A Comparison of Word- and Term-based Methods for Automatic Web Site Summarization

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

ABSTRACT

Automatic Web site summarization is an effective means of making the content of a web site easily accessible to Web users. We demonstrate that a content-based approach to summarization, which is based on keyword and key sentence extraction from narrative text, is able to generate summaries that are as informative as human authored summaries. This work is directed towards summary generation based on n-gram terms extracted by 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 the keyterm-based method is significantly better than the keyword-based method.

Keywords

automatic Web site summarization, machine learning, text classification, keyword extraction, automatic term extraction, key sentence extraction

1. INTRODUCTION

The information overload problem on the World Wide Web has brought Web users great difficulty in information seeking. 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 key topics 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.

Web document summarization techniques are derived from traditional text summarization techniques. Existing text summarization systems generate summaries automatically by either "extraction" or "abstraction". Extraction-based systems [6, 11] analyze source documents using techniques such as frequency analysis to determine significant sentences based on features such as the density of keywords [22] and rhetorical relations [16] in the context. "Abstraction" [3], 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 [10].

Unlike traditional documents with well-structured discourse, Web documents are often not well-structured, and have more diverse contents than narrative text, such as bullets, short sentences, emphasized text and anchor text associated with hyperlinks. Hence, summarizing Web documents differs from traditional text summarization. Research work in [22] has shown that identification of narrative text for summary generation is a key component of Web site summarization.

The aim of this paper is to extend the keyword-based method described in [22] by using automatically extracted n-gram terms in identifying key sentences in the narrative text of a Web site. Keyterms and key sentences are selected to be part of a Web site summary. The keyterm-based summaries for a list of test Web sites are experimentally compared with the keyword-based summaries [22].

The rest of the paper is organized as follows. Section 2 reviews published Web document summarization approaches, and Section 3 explains how to generate term-based summaries. Section 4 discusses the design of our experiments and shows the evaluation results. Finally, Section 5 concludes our work and describes future research directions.

2. RELATED WORK

Research on Web document summarization to date has either been *content-based* or *context-based*. Content-based systems [3, 5] analyze the contents and extract the significant sentences to construct a summary, while context-based systems [2, 7] 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 [3] 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. [5] 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 best performance.

Amitay and Paris [2] propose an innovative 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 sending queries of the type "link:URL" to search engines. Experiments show that on average users prefer this system to search engines. Delort et al. [7] 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.

Zhang et al. [22] extend single Web document summarization to the summarization of complete Web sites. The "Keyword/Summary" idea of [5] 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, the top 10 bigrams and the top 5 key sentences. Since Web docu-

WWW2004, May 17–22, 2004, New York, NY USA.
ACM xxx.xxx.

ments 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, and then extracts key phrases, i.e. keywords and key bigrams, 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 key phrases. The evaluation shows that the automatically generated summaries are as informative as human authored summaries (e.g. DMOZ[1] summaries).

3. 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 n-gram terms automatically and use identified keyterms to summarize a Web site based on the framework of the keyword-based approach.

3.1 Keyword-based AWSS

Zhang et al. [22] 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.

3.1.1 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, namely level (depth) one. The number 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 Web sites (identified in DMOZ). 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.

3.1.2 Plain Text Extraction

After the URLs of the Web pages have been collected, plain text is extracted from these Web pages by the text browser *Lynx*[2], which was found to outperform several alternative text extraction tools such as *HTML2TXT*[3] by Thomas Sahlin and *html2txt*[4] by Gerald Oskoboiny. Another advantage of Lynx is that it has a built-in mechanism to segment text extracted from a Web page into text paragraphs automatically.

3.1.3 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 identified and discarded. Second, among the long paragraphs, narrative ones provide more coherent and meaningful contents than non-narrative ones, so additional criteria are defined to classify *long* paragraphs into *narrative* or *non-narrative*. Only narrative paragraphs are used in summary generation.

