A Comparative Study on Key Phrase Extraction Methods in Automatic Web Site Summarization

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

Abstract

Web Site Summarization is the process of automatically generating a concise and informative summary for a given Web site. It has gained more and more attention in recent years as effective summarization could lead to enhanced Web information retrieval systems such as searching for Web sites. Extraction-based approaches to Web site summarization rely on the extraction of the most significant sentences from the target Web site based on the density of a list of key phrases that best describe the entire Web site. In this work, we benchmark five alternative key phrase extraction methods, TFIDF, KEA, Keyword, Keyterm, and Mixture, in an automatic Web site summarization framework we previously developed. We investigate the performance of these underlying methods via a formal user study and demonstrate that Keyterm is the best choice for key phrase extraction while Mixture should be used to obtain key sentences. We also discuss why one method performs better than another and what could be done to further improve the summarization system.

1 Introduction

As the amount of information continues to grow on the World Wide Web (WWW), effective management of online information becomes more and more critical. In the WWW context, many approaches to information management, such as indexing, categorization, clustering, and summarization, have been proposed.

Automatic Web site summarization, which is derived from multi-document summarization [23], is playing an important role in Web information management. A concise, informative, and meaningful Web site summary can help Web users understand the essential topics and main contents covered in the target Web site quickly without spending much browsing time [35]. The automatic Web site summarization method can be used in information retrieval systems to generate descriptions of the documents returned by a query, or as a way to

browse special Web page collections. It can also be applied to query expansion and query reformulation tasks.

Up to date approaches to Web document summarization have often been extraction-based [2, 4, 35]. They apply statistical and linguistic analysis to extract key phrases[1] which best describe the source documents, and further extract the most significant sentences based on the presence density of key phrases [4, 35]. Hence, the performance of an extraction-based Web site summarization system is mainly determined by its underlying key phrase extraction method.

In our previous work [35], we extend single Web document summarization to the summarization of a complete Web site. The "Keyword/Summary" idea of [4] is adopted, and the methodology is substantially enhanced and extended to Web sites. This extraction-based approach generates a Web site summary consisting of 25 keywords and 5 key sentences, where the numbers are determined by the requirement for a single-page summary and the informativeness of these summary elements. Since Web documents often contain diverse contents such as bullets and short sentences, the system applies machine learning and natural language processing techniques to extract the "narrative" content, and then extracts keywords from the narrative text together with anchor text and special text (e.g., emphasized text). The key sentences are then identified based on the density of keywords. The evaluation shows that the automatically generated summaries are as informative as human authored summaries (e.g. DMOZ[2] summaries).

In this work, we investigate five key phrase extraction methods, i.e., Term Frequency Inverse Document Frequency (TFIDF) [25], Automatic Keyphrase Extraction (KEA) [32], Keyword [35], C-value/NC-value (renamed Keyterm) [9], and Mixture. These methods have been well studied in related literature [8, 12, 13, 14, 17, 21, 22, 24, 29, 30, 33, 34, 36] and can be used in the key phrase extraction stage of our Web site summarization framework [35] described above.

- The TFIDF method captures a word's frequency in a single document compared to its rarity in the whole document collection. It has been widely studied in many information retrieval tasks so we use it as the baseline method.

- The second method, KEA, builds a Naïve Bayes learning model using training documents with known key phrases, and then uses the model to find key phrases in new documents.

  We acknowledge that both TFIDF and KEA were originally designed for key phrase extraction from single documents so we extend them to the application on an entire document collection.

- The Keyword method constructs a C5.0[3] classifier using Web pages with

[1] A phrase can be either a single word or a multi-word term. Throughout the paper, we use keywords, keyterms, and key phrases interchangeably, depending on the method context.

[2] `http://www.dmoz.org`

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

known single keywords, and then uses this model to identify keywords from a new Web site.

- The fourth method, Keyterm, consists of both linguistic and statistical analysis to extract multi-word keyterms automatically. Both Keyword and Keyterm are designed for key phrase extraction from an entire document collection.
- Finally, the Mixture method combines Keyword and Keyterm to obtain a list of key phrases.

We aim to investigate how well each key phrase extraction method performs in the automatic Web site summarization task via a formal user study, i.e., we are interested in learning which method will yield summaries with the best quality. We compare the key phrases generated by different methods in terms of "acceptable percentage", which is the ratio of key phrases that are reasonably related to the most essential topic of a given Web site. We also quantify them to measure the quality difference between any two methods. One-Way Fully Repeated Measures ANOVA is used to conduct the statistical analysis. The evaluation shows that key phrases extracted by the Keyterm method achieve the best quality, and that the Mixture method can lead to key sentences with the best quality.

The rest of the paper is organized as follows. Section 2 reviews related literature. Section 3 outlines our framework of automatic Web site summarization and Section 4 describes the five key phrase extraction methods in detail. In Section 5, we discuss the design of our experiments and show the evaluation results. Finally, we conclude our work and discuss future research directions for improvement of the summarization system in Section 6.

2 Related Work

Web document summarization techniques are mostly derived from traditional plain text summarization techniques. Existing text summarization systems generate summaries automatically either by *extraction* or *abstraction*. Extraction-based systems [5, 11] analyze source documents using techniques such as frequency analysis to determine the most significant sentences based on features such as the density of keywords [35] and rhetorical relations [19] in the context. Abstraction [2], 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].

Research on Web document summarization to date has been extraction-based. Some systems [2, 4] analyze the contents and extract the most significant phrases and sentences to construct a summary.

Berger and Mittal [2] 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. [4] propose five 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 based on the density of keywords. The test shows that the combined *Keyword/Summary* method provides the best performance.

