A Comparison of Keyword- and Keyterm-based Methods for Automatic Web Site Summarization

Yongzheng Zhang and **Evangelos Milios** and **Nur Zincir-Heywood**
Faculty of Computer Science, Dalhousie University
6050 University Ave., Halifax, NS, Canada B3H 1W5
{yongzhen, eem, zincir}@cs.dal.ca

Abstract

Automatic Web site summarization, which is based on keyword and key sentence extraction from narrative text, is an effective means of making the content of a Web site easily accessible to Web users. This work is directed towards summary generation based on multi-word terms extracted by the *C-value*/*NC-value* method. Keyterm-based summaries are compared with keyword-based summaries for a list of test Web sites. The evaluation indicates that keyterm-based summaries are significantly better than keyword-based summaries, which have previously been shown to be as informative as human-authored summaries.

Introduction

The information overload problem on the World Wide Web has brought Web users great difficulty in information seeking tasks. Automatic Web site summarization is one of the effective ways to alleviate the information overload problem. An automatically generated Web site summary can help users get an idea of the main contents covered in the site without spending a lot of browsing time. However, to generate summaries as coherent as human authored summaries is a great challenge (Zhang, Zincir-Heywood, & Milios 2003).

Web document summarization techniques are derived from traditional text summarization techniques. Existing text summarization systems generate summaries automatically either by "extraction" or "abstraction". Extraction-based systems (Chuang & Yang 2000; Goldstein *et al.* 2000) analyze source documents using techniques such as frequency analysis to determine significant sentences in the context. Abstraction (Berger & Mittal 2000), on the other hand, requires a thorough understanding of the source text using knowledge-based methods and is normally more difficult to achieve with current natural language processing techniques (Goldstein *et al.* 1999).

Unlike traditional documents with well-structured discourse, Web documents are often semi-structured, and have more diverse contents than narrative text, such as bullets, short sentences, emphasized text and anchor text associated with hyperlinks. Consequently, Web site summarization is a non-trivial extension of the plain document summarization task due to the greater variety of possible feature sets. Research work in (Zhang, Zincir-Heywood, & Milios 2003) has shown that the identification of narrative text for summary generation is a key component of automatic Web site summarization.

The aim of this paper is to extend the keyword-based method described in (Zhang, Zincir-Heywood, & Milios 2003) by using automatically extracted multi-word terms in identifying key sentences in the narrative text of a Web site. Keyterms and key sentences are selected to form a Web site summary. The keyterm-based summaries for a list of test Web sites are experimentally compared with the keyword-based summaries (Zhang, Zincir-Heywood, & Milios 2003). We statistically evaluate the performance of automatic Web site summarization under different feature sets, namely, keywords or keyterms.

The rest of the paper is organized as follows. First, we review previous Web document summarization approaches. Second, we explain how to generate term-based summaries. Third, we discuss the design of our experiments and show the evaluation results. Finally, we conclude our work and describe future research directions.

Related Work

Research on Web document summarization to date has either been *content-based* or *context-based*. Content-based systems (Berger & Mittal 2000; Buyukkokten, Garcia-Molina, & Paepcke 2001) analyze the contents and extract the significant sentences to construct a summary, while context-based systems (Amitay & Paris 2000; Delort, Bouchon-Meunier, & Rifqi 2003) analyze and summarize the context of a Web document (e.g., brief content descriptions from search engine results) instead of its contents.

Berger and Mittal (Berger & Mittal 2000) propose a system named OCELOT, which applies the Expectation Maximization (EM) algorithm to select and order words into a "gist", which serves as the summary of a Web document. Buyukkokten et al. (Buyukkokten, Garcia-Molina, & Paepcke 2001) compare alternative methods for summarizing Web pages for display on handheld devices. The *Keyword* method extracts keywords from the text units, and the *Summary* method identifies the most significant sentence of each text unit as the summary for the unit. The test indicates that the combined *Keyword*/*Summary* method provides the

Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

best performance.

