World Wide Web Site Summarization

Yongzheng Zhang, Nur Zincir-Heywood, Evangelos Milios

Faculty of Computer Science,

Dalhousie University,

Halifax, N.S., Canada B3H 1W5

Tel: (902) 494 7111, Fax: (902) 492 1517

{*yongzhen, zincir, eem*}*@cs.dal.ca*

August 28, 2003

Abstract

Summaries of Web sites help Web users get an idea of the site contents without having to spend time browsing the sites. Currently, manually constructed summaries of Web sites by volunteer experts are available, such as the DMOZ Open Directory Project. This research is directed towards automating the Web site summarization task. To achieve this objective, an approach which applies machine learning and natural language processing techniques is developed to summarize a Web site automatically. The information content of the automatically generated summaries is compared, via a formal evaluation process involving human subjects, to DMOZ summaries, home page browsing and time-limited site browsing, for a number of academic and commercial Web sites. Statistical evaluation of the scores of the answers to a list of questions about the sites demonstrates that the automatically generated summaries convey the same information to the reader as DMOZ summaries do, and more information than the two browsing options.

1 Introduction

As the size and diversity of the World Wide Web grow, vast amounts of online information are accumulating at an incredible rate, leading to "information overload" [27]. Search engines such as Google[1] have been developed to help Web users quickly find the information they need. However, users are often unclear about the information they are looking for and consequently they give poor query formulations. Moreover, search engines typically return too many documents, especially if the query pertains to popular topics on the Web [17]. Manually reviewing long lists of returned documents is a time consuming process.
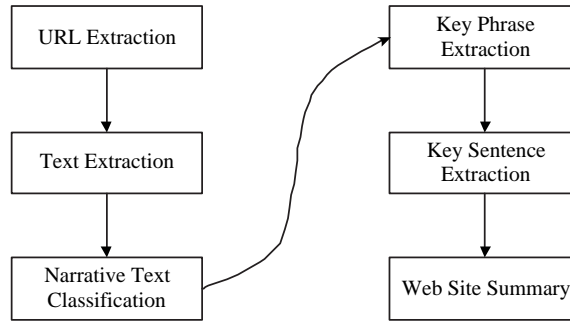
---

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

Figure 1: Web site summarization process

On the other hand, the size and complexity of Web sites continue to grow [20], while their design is not keeping up with their complexity. This makes it often difficult for Web users to skim over a Web site and get an idea of its contents. Availability of Web site summaries would alleviate this problem.

Currently, manually constructed summaries of Web sites by volunteer experts are available, such as the DMOZ Open Directory Project[2]. These human-authored summaries give a concise and effective description of popular Web sites. However, they are subjective, expensive to build and maintain, and not available on demand.

In this paper, our objective is to automate the summarization of Web sites. A summary is produced in a sequence of stages as indicated in Figure 1. First, a given number of Web pages is collected from a given Web site via a breadth-first search starting at the home page. Second, plain text is extracted from these pages and partitioned into text paragraphs by the text browser *Lynx*. In the third step, *short* paragraphs are first filtered out, then *long* paragraphs are classified into *narrative* or *non-narrative*. Machine learning based on features extracted by shallow natural language processing is used to generate classifiers for these problems. Fourth, classifiers are trained to extract key-phrases from narrative text, anchor text and special text (e.g., italic text). These classifiers learn the significance of each category of key-phrases. After key-phrases are extracted, narrative paragraphs are scanned to extract the most significant sentences, as the ones containing a high density of key-phrases. Finally, a summary is generated, which consists of top 25 key-words, top 10 key-terms and top 5 key-sentences.

In Section 2 of the paper, we review current literature in automatic text summarization. In Sections 3 and 4 we discuss the extraction of text from Web sites and its classification as narrative. In Section 5 we explain how to extract key-words and key-terms from narrative text. In Section 6 we describe the extraction of key-sentences. In Section 7 we discuss the design of our experiments and the evaluation results, comparing our approach with human authored DMOZ summaries, home page browsing and time-limited site browsing, for a number of academic and commercial Web sites. In Section 8, we conclude our work and describe future research directions.

---

[2]http://dmoz.org

2 Automatic Text Summarization

The technology of automatic text summarization is maturing and may provide a solution to the information overload problem [25, 27]. Automatic text summarization produces a concise summary of source documents. The summary can either be a *generic* summary [41], which shows the main topics and key contents covered in the source text, or a *query-relevant* summary [5], which locates the contents pertinent to user's seeking goals [15].

Research in automatic text summarization area dates back at least to 1958, when Luhn [24] proposed a simple approach which extracts significant sentences to form a summary based on features such as average term frequency and sentence location. Since then, more approaches have been proposed, either by *abstraction* or *extraction* of important text.

The goal of abstraction [4] is to understand the text using knowledge-based methods and compose a coherent summary comparable to a human authored summary. This is very difficult to achieve with current natural language processing techniques [15]. An easier alternative, extraction, has been the focus of automatic text summarization research in recent years [18, 21]. Extraction systems [9, 12, 16, 39] analyze a source document using techniques derived from information retrieval (e.g. frequency analysis and keyword identification) to determine significant sentences, that constitute the summary. The significance of a sentence is determined by features such as the density of keywords [41] and rhetorical relations [28] in the context. Chuang and Yang [9] propose an approach which generates a summary automatically. First, sentences are broken into segments by special cue phrases. Next, each segment is represented by using a set of pre-defined features, both unstructured (e.g. title words) and structured (e.g. rhetorical relations). Finally, machine learning algorithms are applied to the feature set to extract the most important sentence segments for summary inclusion. Post-processing of the extracted sentences has been used to produce succinct summaries without redundant information. For example, clustering has been applied to find clusters of closely related sentences, and only "core" sentences from all the clusters are used to form the output summary [19]. As a second example, the identification of named entities can help the system rephrase the pronouns used in order to create a meaningful summary [30].

Evaluation of automatically generated summaries can proceed in either of two different 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) [26, 37]. Extrinsic evaluation is also called task-based evaluation and it has become increasingly popular recently [33]. In the following two subsections, we will review literature related more closely to this paper, namely multi-document summarization, and summarization of Web pages.

2.1 Multi-document Summarization

Multi-document summarization (MDS) is an extension of single-document summarization into collections of related documents [25]. Multi-document summaries can save users significant time in reading relevant text documents or browsing Web sites. Many of the single-document summarization techniques can also be used in multi-document summarization. However, issues such as *anti-redundancy* and *cohesion and coherence* become critical in MDS [16, 23]. Moreover, multi-document summarization lacks standard procedures and methodologies for evaluation, in contrast to single-document summarization task [36].