On the other hand, there are systems [1, 6] that analyze and summarize the context of a Web document (e.g. brief content descriptions from search engine results) instead of its contents.

Amitay and Paris [1] propose an approach that generates single-sentence long coherent textual snippets for a given Web page based on the context of the Web page, which is obtained by sending queries of the type "link:URL" to search engines such as Google. Experiments show that on average users prefer the snippets thus generated to the standard snippets provided by search engines.

Delort et al. [6] 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.

The drawback of the systems that rely on context analysis is that context information of the target Web documents is not always available and accessible. Consequently, approaches which analyze source contents have been gaining more popularity. However, they rely on the underlying key phrase extraction method to generate key phrases in order to further identify key sentences.

Automatic key phrase extraction has been a useful tool in many text related applications such as text clustering and document similarity analysis [21]. Traditional approaches to key phrase extraction are focused on frequency analysis such as TFIDF and collocation detection based on mutual information [18].

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 [16]. Turney proposes GenEx, a key phrase extraction system, which consists of a set of parameterized heuristic rules that are tuned to the training documents by a genetic program [28]. However, these methods heavily depend on heuristic rule pre-defining and tuning.

Research in [27, 32] evaluates key phrase extraction methods by matching automatically extracted key phrases with human authored ones. In [30], Turney defines *acceptable* key phrases as good and fair key phrases, which are rated by human subjects. In the WWW context, manually identifying key phrases is time-consuming because of the diversity and complexity nature of Web documents. Thus in our work, we ask human subjects to rate automatically extracted key phrases and then we are able to evaluate different methods using quantitative measures.

3 Automatic Web Site Summarization

Our automatic Web site summarization framework [35] is a multi-stage process as follows:

1. **Web Site Crawling** In order to summarize a given Web site, a certain number of Web pages within a short distance from the root (home page) of the target site, which are assumed to describe the main contents of the site in general terms, are collected by a specific Web crawler via the breadth-first search starting at the home page.

2. **Plain Text Extraction** After the Web pages have been collected, plain text is extracted from these pages and segmented into text paragraphs by the text browser *Lynx*[4], which is found to outperform several alternative text extraction tools such as *HTML2TXT*[5] and *html2txt*[6], in terms of more effective selection of plain text.

3. **Narrative Text Classification** Since Web documents are not well-structured and they often contain diverse contents such as short phrases and images, it is beneficial to have rules that can identify the text considered for summarization. This is achieved in two steps. First, a C5.0 classifier *LONGSHORT* is used to filter out *short* text paragraphs. Second, *long* paragraphs are classified into *narrative* or *non-narrative* by another C5.0 classifier *NARRATIVE*, and only narrative paragraphs are used in summary generation. These two classifiers are built based on features (e.g., number of words, part of speech tag) extracted by shallow natural language processing. The cross-validation shows a mean error of 5.9% and 11.3% for LONGSHORT and NARRATIVE respectively, which indicates the classification accuracy of the classifiers.

4. **Key Phrase Extraction** Traditionally, key phrases for the entire document corpus are extracted from plain text in order to generate a summary. Based on such key phrases, the most significant sentences, which best describe the source documents, can be retrieved. Key phrase extraction from a body of text relies on an evaluation of the importance of each candidate key phrase [4]. In this work, we experiment with five key phrase extraction methods on narrative text as detailed in Section 4, and investigate their performance in the automatic Web site summarization task.

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 density of key phrases [5]. The significance of a sentence is measured by calculating a weight value, which is the maximum of weights of all *word clusters* within the sentence.

[4] http://lynx.isc.org
[5] http://user.tninet.se/~jyc891w/software/html2txt/
[6] http://cgi.w3.org/cgi-bin/html2txt

A word cluster is defined as a sequence of words which starts and ends with a key phrase and at most 2 non-key-phrases must separate any two neighboring key phrases [4]. The weight of a word cluster is computed by adding the weights of all key phrases within the word cluster, and dividing this sum by the total number of key phrases. The weights of all sentences in all narrative text paragraphs are computed and the top five sentences (ranked by sentence weight) are the key sentences to be included in the summary.

6. **Summary Formation** The overall summary is formed by the top 25 key phrases and the top 5 key sentences. These numbers are empirically determined based on the fact that key sentences are more informative than key phrases, and the whole summary should fit in a single page. We aim to compare the five key phrase extraction methods under the summarization frame.

4 Key Phrase Extraction

In this section, we explain in detail the five key phrase extraction methods, i.e., TFIDF, KEA, Keyword, Keyterm, and Mixture. These methods generate single keywords or multi-word keyterms or a mixture of the above two by a critical evaluation of the significance of each candidate key phrase in the source documents.

We realize that these methods have been designed to extract key phrases from traditional well-structured text such as technical papers and news articles. Research in [36] demonstrates that application of key phrase extraction on Web documents relies on the identification of *narrative text*, which often contains more structured, informative and coherent information than non-narrative text. Here is an example of a narrative paragraph: *The Software Engineering Process Group (SEPGSM) Conference is the leading international conference and exhibit showcase for software process improvement (SPI).* In contrast, a non-narrative paragraph often consists of short phrases or bullets, e.g., *First created on 10 May 2000. Last Modified on 22 July 2003. Copyright ©2000-2003 Software Archive Foundation. All rights reserved.* Thus, we apply the *NARRATIVE* classifier introduced in [35] to extract narrative text. Then each key phrase extraction method will work on the narrative text only instead of all plain text.