Amitay and Paris (Amitay & Paris 2000) propose an approach, which generates single-sentence long coherent textual snippets for a target Web page based on the context of the Web page, which is obtained by tracing backlinks, a service offered by search engines like Google. Experiments show that on average users prefer the summary created by this system compared to the textual snippets provided by search engines. Delort et al. (Delort, Bouchon-Meunier, & Rifqi 2003) address three important issues, *contextualization*, *partiality*, and *topicality* in any context-based summarizer and propose two algorithms, the efficiency of which depends on the size of the text content and the context of the target Web page.

In our previous work (Zhang, Zincir-Heywood, & Milios 2003), we extend single Web document summarization to the summarization of complete Web sites. The "Keyword/Summary" idea of (Buyukkokten, Garcia-Molina, & Paepcke 2001) is adopted, and the methodology is substantially enhanced and extended to Web sites by applying machine learning and natural language processing techniques. This approach generates a summary of a Web site consisting of the top 25 keywords and the top 5 key sentences. Since Web documents often contain diverse contents such as bullets and short sentences, the system applies machine learning and natural language processing techniques to extract the narrative content, defined as coherent and informative text, and then extracts keywords from the narrative text together with anchor text and special text (e.g., emphasized text). The key sentences are identified based on the density of keywords. Evaluation by users shows that the automatically generated summaries are as informative as human authored summaries (e.g., DMOZ[1] summaries).

Automatic Web Site Summarization (AWSS)

In this section we first describe the keyword-based approach to automatic Web site summarization. Then we discuss how to generate multi-word terms automatically and use identified keyterms to summarize a Web site based on the framework of the keyword-based approach.

Keyword-based AWSS

In our previous work (Zhang, Zincir-Heywood, & Milios 2003), we propose a content-based approach to summarizing an entire Web site automatically based on keyword and key sentence extraction. The system consists of a sequence of stages as follows.

URL Extraction In order to summarize a given Web site, Web pages within a short distance from the root of the site, which are assumed to describe the content of the site in general terms, are collected. A Web site crawler is designed to collect the top 1000 Web pages from the Web site domain via a breadth-first search starting at the home page, assumed to be at level (depth) one. The choice of a limit of 1000 is based on the observation that there is an average of 1000 pages up to and including depth equal to 4 after crawling 60 DMOZ Web sites. The selected depth of 4 is based on a tradeoff between crawling cost and informativeness of Web pages. For each Web site, the crawler will stop crawling when either 1000 pages have been collected, or it has finished crawling depth 4, whichever comes first.

Plain Text Extraction After the URLs of the Web pages have been collected, plain text is extracted from these Web pages and segmented into text paragraphs by the text browser *Lynx*[2], which was found to outperform several alternative text extraction tools such as *HTML2TXT*[3] and *html2txt*[4].

Narrative Text Classification The Web site summary is created on the basis of the text extracted by Lynx. However, due to fact that Web pages often contain tables of contents, link lists, or "service" sentences (e.g., copyright notices, webmaster information), it is important to identify rules for determining the text that should be considered for summarization. This is achieved in two steps. First, text paragraphs which are too short for summary generation are filtered out by the classifier LONG. Second, another classifier, NARRATIVE, is in turn used to extract *narrative* paragraphs from *long* paragraphs identified in the previous step. These two classifiers are trained by the decision tree tool C5.0[5] based on features extracted by shallow natural language processing.

Key Phrase Extraction Traditionally, keywords are extracted from the documents in order to generate a summary. In this work, single keywords are extracted via supervised learning. Based on such keywords, the most significant sentences, which best describe the document, are retrieved.

Keyword extraction from a body of text relies on an evaluation of the importance of each candidate keyword (Buyukkokten, Garcia-Molina, & Paepcke 2001). For Web site summarization, a candidate keyword is considered as a true keyword if and only if it occurs frequently in the Web pages of the site, i.e., the total frequency of occurrence is high.

As discussed before, Web pages are different from traditional documents. The existence of *anchor text* and *special text* (e.g., title, headings, italic text) contributes much to the difference. Anchor text is the text associated with hyperlinks, and it is considered to be an accurate description of the Web page linked to. A supervised learning approach is applied to learn the significance of each category of keywords.