3.1.3.1 Long Paragraph Classification.

The decision tree learning program C5.0[5] is applied to generate decision tree rules for filtering out *short* paragraphs, which are observed to be too short (in terms of number of words, number of characters, etc.) for summary generation, e.g., *This Web page is maintained by David Alex Lamb of Queen's University. Contact: dalamb@spamcop.net.*

For this purpose, a total of 700 text paragraphs is extracted from 100 Web pages (collected from 60 DMOZ Web sites). Statistics of three attributes *length of paragraph*, i.e. total number of characters including punctuation, *number of words*, and *number of characters in all words* (without punctuation), are recorded for each text paragraph. Then each text paragraph is manually labelled as *long* or *short*, and C5.0 is used to construct a classifier, *LONGSHORT*, for this task.

The training set consists of 700 instances. Each instance consists of the values of three attributes and the associated class. The resulting decision tree is simple: if the number of words in a paragraph is less than 20, then it is a *short* paragraph, otherwise it is classified as *long*. Among the 700 cases, there are 36 cases misclassified, leading to an error of 5.1%. The cross-validation of the classifier LONGSHORT shows a mean error of 5.9%, which indicates the classification accuracy of this classifier.

3.1.3.2 Narrative Paragraph Classification.

Informally, whether a paragraph is narrative or non-narrative is determined by the coherence of its text. Analysis of part-of-speech patterns has proved to be effective in several Web-based applications such as query ambiguity reduction [1] and question answering [18]. It is hypothesized that the frequencies of the part-of-speech tags of the words in a paragraph contain sufficient information to identify the paragraph as narrative or non-narrative. To test the hypothesis, a training set is generated as follows: First, 1000 Web pages are collected from 60 DMOZ Web sites, containing a total of 9763 text paragraphs identified by Lynx, among which 3243 paragraphs are classified as long. Then, the part-of-speech tags for all words in these paragraphs are computed using a rule-based part-of-speech tagger [4].

After part-of-speech tagging, attributes of percentage values of 32 part-of-speech tags [4] are extracted from each paragraph. Two more attributes are added to this set, *number of characters* and *number of words* in the paragraph. Then each paragraph is manually labelled as *narrative* or *non-narrative*. Finally, a C5.0 classifier it NARRATIVE is trained on the training set of 3243 cases.

Among the 3243 cases, about 63.5% of them are following this rule: if the percentage of *Symbols* is less than 6.8%, and the percentage of *Preposition* is more than 5.2%, and the percentage of *Proper Singular Nouns* is less than 23.3%, then this paragraph is *narrative*. There are 260 cases misclassified, leading to an error of 8.0%. The cross-validation of the classifier NARRATIVE shows a mean error of 11.3%, which indicates the predictive accuracy of this classifier.

3.1.4 Key Phrase Extraction

Traditionally, key phrases are extracted from the documents in order to generate a summary. In this work, a key phrase can either be a keyword (single word) or a key bigram (two-word phrase). Based on such key phrases, the most significant sentences, which

[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

best describe the document, are retrieved.

Key phrase extraction from a body of text relies on an evaluation of the importance of each candidate phrase [5]. For Web site summarization, a candidate phrase is considered as key phrase if and only if it occurs very frequently in the Web pages of the site, i.e., the total frequency of occurrences is very 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 key phrases.

3.1.4.1 Keyword Extraction.

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 basically the same. If a word happens to appear in an anchor text, which is also italicized, then it is counted twice. This in turn, indirectly, gives more weight to this word. Moreover, a standard set of 425 stop words (*a, about, above, ...*) [8] is discarded in this stage.

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. Next, each candidate keyword is labelled manually 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 of these candidate keywords, a C5.0 classifier *KEYWORD* is constructed.

Among the total 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.