4.1 TFIDF Method

TFIDF is a standard keyword identification method in information retrieval tasks. It gives preference to words that have high frequency of occurrence in a single document but rarely appear in the whole document collection. In this work, we aim to use TFIDF as a baseline method to extract keywords from pages of a given Web site. This involves in the following steps:

1. For each Web page of the target Web site, identify the narrative text and convert it to lower case.

2. Extract all tokens in the narrative text, i.e., identify single words by removing punctuation marks and numbers. A standard set of 425 stop words (*a, about, above, ...*) [7] are discarded at this stage.

3. Apply Porter stemming to obtain word stems and update the number of documents in which each word stem appears.

4. Once all Web pages are processed using the above three steps, calculate the TFIDF value $w_{i,j}$ of word stem i in page j using the following equation:

$$w_{i,j} = \frac{n_{i,j}}{|p_j|} \cdot \log \frac{N}{n_i} \tag{1}$$

   where $\frac{n_{i,j}}{|p_j|}$ is the normalized term frequency of word stem i in page j, n_i is the number of pages that contain word stem i, and N is the total number of Web pages in consideration.

5. For each Web page j, TFIDF values of all word stems in this page are normalized to unit length as follows:

$$W_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_i {w_{i,j}}^2}}. \tag{2}$$

6. Finally, choose the top five word stems ranked by $W_{i,j}$ for each page. The number 5 is chosen based on the fact that often 3 to 5 key phrases are included in a technical article. Then replace each word stem with its original form which appears most frequently in the collection (e.g., "engin" ("engineering": 8, "engineer": 2) → engineering).

4.1.1 Application of TFIDF on a Web Site

TFIDF is aimed towards extracting keywords from individual documents in a document collection rather than from the whole collection. Hence in order to generate a keyword list for an entire Web site, the output keywords from all pages should be combined properly. We aim to do the following:

1. Unite the 5 keywords from each Web page to obtain a single list. Each keyword (more precisely, its stem) i has a normalized weight $W_{i,j}$, as shown in Equation 2.

2. Record f_i, which is the number of pages in which keyword i appears[7]. Let W_i be the overall weight of keyword i in the Web site and A_i be its average weight. So $W_i = \sum_j W_{i,j}$, and $A_i = W_i / f_i$.

[7]This is again a document frequency concept. However, only those documents in which word stem i serves as a keyword are counted.

3. Now three features, i.e., W_i, A_i, and f_i can be used to re-rank the list in order to select the top 25 keywords for the target Web site. The number 25 is an empirical number set in the summarization framework [35]. Preliminary tests show that in terms of acceptable percentage (see 5.2.1), f_i is the best feature.

4. The top 25 keywords are taken as the keywords for the target Web site and their weights are re-normalized for the purpose of key sentence extraction.

4.2 KEA Method

KEA [32] is an efficient and practical algorithm for extracting key phrases, i.e., single keywords and multi-word keyterms. It consists of two stages: "training" and "extraction". In the training stage, KEA builds a Naïve Bayes classifier using training documents with human-authored key phrases. More explicitly, KEA chooses a set of candidate key phrases from input documents. For each candidate, two feature values, $tfidf$ and *first occurrence*, are calculated. First occurrence is calculated as the number of words that precede the candidate's first appearance, divided by the total number of words in the document. This is the normalized distance to the beginning of the document. Those candidates that are human-authored key phrases are positive examples in the KEA model construction. In the extraction stage, KEA uses the classifier to find the best set of (by default 5) key phrases in new documents. More explicitly, KEA chooses a set of candidate key phrases from new documents and calculate the two feature values as above. Then each candidate is assigned a weight, which is the overall probability that this candidate is a key phrase.

4.2.1 KEA Training

KEA is originally designed for key phrase extraction from traditional coherent text such as technical reports. In order to obtain a good KEA model for key phrase extraction from Web documents, we need to investigate whether KEA works well on diverse Web documents instead of traditional coherent text. Hence we build two KEA models as follows.

- The training set bundled with the Java-based KEA package (Version 3.0)[8] is used to train a *CSTR* KEA learning model. This data set contains 80 abstracts of Computer Science Technical Reports (CSTR) from the New Zealand Digital Library project[9]. Each abstract has 5 human-authored key phrases. The input to the Java program consists of text files with the corresponding key phrases. Research in [32] shows that a training set of 25 or more documents can achieve good performance.

- A total of 80 Web pages are randomly collected from 60 DMOZ Web sites. The criterion is that the Web page must have at least one narrative

[8] `http://www.nzdl.org/Kea`
[9] `http://www.nzdl.org`

paragraph identified by the NARRATIVE classifier described in [35]. We browse each Web page and extract up to five key phrases from its narrative text. Then a *NTXT* (Narrative TeXT) KEA model is constructed.

Web pages are different from technical reports in terms of the diversity of contents and discourse structure. Hence, we intentionally choose technological Web pages in order to eliminate the potential bias that the technical reports could have on building the CSTR model and to make these two models more comparable to each other.

We apply separately the CSTR model and the NTXT model to extract key phrases from the narrative text of Web pages. Preliminary experiments show that the NTXT model can extract key phrases with higher acceptable percentage so we use this model for key phrase extraction from Web pages of a given Web site.

4.2.2 Application of KEA on a Web Site

For the same reason as the application of TFIDF on an entire Web site, we aim to do the following:

1. Unite the 5 key phrases from each Web page to obtain a single list. Each key phrase i has a normalized weight $w_{i,j}$ in page j, which is the overall probability value provided by the KEA model.
2. Compute the same three features, i.e., W_i, A_i, and f_i, as in the application of TFIDF. Preliminary tests again show that f_i is the best feature in terms of acceptable percentage.
3. The top 25 phrases are chosen as the key phrases for the target Web site and their weights are re-normalized.