In order to produce decision tree rules for determining the keywords of a Web site, a data set of 5454 candidate keywords (at most 100 for each site) from 60 DMOZ Web sites are collected. For each site, the frequency of each word in narrative text, anchor text and special text, is measured. Then the total frequency of each word over these three categories is computed, where the weight for each category is

[1] http://dmoz.org

[2] http://lynx.isc.org

[3] http://user.tninet.se/∼jyc891w/software/html2txt/

[4] http://cgi.w3.org/cgi-bin/html2txt

[5] http://www.rulequest.com/see5-unix.html

the same. Moreover, a standard set of 425 stop words (*a, about, above, ...*) (Fox 1992) is discarded in this step.

For each Web site, at most the top 100 candidate keywords are selected. For each candidate keyword, eight features of its frequency statistics (e.g., ratio of frequency to sum of frequency, ratio of frequency to maximum frequency in anchor text) in three text categories and the part-of-speech tag are extracted. In particular, the weight, w, of a candidate keyword is defined as the ratio of its frequency (over three categories of text) to the sum of frequency of all candidate keywords.

Next, each candidate keyword is manually labelled as *keyword* or *non-keyword*. The criterion to determine if a candidate keyword is a true keyword is that the latter must provide important information about the Web site. Based on frequency statistics and part-of-speech tags (Brill 1992) of these candidate keywords, a C5.0 classifier *KEYWORD* is constructed.

Among the total of 5454 cases, 222 cases are misclassified, leading to an error of 4.1%. The cross-validation of the classifier shows a mean error of 4.9%, which indicates the predictive accuracy of this classifier.

Once the decision tree rules for determining keywords have been built, they are applied to automatic keyword extraction from the Web pages of a new Web site. The top 25 keywords (ranked by w) for each site are kept as part of the summary. It is observed that 40% to 70% of keywords appear in the home page of a Web site.

Key Sentence Extraction Once the keywords are identified, the most significant sentences for summary generation can be retrieved from all narrative paragraphs based on the presence of keywords (Chuang & Yang 2000). The significance of a sentence is measured by calculating a weight value, which is the maximum of the weights for word clusters within the sentence. A word cluster is defined as a list of words which starts and ends with a keyword and less than 2 non-keywords must separate any two neighboring keywords (Buyukkokten, Garcia-Molina, & Paepcke 2001). The weight of a word cluster is computed by adding the weights of all keywords within the word cluster, and dividing this sum by the total number of keywords within the word cluster.

The weights of all sentences in all narrative text paragraphs are computed and the top five sentences (ranked according to sentence weight) are the key sentences to be included in the summary.

Summary Generation The overall summary is formed by the top 25 keywords and the top 5 key sentences. These numbers are determined based on the fact that key sentences are more informative than keywords, and the whole summary should fit in a single page.

Keyterm-based AWSS

The keyword identification in (Zhang, Zincir-Heywood, & Milios 2003) is based on word frequency analysis against three different categories of text, narrative text, anchor text, and special text. However, this method is unable to extract terms consisting of two or more component words. Since terms are more informative than single words, we aim to extract multi-word keyterms via automatic term extraction techniques and further identify key sentences based on the density of keyterms only.

This work introduces a keyterm-based approach which applies the same process as the keyword-based approach except in the key phrase (keyword or keyterm) extraction phase. In the keyterm-based method, multi-word terms are extracted from narrative text automatically and the top 25 keyterms are used to identify the top 5 key sentences in the narrative text for summary generation.

Automatic Term Extraction Terms are known to be linguistic descriptors of documents. Automatic term extraction is a useful tool for many text related applications such as text clustering and document similarity analysis (Milios *et al.* 2003). Effective systems for automatic term extraction have been developed. Turney proposes a key phrase extraction system GenEx which consists of a set of parameterized heuristic rules that are tuned to the training documents by a genetic program (Turney 2000). Witten et al. propose a system called KEA which builds a Naïve Bayes classifier using training documents with known key phrases, and then uses the classifier to find key phrases in new documents (Witten *et al.* 1999). Both GenEx and KEA generalize well across domains. However, they are aimed towards extracting key phrases from a single document rather than a whole document collection.

