Efficient Text Summarization for Web Browsing On Mobile Devices

Gaël Dias[1] and Bruno Conde[2]

Abstract. In this paper, we propose an automatic summarization server-based architecture for web browsing on handheld devices. In particular, we introduce different efficient methods for summarizing parts of web pages in real-time. Two main approaches have already been proposed in the literature. First, some methodologies such as [1] [5] use simple summarization techniques to produce results in real-time but clearly lack linguistic treatment for reliable content visualization. Second, some works apply linguistic processing and rely on *ad hoc* heuristics [2] to produce compressed contents but can not be used in real-time environment. As a consequence, we propose a new architecture for summarizing Semantic Textual Units [1] based on efficient algorithms for linguistic treatment that allow real-time processing and deeper linguistic analysis of web pages, thus allowing quality content visualization.

1 INTRODUCTION

The shift in human-computer interaction from desktop computing to mobile real-world interaction highly influences the needs for future decentralized user-adaptive systems. Designing personalized Web Services such as text summarization for web browsing on mobile devices is one of many challenges for the success of ubiquitous computing.

For handheld devices, screen size limitation is clearly the issue as most web pages are designed to be viewed on desktop displays. Indeed, the smallest web page excerpts displayed on any mobile device screen can interfere with users' comprehension, and the resulting scrolling is time consuming.
Some solutions have been proposed to overcome these limitations. They usually require an alternate trimmed-down version of documents prepared beforehand (e.g. WAP Browsers) or the definition of specific formatting styles (e.g. XML Schemas). However, this situation is undesirable as it involves an increased effort in creating and maintaining alternate versions of a web site.

To solve this problem, we propose an automatic summarization server-based architecture for web browsing on handheld devices. In particular, we introduce four 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 [1] [5] 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 [2] 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 [1] based on efficient algorithms for linguistic treatment [3] [4] that allow real-time processing and deeper linguistic analysis of web pages, thus producing quality content visualization as illustrated in Figure 1.

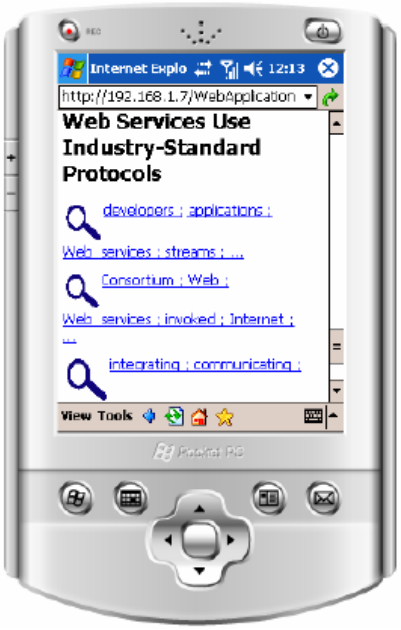

Figure 1. Screenshot of the XSMobile architecture

This paper is divided into five sections. First, we present the relevant related work in the area. Second, we talk about text unit identification and review the concept of Semantic Textual Units proposed by [1]. Third, we emphasize the linguistic treatment we apply on each Semantic Textual Units. Fourth, we present some implemented summarization techniques. And finally, we explain how the information is displayed on the mobile device.

2 RELATED WORK

[1] is certainly the most relevant first appearing paper of this field. They introduced two methods for summarizing parts of web pages. Each web page is broken into Semantic Textual Units that can each be hidden, partially displayed, made fully visible, or summarized. However, their work is built on old well known techniques for text summarization and do not introduce linguistic processing (except stemming) to remain real-time adaptable as processing is handled by the mobile device.

In order to introduce more knowledge compared to the previous model, [5] propose a fractal summarization model based on statistical and structure analysis of web pages. Thus, thematic features, location features, heading features, and cue features are

[1] Centre of Human Language Technology and Bioinformatics, University of Beira Interior, Portugal, email: ddg@di.ubi.pt
[2] Centre of Human Language Technology and Bioinformatics, University of Beira Interior, Portugal, email: countbruno@gmail.com

adopted. Their architecture first generates a skeleton of a summary and its details are generated on demands of users. Comparatively to [1], [5] propose a more organised structure but do not use any linguistic processing although they work on basis of a three-tier architecture which provides more processing power.

