Accessing the Web on Handheld Devices for Visually Impaired People

Gaël Dias[1], and Bruno Conde[1]

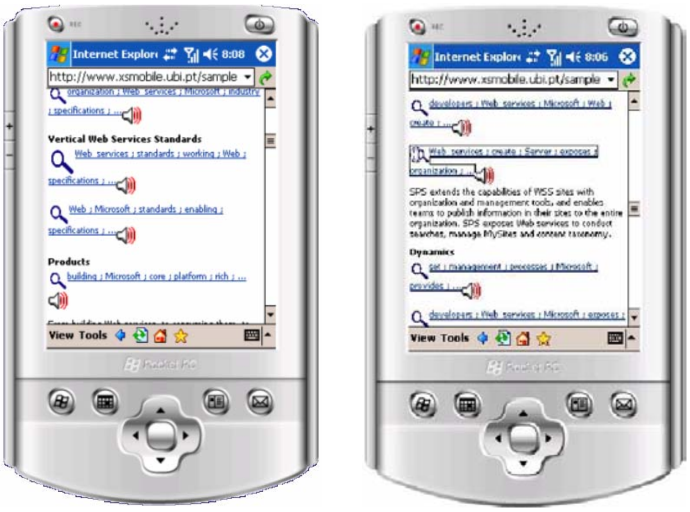
[1] Centre for Human Language Technology and Bioinformatics, University of Beira Interior,
6200-001 Covilhã, Portugal
{ddg,bruno}@hultig.di.ubi.pt
http://hultig.di.ubi.pt

Abstract. In this paper, we propose an automatic summarization system to ease web browsing for visually impaired people on handheld devices. In particular, we propose a new architecture for summarizing Semantic Textual Units [2] based on efficient algorithms for linguistic treatment [3][6] which allow real-time processing and deeper linguistic analysis of web pages, thus allowing quality content visualization. Moreover, we present a text-to-speech interface to ease the understanding of web pages content. To our knowledge, this is the first attempt to use both statistical and linguistic techniques for text summarization for browsing on mobile devices.

1 Introduction

Visually impaired people are info-excluded due to the overwhelming task they face to read information on the web. Unlike fully capacitated people, blind people can not read information by just scanning it quickly i.e. they can not read texts transversally. As a consequence, they have to come through all sentences of web pages to understand if a document is interesting or not.

To solve this problem, we propose an automatic summarization server-based architecture for web browsing on handheld devices. In particular, we introduce five different efficient methods for summarizing subparts of web pages in real-time. Two main approaches have already been proposed in the literature. First, some methodologies such as [2][14] use simple but fast summarization techniques to produce results in real-time. However, they show low quality contents for visualization as they do not linguistically process the web pages. Second, some works apply linguistic processing and rely on *ad hoc* heuristics [7] to produce compressed contents but can not be used in a real-time environment. Moreover, they do not use statistical evidence which is a key factor for high quality summarization. As a consequence, we propose a new architecture, called XSMobile, for summarizing Semantic Textual Units [2] based on efficient algorithms for linguistic treatment [3][6] that allow real-time processing and deeper linguistic analysis of web pages, thus producing quality content visualization as illustrated in Figure 1.



Internet Explore 8:08
http://www.xsmobile.ubi.pt/sample
View Tools
Internet Explore 8:06
http://www.xsmobile.ubi.pt/sample
View Tools

organization ; Web services ; Microsoft ; industry ; specifications ; ...

Vertical Web Services Standards

Web services ; standards ; working ; Web ; specifications ; ...

Web ; Microsoft ; standards ; enabling ; specifications ; ...

Products

building ; Microsoft ; core ; platform ; rich ; ...

developers ; Web services ; Microsoft ; Web ; create ; ...

Web services ; create ; Server ; exposes ; organization ; ...

SPS extends the capabilities of WSS sites with organization and management tools, and enables teams to publish information in their sites to the entire organization. SPS exposes Web services to conduct searches, manage MySites and content taxonomy.