In this work, we apply a state-of-the-art method *C-value/NC-value* (Frantzi, Ananiadou, & Mima 2000) to extract multi-word terms from a Web site automatically. The *C-value/NC-value* method consists of both linguistic analysis (linguistic filter, part-of-speech tagging (Brill 1992), and stop-list) and statistical analysis (frequency analysis, *C-value/NC-value*). A linguistic filter is used to extract word sequences likely to be terms, such as noun phrases and adjective phrases.

The *C-value* is a domain-independent method used to automatically extract multi-word terms from the whole document corpus. It aims to get more accurate terms than those obtained by the pure frequency of occurrence method, especially terms that may appear as nested within longer terms. *C-value* is formally represented in Equation 1.

$$Cv(a) = \begin{cases} \log_2 |a| f(a), \textit{ a is not nested.} \\ \log_2 |a| (f(a) - \frac{\sum_{b \in T_a} f(b)}{P(T_a)}), \textit{ otherwise.} \end{cases} \quad (1)$$

where, a is a candidate term; $|a|$ is the number of words in a; $f(a)$ is the frequency of occurrence of a in the corpus; T_a is the set of extracted candidate terms that contain a; and $P(T_a)$ is the number of these longer candidate terms.

The *NC-value* is an extension to *C-value*, which incorporates information of context words into term extraction. Context words are those that appear in the vicinity of candidate terms, i.e. nouns, verbs and adjectives that either precede or follow the candidate term. Each context word is assigned a weight by Equation 2.

$$weight(w) = \frac{t(w)}{n}. \quad (2)$$

where, w is a term context word (noun, verb or adjective); $weight(w)$ is the assigned weight to the word w; $t(w)$ is the number of terms the word w appears with; and n is the total number of terms considered and it expresses the weight as the probability that the word w might be a term context word.

NC-value is formally given by Equation 3.

$$NCv(a) = 0.8 \cdot Cv(a) + 0.2 \cdot \sum_{b \in C_a} f_a(b) \cdot weight(b). \quad (3)$$

where, a is a candidate term; C_a is the set of distinct context words of a; b is a word from C_a; $f_a(b)$ is the frequency of b as a term context word of a; and $weight(b)$ is the weight of b as a term context word. The two components of the *NC-value*, i.e., *C-value* and the context information factor, have been assigned the weights 0.8 and 0.2, respectively. These two coefficients were derived empirically (Frantzi, Ananiadou, & Mima 2000).

Experiments in (Frantzi, Ananiadou, & Mima 2000; Milios *et al.* 2003) show that *C-value*/*NC-value* method performs well on a variety of special text corpora. In particular, with linguistic filter 2 `(Adjective|Noun)+Noun` (one or more adjectives or nouns followed by one noun), *C-value*/*NC-value* method extracts more terms than with linguistic filter 1 `Noun+Noun` (one or more nouns followed by one noun) without much precision loss. For example, terms such as *artificial intelligence* and *natural language processing* will be extracted by linguistic filter 2. Hence, in our work, we experiment with both linguistic filters to extract terms from a Web site. Finally, the resulting keyterms from each linguistic filter are used to extract key sentences to summarize the target Web site.

Keyterm Identification The candidate term list C (ranked by *NC-value*) of a Web site contains some noun phrases (e.g., *Web page*, *Web site*, *home page*, *credit card*, *privacy statement*), which appear frequently in Web sites. These noun phrases are not relevant to the content of the Web sites and hence must be treated as stop words. We experimented with 60 DMOZ Web sites and identified a stop list, L, of 51 noun phrases. The candidate term list C is filtered through the noun phrase stop list L, and only the top 25 terms are selected as keyterms.

Experiments and Evaluation

In this section, we discuss how to evaluate and compare the quality of keyword-based and keyterm-based summaries.

KWB and KTB Summaries

In our work, both keyword-based (KWB) and keyterm-based (KTB) approaches are used to generate summaries for 20 DMOZ Web sites (in four subdirectories), which are selected randomly and are of varying size and focus (Zhang, Zincir-Heywood, & Milios 2003).

