.cs.umd.edu/hcil/pad++/

[4] Bederson, B., Meyer, J., Good, L., Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java, *In* ACM UIST 2000, pp.171-180, available on the web ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2000-13html/2000-13.pdf

[5] Stuart, P.: Context and Interaction in Zoomable User Interfaces, Published in the AVI 2000 Conference Proceedings, pp 227-231 317, 24-26 May, available on the web at http://citeseer.nj.nec.com/304079.html

[6] Ait-Molahtar, S. "L'analyse pré syntaxique en une seule etape". PhD Thesis, Université Blaise Pascal, GRIL, 1998

[7] Euralex 2000 Tutorial – Homepage, European Association for Lexicography, http://www.ims.uni-stuttgart.de/euralex/conferences/elx2000/tutorial/

[8] Edinburgh Language Technology Group (LTG) http://www.ltg.ed.ac.uk/index.html

[9] University Centre For Computer Corpus Research On Language homepage http://www.comp.lancs.ac.uk/ucrel/