Universal Mobile Information Retrieval

David Machado, Tiago Barbosa, Sebastião Pais, Bruno Martins, Gaël Dias

Centre of Human Language Technology and Bioinformatics, University of Beira Interior
6201-001, Covilhã, Portugal
{david, tiago, sebastiao, brunom, ddg}@hultig.di.ubi.pt

Abstract. The shift in human computer interaction from desktop computing to mobile interaction highly influences the needs for new designed interfaces. In this paper, we address the issue of searching for information on mobile devices, an area also known as Mobile Information Retrieval. In particular, we propose to summarize as much as possible the information retrieved by any search engine to allow universal access to information.

Keywords: Mobile Information Retrieval, Clustering of Web Page Results, Automatic Summarization.

1 Introduction and Related Work

The shift in human computer interaction from desktop computing to mobile interaction highly influences the needs for new designed interfaces. In this paper, we address the issue of searching for information on mobile devices, an area also known as Mobile Information Retrieval.

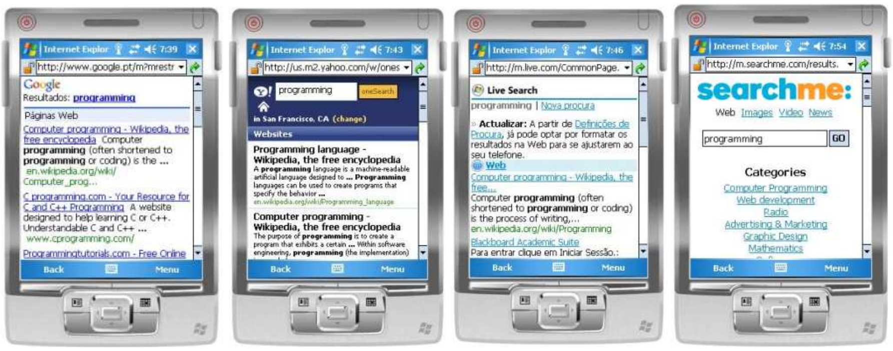
Within this scope, two issues must be specifically tackled: web search and web browsing. On the one hand, small size screens of handheld devices are a clear limitation to displaying long lists of relevant documents which induce repetitive scrolling. On the other hand, as most web pages are designed to be viewed on desktop displays, web browsing can interfere with users' comprehension as repetitive zooming and scrolling are necessary.

To overcome the limitations presented by current search engines to handle information on mobile devices, we propose a global solution to web search and web browsing based on clustering of web page results and web page summarization.

Most of the projects on mobile search deal with organizing the information to fit into small screens without benefiting from new trends in Information Retrieval presented in [1] and [2]. Indeed, projects such as Yahoo Mobile[1], Google Mobile[2] or Live Search Mobile[3] present information in a classic way by listing web page results as it is shown in Figure 1. In order to show as many results as possible on the screens of PDAs or smart phones, layout structures are usually redesigned to keep to their

[1] http://mobile.yahoo.com/yahoo
[2] http://www.google.com/mobile/
[3] http://www.livesearchmobile.com/

basics. In fact, commercial projects have mainly privileged services over location on mobile devices such as news, weather forecast or maps rather than providing new ways of searching for information, maybe to the exception of local search facilities.



Google
Resultados: programming
Páginas Web
Computer programming - Wikipedia, the free encyclopedia Computer programming (often shortened to programming or coding) is the ...
en.wikipedia.org/wiki/Computer_prog...
C programming.com - Your Resource for C and C++ Programming A website designed to help learning C or C++. Understandable C and C++ ...
www.cprogramming.com/
Programmingtutorials.com - Free Online

programming
in San Francisco, CA (change)
Websites
Programming language - Wikipedia, the free encyclopedia
A programming language is a machine-readable artificial language designed to ... Programming languages can be used to create programs that specify the behavior ...
en.wikipedia.org/wiki/Programming_language
Computer programming - Wikipedia, the free encyclopedia
The purpose of programming is to create a program that exhibits a certain ... Within software engineering, programming (the implementation)