4.3 Keyword Method

The Keyword method introduced in [35] consists of two stages: C5.0 learning model construction and keyword identification. In both stages a set of candidate keywords are chosen from the target Web site, and then the values of certain features (e.g., frequency, part-of-speech tag) for each candidate keyword are calculated.

4.3.1 Learning Keywords

As discussed before, Web pages are different from traditional plain text documents. The existence of *anchor text* and *special text* (e.g., titles, headings, italic text) contributes much to the difference. Anchor text is the text associated with hyperlinks, and is often 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 candidate keywords.

In order to produce decision tree rules for determining the keywords of given Web site, a data set of 5454 candidate keywords from the 60 DMOZ Web sites is collected. For each Web site, the frequency of each unique word (after stemming) 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 the same. Stop words are discarded at this stage.

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 [3] are extracted. In particular, the weight of a candidate keyword is defined as *the ratio of its frequency (over three categories of text) to the sum of frequency of all candidate keywords*.

Next, each candidate keyword is labelled manually as *keyword* or *non-keyword*. The criterion to determine whether a candidate keyword is a true keyword is that the candidate 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.

The resulting decision tree shows that anchor text and special text do play an important role in determining keywords of a Web site [35]. Among the total 5454 cases, 222 cases are misclassified, leading to an error of 4.1%. The ten-fold cross-validation of the classifier shows a mean error of 4.9%, which indicates the accuracy of this classifier.

4.3.2 Keyword Identification

Once the decision tree rules for determining keywords have been built, they can be used to automatically extract keywords from a new Web site. First a list of candidate keywords is selected based on the same frequency analysis shown above and ranked by the weight. Then the classifier KEYWORD identifies all keywords in the list and the top 25 keywords are kept and used for key sentence extraction. It is observed that 40% to 70% of keywords appear in the home page of a Web site.

4.4 Keyterm Method

The Keyword method is based on word frequency analysis against three different categories of text, i.e., narrative text, anchor text, and special text. This method is unable to extract terms consisting of multiple words. Since multi-word terms are more informative than single words [21], we aim to apply a state-of-the-art method C-value/NC-value [9] (we rename it Keyterm) to extract multi-word keyterms from a Web site automatically, and further identify key sentences for summary generation.

4.4.1 Automatic Term Extraction

The C-value/NC-value method consists of both linguistic analysis (linguistic filter, part-of-speech tagging [3], and stop-list) and statistical analysis (frequency analysis, *C-value / NC-value*) to extract and rank a list of terms by *NC-value*. A linguistic filter is used to extract word sequences likely to be terms, such as noun phrases and adjective phrases.

The C-value is a domain-independent method used to automatically extract multi-word terms from the whole document corpus. It aims to get more accurate terms than those obtained by the pure frequency of occurrence method, especially terms that may appear as nested within longer terms. *C-value* is formally represented in Equation 3.

$$Cv(a) = \begin{cases} \log_2 |a| f(a), \ a \text{ is not nested.} \\ \log_2 |a| (f(a) - \frac{\sum_{b \in T_a} f(b)}{P(T_a)}), \text{ otherwise.} \end{cases} \tag{3}$$

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

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

$$weight(w) = \frac{t(w)}{n} \tag{4}$$

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

NC-value is formally given by Equation 5.

$$NCv(a) = 0.8 \times Cv(a) + 0.2 \times \sum_{b \in C_a} f_a(b) \cdot weight(b) \tag{5}$$

where a is a candidate term; C_a is the set of distinct context words of a; b is a word from C_a; $f_a(b)$ is the frequency of b as a term context word of a; and $weight(b)$ is the weight of b as a term context word. The two components of the *NC-value*, i.e., *C-value* and the context information factor, have been assigned the weights 0.8 and 0.2, respectively. These two coefficients were derived empirically [9].

Experiments in [9, 21] show that the C-value/NC-value method performs well on a variety of special text corpora. In particular, with the open linguistic filter `(Adj.|Noun)`$^+$`Noun` (one or more adjectives or nouns followed by one noun), the C-value/NC-value method extracts more terms than with the closed linguistic filter `Noun`$^+$`Noun` (one or more nouns followed by a noun) without much precision loss. For example, terms such as *artificial intelligence* and *natural language*

processing will be extracted by the open linguistic filter. Hence, in our work, we use this linguistic filter to extract terms from a Web site.

4.4.2 Keyterm Identification

The candidate term list C (ranked by *NC-value*) of a Web site contains some noun phrases (e.g. *privacy statement*), which, although they appear frequently in various Web sites, are not relevant to the core content of the Web sites and hence must be treated as Web-specific stop words [26]. We experimented with the 60 DMOZ Web sites used in the Keyword method and manually identified a stop list, L, of 51 noun phrases (e.g., *Web site*) [33]. The candidate term list C is filtered through the noun phrase stop list L, and only the top 25 terms (ranked by *NC-value*) are selected as keyterms.

4.5 Mixture Method

It is interesting to combine keywords and keyterms and see whether the mixed list of key phrases will bring in more benefit compared to using either keywords or keyterms alone in key sentence extraction. Our Mixture method works as follows:

1. Normalize the weights of 25 keywords to unit length. Do the same for 25 keyterms.
2. Combine 25 keywords and 25 keyterms to obtain a single list of 50 key phrases. In particular, the weight of each keyterm is assigned a factor λ, i.e., the new weight is $\lambda \cdot W_{Keyterm_i}$.
3. Our objective is to investigate whether keyterms should be given more weight than keywords when they are combined, i.e., determining $\lambda < 1$, $\lambda = 1$, or $\lambda > 1$. We experimented with various values of λ and found the best empirical value is 1.5 in terms of the acceptable percentage.
4. Sort the list of 50 key phrases with new weights and select the top 25 key phrases.
5. Re-normalize the new weights of the top 25 key phrases.