3.1.4.2 Key Bigram Extraction.

It is observed that bigrams which consist of two of the top 100 candidate keywords from each Web site exist with high frequency. Such a bigram could be useful as part of the description of the Web site. Thus, a similar approach to automatic keyword extraction is developed to identify key bigrams of the Web site.

The algorithm heuristically combines any two of the top 100 candidate keywords and searches for these bigrams in collocation over narrative text, anchor text and special text. Then these bigrams are sorted by frequency and the top 30 are selected as candidate key bigrams. A C5.0 classifier *KEYBIGRAM* is constructed based on frequency statistics and tag features of 1360 candidate bigrams, which are extracted from 60 DMOZ Web sites. The C5.0 classifier KEYBIGRAM is similar to the KEYWORD classifier except that it has two part-of-speech tags, one for each component word.

Once the decision tree rules for determining key bigrams have been built, they are applied to automatic key bigram extraction from the Web pages of a Web site. The top 10 key bigrams (ranked by overall frequency) for each site are kept as part of the summary. Then the frequency of the candidate keywords forming the top 10 key bigrams is reduced by subtracting the frequency of the corresponding key bigrams. Then candidate keywords of the Web site are classified into keyword or non-keyword by applying the KEYWORD classifier. Finally, the top 25 keywords (ranked by frequency) are kept as part of the summary. It is observed that 40% to 70% of keywords and 20% to 50% of key bigrams appear in the home page of a Web site.

3.1.5 Key Sentence Extraction

Once the key phrases are identified, the most significant sentences for summary generation can be retrieved from all narrative paragraphs based on the presence of key phrases [6]. The significance of a sentence is measured by calculating a weight value, which is the maximum of the weights for clusters within the sentence. A cluster is defined as a list of words which starts and ends with a key phrase and less than 2 non-key-phrases must separate any two neighboring key phrases [5]. A cluster's weight is computed by adding the weights of all key phrases within the cluster, and dividing this sum by the total number of key phrases within the 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.

3.1.6 Summary Generation

The overall summary is formed by the top 25 keywords, the top 10 bigrams and the top 5 key sentences. These numbers are experimented and determined based on the fact that key bigrams are more informative than keywords and key sentences are more informative than key bigrams, and the whole summary should fit in a single page. Table 1 shows the generated summary of the Software Engineering Institute (SEI) Web site[6].

Table 1: An example of keyword-based summary.

Part I. top 25 keywords
system, product, information, organization, institute, architecture, program, course, research, carnegie, defense, development, team, department, term, component, sponsor, process, design, management, education, method, technology, service, acquisition
Part II. top 10 key bigrams
software engineering, mellon university, software process, development center, software architecture, maturity model, software product, staff page, process improvement, contact information
Part III. top 5 key sentences
1. Explore the topics listed on the left for more information about software engineering practices, SEI projects, and software engineering. 2. 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. 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 Software Engineering Institute (SEI) sponsors, cosponsors, and is otherwise involved in many events throughout the year.

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

3.2 Keyterm-based AWSS

The key phrase identification in [22] is based on phrase frequency analysis against three different categories of text, narrative text, anchor text, and special text. Bigrams are identified in a heuristic way, i.e., combining any two of the top 100 candidate keywords to form a candidate bigram and determining if it is a true bigram using decision tree rules. This might not work well if component words of informative terms do not appear as candidate keywords and consequently such terms fail to be extracted. Moreover, the method is unable to extract key phrases consisting of three or more component words. Since terms (phrases with two or more words) are more informative than single words, the authors aim to extract n-gram ($n \geq 2$) 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 keyword-based approach except in the key phrase extraction phase. In the keyterm-based method, n-gram 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.