Live Search
programming | Nova procura
Actualizar: A partir de Definições de Procura, já pode optar por formatar os resultados na Web para se ajustarem ao seu telefone.
Web
Computer programming - Wikipedia, the free...
Computer programming (often shortened to programming or coding) is the process of writing,...
en.wikipedia.org/wiki/Programming
Blackboard Academic Suite
Para entrar clique em Iniciar Sessão.:

searchme:
Web Images Video News
programming GO
Categories
Computer Programming
Web development
Radio
Advertising & Marketing
Graphic Design
Mathematics

(a) (b) (c) (d)

Fig. 1. (a) Google Mobile. (b) Yahoo Mobile. (c) Live Search Mobile. (d) Searchme Mobile.

Other projects have proposed different directions. In particular, Searchme Mobile[4] is certainly one of the first mobile search engine to categorize web page results as shown in Figure 1d. By doing so, it is clear that web search is made easier to the user. Indeed, the more the information is condensed into chunks of valuable information the more it is accessible to any user (paired or impaired) in any location (car, home, street, etc.). However, the solution implemented by Searchme is based on a set of pre-defined categories for each query term. As a consequence, the categorization can only be performed for well-known queries. In the case the category is not known, no search results are displayed. This solution is clearly unsatisfactory as one may want to query any term in any language over the all web.

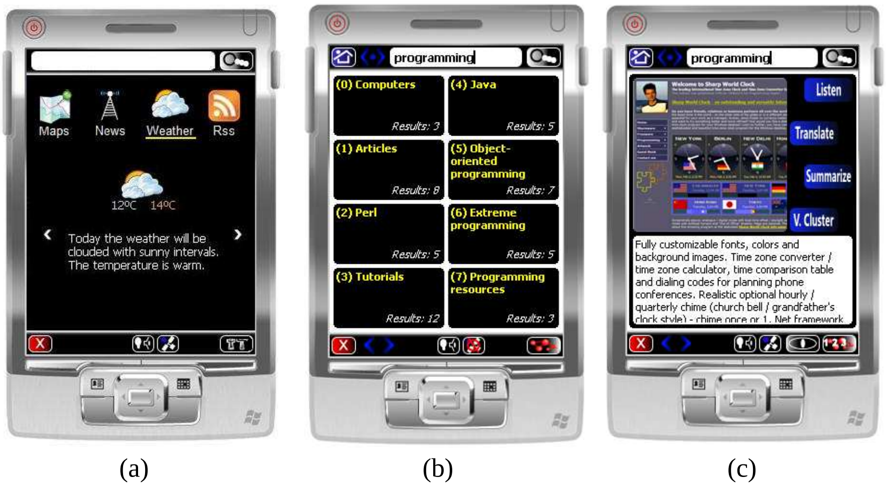
Within the VipAccess project[5], we propose to cluster web page results "on the fly" independently of the language thus allowing to searching for any query in any language over the entire web and providing a user-friendly interface for mobile devices (Figure 2b). For that purpose, we propose to cluster web page results based on a new clustering algorithm CBL (Clustering by Labels) especially designed for web page results. Comparatively to Searchme, we propose a more sophisticated way to cluster web page results, which does not depend on pre-defined categories, and as such can be applied "on the fly" as web page results are retrieved from any domain, any language or any search engine.

In terms of visualization of web page results, clustering may drastically improve users' satisfaction rates as only few selection items are presented to the user. However, an extra-step in the search process is introduced which may interfere with the users' habits to scroll lists of web page results. In this paper, we also propose

[4] http://m.searchme.com/

[5] This project is funded by the Portuguese *Fundação para a Ciência e a Tecnologia* with the reference PTDC/PLP/72142/2006.

different visualizations which try to make the most of both techniques i.e. lists of web page results and lists of clusters of web pages results.



Today the weather will be clouded with sunny intervals. The temperature is warm.