[2] are the first to introduce some linguistic knowledge into the process of text summarization. They use a parser to perform text segmentation and morphological analysis. In particular, they apply linguistic patterns for sentence compression rather than for sentence extraction. For example, some names are replaced with their acronyms and some adjectives may also be removed. The major drawback of this approach is the lack of statistical analysis which is a key factor for high quality summarization.

In XSMobile, our objective is to use both statistical evidence and linguistic processing for sentence extraction in real-time. For that purpose, we use two efficient linguistic softwares (the TnT tagger [3] and the SENTA multiword unit extractor [4]) and propose new sentence weighting schemes. To our knowledge, this is the first attempt to use both statistical and linguistic techniques for text summarization for browsing on mobile devices.

3 TEXT UNITS IDENTIFICATION

One main problem to tackle is to define what to consider as a relevant text in a web page. Indeed, the summary of a web page will be created on the basis of the text extracted by the web server. However, web pages often do not contain a coherent narrative structure [7]. 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 that purpose, [8] 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, [9] propose to use a language model based on Hidden Markov Models (HMM) using the SRILM toolkit [10]. 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[3].

Finally, [1] 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. It is clear that the STU methodology is not as reliable as any language model for content detection but on the opposite it allows fast processing of web pages.

So, any requested web page is first divided into STUs (i.e. narrative paragraphs) so that further linguistic processing can be performed to identify relevant information about the text.

4 LINGUISTIC PROCESSING

On the one hand, single nouns and single verbs usually convey most of the information in written texts. They are the main contributors to the "aboutness" of any text. 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. So, compound nouns and phrasal verbs provide good clues for text content description. As a consequence, identifying these lexical items is likely to contribute to the performance of the extractive summarization process [11]. For that purpose, we apply to each STU the following linguistic treatment.

Each STU in the web page is first morpho-syntactically tagged with the TnT tagger [3] which is an implementation of the Viterbi algorithm for second order Markov Models [12]. The main paradigm used for smoothing is linear interpolation and respective weights are determined by deleted interpolation. Unknown words are handled by a suffix trie and successive abstractions. As a summary, TnT is an efficient tagger in terms of processing power and reaches precision results around 96% to 99%.

Once morpho-syntactically tagged, each STU is processed by the SENTA multiword unit extractor [4]. SENTA combines an association measure called Mutual Expectation with an acquisition process based on an algorithm of local maxima called GenLocalMaxs over a data set of positional ngrams. Its efficient implementation shows time complexity $\Theta(N \log N)$ where N is the number of words to process. It is based on the definition of masks that virtually represent any positional ngram in the text and applies a suffix-array data structure coupled with the Multikey Quicksort algorithm [13] to compute positional ngram frequencies in real-time.

Both softwares are freely available and flexible for any language as the TnT can be trained on any tag set and SENTA is an unsupervised statistical parameter-free architecture. This is an important remark as our architecture can easily be adapted to other languages and as a consequence is totally portable.

Then, we apply some heuristics to define quality multiword units for content visualization. So, multiword units that do not respect the following regular expression are filtered out:

[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 would at most consider both words “Web” and “Services” separately. This would lead to less expressiveness of the content of the STU and may imply text understanding errors.

Finally, we remove all stop words present in the STU. This process allows faster processing of the summarizing techniques as the Zipf’s Law [15] shows that stop words represent 1% of all the words in texts but cover 50% of its surface.

5 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 chosen by the user defines the number of sentences to present on the screen of the device.

[3] By the time of implementation, this solution was unknown to us and as a consequence was not considered, but will be tested in future work.

For that purpose, we implement four basic extractive techniques: the simple tf.idf, the enhanced tf.idf and the two methodologies proposed by [1]. It is clear that more powerful methodologies exist. However, there are not still tailored for fast processing [11], although some research is done in this direction [16].

In the following subsections, we will explain the simple tf.idf and the enhanced tf.idf methodologies and introduce the cluster methodology proposed by [1].

5.1 Simple tf.idf

This methodology is simple and mainly used in Information Retrieval [6]. 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 [17]. The idea of the tf.idf score is to evaluate the importance of a word within a document based on its frequency and its distribution across a collection of documents. 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.