3.2.1 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 [17]. Traditional approaches to automatic term extraction were focused on information-theoretic approaches based on mutual information in detecting collocations [15]. Recently more effective systems have been developed. Krulwich and Burkey use heuristic rules such as the use of acronyms and the use of italics to extract key phrases from a document for use as features of automatic document classification [12]. 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 [19]. Witten et al. propose a system called KEA which builds a Naive Bayes learning model using training documents with known key phrases, and then uses the model to find key phrases in new documents [21]. 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* [9] to extract n-gram terms from a Web site automatically. This term extraction approach consists of both linguistic analysis (linguistic filter, part-of-speech tagging [4], and stop-list) and statistical analysis (frequency analysis, *C-value*/*NC-value*).

Experiments in [9, 17] 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 Web site as described in Section 3.1.5.

3.2.2 Keyterm Identification

The candidate term list C (ranked by *NC-value*) of a Web site generated by *C-value*/*NC-value* method contains some noun phrases (e.g. *Web page*), 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 81 noun phrases (e.g., *Web site*, *home page*, *credit card*, *privacy statement*, ...). The candidate term list C is filtered through the noun phrase stop list L, and only the top 25 terms are selected as keyterms. The choice of number 25 is based on the assumption that the informativeness of the top 25 keyterms is comparable to that of 25 keywords and 10 key bigrams in the keyword-based approach.

4. EXPERIMENTS AND EVALUATION

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

4.1 KWB and KTB Summaries

In our work, both keyword-based (KWB) and keyterm-based (KTB) approaches are used to generate summaries for 20 test Web sites used in [22]. 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, the top 10 key bigrams and the top 5 key sentences as shown in Table 1. Each KTB (KTB_1 or KTB_2) summary consists of the top 25 keyterms and the top 5 key sentences. Table 2 gives an example of KTB_1 and KTB_2 summaries for the Software Engineering Institute Web site.

As we can see in Tables 1 and 2, the heuristic bigram extraction method can catch key phrases such as *software engineering* and *software architecture*. However, it cannot extract terms consisting of more than two words such as *software engineering institute*, which in this case is a very important term to be extracted. Another drawback is that it also extracts untrue keyterms such as *staff page* and *contact information* which will be filtered out by noun phrase stop list in automatic term extraction method.

Also it is observed that there are 19 matches out of 25 keyterms in KTB_1 and KTB_2 summaries, but the order of these keyterms is significantly different from each other according to their weights (*NC-values*). Among the top 5 key sentences generated based on these key phrases, there are 4 matches out of 5 between any two of the three summaries, but again the order of sentences is different due to its significance value.

4.2 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*. Intrinsic evaluation compares automatically generated summaries against a gold standard (ideal summaries), which is very hard to construct. Extrinsic evaluation measures the utility of automatically generated summaries in performing a particular task (e.g., classification) [14]. In this work, however, we evaluate the quality of KWB and KTB summaries in a different way which has been extensively used in related work [13, 17, 20]. Domain experts are asked to read 20 summaries and 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, key bigram, keyterm, or key sentence) into *good*, *fair* or *bad* using the following rules:

Table 2: An example of KTB_1 and KTB_2 summaries.

Part I. KTB_1 top 25 keyterms	**Part I. KTB_2 top 25 keyterms**
engineering institute, software engineering, software engineering institute, product line, carnegie mellon, development center, software architecture, software development, software product, software product line, software process, system component, process improvement, design decision, coordination pattern, reference architecture, software system, coordination protocol, infrastructure capability, application developer, whiteboard course attendees stryker infantry carrier vehicle, system architecture, capability maturity, target system, risk management	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. KTB_1 top 5 key sentences	**Part II. KTB_2 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. 2002 SEI Annual Report Published The online version of the Annual Report of the Software Engineering Institute (SEI), reporting on fiscal year 2002, is available at http://www.sei.cmu.edu/annual-report/. 3. The Software Engineering Institute (SEI) helps organizations and individuals to improve their software engineering management practices. 4. The Software Engineering Institute (SEI) sponsors, cosponsors, and is otherwise involved in many events throughout the year. 5. The Software Engineering Institute offers a number of courses and training opportunities.	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, cosponsors, 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.