We denote KTB summaries based on terms extracted by linguistic filter 1 as KTB_1 and KTB summaries based on terms extracted by linguistic filter 2 as KTB_2. Each KWB summary consists of the top 25 keywords and the top 5 key sentences. Each KTB (KTB_1 or KTB_2) summary consists of the top 25 keyterms and the top 5 key sentences. Table 1 gives an example of the KTB_2 summary for the Software Engineering Institute Web site[6].

Part I. top 25 keyterms
engineering institute, software engineering institute, software engineering, product line, software architecture, carnegie mellon university, capability maturity, capability maturity model, carnegie mellon, maturity model, software process, mellon university, process improvement, development center, system component, software development, software system, reference architecture, personal software process, software product line, capability maturity model integration, target system, design decision, software product, team software process
Part II. top 5 key sentences
1. The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. 2. The Software Engineering Institute (SEI) sponsors, co-sponsors, and is otherwise involved in many events throughout the year. 3. The Software Engineering Institute offers a number of courses and training opportunities. 4. The Software Engineering Institute (SEI) helps organizations and individuals to improve their software engineering management practices. 5. The SEI provides the technical leadership to advance the practice of software engineering so the DoD can acquire and sustain its software-intensive systems with predictable and improved cost, schedule, and quality.

Table 1: An example of KTB_2 summary of the SEI site.

Summary Evaluation

In this subsection, we describe how to compare the quality of KWB summaries with that of KTB summaries. Evaluation of automatically generated summaries often proceeds in either of two main modes, *intrinsic* and *extrinsic* (Mani *et al.* 1999). Intrinsic evaluation compares automatically generated summaries against a gold standard (ideal summaries), which is very expensive to construct. Extrinsic evaluation measures the utility of automatically generated summaries in performing a particular task, e.g., (Lu *et al.* 2001; Milios *et al.* 2003; Turney 2003).

In this work, we apply the extrinsic evaluation to investigate how well KWB and KTB summaries can help Web users in understanding the main contents of target Web sites. Four human subjects who specialize in Web domain research (such as finding related Web pages, citation graph analysis) are asked to read and evaluate summaries. In order to avoid bias towards a particular type of summary, each subject reads 5 KWB, 5 KTB_1 and 5 KTB_2 summaries, which

[6] http://www.sei.cmu.edu

are different from the summaries assigned to other subjects. Then they judge the relatedness of key phrases and key sentences to the essential topics covered in the Web site as follows:

1. Browse the Web site for a sufficient time in order to extract two essential topics from each test Web site.
2. Read KWB and KTB summaries and rank each **summary item** (i.e. keyword, keyterm, or key sentence) into *good*, *fair* or *bad* using the following rules:
   - If it is pertinent to both of the two topics of the Web site, rank it *good*.
   - If it is strongly pertinent to one of the two topics, rank it *good*.
   - If it is pertinent to one of the two topics, rank it *fair*.
   - If it is not pertinent to any of the two topics at all, rank it *bad*.
3. Count the number of *good*/*fair*/*bad* items in each summary.

Let n_g, n_f, and n_b be the number of good, fair, and bad summary items, respectively. For example, in the summary shown above, the two essential topics for the Web site could be: 1) Software Engineering Institute at Carnegie Mellon University, and 2) software engineering management and practice. And in the KTB_2 summary, there are 23 good, 2 fair, and 0 bad keyterms; and 5 good, 0 fair, and 0 bad key sentences.

Further we assign weights 1.0, 0.5 and 0 to good, fair, and bad summary items, respectively. Let kp be the quality value of key phrases and ks be the quality value of key sentences in each summary, respectively. These values are formally represented by Equation 4.

$$kp, ks = \frac{1.0 \times n_g + 0.5 \times n_f + 0.0 \times n_b}{n_g + n_f + n_b}. \quad (4)$$

Finally let s be the quality value of KWB and KTB summaries. We assign equal weights (after experimentation) to key phrases and key sentences when calculating the summary value, which is formally represented by Equation 5.

$$s = 0.5 \times kp + 0.5 \times ks. \quad (5)$$

Figure 1 shows the quality values of key phrases from three different approaches. As we can see, key phrases in KTB_1 summaries achieve higher scores than those in KWB summaries in 11 out of 20 Web sites. Key phrases in KTB_2 summaries achieve higher scores than those in KTB_1 summaries in 12 out of 20 Web sites. This indicates that key phrases in KTB_2 summaries are generally better than those in KTB_1 summaries, which are further better than those in KWB summaries.