Dynamics

set ; management ; processes ; Microsoft ; provides ; ...

developers ; Web services ; Microsoft ; exposes ;

Fig. 1. Screenshot of the XSMobile architecture

2 Text Unit Identification

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 [1].

For that purpose, [15] propose a C5.0 classifier to differentiate narrative paragraphs from non narrative ones. However, 34 features need to be calculated for each paragraph which turns this solution impractical for real-time applications. In the context of automatic construction of corpora from the web, [5] propose to use a language model based on Hidden Markov Models using the SRILM toolkit [12]. This technique is certainly the most reliable one as it is based on the essence of the language but still needs to be tested in terms of processing time. Finally, [2] propose Semantic Textual Unit (STU) identification. In summary, STUs are page fragments marked with HTML markups which specifically identify pieces of text following the W3 consortium specifications. However, not all web pages respect the specifications and as a consequence text material may be lost. In this case, unmarked strings are considered STUs if they contain at least two sentences.

3 Linguistic Processing

On the one hand, single nouns and single verbs usually convey most of the information in written texts. On the other hand, compound nouns (e.g. *hot dog*) and phrasal verbs (e.g. *take off*) are also frequently used in everyday language, usually to precisely express ideas and concepts that cannot be compressed into a single word. As a consequence, identifying these lexical items is likely to contribute to the performance of the extractive summarization process.

Subsequently, each STU in the web page is first morpho-syntactically tagged with the efficient TnT tagger[1] [3]. Then, multiword units are extracted from each STU based on an efficient implementation of the SENTA[2] multiword unit extractor [6] which shows time complexity Θ(N log N) where N is the number of words to process. Then, multiword units which respect the following regular expression are selected for quality content visualization:

```
[Noun Noun* | Adjective Noun* | Noun Preposition Noun | Verb Adverb].
```

This technique is usual in the field of Terminology [14]. A good example can be seen in Figure 1 where the multiword unit "Web Services" is detected, where existing solutions [2][7][14] would at most consider both words "Web" and "Services" separately. Finally, we remove all stop words present in the STU. This process allows faster processing of the summarizing techniques as the Zipf's Law shows that stop words represent 1% of all the words in texts but cover 50% of its surface.

4 Summarization Techniques

Once all STUs have been linguistically processed, the next step of the extractive summarization architecture is to extract the most important sentences of each STU. In order to make this selection, each sentence in a STU is assigned a significance weight. The sentences with higher significance become the summary candidate sentences. Then, the compression rate defines the number of sentences to be visualized.

Simple *tf.idf*: This methodology is mainly used in Information Retrieval [13]. The sentence significance weight is the sum of the weights of its constituents divided by the length of the sentence. A well-known measure for assigning weights to words is the *tf.idf* score [11]. The *tf.idf* score is defined in Equation 1 where *w* is a word, *stu* a STU, *tf(w, stu)* the number of occurrences of *w* in *stu*, |*stu*| the number of words in the *stu* and *df(w)* the number of documents where *w* occurs.

$$tf.idf(w,stu) = \frac{tf(w,stu)}{|stu|} \times \log_2 \frac{N}{df(w)} \tag{1}$$

In our case, a dictionary of *idf* values is processed for each website where XSMobile is installed based on the collection of texts present in it. The process is web-based. All texts in the collection of the website are first linguistically processed as explained in Section 3. Then, the *n* most frequent words of the collection are extracted to produce query samples sent to the web search engine Google™. For each query, the first 10 most relevant urls are gathered given rise to 10*n urls. Then, a web spider processes each url as deeply as possible in the hypertext structure and extracts all texts related to the initial query. Finally, after automatically gathering huge quantities of texts to approximate as best as possible the ideal *idf* values of the words, a XML dictionary of <*word, idf*> entries is produced.

[1] http://www.coli.uni-saarland.de/~thorsten/tnt/.

[2] http://senta.di.ubi.pt/.