- 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 example 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. In the KWB summary, there are 13 good, 10 fair, and 2 bad keywords; 6 good, 2 fair, and 2 bad key bigrams; and 4 good, 1 fair, and 0 bad key sentences. Details of KWB, KTB_1, and KTB_2 summary item numbers are listed in Table 3.

The average number of good, fair, and bad summary items per Web site summary is listed in the bottom line of Table 3. Related research in [20] defines *acceptable* terms as good and fair terms. The percentage of acceptable terms, p, is formally represented by Equation 1.

$$p = \frac{n_g + n_f}{n_g + n_f + n_b}. \quad (1)$$

The values of p in KWB, KTB_1, and KTB_2 summaries above are $\frac{13+8+6+2}{25+10} = 82.9\%$, $\frac{14+7}{25} = 84.0\%$, and $\frac{16+6}{25} = 88.0\%$, respectively.

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 KWB and KTB summaries, respectively. These values are formally represented by Equation 2.

$$kp, ks = \frac{1.0 \times n_g + 0.5 \times n_f + 0.0 \times n_b}{n_g + n_f + n_b}. \quad (2)$$

For example, the key phrase quality value kp for the KWB summary above is calculated as $\frac{1.0\times13+0.5\times10+1.0\times6+0.5\times2}{13+10+2+6+2+2} = 0.71$, and the key sentence quality value is $\frac{1.0\times4+0.5\times1}{4+1} = 0.90$.

Finally let s be the quality value of KWB and KTB summaries. We give equal weights to key phrases and key sentences when calculating the summary value, which is formally represented by Equation 3.

$$s = 0.5 \times kp + 0.5 \times ks. \quad (3)$$

Table 4 summarizes the quality values of 20 Web site summaries.

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 12 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 summary are generally better than those in KTB_1 summary, which are further better than those in KWB summary.

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

Table 3: Details of summary item numbers.

Method	**KWB**									**KTB$_1$**						**KTB$_2$**					
Item	25 keywords			10 key bigrams			5 key sentences			25 keyterms			5 key sentences			25 keyterms			5 key sentences		
Site	n_g	n_f	n_b	n_g	n_f	n_b	n_g	n_f	n_b	n_g	n_f	n_b	n_g	n_f	n_b	n_g	n_f	n_b	n_g	n_f	n_b
1	19	3	3	7	2	1	4	1	0	20	3	2	4	1	0	19	5	1	4	1	0
2	12	10	3	4	4	2	2	2	1	15	5	5	4	1	0	16	4	5	4	1	0
3	14	7	4	6	2	2	2	1	2	13	7	5	2	2	1	15	9	1	3	1	1
4	16	5	4	7	2	1	1	3	1	14	6	5	2	2	1	14	7	4	2	3	0
5	13	10	2	6	2	2	4	1	0	20	4	1	4	1	0	23	2	0	5	0	0
6	13	9	3	7	2	1	1	3	1	14	7	4	1	4	0	18	3	4	1	4	0
7	11	9	5	6	2	2	1	3	1	15	5	5	1	3	1	17	5	3	3	2	0
8	13	10	2	6	3	1	3	2	0	12	8	5	1	3	1	16	5	4	3	2	0
9	11	9	5	5	3	2	3	1	1	13	8	4	3	1	1	14	7	4	4	1	0
10	12	9	4	6	1	3	1	3	1	14	6	5	1	3	1	12	7	6	0	5	0
11	14	9	2	1	0	0	2	2	1	12	9	4	2	2	1	14	7	4	2	3	0
12	12	4	4	3	3	4	3	1	1	13	8	4	3	1	1	14	10	1	2	3	0
13	13	7	5	7	1	2	1	3	1	13	7	5	1	4	0	20	4	1	0	5	0
14	11	10	4	1	5	3	1	2	2	12	9	4	4	1	0	10	7	8	2	2	1
15	10	6	2	3	2	1	1	2	2	12	10	3	2	2	1	12	7	6	1	3	1
16	11	8	6	6	2	2	2	3	0	16	7	2	1	3	1	16	3	6	3	2	0
17	18	4	3	8	2	0	4	1	0	18	4	3	4	1	0	17	5	3	5	0	0
18	10	9	6	4	3	3	1	3	1	10	12	3	4	1	0	17	5	3	3	1	1
19	13	8	4	5	3	2	0	5	0	15	9	1	0	5	0	13	8	4	0	4	1
20	8	7	6	6	2	2	1	3	1	14	9	2	2	3	0	14	7	4	2	2	1
Average	13	8	4	6	2	2	1.9	2.2	0.9	14	7	4	2.3	2.2	0.5	16	6	3	2.4	2.3	0.3