The National Institute of Standards and Technology (NIST) sponsored the Document Understanding Conference[3] starting in 2001, which aims towards both providing standard training and test document collections (mostly news articles) which can be shared among the research community and evaluations in single- and multi-document summarization for the conference participants [23].

Current MDS systems often apply a two-phase process, i.e., *topic identification* and *summary generation*. In the first phase, main topics (or events) covered in the multiple source documents are identified. Documents regarding the same topic (or event) with variations in presentation are put into the same set. Then each set of closely related documents is used to produce representative passages for the final summary by extraction or by reformulation [19, 38]. Radev et al. [33] present a MDS system called MEAD, which first uses modified TF-IDF measure to form clusters of documents on the same topic, and then uses centroids of the clusters to identify which sentences are most likely to be relevant to the cluster topic, rather than individual articles. Evaluation demonstrates that summaries generated by MEAD are as good as human created summaries. Stein et al. [38] propose a different approach which first summarizes single documents and groups summaries in clusters, then selects representative passages from clusters, and finally organizes passages into a coherent summary. McKeown et al. [30] introduce a system which first identifies the type of document sets, i.e. single-event, person-centered (or biographical), or multi-event, and then summarizes them accordingly.

2.2 Summarizing Web Pages

Summarization of Web pages has been either *context-based* or *content-based*. Context-based systems [3, 11] analyze and summarize the context of a Web document (e.g. brief content descriptions from search engine results) instead of its contents. Content-based systems [4, 8] derive from traditional text summarization techniques. The great challenge in Web page summarization is the diversity of contents and the frequent lack of a well-defined discourse structure compared to traditional text [4]. Approaches based on implicit document association (rhetorical relation) analysis [28] are difficult to apply to Web page summarization.

Amitay and Paris [3] propose an innovative approach, which relies on the hypertext structure and the way information is described using it. Instead of analyzing the Web page itself, this approach collects the context of the document by sending queries of the type "link:URL" to search engines. Text units which

---

[3]http://duc.nist.gov

contain the link to the target Web page are then extracted. Finally, an automatic filter is used to select the best description for the Web page (URL). Single-sentence sized coherent textual snippets are generated and presented to the user together with results from search engines Google and AltaVista[4]. The experiments show that on average users prefer the system to search engines. Delort et al. [11] address three important issues, *contextualization*, *partiality*, and *topicality* faced by any context-based summarizer and propose two algorithms whose efficiency depends on the size of the text contents and the context of the target Web page.

Berger and Mittal [4] propose a system called OCELOT, which applies standard statistical models (in particular, 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. [8] 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 a summary for the unit. They test the performance of these methods by asking human subjects to perform specific tasks using each method, and conclude that the combined *Keyword/Summary* method provides the best performance in terms of access times and number of pen actions on the hand held devices.

In this paper, we extend single Web document summarization to the summarization of complete Web site. The "Keyword/Summary" idea of [8] is adopted, and the methodology is substantially enhanced by applying machine learning and natural language processing techniques and extended to Web sites.

3 Web Page and Text Extraction

In general, the structure of a Web site is hierarchical. In a breadth-first traversal of a Web site, the home page is the root of the hierarchy, i.e., first level. All Web pages pointed at from the home page are in the second level, and so on. Intuitively, the content of pages near the root is more representative of the content of the site than pages deeper into the site. The home page of a Web site often presents a brief description of what this site is about. When we go deeper into the Web site, Web pages tend to discuss specific topics in detail.

Since our objective is to summarize the Web site, we want to focus on top-level pages in order to extract the contents which describe the Web site in a general sense. A module called *Site Crawler* is developed, that crawls within a given Web site using breadth-first-search. This means that only Web pages physically located in the site (in the same domain) will be crawled and analyzed. Besides tracking the URLs of these Web pages, the Site Crawler records the depth (i.e. level) and length of each page. Depth represents the number of "hops" from the home page to the current page. For example, if we give the home page depth 1, then all pages which can be reached by an out-link of the home page are assigned depth 2. Length of a Web page is the number of characters in the Web page source file. The Site Crawler only keeps known types of Web pages, such as .htm, .html, .shtml, .php, etc. Handling other types of text and non-text files is a topic

---

[4]http://www.altavista.com

for future research.

After crawling 60 Web sites (identified in DMOZ subdirectories), according to a breadth-first traversal starting from the home page, it is observed that there is an average of about 1000 pages up to and including depth equal to 4. The depth number 4 is based on a tradeoff between crawling cost and informativeness of Web pages. For each Web site, the Site Crawler keeps crawling until either 1000 pages have been collected, or it has finished crawling depth 4, whichever comes first.

After the URLs of the Web pages have been collected, the plain text is extracted from them. Several packages are available for this purpose. Two of them, *HTML2TXT* [34] by Thomas Sahlin and *html2txt* [31] by Gerald Oskoboiny are compared against the text browser *Lynx* [10] and an *HTML Parser* developed by the authors. Our *HTML Parser* identifies all HTML tags in a Web page, removes dynamic content such as JavaScript, and keeps only plain text. We compared these four packages and settled on Lynx. Another advantage of Lynx is that it has built-in mechanism to segment text extracted from a Web page into text paragraphs automatically [10].

4   Narrative Text Classification

The summary of the Web site will be created on the basis of the text extracted by Lynx. However, Web pages often contain isolated phrases, bullets or very short sentences, instead of a coherent narrative structure. Such text provides little foothold for a coherent and meaningful summary [4], so our aim is to identify rules for determining which paragraphs should be considered for summarization and which ones should be discarded. This is achieved in two steps: First, criteria are defined for determining if a paragraph is long enough to be considered for analysis. Then, additional criteria are defined to classify long paragraphs into narrative or non-narrative. Only narrative paragraphs are used in summary generation. The criteria are defined automatically using supervised machine learning.