Fully customizable fonts, colors and background images. Time zone converter / time zone calculator, time comparison table and dialing codes for planning phone conferences. Realistic optional hourly / quarterly chime (church bell / grandfather's clock style) - chime once or 1. Net framework

Fig. 2. (a) VipAccess Mobile interface. (b) VipAccess Mobile with clusters. (c) VipAccess Mobile for summarization.

In terms of summarization of web contents, accessing summaries of information instead of full information may be a great asset for users of mobile devices. Indeed, most web pages are designed to be viewed on desktop displays. As a consequence, users may find it hard to evaluate the importance of a document as they have to come through all of it by repetitive zooming and scrolling. Some solutions have been proposed by content providers to overcome these drawbacks. They usually require an alternate trimmed-down version of documents prepared beforehand or the definition of specific formatting styles. However, this situation is undesirable as it involves an increased effort in creating and maintaining alternate versions of a web site. Within the VipAccess project, we propose to automatically identify the text content of any web page and summarize it in an efficient way so that web browsing is limited to its minimum. For that purpose, we propose a new architecture for summarizing Semantic Textual Units [3] based on efficient algorithms for semantic treatment such as the SENTA multiword extractor [4] which allows real-time processing and language-independent analysis of web pages thus proposing quality content extraction and visualization (Figure 2c).

2 Clustering Web Page Results

The categorization of web page results is obtained by the implementation of a new clustering algorithm called CBL (Clustering by Label) which is specifically designed to cluster web page results and is inspired from the label-derived approach. In terms of clustering algorithms, two different approaches have been proposed: label-derived clustering [1][5][6][7] and document-derived clustering [2][8][9]. The first approach defines potential labels and agglomerates documents which share common labels and the second groups similar documents based on text similarities and extracts potential labels at the end of the process. CBL is a label-derived clustering algorithm and as such, the first step of the clustering process aims at identifying potential labels.

2.1 Label Identification

Most methodologies identify potential labels based on the extraction of frequent itemsets. A frequent itemset is a set of items that appear together in more than a minimum fraction of the whole document set. For that purpose, different language-independent and language-dependent approaches have been proposed. In the first case, [5] implement a suffix tree-like structure and [6] use association rules. In the second case, [1] propose to extract common gapped sentences from linguistically enriched web snippets and [7] extract frequent word sequences based on suffix-arrays which are weighted using the well-know tf.idf score.

As one may want to search over the entire web in any language and any domain, it is important that the clustering algorithm only depends on language-independent features. Within this scope, the identification of relevant labels based on frequent itemsets mainly takes frequency of occurrence as a clue for extraction. However, this methodology suffers from the poor quality of web snippets which mainly contain ill-formed sentences with many repetitions. To overcome this drawback, we propose to weight strings based on three different word distributions and consequently extract potential labels.

Internal Value of a String. If a string[6] appears alone in a chunk of text separated on both sides by any given delimiter (such as a HTML tag or a comma), this string is likely to be meaningful. This characteristic is weighted in Equation (1) where *w* is any string, *A*(*w*) is the number of occurrences where *w* appears alone in a chunk and *F*(*w*) is the total number of occurrences of *w*.

$$W_1(w) = \frac{A(w)}{\ln\left(F(w)\right)}. \tag{1}$$

External Value of a String 1. The bigger the number of strings that co-occur with any string *w* both on its left and right contexts, the less meaningful this string is likely to be. This characteristic is weighted in Equation (2) where *w* is any string and *WIL*(*w*) (resp. *WIR*(*w*)) is the number of strings which appear on the immediate left (resp. right) context of string *w*.

$$W_2(w) = \frac{WIL(w)+WIR(w)}{2\times F(w)}. \tag{2}$$

[6] In our context, a string is any sequence of characters separated by spaces or other common linguistic delimiters such as dots, commas, etc.

