Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora

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

ABSTRACT

Automatic key phrase extraction is a useful tool in many text related applications such as clustering and summarization. State-of-the-art methods are aimed towards extracting key phrases from traditional text such as technical papers. Application of these methods on Web documents, which often contain diverse and heterogeneous contents, is of particular interest and challenge in the information age. In this work, we investigate the significance of narrative text classification in the task of automatic key phrase extraction in Web document corpora. We benchmark three methods, TFIDF, KEA, and Keyterm, used to extract key phrases from all the plain text and from only the narrative text of Web pages. ANOVA tests are used to analyze the ranking data collected in a user study using quantitative measures of acceptable percentage and quality value. The evaluation shows that key phrases extracted from the narrative text only are significantly better than those obtained from all plain text of Web pages. This demonstrates that narrative text classification is indispensable for effective key phrase extraction in Web document corpora.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*performance evaluation (efficiency and effectiveness)*; H.3.1 [**Information Storage and Retrieval**]: Systems and Software—*linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*

General Terms

experimentation, performance

Keywords

narrative text classification, key phrase extraction, acceptable percentage

1. INTRODUCTION

WIDM'05, November 5, 2005, Bremen, Germany.

Key phrases, which can be either single keywords or multi-word keyterms[1], are known to be linguistic descriptors of documents [6]. They are often sufficiently informative to help human readers get a feel of the essential topics and main contents included in the source documents [18]. Key phrases have also been used as features in many text related applications such as text clustering [20], document similarity analysis [11, 17], and document summarization [2, 4, 19, 21].

Manually extracting key phrases from a number of documents is too expensive. Instead, automatic key phrase extraction is maturing and can be a good practical alternative. State-of-the-art methods [6, 8, 13, 15, 18]) are aimed towards automatic key phrase extraction from traditional text corpus such as a collection of technical papers in the same domain. In the information age, application of these methods on Web documents is of particular interest and significance.

1.1 Background and Motivation

Key phrase extraction in Web document corpora is a challenging task as Web documents are often less structured and contain more diverse contents (e.g., images, bullets, short phrases) than traditional text. Moreover, Web pages often contain many uninformative fragments (e.g., navigation bars, copyright notices). The text in these fragments may spoil the performance of key phrase extraction, which relies on term frequencies and similar statistics. Hence, the main idea of our work is to look only at the informative parts of Web pages.

In our previous work [21], we developed an extraction-based Web site summarization framework, which generates a concise summary by means of key phrase and key sentence extraction. One of the main contributions is the definition and design of "narrative text classification", behind which the main objective is to identify the narrative paragraphs from plain text[2] of Web pages to facilitate summary generation, since narrative text is more coherent and informative than non-narrative text.

Therefore, we propose a two-phrase extraction approach: first we filter out uninformative text using the "NARRATIVE" classifier reported in [21]; then we apply the usual key phrase extraction methods to the narrative text left. We are interested in learning the impact of narrative text clas-

[1]Hereafter, we use keywords, keyterms, and key phrases interchangeably, depending on the method context.
[2]By plain text we mean the text extracted from the HTML source by a HTML-to-text tool. The plain text often consists of both narrative and non-narrative paragraphs.

sification on the automatic key phrase extraction task, i.e., whether key phrase extraction methods can perform better by working on the narrative text only instead of all plain text.

1.2 Related Work

Traditional approaches to automatic key phrase extraction are focused on frequency analysis such as TFIDF and collocation detection based on mutual information [10]. Recently more effective systems have been developed. Krulwich and Burkey use a set of heuristic rules such as the use of acronyms and italics to extract key phrases from a document for use as features of automatic document classification [8]. Turney [15] 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. However, these two methods heavily depend on heuristic rule pre-defining and tuning. Song et al. [13] introduce a method which uses the information gain measure to rank the candidate key phrases based on the *tf-idf* and *distance* features, which were first proposed in KEA [18].