4.1   Long Paragraph Classification

During our experiments with the crawler, it is observed that some paragraphs are 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.*

Intuitively, whether a paragraph is long or short is determined by its length (i.e., the number of characters). However, two more features, number of words, and number of characters in all words, might also play key roles. Thus instead of setting up a simple threshold for number of words, we let the decision tree learning program C5.0 [1] determine which feature is the most important. A total of 700 text paragraphs is extracted from 100 Web pages (collected from 60 Web sites). Statistics of three attributes *Length, NumberOfWords* and *NumberOfChars* are recorded from each paragraph. *Length* is the number of all characters (including punctuation) in the paragraph. *NumberOfWords* is the number of words in this paragraph, and

No.	Length	NumberOfWords	NumberOfChars	LONGSHORT
1	423	58	372	long
2	35	5	31	short
3	913	125	802	long
...	...	...	...	...
700	89	12	71	short

Table 1: Training data of C5.0 classifier *LONGSHORT*

Fold	1	2	3	4	5	6	7	8	9	10	Mean
Size	2	2	2	2	2	2	2	2	2	2	2.0
Error(%)	5.7	5.7	11.4	4.3	2.9	4.3	4.3	7.1	2.9	10.0	5.9

Table 2: Cross-validation of C5.0 classifier *LONGSHORT*

NumberOfChars is the total number of characters in all words (excluding punctuation). 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, as shown in Table 1. 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 is listed in Table 2. The mean error rate 5.9% indicates the classification accuracy of this classifier.

4.2 Narrative Paragraph Classification

Not all long paragraphs provide coherent information in terms of generating a meaningful summary. Intuitively, among the long paragraphs, narrative ones provide more coherent and meaningful content than non-narrative ones.

An example of a narrative paragraph is: *The user's login file usually defines the command aliases and author identification (for the update history). Then one normally opens one or more CMZ files depending on the size of the project.*

An example of a non-narrative paragraph: ** ESTIMATE Professional (software project planning and estimation); * EssentialSET (documentation templates, process framework); * ISOplus (quality systems documentation)*

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 [2] and question answering [32]. We hypothesize that the frequencies of the part-of-

No.	Length	Number	P_1	P_2	...	P_{32}	NARRATIVE
1	2010	201	0.040	0.020	...	0.003	narrative
2	1068	189	0.042	0.011	...	0.001	non-narrative
3	950	166	0.067	0.0	...	0.012	narrative
...	...	...	...	...	...	...	...
3243	572	108	0.020	0.049	...	0.020	non-narrative

Table 3: Training data of C5.0 classifier *NARRATIVE*

Fold	1	2	3	4	5	6	7	8	9	10	Mean
Size	5	5	3	4	4	5	4	3	4	3	4.0
Error(%)	11.1	9.3	13.6	11.1	9.9	7.4	9.3	16.0	10.5	14.7	11.3

Table 4: Cross-validation of C5.0 classifier *NARRATIVE*

speech tags of the words in a paragraph contain sufficient information to classify the paragraph as narrative. To test our hypothesis, a training set is generated as follows: First, 1000 Web pages are collected from 60 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 [6].

After part-of-speech tagging, the following attributes are extracted from each paragraph. Let n_i (i = 1, 2, ... , 32) be the number of occurrences of tag i, and S be the total number of tags (i.e. words) in the paragraph. Let P_i be the fraction of S, that n_i represents.

$$
\begin{aligned}
S &= \sum_{i=1}^{32} n_i \\
P_i &= n_i/S \ \ (i = 1, 2, ..., 32)
\end{aligned}
\tag{1}
$$

Two more attributes are added to this set, the number of characters and the number of words in the paragraph. Then each paragraph is manually labelled as *narrative* or *non-narrative*. Finally, a C5.0 classifier *NARRATIVE* is trained on the training set of 3243 paragraphs, shown in Table 3.

The decision tree and its evaluation generated by the C5.0 program are presented in Figure 2. 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* is listed in Table 4. The mean error rate 11.3% indicates the predictive accuracy of this classifier.

Decision Tree:

```
SYM > 0.068: non-narrative (354/14)
SYM <= 0.068:
:...IN <= 0.052: non-narrative (284/38)
   IN > 0.052:
   :...NNP <= 0.233: narrative (2058/90)
      NNP > 0.233:
      :...DT <= 0.075: non-narrative (236/72)
         DT > 0.075: narrative (210/46)
```

Evaluation on training data (3242 cases):

Decision Tree		
Size	Errors	
5	260	(8.0%)

(a)	(b)	<< classified as
2232	124	(a): class *narrative*
136	750	(b): class *non-narrative*

Figure 2: Decision tree of *NARRATIVE* and its evaluation

5   Key-Phrase Extraction

Traditionally, key-phrases (key-words and key-terms) are extracted from the documents in order to generate a summary. Based on these phrases, the most significant sentences, which 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 [8]. Traditionally, TF-IDF measure [35] is widely used to identify key phrases, which occur frequently within the given text, but infrequently in the larger collection of documents [8]. However, this might not be true when summarizing a Web site. Suppose there is a page talking about *Linux*, and Linux occurs very frequently in this page, but rarely in the rest of the pages, then Linux is a good key-phrase based on TF-IDF, but a bad phrase for describing the whole Web site. Thus in our approach, 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 occurrence is very high.

In this work, a *key-phrase* can be either *key-word* or *key-term*. A key-word is a single word with high frequency over the set of Web pages, and a key-term is a two-word term with very high frequency.

As we discussed in the previous section, Web pages are quite different from traditional documents. The existence of *anchor text* and *special 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 [7, 29]. Special text includes title, headings and bold or italicized text. The assumption is that both anchor text and special text may play a key role in describing important topics of Web pages. Therefore a supervised learning approach is applied to test this assumption.

5.1   Key-word Extraction

First we discuss how to produce decision tree rules for determining the key-words of a Web site. A data set of 5454 candidate key-words (at most 100 for each site) from 60 Web sites are collected. The sites are taken from DMOZ subdirectories. For each site, the frequency of each word in narrative text, anchor text and

No.	Word	f	fn	fa	fs
1	system	5450	4648	310	492
2	software	5313	3643	1092	578
3	product	2953	2033	668	252
...	...	...	...	...	...
5454	process	2294	1237	632	425

Table 5: Example of candidate key-words

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.

As it can be seen in Table 5, f is the total frequency of a candidate key-word, while fn, fa, and fs are the frequencies of a candidate key-word in narrative text, anchor text and special text, respectively, hence $f = fn + fa + fs$. For example, the word *system* occurs 4648, 310 and 492 times in narrative text, anchor text and special text, respectively. This yields a total frequency of 5450. 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, it should be noted that a standard set of 425 stop words (*a, about, above, across, after, again, against, all, almost, alone, along, already, also, although, always, among, an, and, ...*) [13] is discarded in this stage.