External Value of a String 2. The bigger the number of different strings that co-occur with any string *w* both on its left and right contexts compared to the number of co-occurring strings on both contexts, the less meaningful this string is likely to be. This characteristic is weighted in Equation (3) where *w* is any string, *WDL*(*w*) (resp. *WDR*(*w*)) is the number of different strings which appear on the immediate left (resp. right) context of string *w* and *FH*(*w*) is equal to max[F(*w*)], for all *w*.

$$W_3(w) = \left(\frac{WDL(w)}{WIL(w)} + \frac{WDL(w)}{FH(w)}\right) \times \frac{WIL(w)}{F(w)} + \left(\frac{WDR(w)}{WIR(w)} + \frac{WDR(w)}{FH(w)}\right) \times \frac{WIR(w)}{F(w)}. \qquad (3)$$

Based on these three characteristics, we propose to weight all strings from the web snippets as in Equation (4) such that the smaller the *W*(*w*) value, the more meaningful the string *w*.

$$W(w) = \begin{cases} W_2(w) \times W_3(w), W_1(w) < 0.5 \\ \dfrac{W_2(w) \times W_3(w)}{1 + W_1(w)}, W_1(w) \geq 0.5 \end{cases} \qquad (4)$$

In Table 1, we present the 30 most relevant results of our weighting score *W*(.) for the query term "programming" searched over Google search engine[7], Yahoo search engine[8] and MSN search engine[9] accessed via respective web services.

Table 1. The first 30 strings ordered by *W*(.) for the query "programming".

String (1-5)	String (6-10)	String (11-15)	String (16-20)	String (21-25)	String (26-30)
articles	perl	tutorials	cgi	documentation	tips
wikibooks	java	c	category	news	science
computers	php	wiki	knuth	net	object-oriented
compilers	training	security	home	unix	site
subject	forums	database	advanced	internet	downloads

2.2 Clustering by Labels

Once all important words have been identified, these are going to play a crucial role in the process of clustering following the label-derived approach. Within this scope, many algorithms have been proposed based on frequent item sets [1][5][6][7]. In this paper, we propose a new algorithm called Clustering by Label (CBL) which objective is to group similar documents around meaningful word anchors i.e. labels. The algorithm is based on three steps: pole creation, unification and absorption, and labeling.

[7] http://www.google.com

[8] http://www.yahoo.com

[9] http://www.msn.com

Pole Creation. We first need to initialize the algorithm so that we can start from potential meaningful labels. For that purpose, all words with less than a given threshold α[10], which cover more than two urls, are proposed as initial cluster centers i.e. poles. For each start pole, a list of urls is built. An url is added to the list if it contains the pole word before a β position of the sorted relevant word list of each url. In particular, this allows to controlling the number of urls which are added to each pole since low β will produce smaller clusters and on the opposite, high values will join more results.

Union and Absorption. The next stage aims at iteratively unifying clusters which contain similar urls. For that purpose, we define two types of agglomerations: Union, when two clusters contain a significant number of common urls and share similar size in terms of cluster number; Absorption, when they share many common urls but are dissimilar in size. As a consequence, we define two proportions: P1, the number of common urls between two clusters divided by the number of urls of the smaller cluster and P2, the number of urls in the smaller cluster divided by the number of urls in the bigger cluster. The following algorithm is then iterated.
For each cluster, P1 is calculated over all other clusters. Then for each pair of clusters, if P1 is higher than a constant γ, then we evaluate P2 between both clusters. If P2 is higher than a constant δ, then the pair of clusters is added to the Union list otherwise it is integrated in the absorption list. Once all clusters have been covered, both union lists and absorption are treated. The union list is first processed as follows.

For each cluster pair in the union list[11], each two clusters are joined into the original cluster with the highest *W*(.) score for its label. At each step of this process, clusters indexes are substituted and unified clusters are removed in the union list to keep a list of updated clusters. Then the absorption list is processed.

Iteratively select the pair of clusters which contains any cluster which cannot be absorbed by any other one in the absorption list. Once encountered, this cluster absorbs the cluster which forms the pair with it, cluster indexes are updated and useless clusters removed. Both lists have been updated and the initial process iterates, thus enabling flat clustering (first step of the algorithm) or hierarchical clustering (all steps of the algorithm). Moreover, the CBL algorithm allows soft clustering as urls may be contained in different clusters. Finally, clusters are labeled.