5 Experiments and Evaluation

In this section, we describe the methodology of our user study and present the evaluation results.

5.1 Experimental Methodology

We first present in this subsection how summaries of test Web sites are generated, and then discuss our study design.

5.1.1 Summaries of Test Web Sites

In our work, all five key phrase extraction methods are used to generate key phrases for 20 DMOZ Web sites, which are used in our previous summarization research [35]. These sites are randomly selected from four DMOZ subdirectories because they are either academic or commercial, and users have more familiarity with them. Also these sites are of varying size. The URLs are listed in Table 1.

Software/Software Engineering
1. http://www.ispras.ru/groups/case/case.html
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-global.com
18. http://www.commerceone.com
19. http://www.getgamma.com
20. http://www.rdmcorp.com

Table 1: URLs of the 20 test Web sites selected from four DMOZ subdirectories.

Furthermore, key sentences are extracted from each of the 20 Web sites based on the presence of key phrases [35]. Each Web site summary consists of 25 key phrases and 5 key sentences. Table 2 presents a Mixture-based summary for the Software Engineering Institute (SEI) Web site[10]. These summaries are printed out and presented to the human subjects in our user study outlined below.

5.1.2 Study Design

Evaluation of automatically generated summaries often proceeds in *intrinsic* mode, where summaries are compared against a gold standard, or in *extrinsic*

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

Part I. top 25 key phrases
engineering institute, software engineering institute, software engineering, system, software, product line, product, information, software architecture, carnegie mellon university, organization, architecture, capability maturity, institute, program, course, research, carnegie, capability maturity model, defense, development, team, department, term, component
Part II. top 5 key sentences
1. The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. 2. The 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) sponsors, co-sponsors, and is otherwise involved in many events throughout the year. 4. The Software Engineering Institute offers a number of courses and training opportunities. 5. The Software Engineering Institute (SEI) helps organizations and individuals to improve their software engineering management practices.

Table 2: A Mixture-based summary for the Software Engineering Institute Web site, consisting of 25 key phrases and 5 key sentences.

mode, which measures the utility of summaries in performing a particular task (e.g., site browsing).

In this work, we aim to investigate how well different types of summaries reveal the main contents of a given Web site[11]. In other words, we are interested in the correctness and completeness of the automatically generated summaries. Our assumption is that the subjects can define the most essential topic of a given Web site well enough for the most essential topic to be used as gold standard. To do so, we conducted a user study where summaries are judged by subjects using a golden standard of their own.

More explicitly, we conduct a user study in a "within-subjects" fashion where human subjects read and rate all five summaries of a given Web site (in sheets of paper) based on their understanding of how these summaries relate to the most essential topic of the target Web site. Our study is close to the intrinsic evaluation in the sense that human subjects rate the summaries against a hypothetical gold standard of their own. The study makes sense in that Web site summaries are expected to reveal the main contents of Web sites. Similar studies in which human subjects rate documents or phrases have been reported in [15, 17, 21, 30].

[11]We acknowledge that there are other critical factors in multi-document summarization such as coherence, redundancy deduction, and compression rate, which we leave for future research.

In our study, we focus on the "method" factor only. Other factors such as "subject" (inter-rater reliability) and "Web site" (e.g., academic vs. commercial) might also play a role in this learning task. Inter-rater reliability measures the rating agreement between subjects in a user study. It often calculates a score of how much consensus there is in the ratings given by subjects. There are a number of statistics that can be used to determine the inter-rater reliability. For example, the *joint-probability of agreement* is a simple measure, which takes the number of times each rating (e.g., 1, 2, ..., 5) is given by each subject and then divides this number by the total number of ratings. [31]. Investigation of these factors is a topic of future research.

For each given Web site, subjects are asked to do the following:

1. Browse the Web site and extract *the most essential topic*, which is defined as *the entity behind the Web site and its main activity*. The most essential topic serves as the representation of core contents of the target Web site. For example, the most essential topic for the SEI Web site could be extracted as "Software Engineering Institute at CMU for improvement of software engineering management and practice".

2. Read each of the five summaries of the target Web site, which are generated based on the five key phrase extraction methods, respectively.

3. Based on the *relatedness*, which is defined as *the extent to which a summary element (key phrase or key sentence) is related to the most essential topic*, rate summary elements using a 1-to-5 scale (1 = not related, 2 = poorly related, 3 = fairly related, 4 = well related, and 5 = strongly related).

We note that there are several "effects" such as fatigue and practice (warm-up) that could lead to "systematic bias", which means subjects give bias to a particular type of summary. One way to prevent such bias is to randomize the order in which five different summaries of a Web site are presented to subjects. More specifically, for each subject we choose 10 different presentation orders out of 120 possible permutations of five summaries such that the five summaries for each of the 10 Web sites are presented in a different order.

5.1.3 Study Recruitment

A related research reported in [4] asks 15 subjects to evaluate five summarization methods by collecting data such as number of pen movements in the task of browsing Web pages using handheld devices.

In another study [15], 37 subjects are asked to rate Web pages, which are returned by three different search engines, into "bad", "fair", "good", and "excellent" in terms of their utility in learning about the search topic. However, no specific statistical analysis methods are reported in these two studies.

In [20], 45 subjects are divided into four groups to perform task-based evaluation of multi-document summaries in order to determine whether multi-

document summaries measurably improve user performance when using online news browsing systems for directed research.

A size of 20 subjects is sufficient for our study. Each subject is asked to review 10 out of 20 Web sites such that each Web site is covered by exactly 10 subjects. This means that for each method, we have a sample size of 200 with replication.