There are about 5,100 different words (excluding stop words) on the average within the text body of the top Web pages of a site. The 5 words with the highest frequency often have a frequency of more than 2,000, whereas the words with the lowest frequencies may occur only once or twice. Figure 3 shows the rank and frequency statistics of these words in log-log scale (base 10). The data is approximated well by a straight line, except at the two ends. This indicates that the data fits Zipf's Law [22]. The words with the lowest frequencies are obviously not key-words, hence only those words whose frequency is more than 5% of the maximum frequency are kept as *candidate key-words*. This step eliminates about 98% of the original words, leaving about 100 candidate key-words per site. As a result, at most the top 100 candidate key-words are selected. For each candidate key-word C_i, nine features are extracted, as shown in Table 6. Since a word often occurs in the text with several different part-of-speech tags, the most frequently appeared part-of-speech tag is assigned to the word.

Next, each candidate key-word is labelled manually as *key-word* or *non-key-word*. The criterion to determine if a candidate key-word is a true key-word is that a key-word must provide important information about the Web site. Based on frequency statistics and part-of-speech tags of these candidate key-words, a C5.0 classifier *KEY-WORD* is constructed as shown in Table 7.

The decision tree and its evaluation generated by the C5.0 program are presented in Figure 4. Among
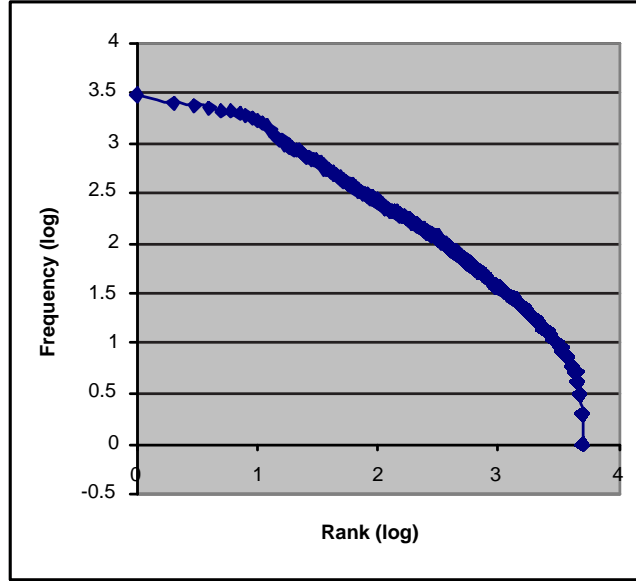
Figure 3: Rank-Frequency data and Zipf's Law

No.	Feature	Value	Meaning
1	W	$W_i = f_i / \sum_{i=1}^{100} f_i$	Weight of candidate key-word C_i
2	R	$R_i = f_i / \max_{i=1}^{100} f_i$	Ratio of frequency to max frequency
3	WN	$WN_i = fn_i / \sum_{i=1}^{100} fn_i$	Weight in narrative text only
4	RN	$RN_i = fn_i / \max_{i=1}^{100} fn_i$	Ratio in *narrative* text only
5	WA	$WA_i = fa_i / \sum_{i=1}^{100} fa_i$	Weight in *anchor* text only
6	RA	$RA_i = fa_i / \max_{i=1}^{100} fa_i$	Ratio in *anchor* text only
7	WS	$WS_i = fs_i / \sum_{i=1}^{100} fs_i$	Weight in *special* text only
8	RS	$RS_i = fs_i / \max_{i=1}^{100} fs_i$	Ratio in *special* text only
9	Tag	$CC, CD, ..., WRB$	Part-of-speech tag ([6])

Table 6: Feature list of candidate key-words

W	R	WN	RN	WA	RA	WS	RS	Tag	KEY-WORD
0.072	1.0	0.067	1.0	0.080	1.0	0.096	1.0	NN	key-word
0.047	0.651	0.055	0.824	0.017	0.214	0.039	0.403	NN	key-word
0.015	0.320	0.012	0.388	0.013	0.028	0.055	0.211	NN	key-word
...	...	...	...	...	...	...	...	...	...
0.010	0.136	0.007	0.104	0.026	0.323	0.005	0.051	VB	non-key-word

Table 7: Training data of C5.0 classifier *KEY-WORD*

Fold	1	2	3	4	5	6	7	8	9	10	Mean
Size	22	20	20	30	23	18	20	27	20	20	22.0
Error(%)	4.0	5.1	5.5	4.4	4.0	5.1	5.1	5.9	5.5	4.0	4.9

Table 8: Cross-validation of C5.0 classifier *KEY-WORD*

the total 5454 cases, 222 cases are misclassified, leading to an error of 4.1%. In the decision tree, about 35% of cases are following this rule: if R (defined as the ratio of a candidate key-word's frequency to the maximum frequency in Table 6) is less than or equal to 0.1, then this candidate key-word is a non-key-word. Another main stream of cases follows the second rule: if R is greater than 0.1, and part-of-speech tag is NN (common singular nouns [6]), and RA (ratio in anchor text) is less than or equal to 0.798, then the candidate key-word is a key-word. This case covers 45% of the data set. Another interesting rule is: if R is greater than 0.1, and part-of-speech tag is NNS (common plural nouns [6]), then the candidate key-word should be classified as key-word. However, among these 138 cases, 50 are misclassified. This means that this training set is not effective in identifying common plural nouns, due to an insufficient number of such cases. A fourth interesting rule is: if R is greater than 0.1 and part-of-speech tag is NN (common singular nouns) or VBG (verb -ing [6]), then WA (weight in anchor text), RA (ratio in anchor text) and/or WS (weight in special text) will determine if a candidate key-word should be classified as key-word or non-key-word. This demonstrates that our assumption is true, that anchor text and special text do play an important role in determining key-words of a Web site. The cross-validation results of the classifier *KEY-WORD* is listed in Table 8. The mean error rate 4.9% indicates the predictive accuracy of this classifier.