Labeling. By union and absorption, each cluster may contain different candidate labels. However, it may be the case that urls in the cluster contain more meaningful words (i.e. multiword units) than the highly scored single words. As a consequence, multiword units are extracted from the web snippets agglomerated in the clusters by applying a methodology proposed by [18] implemented with suffix-arrays for real-time processing[12]. Then, each multiword unit is compared to the potential labels and if it contains one of the single words it is evaluated by frequency if it must replace the single word label. Finally, the best scoring labels, with a given threshold, are chosen as final labels.

[10] Most meaningful strings.

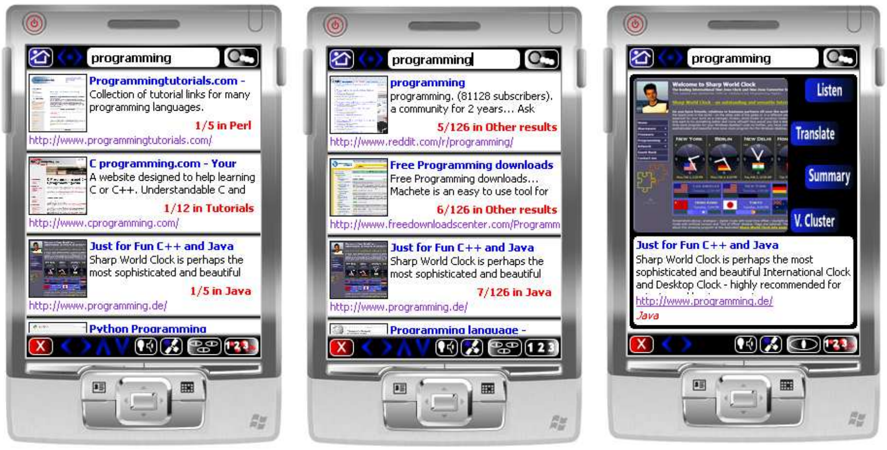
[11] Both the union and the absorption lists are ordered by *W*(.) score of the label.

[12] This method has proved to be particularly suited for web snippets processing.

2.3 Visualization

In terms of visualization of web page results, clustering may drastically improve users' satisfaction rates as only few selection items are presented to the user on the small screens of mobile devices (Figure 2b). However, an extra-step in the search process is introduced which may interfere with the users' cognitive process to search for information. Indeed, the user is used to find web page results after the first selection. In order to avoid the gap between the classic view (lists of web pages) and the cluster view (list of clusters), we propose to display the most relevant web page result of each encountered cluster in the form of a list as shown in Figure 3a. As such, the user is proposed the best possible coverage of its query with the minimum number of web page results thus reducing scrolling and maintaining the cognitive process for information search. If the user wants to keep the classic view, this option is available but always with the indication of the name of the belonging cluster so that the user can navigate to any given cluster and visualize only its members (Figure 3b).

In order to take into account that the users of mobile devices may use their device in different contexts, such as car, classroom or street, we also propose a full-screen visualization (Figure 3c). In this case, the best first web page result of the most relevant cluster is presented to the user. The next result is obviously the best first web page result of the second most relevant cluster, and so on and so forth.



programming

Programmingtutorials.com -
Collection of tutorial links for many programming languages.
1/5 in Perl
http://www.programmingtutorials.com/

C programming.com - Your
A website designed to help learning C or C++. Understandable C and
1/12 in Tutorials
http://www.cprogramming.com/

Just for Fun C++ and Java
Sharp World Clock is perhaps the most sophisticated and beautiful
1/5 in Java
http://www.programming.de/

Python Programming

(a)

programming

programming
programming. (81128 subscribers). a community for 2 years... Ask
5/126 in Other results
http://www.reddit.com/r/programming/

Free Programming downloads
Free Programming downloads... Machete is an easy to use tool for
6/126 in Other results
http://www.freedownloadscenter.com/Programm

Just for Fun C++ and Java
Sharp World Clock is perhaps the most sophisticated and beautiful
7/126 in Java
http://www.programming.de/

Programming language -

(b)

programming

Just for Fun C++ and Java
Sharp World Clock is perhaps the most sophisticated and beautiful International Clock and Desktop Clock - highly recommended for
http://www.programming.de/
Java

(c)

Fig. 3. (a) Clustering visualization. (b) List visualization. (c) Full-screen visualization.