Table 4: Quality values of KWB and KTB summaries.

Site	kp_w	kp_{t1}	kp_{t2}	ks_w	ks_{t1}	ks_{t2}	s_w	s_{t1}	s_{t2}
1	0.81	0.86	0.86	0.90	0.90	0.90	0.86	0.88	0.88
2	0.66	0.70	0.72	0.60	0.90	0.90	0.63	0.80	0.81
3	0.70	0.66	0.78	0.50	0.60	0.70	0.60	0.63	0.74
4	0.76	0.68	0.70	0.50	0.60	0.70	0.63	0.64	0.70
5	0.71	0.88	0.96	0.90	0.90	1.00	0.81	0.89	0.98
6	0.73	0.70	0.78	0.50	0.60	0.60	0.61	0.65	0.69
7	0.64	0.70	0.78	0.50	0.50	0.80	0.57	0.60	0.79
8	0.73	0.64	0.74	0.80	0.50	0.80	0.76	0.57	0.77
9	0.63	0.68	0.70	0.70	0.70	0.90	0.66	0.69	0.80
10	0.66	0.68	0.62	0.50	0.50	0.50	0.58	0.59	0.56
11	0.75	0.66	0.70	0.60	0.60	0.70	0.68	0.63	0.70
12	0.62	0.68	0.76	0.70	0.70	0.70	0.66	0.69	0.73
13	0.69	0.66	0.88	0.50	0.60	0.50	0.59	0.63	0.69
14	0.57	0.66	0.54	0.40	0.90	0.60	0.49	0.78	0.57
15	0.71	0.68	0.62	0.40	0.60	0.50	0.55	0.64	0.56
16	0.63	0.78	0.70	0.70	0.50	0.80	0.66	0.64	0.75
17	0.83	0.80	0.78	0.90	0.90	1.00	0.86	0.85	0.89
18	0.57	0.64	0.78	0.50	0.90	0.70	0.54	0.77	0.74
19	0.67	0.78	0.68	0.50	0.50	0.40	0.59	0.64	0.54
20	0.60	0.74	0.70	0.50	0.70	0.60	0.55	0.72	0.65
Average	0.683	0.713	0.739	0.605	0.680	0.715	0.644	0.697	0.727

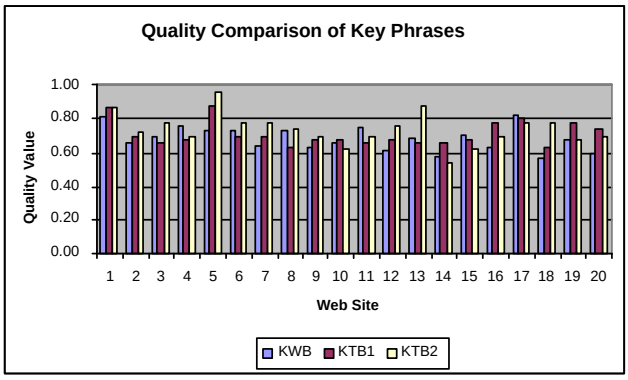

Figure 1: Comparison of quality values of key phrases in KWB summaries and KTB summaries of 20 test Web sites.