5.2 Key-term Extraction

It is observed that terms which consist of two of the top 100 candidate key-words from each Web site may exist with high frequency. Such a term could be good as part of the description of the Web site. For example, at the Software Engineering Institute Web site[5], the words *software* and *engineering* have frequency 7805

---
[5]http://www.sei.cmu.edu

Decision Tree:

R <= 0.1: *non-key-word* (1908)

R > 0.1:

:...Tag in {CC,CD,EX,FW,IN,JJR,NNPS,NNP,

:   PRP$,PRP,RBR,RBS,SYM,TO,VBZ,

:   WDT,WP$,WP,WRB}:*non-key-word* (0)

  Tag = DT: *non-key-word* (2)

  Tag = JJS: *non-key-word* (4)

  Tag = JJ: *non-key-word* (350/6)

  Tag = MD: *non-key-word* (2)

  Tag = NNS: *key-word* (138/50)

  Tag = RB: *non-key-word* (18)

  Tag = UH: *non-key-word* (2)

  Tag = VBD: *non-key-word* (36)

  Tag = VBN: *non-key-word* (94/4)

  Tag = VBP: *non-key-word* (26)

  Tag = VB: *non-key-word* (292)

  Tag = NN:

:...RA <= 0.798: *key-word* (2438/138)

:   RA > 0.798:

:   :...WA > 0.192: *non-key-word* (12)

:   WA <= 0.192:

:   :...RA <= 0.833: *non-key-word* (6)

:   RA > 0.833: *key-word* (52/14)

  Tag = VBG:

:...WS <= 0.004: *non-key-word* (40/6)

  WS > 0.004:

  :...WS > 0.105: *non-key-word* (4)

    WS <= 0.105:

    :...R <= 0.121: *non-key-word* (4)

      R > 0.121: *key-word* (26/4)

Evaluation on **training data** (5454 cases):

Decision Tree

Size	Errors	
20	222	(4.1%)
(a)	(b)	<< classified as
2448	16	(a): class *key-word*
206	2784	(b): class *non-key-word*

Evaluation on **test data** (2718 cases):

Decision Tree

Size	Errors	
20	160	(5.9%)
(a)	(b)	<< classified as
1208	30	(a): class *key-word*
130	1350	(b): class *non-key-word*

Figure 4: Decision tree of *KEY-WORD* and its evaluation

and 3430, respectively, and the term *software engineering* occurs 2492 times. Thus, a similar approach to automatic key-word extraction is developed to identify key-terms of the Web site.

The algorithm combines any two of the top 100 candidate key-words and searches for these terms in collocation over narrative text, anchor text and special text. Then these terms are sorted by frequency and the top 30 are kept as candidate key-terms. A C5.0 classifier *KEY-TERM* is constructed based on frequency statistics and tag features of 1360 candidate key-terms, which are extracted from 60 Web sites (collected from DMOZ subdirectories). The C5.0 classifier KEY-TERM is similar to the KEY-WORD classifier except that it has two part-of-speech tags *Tag1* and *Tag2*, one for each component word.

Once the decision tree rules for determining key-terms have been built, they are applied for automatic key-term extraction to the Web pages of a Web site. The top 10 key-terms (ranked by overall frequency) for each site are kept as part of the summary. Then the frequency of the candidate key-words forming the top 10 key-terms is reduced by subtracting the frequency of the corresponding key-terms. For example, the actual frequency of the words *software* and *engineering* above becomes $7805 - 2492 = 5313$ and $3430 - 2492 = 938$, respectively. Then candidate key-words of the Web site are classified into key-word or non-key-word by applying the KEY-WORD classifier shown in Figure 4. Finally, the top 25 key-words (ranked by frequency) are kept as part of the summary. It is observed that 40% to 70% of key-words and 20% to 50% of key-terms appear in the home page of a Web site.

6 Key-Sentence Extraction

Once the key-words and key-terms are identified, the most significant sentences for summary generation can be retrieved from all narrative paragraphs based on the presence of key-phrases [9]. Each sentence is assigned a significance factor or sentence weight. The top five sentences, ranked according to sentence weight, are chosen as part of the summary. In order to achieve this goal, we applied a modified version of the procedure in [8].

First, the sentences containing any of the set L of the extracted key-phrases (the top 25 key-words and top 10 key-terms) are selected.

Second, all clusters in each selected sentence S are identified. A *cluster C* is a sequence of consecutive words in the sentence for which the following is true: (1) the sequence starts and ends with a key-phrase in L, and (2) less than D non-key-phrases must separate any two neighboring key-phrases within the sentence. D is called the "distance cutoff", and we used a value of 2 as in [8]. Table 9 gives an example of clustering, where key-words, key-terms and clusters are listed.

Third, the weight of each cluster within S is computed. The maximum of these weights is taken as the sentence weight. A cluster weight is computed by adding the weights of all key-phrases within the cluster, and dividing this sum by the total number of words within the cluster [8]. The weight of key-phrase i is defined as $W_i = f_i / \sum_{i=1}^{100} f_i$, where f_i is the frequency of the key-phrase in the Web site (Table 6). For

Candidate Sentence			
The Software Engineering Information Repository (SEIR) is a Web-based repository of information on software engineering practices that lead to improved organizational performance.			
Key-Phrase	**Weight**	**Cluster**	**Weight**
information	0.021	1. Software Engineering Information	0.157
software engineering	0.293	2. information on software engineering practices	0.109
practice	0.013	Sentence Weight: 0.157	

Table 9: Example of clustering

example, the second cluster's weight in Table 9 is $(0.021 + 0.293 + 0.013)/5 = 0.065$.

However, division by the total number of words in the cluster decreases the weight too much when there are several key-phrases present together with a large number of non-key-phrases. After some informal experimentation, it is decided to modify the weighting: the best cluster weighting (in terms of assigning the highest weight to the most informative sentences) is obtained 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. Hence the second cluster's weight in Table 9 will now be $(0.021 + 0.293 + 0.013)/3 = 0.109$ and the first cluster's weight is $(0.293 + 0.021)/2 = 0.157$, therefore the sentence weight is 0.157.

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

7 Experiments and Evaluation

In order to measure the overall performance of our approach, four sets of experiments are performed. During these experiments, automatically generated summaries are compared with human-authored summaries, home page browsing and time-limited site browsing, in terms of their effectivess in helping a human answer a set of questions about the Web site.

Part 1. Top 25 Key-words				
sei	system	software	cmu	product
component	information	process	architecture	organization
course	program	report	practice	project
method	design	institute	development	research
document	management	defense	technology	team

Part 2. Top 10 Key-terms				
software engineering	carnegie mellon	development center	software process	software architecture
maturity model	risk management	software development	process improvement	software system

Part 3. Top 5 Key-sentences

1. The Software Engineering Information Repository (SEIR) is a Web-based repository of information on software engineering practices that lead to improved organizational performance.