Participants are graduate students in computer science with strong reading comprehension skills and Web browsing experiences. They are recruited because of the technical nature of the Web sites being summarized. Each subject is provided a computer with Internet access and summaries in hard copies. They are required to finish the study in a session of two hours.

5.2 Summary Evaluation

In this subsection, we explain how to compare the quality of key phrases and key sentences obtained by different methods based on statistical analysis of rating data collected in the user study. Our main objective is to benchmark the five key phrase extraction methods and investigate which method yields a Web site summary with the best quality.

For each key phrase extraction method, we have a sample size of 200 with replication. Let n_1, n_2, n_3, n_4, and n_5 be the number of summary elements that receive a score of 1, 2, 3, 4, and 5, respectively. Hence for each summary, $\sum_{i=1}^{5} n_i$ will be 25 for key phrases and 5 for key sentences, respectively.

5.2.1 Comparison of Key Phrases

We aim to evaluate and compare the five key phrase extraction methods by an analysis of both *acceptable percentage* and *quality value*, which are both calculated based on the rating data obtained in the user study.

Analysis of Acceptable Percentage Related research in [30] defines acceptable key phrases as those that are rated good or fair by human subjects. In our work, acceptable key phrases and key sentences are those that receive a score of 3, 4, or 5. These summary elements are reasonably related to the most essential topic of a given Web site. In other words, they correctly and completely reveal the main contents of the target Web site. The percentage, p, is then formally defined as:

$$p = \frac{n_3 + n_4 + n_5}{\sum_{i=1}^{5} n_i}. \tag{6}$$

The five methods TFIDF, KEA, Keyword, Keyterm, and Mixture achieve an average acceptable percentage of 0.55, 0.67, 0.59, 0.78, and 0.72, respectively. This indicates that the five methods can be ranked as TFIDF, Keyword, KEA, Mixture, and Keyterm, in ascending order of acceptable percentage of key phrases.

We apply the One-Way Fully Repeated Measures ANOVA on the acceptable percentage data and a significant difference between the five methods ($F_{4,995} = 23.421$, $P = 1.58E^{-18}$) is found at the 5% level. Further, we apply ANOVA on each pair of the five methods. The ANOVA results are presented in Table 3, which can be summarized as TFIDF $\sim$ Keyword $<<$ KEA $\sim$ Mixture $<$ Keyterm and KEA $<<$ Keyterm[12].

Method	KEA	Keyword	Keyterm	Mixture
TFIDF	$F = 19.121$ $P = 1.57E^{-5}$	$F = 2.224$ $P = 0.137$	$F = 72.495$ $P = 3.47E^{-16}$	$F = 37.071$ $P = 2.68E^{-9}$
KEA		$F = 8.048$ $P = 0.005$	$F = 17.765$ $P = 3.09E^{-5}$	$F = 3.051$ $P = 0.082$
Keyword			$F = 48.312$ $P = 1.50E^{-11}$	$F = 20.634$ $P = 7.38E^{-6}$
Keyterm				$F = 6.079$ $P = 0.014$

Table 3: $F_{1,398}$ and P values of applying ANOVA on each pair of the five key phrase extraction methods, i.e., TFIDF, KEA, Keyword, Keyterm, and Mixture, using the measure of acceptable percentage of key phrases.

Analysis of Quality Value In addition to the acceptable percentage measure, we also aim to compare the five methods using the *quality value* measure, which calculates the average correctness score of summary elements. The quality value, q, of 25 key phrases in a summary is defined as follows:

$$q = \frac{\sum_{i=1}^{5} n_i \times i}{\sum_{i=1}^{5} n_i}. \tag{7}$$

The higher the quality value, the more accurately the summary reveals the main contents of a site overall.

The acceptable percentage measure and the quality value measure are intrinsically related to each other as they are both based on users' ratings. The only difference is that the former gives equal weight to (i.e., a summation of) the number of summary elements with scores 3, 4, and 5, while the latter gives different weights to summary elements with different scores (i.e., number of such elements times the score they receive).

The average quality values of key phrases extracted by TFIDF, KEA, Keyword, Keyterm, and Mixture, are 2.85, 3.55, 3.46, 3.96, and 3.87 out of a possible 5.0, respectively. Hence the ordering of methods in terms of quality values is exactly the same as that obtained by the acceptable percentage measure. We

[12] $<<$ indicates a significant difference with $P_{value} \leq 0.01$; $<$ indicates a significant difference with $P_{value} \in (0.01, 0.05]$; $\sim$ indicates no significant difference with $P_{value} > 0.05$.

also apply ANOVA on the quality values data. We obtain the same result as using the acceptable percentage measure with the only exception that there is no significant difference between Mixture and Keyterm, i.e., Mixture $\sim$ Keyterm.

5.2.2 Comparison of Key Sentences

In our Web site summarization framework [35], once key phrases are identified by a particular method, we further extract key sentences based on the density of key phrases. We are interested in learning how good key sentences, which are obtained by using different key phrase extraction methods, will be from the user's point of view. Again, we are using both the acceptable percentage and quality value measures introduced in Equations 6 and 7, respectively.

The key sentences resulted from the five methods TFIDF, KEA, Keyword, Keyterm, and Mixture achieve an average acceptable percentage of 0.88, 0.90, 0.89, 0.90, and 0.91, respectively. The One-Way Fully Repeated Measures ANOVA on the acceptable percentage data shows that there is no significant difference between the five methods ($F_{4,995} = 0.490$, $P = 0.743$).