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

In our case, we processed all idf[4] values from a collection of texts: the DUC 2004 collection[5] plus all the texts in our test website. In particular, all texts of the collection have been linguistically processed as explained in Section 4.

So, the sentence significance weight, $weight_1(S, stu)$, is defined straightforwardly in Equation 2

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

where |*S*| stands for the number of words in *S* and w_i is a word in *S*.

5.2 Enhanced tf.idf

In the field of Relevant Feedback, [6] 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(.,.)*. In fact, this is a refinement of the tf.idf measure and it is defined in Equation 3

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

where argmax(*tf*(*w,stu*)) corresponds to the word with the highest frequency in the STU.

Based on this weighting factor, [9] define a new sentence significance factor $weight_2(S,stu)$ that 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 i.e. which contain a higher proportion of words with high tf.idf as shown in Equation 4

$$weigth_2(S,stu) = \frac{\sum_{i=1}^{|S|} W(w_i,stu)}{\left(\frac{\arg\max_s(|S|)}{|S|}\right)} \quad (4)$$

where argmax(|*S*|) is the length of the longest sentence in the STU.

5.3 Cluster methodologies

Luhn suggested in [19] 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. [1] based their sentence ranking module on this paradigm.

The procedure proposed by [1], 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. This is illustrated in Figure 2 where "*" are significant words and *D=2*.

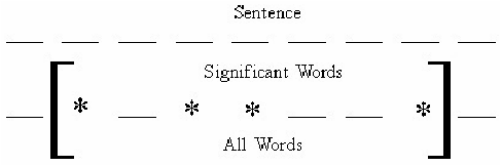

Figure 2. Cluster representation taken from [1].

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 weight of its clusters is taken as the sentence weight.

6 VISUALIZATION

The last part of the process is the visualization phase. For that purpose, the user can choose one option from a set of five levels of visualization for each summarization methodology as shown in Figure 3. In particular, at installation time, a link to this configuration page is automatically inserted in each page of the website. As a consequence, the user can choose a different visualization mode for each browsed web page. This mechanism is handled by cookies.

Following the same strategy as in [1], the user can choose between the following five options: (1) first characters of the most relevant sentence in the STU[6] and no summarization, (2) five most relevant keywords[7] in the STU and no summarization, (3) first characters of the most relevant sentence in the STU and summarization, (4) five

[4] The idf is the second argument of the product in Equation 1.
[5] The DUC 2004 corpus is available at http://duc.nist.gov/.
[6] This is the same idea as web snippets.
[7] Here, keyword stands for the most relevant lexical items in the STU according to the word weighting factor.

most relevant keywords in the STU and summarization, (5) no processing of the web page.

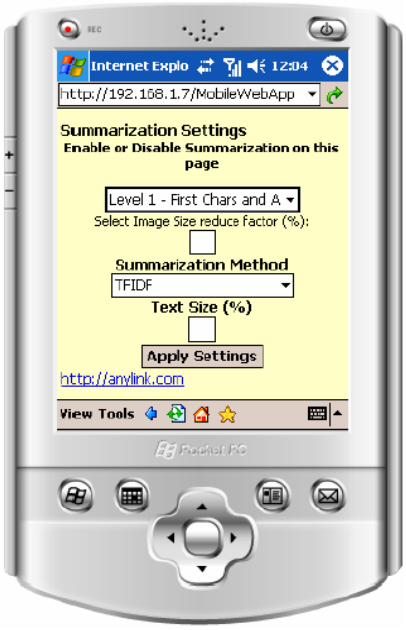

Figure 3. Screenshot of the XSMobile configuration page.

It is clear that for levels 3 and 4, the user must define the summarization compression rate, *C*. Each STU is then summarized according to *C* i.e. *E(C.|S|)* significant sentences are presented in order of relevance where *E(.)* is the floor function, *C*, the compression rate and |S| the number of sentences in the STU.