2. Because of its mission to improve the state of the practice of software engineering, the SEI encourages and otherwise facilitates collaboration activities between members of the software engineering community.

3. The SEI mission is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.

4. The Software Engineering Institute is operated by Carnegie Mellon University for the Department of Defense.

5. The Software Engineering Institute (SEI) sponsors, co-sponsors, and is otherwise involved in many events throughout the year.

Table 10: Summary of Software Engineering Institute Web site

Subdirectory	Site URL
Software/ Software Engineering	1. http://case.ispras.ru
	2. http://www.ifpug.org
	3. http://www.mapfree.com/sbf
	4. http://www.cs.queensu.ca/Software-Engineering
	5. http://www.sei.cmu.edu
Artificial Intelligence/ Academic Departments	6. http://www.cs.ualberta.ca/~ai
	7. http://www.ai.mit.edu
	8. http://www.aiai.ed.ac.uk
	9. http://www.ai.uga.edu
	10. http://ai.uwaterloo.ca
Major Companies/ Publicly Traded	11. http://www.aircanada.ca
	12. http://www.cisco.com
	13. http://www.microsoft.com
	14. http://www.nortelnetworks.com
	15. http://www.oracle.com
E-Commerce/ Technology Vendors	16. http://www.adhesiontech.com
	17. http://www.asti-solutions.com
	18. http://www.commerceone.com
	19. http://www.getgamma.com
	20. http://www.rdmcorp.com

Table 11: URL list of the Web sites used in the experiments

7.1 W3SS and DMOZ Summaries

From the DMOZ Open Directory Project, 20 manually constructed summaries are selected from four sub-directories. As listed in Table 11, sites 1-5 are in the *Software/Software Engineering*[6] subdirectory. Sites 6-10 are in the *Artificial Intelligence/Academic Departments*[7] subdirectory. Sites 11-15 are in *Major Companies/Publicly Traded*[8] subdirectory. And finally sites 16-20 are in *E-Commerce/Technology Vendors*[9] sub-directory. These sites are selected randomly and are of varying size and focus. Sites 1-10 are academic ones, with a focus on academic or non-profit research and/or development, whereas sites 11-20 are commercial ones, which deliver commercial products and/or services.

Our approach, $W3SS$ (World Wide Web Site Summarization), is used to create summaries of these 20

---

[6]http://dmoz.org/Computers/Software/Software_Engineering/

[7]http://dmoz.org/Computers/Artificial_Intelligence/Academic_Departments/

[8]http://dmoz.org/Business/Major_Companies/Publicly_Traded/

[9]http://dmoz.org/Business/E-Commerce/Technology_Vendors/

Web sites. Each W3SS summary consists of the top 25 key-words, the top 10 key-terms and the top 5 key-sentences. All DMOZ and W3SS summaries are detailed in [41].

7.2 Summarization Evaluation

In this work, extrinsic evaluation is used to evaluate the summaries. The objective is to measure how informative W3SS summaries, DMOZ summaries, home page browsing and time-limited site browsing are in answering a set of questions (see [41]) about the content of the Web site. Each question is meant to have a well-defined answer, ideally explicitly stated in the summary, rather than being open-ended. For example, one question asks for the purpose of the Web site, and the other looks for the main activities of the entity behind the Web site. Four groups of graduate students in Computer Science (5 in each group) with strong World Wide Web experience are asked to take the test as follows:

The first group and the second group are asked to read each W3SS and DMOZ summary respectively and then answer the questions. The third group is asked to browse the home page of each of the 20 Web sites and answer the questions. The last group is asked to browse each Web site for at most 10 minutes (time-limited site browsing) and answer all questions. All answers are then graded in terms of their quality in a scale 0-20. The grades are tabulated in [41].

7.2.1 Evaluation of W3SS Summaries

The average score of all W3SS summaries over five subjects is 15.0. The average score of each W3SS summary over five subjects varies from 8.8 to 19.6, which means the quality of W3SS summaries varies from site to site. Summaries of Nortel Networks and Microsoft Corporation Web sites get the top two average scores 19.6 and 19.2, respectively. These two summaries give a brief but accurate description of the corporations. The summary of the Adhesion Technologies Web site gets the lowest average score 8.8, because it describes specific software products the company delivers but no general information about the company is available. The summary with the second lowest average score 10.0 corresponds to a site that contains archives of Software Engineering but the summary gives a description of specific software tools and cannot convey any sense of that this site is an information resource. However, the variance between the average scores of all summaries over five subjects is only 0.213, which shows that all subjects in this experiment evaluated W3SS summaries consistently.

7.2.2 Evaluation of DMOZ Summaries

The average score of all DMOZ summaries over all subjects is 15.3, hence the overall performance of DMOZ summaries is slightly better than that of W3SS ones (with an overall average 15.0). The average score of each DMOZ summary over five subjects varies from 11.0 to 19.2. However, the variance between the average scores of all DMOZ summaries over five subjects is 1.267, much larger than that of W3SS summaries.
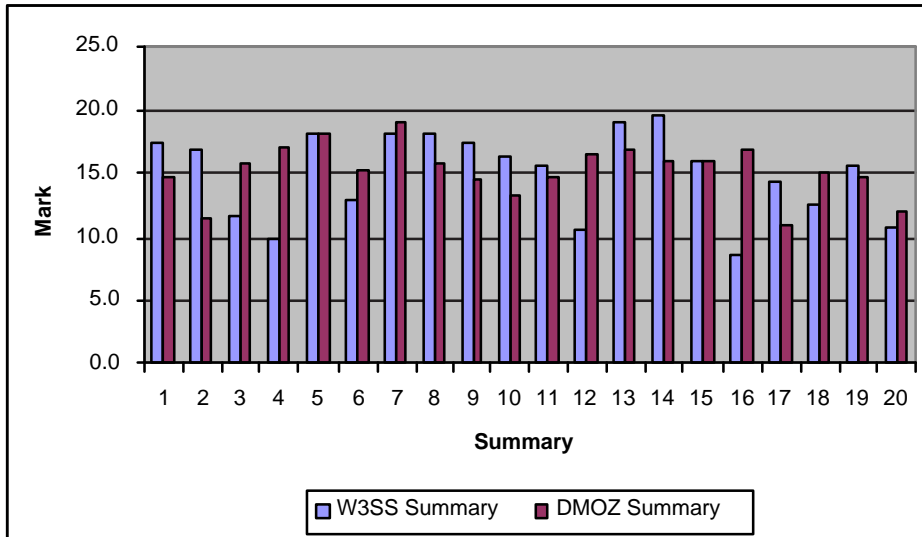
18

Figure 5: W3SS summaries vs. DMOZ summaries

As indicated in Figure 5, there are 11 Web sites whose W3SS summaries are better than DMOZ summaries, and 8 sites whose W3SS summaries are worse than DMOZ summaries. The remaining site has the same quality of W3SS summary and DMOZ summary.