So, the sentence significance weight, $weight_1(S, stu)$, is defined straightforwardly in Equation 2 where $|S|$ stands for the number of words in S and w_i is a word in S.

$$weigth_1(S,stu) = \frac{\sum_{i=1}^{|S|} tf.idf(w_i,stu)}{|S|} \tag{2}$$

Enhanced *tf.idf*: In the field of Relevant Feedback, [13] propose a new score for sentence weighting that proves to perform better than the simple *tf.idf*. In particular, they propose a new weighting formula for word relevance, *W(.,.)*. It is defined in Equation 3 where $\text{argmax}_w(tf(w,stu))$ corresponds to the word with the highest frequency in the STU.

$$W(w,stu) = \left(0.5 + \left(0.5 \times \frac{tf(w,stu)}{\arg\max_{w}(tf(w,stu))}\right)\right) \times \log_2 \frac{N}{df(w)} \tag{3}$$

Based on this weighting factor, [13] define a new sentence significance factor $weight_2(S,stu)$ which takes into account the normalization of the sentence length. The subjacent idea is to give more weight to sentences which are more content-bearing and central to the topic of the STU as shown in Equation 4 where $\text{argmax}(|S|)$ is the length of the longest sentence in the STU.

$$weigth_2(S,stu) = \frac{\sum_{i=1}^{|S|} W(w_i,stu) \times |S|}{\left(\arg\max_{s}(|S|)\right)} \tag{4}$$

The rw.idf: Recently, [10] have proposed the TextRank algorithm. The basic idea of the algorithm is the same as the PageRank algorithm proposed by [4] i.e. the higher the number of votes that are cast for a vertex, the higher the importance of a vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is. The score of a vertex V_i is defined as in Equation 5 where $In(V_i)$ is the set of vertices that point to it, $Out(V_j)$ is the set of vertices that the vertex V_j points to and d is a dumping factor[3].

$$S(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{5}$$

In our case, each STU is represented as an un-weighted oriented graph being each word connected to its successor following sequential order in the text as in Figure 2.

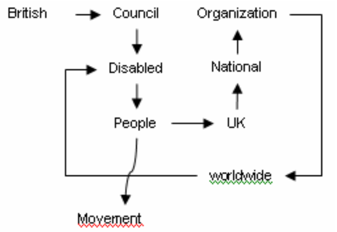


Fig. 2. Graph representation of the text: "*The British Council of Disabled People is the UK's National Organization of the worldwide Disabled People's Movement*".

[3] *d* was set to 0.85 as referred in [4].

After the graph is constructed, the score associated with each vertex is set to an initial value of 1, and the ranking algorithm is run on the graph for several iterations until it converges. So, each word is then weighted as in Equation 6

$$rw.idf(w, stu) = S(w) \times \log_2 \frac{N}{df(w)} \tag{6}$$

and the sentence significance weight, *weight*$_3$(*S, stu*), is defined straightforwardly in Equation 7 where |*S*| stands for the number of words in *S* and w_i is a word in *S*.

$$weigth_3(S, stu) = \frac{\sum_{i=1}^{|S|} rw.idf(w_i, stu)}{|S|} \tag{7}$$

Cluster Methodologies: Luhn suggested in [9] that sentences in which the greatest number of frequently occurring distinct words are found in greatest physical proximity to each other, are likely to be important in describing the content of the document in which they occur[4]. The procedure proposed by [2], when applied to sentence *S*, works as follows. First, they mark all the significant words in *S*. A word is significant if its *tf.idf* is higher than a certain threshold *T*. Second, they find all clusters in *S* such that a cluster is a sequence of consecutive words in the sentence for which the following is true: (i) the sequence starts and ends with a significant word and (ii) fewer than *D* insignificant words must separate any two neighboring significant words within the sequence. Then, a weight is assigned to each cluster. This weight is the sum of the weights of all significant words within a cluster divided by the total number of words within the cluster. Finally, as a sentence may have multiple clusters, the maximum weighted cluster is taken as the sentence weight.