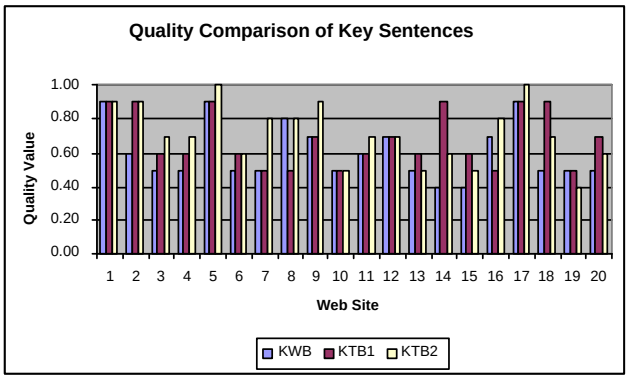

Figure 2: Comparison of quality values of key sentences in KWB summaries and KTB summaries of 20 test Web sites.

than KWB summaries with 16 wins and only 4 losses, and that KTB_2 summaries are generally better than KTB_1 summaries with 13 wins, 1 tie, and 6 losses.

In order to statistically measure if the differences between summaries created by three methods are significant, we apply a t-test analysis, which generally compares two different methods used for experiments carried in pairs. It is the difference between each pair of measurements which is of interest.

For example, when comparing the quality of KTB_1 summaries and KWB summaries, we have 20 pairs of quality values of summaries, s_{t1_i}, s_{w_i} (i = 1, 2, ..., 20), which are independent observations from the two samples in KTB_1 approach and KWB approach, respectively. Then the differences $d_i = s_{t1_i} - s_{w_i}$ (i = 1, 2, ..., 20) will be a sample of size n (n = 20) from a population with mean zero. Furthermore, if the populations, where the above two samples are drawn from, are approximately normally distributed, then the differences will also be approximately normally distributed. If the observed average difference is denoted by $\overline{d}$, the standard deviation of the observed differences by s_d, and the t-test statistic by t,

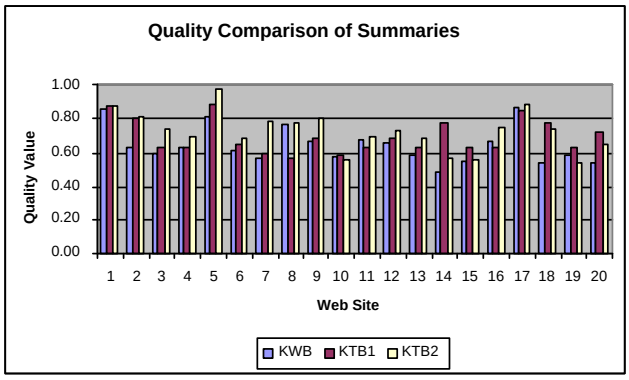

Figure 3: Comparison of quality values of KWB summaries and KTB summaries of 20 test Web sites.

then we have the following equations:

$$\overline{d} = \frac{\sum_{i=1}^{n} d_i}{n} \quad (4)$$

$$s_d^2 = \frac{\sum_{i=1}^{n} (d_i - \overline{d})^2}{n-1} \quad (5)$$

$$t = \frac{\overline{d}}{s_d/\sqrt{n}}. \quad (6)$$

The null hypothesis H_0 and the alternative hypothesis H_1 are given by:
$H_0 : d = 0$ (KTB_1 and KWB have the same performance), and
$H_1 : d > 0$ (KTB_1 is significantly better than KWB).

If H_0 is true, then the distribution of t will be a t-distribution with $n - 1$ degrees of freedom, as the estimate s_d is calculated from n differences.

From Table 4, we have: $\overline{d} = 0.053$, $s_d^2 = 0.011$, $s_d = 0.105$, $t = 2.238$.