Figure 2 shows that key sentences in KTB_1 summaries outperform those in KWB summaries with 9 wins, 9 ties and only 2 losses, and that key sentences in KTB_2 summaries outperform those in KTB_1 summaries with 9 wins, 5 ties and 6 losses.

Figure 3 indicates that KTB_1 summaries are generally better than KWB summaries with 15 wins, 1 tie, and only 4 losses, and that KTB_2 summaries are generally better than KTB_1 summaries with 13 wins, 1 tie, and 6 losses.

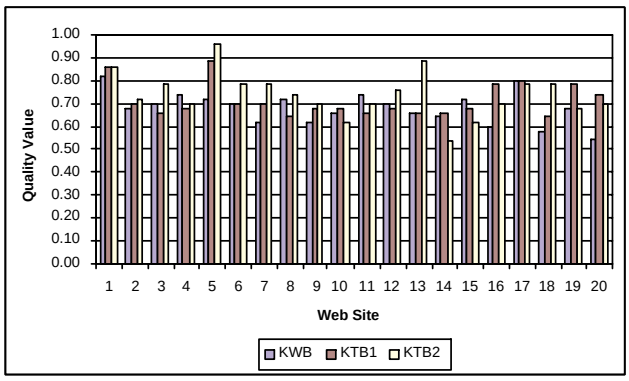

Figure 1: Comparison of quality values of key phrases in KWB summaries and KTB summaries of 20 test Web sites.

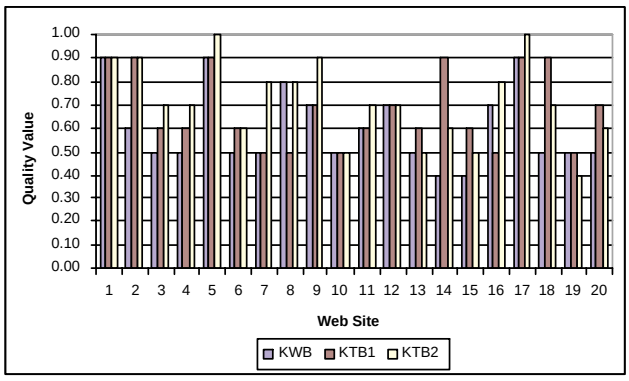

Figure 2: Comparison of quality values of key sentences in KWB summaries and KTB summaries of 20 test Web sites.

In order to statistically measure if the differences between summaries created by three methods are significant, we apply two-tail paired t-tests, which generally compares two different methods used for experiments carried in pairs.

Comparisons of the three methods via t-tests (confidence level $\alpha = 0.05$, $t_{\alpha,19}$ = 2.093) are summarized in Table 2, which shows that both KTB_1 and KTB_2 methods are significantly better than KWB method, and that there is no significant difference between KTB_1 method and KTB_2 method.

Method	KWB	KTB_1
KTB_1	$t_0 = 2.238$ $Pvalue < 0.040$	
KTB_2	$t_0 = 4.951$ $Pvalue < 0.001$	$t_0 = 1.378$ $Pvalue = 0.184$

Table 2: Pairwise t-test results for the three methods.