The visualization issue of web page results has never been addressed as far as we know, although it is at the core of the success or failure of new techniques in Information Retrieval. Indeed, most search engines which propose interfaces with clustering of web page results[13] are not as popular as classic search engines although they provide a better understanding of the retrieved information. A reason for that may be the lack of newly designed interfaces for the sake of information search.

[13] For example, http://www.clusty.com or http://www.searchme.com

3 Web Page Summarization

After clustering web page results, scrolling and zooming must also be kept to its minimum for web browsing. For this purpose, we propose a new architecture to summarize Semantic Textual Units [3] which embeds an efficient algorithm for multiword extraction [4].

3.1 Semantic Textual Units and Multiword Units

One main problem to tackle is to define what to consider as a relevant text in a web page. Indeed, web pages often do not contain a coherent narrative structure. So, the first step of any system is to identify rules for determining which text should be considered for summarization and which should be discarded. For this purpose, [3] propose to identify Semantic Textual Unit (STU). STUs are page fragments marked with HTML markups which specifically identify pieces of text following the W3 consortium specifications. It is clear that the STU methodology is not as reliable as any language model for content detection [10] but on the opposite it allows fast processing of web pages.

Once each STU has been identified in the web page it is processed with the SENTA software [4] to identify and mark relevant phrases in it. SENTA is statistical parameter-free software which can be applied to any language without tuning and as a consequence is totally portable. Moreover, its efficient implementation shows time complexity Θ(N log N) where N is the number of words to process which allows the extraction of relevant phrases in real-time.

3.2 Extractive Text Summarization

Extractive text summarization aims at finding the most significant sentences in a given text. So, a significance score must be assigned to each sentence in a STU. The sentences with higher significance naturally become the summary candidates and a compression rate defines the number of sentences to extract. For this purpose, we implement the TextRank algorithm [11] combined with an adaptation of the well-known inverse document frequency, the inverse STU frequency (*isf*) to weight word relevance. The basic idea is that highly ranked words with high *isf* are more likely to represent relevant words in the text and as a consequence provide good clues to extract relevant sentences for the summary.

Within our purpose, each STU is first represented as an unweighted oriented graph being each word connected to its successor following sequential order in the text. Following the TextRank algorithm, the score *S*(.,.) of any word w_i in any *stu* is defined as in Equation (5) where *In*(w_i) is the set of words that point to w_i, *Out*(w_j) is the set of words that the word w_j points to and *d* is the damping factor set to 0.85.

$$S(w_i, stu) = (1-d) + d \times \sum_{j \in In(w_i)} \frac{S(w_j, stu)}{|Out(w_j)|}. \qquad (5)$$

Then, each word is weighted as in Equation (6) based on its graph-based ranking and its relevance in the text based on its inverse STU frequency where *N* is the number of STUs in the text and *stuf*(*w*) is the number of STUs the word *w* appears in.

$$rw.isf(w,stu) = S(w,stu) \times log_2 \frac{N}{stuf(w)}. \quad (\mathbf{6})$$

Finally, the sentence significance weight is defined as in [12], thus giving more weight to longer sentences, as shown in Equation (7) where |*S*| stands for the number of words in sentence *S*, w_i is a word in *S* and *max*(|*S*|) is the length of the longest sentence in the STU.

$$weight(S,stu) = \frac{\sum_{i=1}^{|S|} rw.isf(w_i,stu) \times |S|}{max(|S|)}. \quad (\mathbf{7})$$

In order to present as much information of the web page as possible so that its understanding is eased, the best scoring sentences of each STUs are retrieved and presented to the user as in Figure 2c[14]. As such, the user gets the most of the web page in a small text excerpt easy to read and scroll.