However, we note that compared with the ordering of key phrase extraction, KEA and Mixture have moved up in the ordering of key sentence extraction, i.e., KEA is tied with Keyterm compared to that KEA is worse than Keyterm in key phrase extraction, and Mixture is better than Keyterm compared to that Mixture is worse than Keyterm in key phrase extraction. This indicates that a mixture of single keywords and multi-word keyterms can improve the key sentence extraction performance. This also implies that key sentence extraction is often dominated by a few "good" key phrases, which are often ranked high in the key phrase list.

The average quality values of key sentences resulted from TFIDF, KEA, Keyword, Keyterm, and Mixture, are 3.87, 3.99, 3.94, 4.01, and 4.02, respectively. Hence the ordering of methods in terms of quality values is similar with that obtained by the acceptable percentage measure, i.e., the Mixture method is the best in terms of key sentence extraction. The ANOVA test shows that there is no significant difference between methods ($F_{4,995} = 1.145$, $P = 0.334$).

5.2.3 Comparison of Summaries

Each summary consists of 25 key phrases and 5 key sentences, so the evaluation of the whole summary depends on users' relative preference to different parts of the summary. A simple survey in our study indicates that users prefer to give equal weight to both parts of the summary. Thus the quality value of a summary will be $\frac{q_p+q_s}{2}$, where q_p and q_s are quality values of key phrases and key sentences (calculated using Equation 7), respectively. The ANOVA based on summary quality values shows that TFIDF $\sim$ Keyword $<$ KEA $\sim$ Mixture $\sim$ Keyterm and KEA $<$ Keyterm.

A thorough user study is needed in future research to see what is the best size for summary elements and how the methods compare with each other when the size of summaries change.

5.2.4 Comparison of Computational Cost

Regarding the computational complexity, it is observed that on average, TFIDF, Keyword and KEA are roughly 12 times faster than Keyterm in extracting key phrases from the narrative text of a given Web site. Keyterm is much slower mainly due to the computational complexity of *NC-value*. Hence in terms of summary quality, Keyterm is the best choice and Mixture is a good alternative in the automatic Web site summarization task, whereas if efficiency is the most important factor, then KEA is the best method.

5.2.5 Discussion

We evaluate the correctness and completeness of summary elements in an intrinsic manner, i.e., we measure how well different types of summaries could reveal the core contents of Web sites. We are interested in learning why and in what circumstances one method outperforms the other in this task.

It is not surprising that TFIDF is the worst method as it is conceptually simple to consider only features of term frequency and document frequency. The Keyword method is able to take advantage of topical information in three categories of text. However, it is mainly based on analysis of a word's overall frequency in the document collection. Consequently, its performance is at the same level as TFIDF. The KEA method utilizes both the TFIDF feature and the first appearance feature. It provides a learning scheme where prior knowledge of key phrases can be easily incorporated as the learning model is conceptually domain-independent. Hence, it can find a better set of key phrases than TFIDF and Keyword. The Keyterm method incorporates both statistical information (frequency, term nesting statistic, and contextual information) and linguistic knowledge. Consequently, it is able to find the best set of key phrases. Finally, the Mixture method has the advantage of obtaining a good mixture of single keywords and multi-word keyterms, which are found to greatly improve the performance of key sentence extraction.

Keyword and KEA are supervised methods which require known phrases from training documents in order to obtain the model. In contrast, TFIDF and Keyterm are unsupervised methods where no learning process is involved. Hence, they are more practical when applied to applications without domain knowledge. However, the Keyterm method is more sensitive to the amount of narrative text than the other three methods as it prefers more narrative text to conduct the *NC-value* calculation.

It will be ideal to apply the Mixture method to obtain a good set of candidate key phrases which can be further processed in consideration of Web-specific features such as availability of phrases in meta data and anchor text. Also more advanced learning algorithm such as Support Vector Machines can be deployed. This will be a direction of our future research.

6 Conclusion and Future Work

In this paper, we benchmark five automatic key phrase extraction methods, TFIDF, KEA, Keyword, Keyterm, and Mixture, in an extraction-based approach to automatic Web site summarization. These methods extract key phrases from the narrative text of a given Web site by applying information retrieval, machine learning and natural language processing techniques. Key phrases are in turn used to extract key sentences from the narrative text that form the Web site summary, together with the key phrases. We demonstrate that Keyterm is significantly better than TFIDF, KEA, and Keyword in the automatic Web site summarization task, i.e., summaries generated based on the Keyterm method can significantly better reveal the main topics and contents covered in the target Web site.

Future research involves several directions: 1) Investigation of the subject learning factor to determine whether there is a significant agreement difference within human subjects; 2) Estimation, via a user study, of the optimal number of key phrases and key sentences in summary formation, as well as their weights in summary quality calculation; 3) Investigation of the utility of different types of summaries in a particular task, e.g., asking human subjects to answer a predefined set of questions based on the Web site contents or a particular type of summaries; 4) Evaluation of other factors in multi-document summarization such as coherence, redundancy deduction, and compression rate.

Acknowledgements

This research has been supported by grants from the Natural Sciences and Engineering Research Council of Canada, GINIus Inc., and IT Interactive Services Ltd. We thank the anonymous reviewers for their detailed and constructive comments, that helped significantly improve the manuscript.

References

[1] E. Amitay and C. Paris. Automatically Summarising Web sites: Is There a Way Around It? In *Proceedings of the Ninth ACM International Conference on Information and Knowledge Management*, pages 173–179, McLean, VA, USA, November 6–11, 2000.

[2] 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*, pages 144–151, Athens, Greece, July 24–28 2000.

[3] E. Brill. A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, March 31–April 3 1992.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In *Proceedings of the Tenth International World Wide Web Conference*, pages 652–662, Hong Kong, China, May 01–05, 2001.