In this work, we choose and benchmark three key phrase extraction methods, **TFIDF**, **KEA**, and **Keyterm**. The first method, TFIDF, captures a candidate keyword's frequency of occurrence 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 (Automatic Keyphrase Extraction) [18], 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. The third method, Keyterm (named C-value/NC-value by the authors [6]), consists of both linguistic and statistical analysis to extract multi-word keyterms automatically. It was designed for key phrase extraction from a whole document corpus. We acknowledge that both TFIDF and KEA were originally designed for extracting key phrases from single documents. We extend them to apply on a whole Web document collection. In doing so, our objective is to learn whether identification of narrative text from a document corpus will improve the performance of key phrase extraction.

1.3 Research Objective

We aim to conduct a formal user study in order to investigate whether there is a significant difference between the two sets of key phrases, which are obtained by each method by working on all plain text and on the narrative text only, respectively. We compare the key phrases extracted from two text sources in terms of "acceptable percentage", which is the ratio of key phrases acceptable[3] to human readers. We also quantify them to measure the quality difference between the two sets of key phrases. The fully repeated measures ANOVA shows that key phrases extracted from the narrative text only are significantly better than those obtained from all plain text of Web pages. We also found that Keyterm is significantly better than KEA, which further significantly outperforms TFIDF.

The rest of the paper is organized as follows. Section 2 explains why and how to identify narrative text in Web documents. In Section 3, we describe the three key phrase extraction methods. Section 4 discusses the design of our experiments and shows the evaluation results. Finally, Section 5 concludes our work and describes future research directions.

[3]By acceptable we mean that the key phrases are sufficiently informative and that they are related to the essential contents of the Web site under consideration.

2. NARRATIVE TEXT CLASSIFICATION

In order for key phrase extraction methods to work on Web documents, the HTML source code of Web pages should be parsed (including removal of HTML tags, scripts, etc.) and converted to **plain text** using HTML-to-text tools. The text browser, *Lynx*[4], is found to outperform alternative text extraction tools such as *html2txt*[5] in this task [21]. A downside of the Lynx approach is that it does not utilize contextual cues provided by HTML tags. It is interesting to consider alternative tools such as Tidy[6] to collect structural and syntax cues in order to better understand the contents embedded in HTML.

As we know, Web pages often contain diverse contents such as tables of contents, link lists, and "service" sentences (e.g., copyright notices, Web master information). Consequently, the extracted plain text is much less coherent and more diverse than traditional text. Hence, we need to determine which part of plain text is suitable for key phrase extraction. The process of narrative text classification [21] is for such a tool to extract narrative content and discard non-narrative text. It consists of two steps, i.e., *long paragraph classification* and *narrative paragraph classification*.

2.1 Long Paragraph Classification

Some text paragraphs in the plain text of Web pages are observed to be too short (in terms of number of words, number of characters, etc.) and to contain insufficient context word information for automatic key phrase extraction, 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 words. However, two more features, *number of characters including punctuation*, and *number of characters without punctuation*, might also play key roles. Thus instead of setting up a simple threshold for each feature, we let the decision tree learning program C5.0 [7] determine which feature is the most important one.

A total of 700 text paragraphs is extracted from 100 Web pages which are randomly collected from 60 DMOZ[8] Web sites. Statistics of the aforementioned three attributes are recorded for each text paragraph. Then each paragraph is manually labelled as *long* or *short*, and C5.0 is used to construct a classifier, *LONGSHORT*, for this task.

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 misclassified cases, leading to an error of 5.1%. The cross-validation of the classifier LONGSHORT shows a mean error of 5.9%, which indicates the accuracy of this classifier.