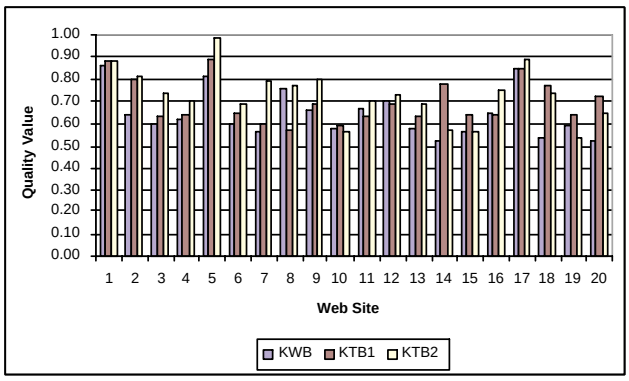

Figure 3: Comparison of quality values of KWB summaries and KTB summaries of 20 test Web sites.

Conclusion and Discussion

In this paper, we apply automatic term extraction techniques in a keyterm-based approach to automatic Web site summarization. Our approach relies on machine learning and natural language processing techniques to extract and classify narrative paragraphs from the Web site, from which keyterms are then extracted. Keyterms are in turn used to extract key sentences from the narrative paragraphs that form the summary, together with the top keyterms. We demonstrate that keyterm-based summaries are significantly better than former keyword-based summaries.

Future research involves several directions: 1) Use of machine learning in setting the relative weights for keywords from narrative, anchor and special text; 2) Application of the keyterm-based approach to summarizing the Web pages returned by a query to a search engine, after clustering the returned pages; 3) Integration of keyword- and keyterm-based methods in Web document corpus summarization.

Acknowledgements This research has been supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

Amitay, E., and Paris, C. 2000. Automatically Summarising Web sites: Is There a Way Around It? In *Proceedings of the Ninth ACM International Conference on Information and Knowledge Management*, 173–179.

Berger, A., and Mittal, V. 2000. OCELOT: A System for Summarizing Web Pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 144–151.

Brill, E. 1992. A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 152–155.

Buyukkokten, O.; Garcia-Molina, H.; and Paepcke, A. 2001. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In *Proceedings of Tenth International World Wide Web Conference*, 652–662.

Chuang, W., and Yang, J. 2000. Extracting Sentence Segments for Text Summarization: A Machine Learning Approach. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 152–159.

Delort, J.; Bouchon-Meunier, B.; and Rifqi, M. 2003. Enhanced Web Document Summarization using Hyperlinks. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, 208–215.

Fox, C. 1992. Lexical Analysis and Stoplists. In Frakes, W., and Baeza-Yates, R., eds., *Information Retrieval: Data Structures and Algorithms*, 102–130.

Frantzi, K.; Ananiadou, S.; and Mima, H. 2000. Automatic Recognition of Multi-word Terms: the *C-value*/*NC-value* Method. *International Journal on Digital Libraries* 3(2):115–130.

Goldstein, J.; Kantrowitz, M.; Mittal, V.; and Carbonell, J. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 121–128.

Goldstein, J.; Mittal, V.; Carbonell, J.; and Callan, J. 2000. Creating and Evaluating Multi-document Sentence Extract Summaries. In *Proceedings of the Ninth ACM International Conference on Information and Knowledge Management*, 165–172.

Lu, W.; Janssen, J.; Milios, E.; and Japkowicz, N. 2001. Node Similarity in Networked Information Spaces. Technical Report CS-2001-03, Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada.

Mani, I.; Firmin, T.; House, D.; Klein, G.; Sundheim, B.; and Hirschman, L. 1999. The TIPSTER SUMMAC Text Summarization Evaluation. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 77–85.

Milios, E.; Zhang, Y.; He, B.; and Dong, L. 2003. Automatic Term Extraction and Document Similarity in Special Text Corpora. In Kešelj, V., and Endo, T., eds., *Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics*, 275–284.

Turney, P. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval* 2(4):303–336.

Turney, P. 2003. Coherent Keyphrase Extraction via Web Mining. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 434–439.

Witten, I.; Paynter, G.; Frank, E.; Gutwin, C.; and Nevill-Manning, C. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, 254–256.

Zhang, Y.; Zincir-Heywood, N.; and Milios, E. 2003. Summarizing Web Sites Automatically. In Xiang, Y., and Chaib-draa, B., eds., *Advances in Artificial Intelligence, Proceedings of the Sixteenth Conference of the Canadian Society for Computational Studies of Intelligence*, 283–296.