[5] 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*, pages 152–159, Athens, Greece, July 24–28, 2000.

[6] J. Delort, B. Bouchon-Meunier, and M. Rifqi. Enhanced Web Document Summarization Using Hyperlinks. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, pages 208–215, Nottingham, UK, August 26–30, 2003.

[7] C. Fox. Lexical Analysis and Stoplists. In *Information Retrieval: Data Structures and Algorithms*, pages 102–130, 1992.

[8] E. Frank, G. Paynter, I. Witten, C Gutwin, and C Nevill-Manning. Domain-specific Keyphrase Extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673, Stockholm, Sweden, July 31–August 06, 1999.

[9] K. Frantzi, S. Ananiadou, and H. Mima. Automatic Recognition of Multi-word Terms: the *C-value/NC-value* Method. *International Journal on Digital Libraries*, 3(2):115–130, August 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*, pages 121–128, Berkeley, CA, USA, August 15–19, 1999.

[11] J. Goldstein, V. Mittal, J. Carbonell, and J. Callan. Creating and Evaluating Multi-document Sentence Extract Summaries. In *Proceedings of the Ninth ACM International Conference on Information and Knowledge Management*, pages 165–172, McLean, VA, USA, November 6–11, 2000.

[12] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank. Improving Browsing in Digital Libraries with Keyphrase Indexes. *Journal of Decision Support Systems*, 27(1–2):81–104, November 1999.

[13] S. Jones and G. Paynter. Automatic Extraction of Document Keyprases for Use in Digital Libraries: Evaluation and Applications. *Journal of the American Society for Information Science and Technology*, 53(8):653–677, April 2002.

[14] S. Jones and G. Paynter. An Evaluation of Document Keyphrase Sets. *Journal of Digital Information*, 4(1), February 19, 2003.

[15] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, Septermber 1999.

[16] B. Krulwich and C. Burkey. Learning User Information Interests through the Extraction of Semantically Significant Phrases. In *AAAI Spring Symposium Technical Report SS-96-05: Machine Learning in Information Access*, pages 110–112, 1996.

[17] W. Lu, J. Janssen, E. Milios, N. Japkowicz, and Y. Zhang. Node Similarity in the Citation Graph. *Knowledge and Information Systems: An International Journal*, 11(1):105–129, January 2007. Available at `http://dx.doi.org/10.1007/s10115-006-0023-9`, last visited on February 12, 2007.

[18] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, June 18 1999. ISBN: 0-262-13360-1.

[19] 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, Madrid, Spain, July 1997.

[20] K. McKeown, R. Passonneau, D. Elson, A. Nenkova, and J. Hirschberg. Do Summaries Help? A Task-Based Evaluation of Multi-Document Summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 210–217, Salvador, Brazil, August 15–19, 2005.

[21] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic Term Extraction and Document Similarity in Special Text Corpora. In *Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics*, pages 275–284, Halifax, NS, Canada, August 22–25, 2003.

[22] H. Nakagawa. Experimental Evaluation of Ranking and Selection Methods in Term Extraction. In D. Bourigault, C. Jacquemin, and M. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 303–325, 2001.

[23] National Institute of Standards and Technology. Document Understanding Conferences. Available at `http://duc.nist.gov`, last visited on May 28, 2005.

[24] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[25] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN: 0070544840.

[26] M. Sinka and D. Corne. Towards Modernized and Web-Specific Stoplists for Web Document Analysis. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 396–402, Halifax, NS, Canada, October 13–16, 2003.

[27] P. Turney. Extraction of Keyphrases from Text: Evaluation of Four Algorithms. Technical Report ERB-1051 (NRC-41550), Institute for Information Technology, National Research Council of Canada, Ottawa, ON, Canada, October 23, 1997. Available at `http://citeseer.ist.psu.edu/turney97extraction.html`, last visited on November 26, 2004.

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

[29] P. Turney. Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data. Technical Report ERB-1096 (NRC-44947), Institute for Information Technology, National Research Council of Canada, Ottawa, ON, Canada, August 13, 2002. Available at `http://citeseer.ist.psu.edu/turney02mining.html`, last visited on November 26, 2004.

[30] P. Turney. Coherent Keyphrase Extraction via Web Mining. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 434–439, Acapulco, Mexico, August 9–15, 2003.

[31] Wikipedia. Inter-rater Reliability. Available at `http://en.wikipedia.org/wiki/Inter-rater_reliability`, last visited on December 6, 2006.

[32] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, pages 254–255, Berkeley, CA, USA, August 11–14, 1999.

[33] Y. Zhang, E. Milios, and N. Zincir-Heywood. A Comparison of Keyword- and Keyterm-Based Methods for Automatic Web Site Summarization. Technical Report CS-2004-11, Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada, October 2, 2004. Available at `http://www.cs.dal.ca/research/techreports/2004/CS-2004-11.shtml`, last visited on November 26, 2004.

[34] Y. Zhang, N. Zincir-Heywood, and E. Milios. Term-Based Clustering and Summarization of Web Page Collections. In *Advances in Artificial Intelligence, Proceedings of the Seventeenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 60–74, London, ON, Canada, May 17–19, 2004.

[35] Y. Zhang, N. Zincir-Heywood, and E. Milios. World Wide Web Site Summarization. *Web Intelligence and Agent Systems: An International Journal*, 2(1):39–53, June 2004.

[36] Y. Zhang, N. Zincir-Heywood, and E. Milios. Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora. In *Proceedings of the Seventh ACM International Workshop on Web Information and Data Management*, pages 51–58, Bremen, Germany, November 5, 2005.