5. Text-to-Speech Interface

The Text-to-Speech module is a crucial issue for accessibility of Visually Impaired People to web page contents. For this purpose, we have integrated the Microsoft Speech Server into our architecture using the SALT markup language following the architecture proposed in Figure 3.

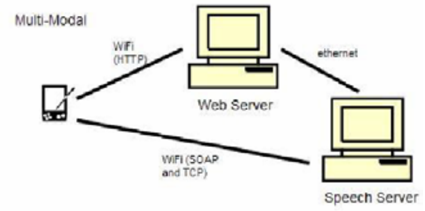


Fig. 3. Text-to-Speech Interface

However, in future work, we will integrate a Speech-to-Speech module on the proper device in order to avoid the overload of the Microsoft Speech Server which has shown limitations for high amounts of requests.

[4] [2] based their sentence ranking module on this paradigm.

6. Conclusion

In this paper, we proposed an automatic summarization system to help web browsing for visually impaired people on handheld devices. Unlike previous works [2][7][14], it is based on efficient algorithms [3][6] for linguistic treatment that allow real-time processing and deeper linguistic analysis for quality content visualization. The first results are every encouraging in terms of (1) quality of the content of the summaries, especially with the rw.idf, (2) processing time although the architecture is not still distributed over different processing units and (3) user interaction satisfaction. However, improvements must be taken into account. In particular, current work involves the integration of a Speech-to-Speech control interface which may provide a solution capable to compete with Braille PDAs that are expensive and difficult to use.

References

1. Berger, A., and Mittal,V.: Ocelot: a System for Summarizing Web Pages. In Proc. of SIGIR. (2000).
2. Buyukkokten, O., Garcia-Molina, H. and Paepcke, A: Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In Proc. of the 10th International World Wide Web Conference. (2000).
3. Brants, T.: TnT - a Statistical Part-of-Speech Tagger. In Proc. of the 6th Applied NLP Conference. (2000).
4. Brin, S., and Page, L.: The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*. 30(1–7). (1998).
5. Dolan, W. B., Quirk, C., and Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In Proc. of COLING. (2004).
6. Gil, A., and Dias, G.: Using Masks, Suffix Array-based Data Structures and Multidimensional Arrays to Compute Positional Ngram Statistics from Corpora. In Proc. of the Workshop on Multiword Expressions of the 41st ACL. (2003).
7. Gomes, P. Tostão, S., Gonçalves, D. and Jorge, J: Web-Clipping: Compression Heuristics for Displaying Text on a PDA. In Proc. of Workshop on HCI with Mobile Devices. (2001).
8. Justeson, J. and Katz, S.: Technical Terminology: some Linguistic Properties and an Algorithm for Identification in text. *Natural Language Engineering*. (1). (1995).
9. Luhn, H.P.: The automatic creation of literature abstracts. *IBM Journal of Research and Development*. (1958).
10. Mihalcea R., and Tarau, P.: TextRank: Bringing Order into Texts. In Proc. of the Conference on Empirical Methods in Natural Language Processing. (2004).
11. Salton, G., Yang, C.S., and Yu, C.T.: A Theory of Term Importance in Automatic Text Analysis. *Amer. Soc. of Inf. Science*. (26)1. (1975).
12. Stolcke, A.: SRILM -- An Extensible Language Modeling Toolkit. In Proc. of International Conference on Spoken Language Processing. (2002).
13. Vechtomova, O., and Karamuftuoglu, M.: Comparison of Two Interactive Search Refinement Techniques. In Proc. of HLT-NAACL. (2004).
14. Yang, C., and Wang, F.L.: Fractal Summarization for Mobile Devices to Access Large Documents on the Web. In Proc. of the International World Wide Web Conference. (2003).
15. Zhang, Y., Zincir-Heywood, N., and Milios, E.: Summarizing Web Sites Automatically. In Proc. Conference of Canadian Society for Computational Studies of Intelligence. (2003).