7.2.3 Evaluation of home page browsing

Since every subject is allowed to browse only the home page, there are a few very poor marks as low as 4.4 and 5.0. The average score of all home pages over five subjects is 12.7, which is less than 15.0 of W3SS summaries and 15.3 of DMOZ summaries.

As indicated in Figure 6, the performance of W3SS summaries is generally better than that of home page browsing. This experiment tells us that the home page alone is often not sufficiently informative, and that digging deeper into the site conveys more complete information about the site than the home page alone. In order to understand the site better, more browsing beyond the home page alone is needed.

7.2.4 Evaluation of time-limited site browsing

In this test, each subject is allowed 10 minutes to browse each Web site, and look for the answers of all questions. For each site, the average score of all subjects varies from 7.0 to 20.0. This implies that either some Web sites are poorly designed, or there is too much non-text (e.g. flash) in top-level pages, which may confuse the user's understanding of the site. The average score of each site browsing over all subjects is 13.4, which is less than that of both W3SS and DMOZ summaries.

As indicated in Figure 7, the performance of W3SS summaries is generally better than that of time-limited site browsing. This means it is not so easy to get a good understanding of the site's main contents
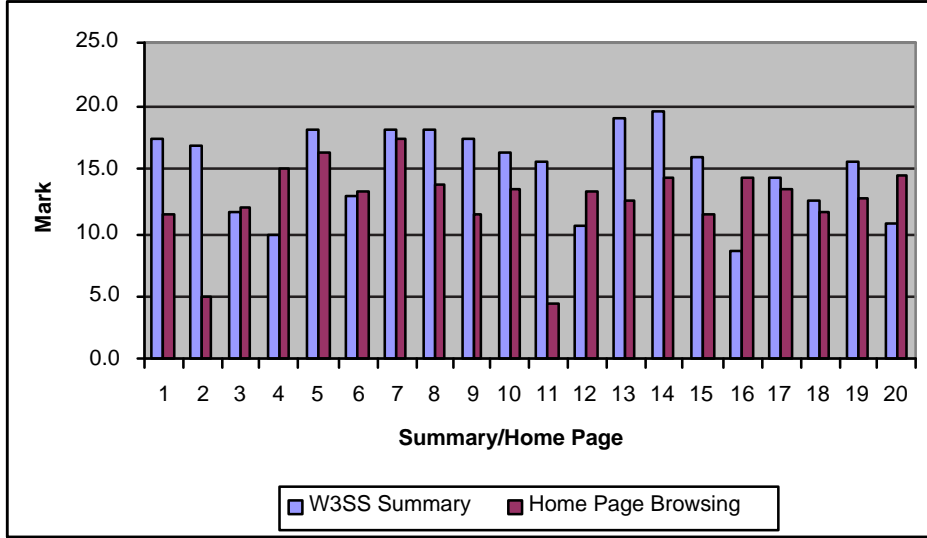
19

Figure 6: W3SS summaries vs. Home page browsing

	W3SS	DMOZ	HPB
DMOZ	$F_{1,190} = 0.18$ $Pvalue = 0.67$		
HPB	$F_{1,190} = 17.42$ $Pvalue < 0.0001$	$F_{1,190} = 23.7$ $Pvalue < 0.0001$	
TLSB	$F_{1,190} = 6.13$ $Pvalue = 0.014$	$F_{1,190} = 8.88$ $Pvalue = 0.003$	$F_{1,190} = 1.62$ $Pvalue = 0.20$

Table 12: Pairwise ANOVA results for the four experiments. W3SS, DMOZ, HPB, TLSP is the performance of our summaries, the human-authored summaries, home-page browsing and time-limited site browsing.

by browsing within a limited time period. If the W3SS summary of a Web site is available, then the reader can know the site's main contents by viewing the summary without spending much time in browsing the site. This indicates that our approach of automatically creating summaries is potentially useful because it saves the reader a lot of time.

To confirm the above intuitive conclusions, we perform a two-factor Analysis of Variance with replications on the raw scores from the above experiments. As shown in Table 12, there is no significant difference between our summaries and the human-authored summaries, and between home-page and time-limited site browsing. However, our summaries and the human-authored summaries are significantly better than home-page and time-limited site browsing.

Since the W3SS summaries are as informative as DMOZ summaries, they could be transformed into
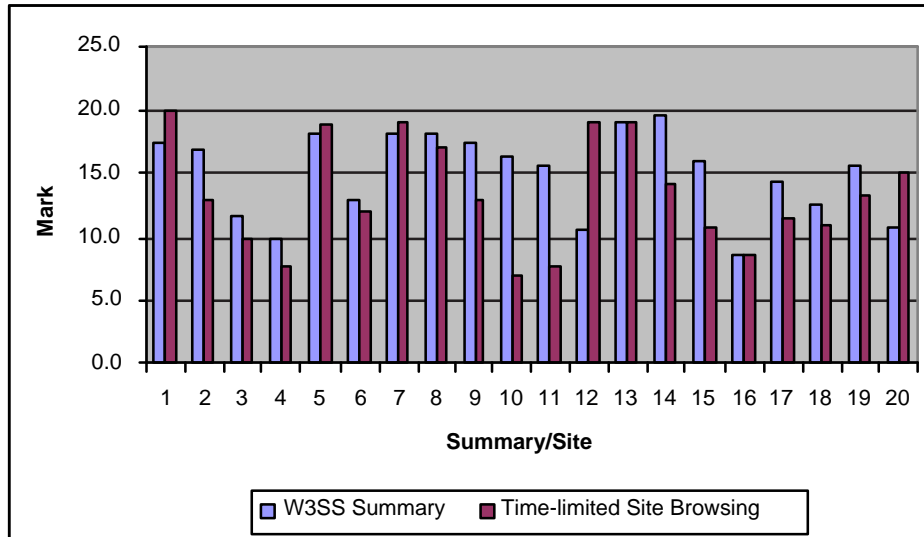
Figure 7: W3SS summaries vs. Time-limited site browsing

proper prose of comparable quality to the latter by human editors without browsing the Web site.