By checking the t-table, we have $t_{0.05,19} = 2.093$. Since $t > t_{0.05,19}$, it's reasonable for us to reject the null hypothesis H_0, i.e., there is a significant difference between the quality values of summaries obtained from the two methods. More precisely, the KTB_1 approach performs significantly better than the KWB approach.

Comparisons of the three methods via t-tests are summarized in Table 5, 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.

Table 5: Pairwise t-test results for the three methods.

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

5. 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 a Web crawler that collects shallow web pages from a Web site and summarizes them off-line.

It applies 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.

- Use of machine learning in setting the relative weights for keywords or key bigrams from narrative, anchor and special text.
- Hierarchical summarization of complex web sites that may include a multitude of topics, for example Web sites of large organizations (e.g., government, university).
- Application of the keyterm-based approach to summarizing the Web pages returned by a query to a search engine, after clustering the returned pages.
- Integration of keyword-based and keyterm-based methods in Web document corpus summarization.
- Refinement of the evaluation process, including extrinsic evaluation.

Acknowledgements

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

6. REFERENCES

[1] J. Allan and H. Raghavan. Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 307–314, 2002.

[2] E. Amitay and C. Paris. Automatically summarising Web sites: is there a way around it? In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM'00)*, pages 173–179, 2000.

[3] A. Berger and V. Mittal. OCELOT: a system for summarizing Web pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 144–151, 2000.

[4] E. Brill. A simple rule-based part of speech tagger. In *3rd Conference on Applied Natural Language Processing (ANLP-92)*, pages 152–155, 1992.

[5] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: text summarization for Web browsing on handheld devices. In *Proceedings of 10th International World Wide Web Conference*, pages 652–662, 2001.

[6] W. Chuang and J. Yang. 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 (SIGIR'00)*, pages 152–159, 2000.

[7] J. Delort, B. Bouchon-Meunier, and M. Rifqi. Enhanced Web document summarization using hyperlinks. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, pages 208–215, 2003.

[8] C. Fox. Lexical analysis and stoplists. In *W. Frakes and R. Baeza-Yates, editors, Information Retrieval: Data Structures and Algorithms*, pages 102–130, Englewood Cliffs, NJ: Prentice Hall, 1992.

[9] K. Frantzi, S. Ananiadou, and H. Mima. Automatic recognition of multiword terms. *International Journal of Digital Libraries*, 3(2):117–132, 2000.

[10] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. 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 (SIGIR'99)*, pages 121–128, 1999.

[11] J. Goldstein, V. Mittal, J. Carbonell, and J. Callan. Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM'00)*, pages 165–172, 2000.

[12] B. Krulwich and C. Burkey. Learning user information interests through the extraction of semantically significant phrases. In *M. Hearst and H. Hirsh, editors, AAAI 1996 Spring Symposium on Machine Learning in Information Access*, Califonia: AAAI Press, 1996.

[13] W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node similarity in networked information spaces. *Technical Report CS-2001-03, Faculty of Computer Science, Dalhousie University*, September 2001.

[14] I. Mani, T. Firmin, D. House, G. Klein, B. Sundheim, and L. Hirschman. The TIPSTER SUMMAC text summarization evaluation. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 77–85, 1999.

[15] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press Cambridge, Massachusetts, 1999.

[16] D. Marcu. From discourse structures to text summaries. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, 1997.

[17] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic term extraction and document similarity in special text corpora. In *Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics (PACLing'03)*, pages 275–284, 2003.

[18] D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering on the Web. In *Proceedings of the 11th International World Wide Web Conference (WWW'02)*, pages 408–419, 2002.

[19] P. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

[20] P. Turney. Coherent keyphrase extraction via Web mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 434–439, 2003.

[21] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL'99)*, pages 254–256, 1999.

[22] Y. Zhang, N. Zincir-Heywood, and E. Milios. Summarizing Web sites automatically. In *Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI'03)*, pages 283–296, 2003.