In order to help the user in its search for information, we also define a degree of significance of each STU. So the more relevant a STU is, the bigger its associated magnifying glass will be as shown in Figure 1. The significance factor of a STU is simply calculated as in Equation 5

$$factor_j(stu) = \sum_{i=1}^{|S|} weight_j(S_i, stu) \quad (5)$$

where *j* (*j=1..4*) defines the significance sentence weight formula. This weight is then normalized among all STUs in the web page so that its value ranges between [0..1] i.e. it represents its percentage of relevance compared to all other STUs relevance weights.

Finally, image compression rate is also accessible to the user. In this case, the process is performed by reformulating the <img> tag i.e. by modifying/inserting the width and height attributes. This process reduces both the size of the picture on the screen and the size of the picture to be transferred on the network. We are aware that this compression rate is not ideal but some improvements will be introduced in future work.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed an automatic summarization server-based architecture for web browsing on handheld devices. Unlike previous works [1] [2] [5], it is based on efficient algorithms [3] [4] 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 enhanced tf.idf, (2) processing time although the architecture is not still distributed over different processing units and (3) user interaction satisfaction. However, many improvements must be taken into account. Immediate future work involves applying a language model for content detection instead of the STU strategy. Another important improvement has to do with document structure. Indeed, hierarchical display is suitable for navigation of large documents and it is ideal for small area displays [5]. But, unlike [5], we intend to apply a hierarchical graph-based overlapping clustering algorithm [18] to automatically infer from text content only the relationships between text subparts.

ACKNOWLEDGMENTS

This work is funded by the Portuguese Foundation for Science and Technology under the SUMO project - POSC/PLP/57438/2004.

REFERENCES

[1] O. Buyukkokten, H. Garcia-Molina and A. Paepcke. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In Proc. of the 10th International World Wide Web Conference. (2000).
[2] P. Gomes, S. Tostão, D. Gonçalves and J. Jorge. Web-Clipping: Compression Heuristics for Displaying Text on a PDA. In Proc. of 3rd Workshop on Human Computer Interaction with Mobile Devices. (2001).
[3] T. Brants. TnT - a Statistical Part-of-Speech Tagger. In Proc. of the 6th Applied NLP Conference. (2000).
[4] A. Gil and G. Dias. 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).
[5] C. Yang and F.L. Wang. Fractal Summarization for Mobile Devices to Access Large Documents on the Web. In Proc. of the International World Wide Web Conference. (2003).
[6] O. Vechtomova and M. Karamuftuoglu. Comparison of Two Interactive Search Refinement Techniques. In Proc. of HLT-NAACL. (2004).
[7] A. Berger and V. Mittal. Ocelot: a System for Summarizing Web Pages. In Proc. of SIGIR. (2000).
[8] Y. Zhang, N. Zincir-Heywood, and E. Milios. Summarizing Web Sites Automatically. In Proc. of the 16th Conference of the Canadian Society for Computational Studies of Intelligence. (2003).
[9] W. B. Dolan, C. Quirk and C. Brockett. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In Proc. of COLING. (2004).
[10] A. Stolcke. SRILM -- An Extensible Language Modelling Toolkit. In Proc. of International Conference on Spoken Language Processing. (2002).
[11] R. Barzilay and M. Elhadad. Using Lexical Chains for Text Summarization. In Proc. of the Intelligent Scalable Text Summarization Workshop of ACL. (1997).
[12] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In IEEE. (77)2. (1989).
[13] J. Bentley and R. Sedgewick. Fast Algorithms for Sorting and Searching Strings. In Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms. (1997).
[14] J. Justeson and S. Katz. Technical Terminology: some Linguistic Properties and an Algorithm for Identification in text. Natural Language Engineering, (1). (1995).
[15] G. K. Zipf. Selective Studies and the Principle of Relative Frequency in Language. Harvard University Press. (1932).
[16] G. Silber and K. McCoy. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. Computational Linguistics. (28)4. (2002).
[17] G. Salton, C.S. Yang, and C.T. Yu. A Theory of Term Importance in Automatic Text Analysis. Amer. Soc. of Inf. Science. (26)1. (1975).
[18] G. Cleuziou, L. Martin and C. Vrain. PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classication and Textual Data. In Proc. of the 16th EACL. (2004).
[19] H.P. Luhn. The automatic creation of literature abstracts. IBM Journal of Research and Development.(1958).