8 Conclusion and Discussion

In this work, we developed a new approach for generating summaries of Web sites. Our approach applies machine learning and natural language processing techniques to extract and classify narrative paragraphs from the Web site, from which key-phrases are then extracted. Key-phrases are in turn used to extract key-sentences from the narrative paragraphs that form the summary, together with the top key-phrases. We demonstrate that our summaries, although not in proper prose, are as informative as human-authored summaries, and significantly better than browsing the home page or the site for a limited time. Our approach should be easy to transform into proper prose by human editors without having to browse the Web site. The performance of our method depends on the availability of sufficient narrative content in the Web site, and the availability of explicit narrative statements describing the site.

Several issues need to be addressed to further improve the performance of our approach.

- Currently the system (mainly written in Java) is running on a UNIX machine with 8 processors (400MHz UltraSPARC II) and 3GB Memory. In general, the amount of time required for the URL Extraction and Text Extraction steps depends on the throughput of the Internet connection. There is an average of 17000 paragraphs (including sentence segments) in the text parts of the Web pages of the sites we considered. Long paragraph classification takes about one minute. Usually it takes 3 seconds for the part-of-speech tagger [6] (written in C) to tag a text file with around 100 words in this environment. So tagging an average of 5000 long paragraphs can last as long as 4 hours. It takes about

21

three minutes to identify 1500 narrative paragraphs. Key-phrase extraction takes about 10 minutes, and key-sentence extraction needs about a minute. So more than 80% of computing time is spent in tagging long paragraphs.

- In the key-term extraction step, we simply combine any two of top 100 candidate key-words. More sophisticated methods, such as the C-value/NC-value method [14] will be considered to automatically recognize multi-word terms and then measure their frequencies for the purpose of extracting key multi-word terms.

- Research [7] indicates that assigning higher weight to anchor text may provide better results in search engines. Further research is required to determine appropriate weights for the key-words from different categories (plain text, anchor text and special text).

- Different subjects gave different marks to the same summary (or home page, site), indicating the potential presence of an inter-rater reliability problem[40]. Redesign of the evaluation process to reduce the inter-rater reliability problem is a topic for future research. Intrinsic evaluation should also be considered.

- Moreover, controls for the experiments should be set to factor out the background knowledge of human subjects in future research.

References

[1] RULEQUEST Research 2003. *C5.0: An Informal Tutorial.* http://www.rulequest.com/see5-unix.html, last accessed on Oct. 9, 2002.

[2] 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.

[3] 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.

[4] 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.

[5] A. Berger and V. Mittal. Query-relevant summarization using FAQs. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 294–302, 2000.

[6] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 152–155, 1992.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, 1998.

[8] 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.

[9] 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.

[10] Internet Software Consortium. *Lynx: a World Wide Web (WWW) client for cursor-addressable, character-cell display devices.* http://lynx.isc.org, last accessed on Oct. 9, 2002.

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

[12] H. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285, 1969.

[13] 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.

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

[15] 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.

[16] 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.

[17] G. Greco, S. Greco, and E. Zumpano. A probabilistic approach for distillation and ranking of Web pages. *World Wide Web, Kluwer Academic Publishers*, 4(3):189–207, 2001.

[18] U. Hahn and I. Mani. The challenges of automatic summarization. *IEEE Computer*, 33(11):29–36, 2000.

[19] V. Hatzivassiloglou, J.L. Klavans, M.L. Holcombe, R. Barzilay, M. Kan, and K.R. McKeown. Simfinder: a flexible clustering tool for summarization. In *NAACL Workshop on Automatic Summarization*, pages 41–49, 2001.

[20] B. Huberman and L. Adamic. Evolutionary dynamics of the World Wide Web. *Nature*, 399(131), 1999.

[21] S. Jones, S. Lundy, and G. Paynter. Interactive document summarisation using automatically extracted keyphrases. In *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02)*, 2002.

[22] W. Li. *Zipf's Law.* http://linkage.rockefeller.edu/wli/zipf, last accessed on Oct. 9, 2002.

[23] C. Lin and E. Hovy. From single to multi-document summarization: a prototype system and its evaluation. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, pages 457–464, 2002.

[24] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research & Development*, 2(2):159–165, 1958.

[25] I. Mani. Recent developments in text summarization. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM'01)*, pages 529–531, 2001.

[26] 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.

[27] I. Mani and M. Maybury. *Advances in Automatic Text Summarization.* MIT Press, 1999.

[28] 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.

[29] O. McBryan. GENVL and WWWW: Tools for taming the Web. In *Proceedings of the 1st International Conference on the World Wide Web*, pages 79–90, 1994.

[30] K. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, M. Yen Kan, B. Schiffman, and S. Teufel. Columbia multi-document summarization: approach and evaluation. In *DUC'01 Workshop on Text Summarization*, 2001.

[31] G. Oskoboiny. *html2txt.* http://cgi.w3.org/cgi-bin/html2txt, last accessed on Oct. 9, 2002.

[32] 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.

[33] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *ANLP/NAACL'00 Workshop on Automatic Summarization*, pages 21–29, 2000.

[34] T. Sahlin. *HTML2TXT*. http://user.tninet.se/~jyc891w/software/html2txt/, last accessed on Oct. 9, 2002.

[35] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[36] J. Schlesinger, J. Conroy, M. Okurowski, and D. OLeary. Machine and human performance for single and multidocument summarization. *IEEE Intelligent Systems*, 18(1):46–54, 2003.

[37] K. Sparck-Jones and J. Galliers. *Evaluating Natural Language Processing Systems: an Analysis and Review*. Springer-Verlag, 1996.

[38] G. Stein, A. Bagga, and G. Wise. Multi-document summarization: methodologies and evaluations. In *Proceedings of the 7th Conference on Automatic Natural Language Processing (TALN'00)*, pages 337–346, 2000.

[39] S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002.

[40] Colorado State University. *Writing Guide: Interrater Reliability*. http://writing.colostate.edu/references/research/relval/com2a5.cfm, last accessed on Oct. 9, 2002.

[41] Y. Zhang, N. Zincir-Heywood, and E. Milios. *World Wide Web Site Summarization*. Technical Report CS-2002-8, Faculty of Computer Science, Dalhousie University, October 2002, http://www.cs.dal.ca/research/techreports/2002/CS-2002-08.pdf, last accessed on Oct. 9, 2002.