5 Conclusions and Future Work

In this paper, we proposed a global solution to web search and web browsing for handheld devices based on web page results clustering, web page summarization and new ideas for visualization.

In order to enable full information access to any users (paired or impaired), we also propose a speech-to-speech interface which is used as the exchange mode which may allow to achieving greater user satisfaction [13] in situations where the hands are not free [14], whenever reading is difficult [15], or in situations of mobility [16].

Moreover, we propose a location search based on Global Positioning System (GPS) which automatically expands the original query with the closest city name to the user's location.

In particular, a test of the interface has been conducted in the context of visually impaired people which received positive feedback although coherent and exhaustive evaluation is still needed in the way [17] explain.

References

1. Ferragina, P., Gulli, A.: A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering, Journal of Software: Practice and Experience, 38(2), 189 -- 225 (2008)

[14] Compression rate is defined by the user in the menu options.

2. Campos, R., Dias, G., Nunes, C., Nonchev, B.: Clustering of Web Page Search Results: A Full Text Based Approach. International Journal of Computer and Information Science, 9(4), (2008)
3. Buyukkokten, O., Garcia-Molina H., Paepcke, A.: Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. 10th International World Wide Web Conference, (2000)
4. Gil, A., Dias, G.: Using Masks, Suffix Array-based Data Structures and Multidimensional Arrays to Compute Positional Ngram Statistics from Corpora. Workshop on Multiword Expressions of the International Conference of the Association for Computational Linguistics, (2003)
5. Zamir, O., Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. 19th Annual International SIGIR Conference, (1998)
6. Fung, P., Wang, K., Ester, M.: Large Hierarchical Document Clustering using Frequent Itemsets. SIAM International Conference on Data Mining, (2003)
7. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on Singular Value Decomposition. Intelligent Information Systems Conference, (2004)
8. Jiang, Z., Joshi, A., Krishnapuram, R., Yi, Y.: Retriever Improving Web Search Engine Results using Clustering. Journal of Managing Business with Electronic Commerce, (2002)
9. Dias, G., Pais, S., Cunha, F., Costa, H., Machado, D., Barbosa, T., Martins, B.: Hierarchical Soft Clustering and Automatic Text Summarization for Accessing the Web on Mobile Devices for Visually Impaired People. 22nd International FLAIRS Conference, (2009)
10. Dolan, W.B., Quirk, C., Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. Interantional Conference on Computational Linguistics, (2004)
11. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. Conference on Empirical Methods in Natural Language Processing, (2004)
12. Vechtomova, O., Karamuftuoglu, M.: Comparison of Two Interactive Search Refinement Techniques. Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting, (2004)
13. Lee, K. W., Lai, J.: Speech versus Touch: A Comparative Study of the Use of Speech and DTMF Keypad for Navigation. International Journal Human-Computer Interaction, 19, 343-360. (2005)
14. Parush, A.: Speech-based Interaction in a Multitask Condition: Impact of Prompt Modality. Human Factors, 47, 591-597. (2005)
15. Fang, X., Xu, S., Brzezinski, J., Chan, S. S.: A Study of the Feasibility and Effectiveness of Dual-modal Information Presentations. International Journal Human-Computer Interaction, 20, 3-17. (2006)
16. Oviatt, S. L., Lunsford, R.: Multimodal Interfaces for Cell Phones and Mobile Technology. International Journal of Speech Technology, 8, 127-132. (2005)
17. Fallman, D., Waterworth, J. A.: Dealing with User Experience and Affective Evaluation in HCI Design: A Repertory Grid Approach. Conference on Human Factors in Computing Systems, (2005)
18. Frantzi K.T., Ananiadou S.: Retrieving Collocations by Co-occurrences and Word Order Constraint. 16th International Conference on Computational Linguistics, (1996)