[4]`http://lynx.isc.org`
[5]`http://cgi.w3.org/cgi-bin/html2txt`
[6]`http://tidy.sourceforge.net`
[7]`http://www.rulequest.com/see5-unix.html`
[8]`http://www.dmoz.org`

2.2 Narrative Paragraph Classification

Intuitively, a narrative paragraph contains informative and coherent text, whereas a non-narrative paragraph often consists of more noise words. 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 such as *First created on 10 May 2000. Last Modified on 22 July 2003. Copyright ©2000-2003 Software Archive Foundation. All rights reserved.*

Analysis of part-of-speech patterns has proved to be effective in several Web-based applications such as query ambiguity reduction [1] and question answering [12]. It is hypothesized that the relative frequencies of the part-of-speech tags of the words in a paragraph contain sufficient information to identify the paragraph as narrative or non-narrative. To test this hypothesis, we generate a training set of 9763 text paragraphs, which are extracted by Lynx from 1000 Web pages randomly collected from the 60 DMOZ Web sites. In this set, there are 3243 paragraphs classified as long. Next, part-of-speech tags for all words in these paragraphs are given using a rule-based part-of-speech tagger [3].

Tag	Meaning & Example
CC	conjunction (and, or)
CD	number (four, fourth)
DT	determiner, general (a, the)
EX	existential (there)
FW	foreign word (ante, de)
IN	preposition (on, of)
JJR	adjective, comparative (lower)
JJS	adjective, superlative (lowest)
JJ	adjective, general (near)
MD	modal auxiliary (might, will)
NNPS	noun, proper plural (Americas)
NNP	noun, proper singular (America)
NNS	noun, common plural (cars)
NN	noun, common singular (car)
PRP$	pronoun, possessive (my, his)
PRP	pronoun, personal (I, he)
RBR	adverb, comparative (faster)
RBS	adverb, superlative (fastest)
RB	adverb, general (fast)
SYM	symbol or formula (US$500)
TO	infinitive marker (to)
UH	interjection (oh, yes, no)
VBD	verb, past tense (went)
VBG	verb, -ing (going)
VBN	verb, past participle (gone)
VBP	verb, (am, are)
VBZ	verb, -s (goes, is)
VB	verb, base (go, be)
WDT	det, wh- (what, which)
WP$	pronoun, possessive (whose)
WP	pronoun (who)
WRB	adv, wh- (when, where, why)

Table 1: The list of 32 part-of-speech tags used in narrative paragraph classification.

After part-of-speech tagging, attributes of percentage values of 32 part-of-speech tags [21] (see Table 1) are extracted from each paragraph. Two more attributes, *number of characters* and *number of words* in the paragraph, are added to this set. 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 cases.

There are 5 rules in the resulting decision tree. Among the 3243 cases, 63.5% are classified using 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 misclassified cases, leading to an error of 8.0%. The cross-validation of the classifier NARRATIVE shows a mean error of 11.3%, which indicates the accuracy of this classifier.

The decision about text informativeness is based on purely text-based features, i.e., paragraph length and part-of-speech statistics. Web-specific features, such as the visual Web page structure and the visual attributes of the text, might be helpful in determining the informative text. It is interesting to see whether taking into account the web-specific features will improve the performance of narrative text classification. We are also interested in investigation of state-of-the-art methods such as Support Vector Machines in the classification tasks described above. These will be part of our future research.

3. KEY PHRASE EXTRACTION

In this section, we describe the three methods we choose, i.e., TFIDF, KEA, and Keyterm. 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. Each method is used to extract key phrases from all plain text and from only the narrative text of a given Web site, respectively.

3.1 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. This involves in the following steps:

1. For each Web page of the target Web site, identify the narrative text[9] 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, ...*) [5] is 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

[9]It is possible that a Web page does not contain any narrative text paragraph. However, a given Web site often has a fair amount of narrative text, from which we aim to extract key phrases.

in page j using the following equation:

$$w_{i,j} = \frac{n_{i,j}}{|p_j|} \cdot \log_2 \frac{N}{n_i} \quad (1)$$

where $n_{i,j}$ is the frequency of word stem i in page j, $|p_j|$ is the number of word stems 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 1.0 as follows:

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

6. Finally, choose the top five word stems ranked by normalized TFIDF values 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.

3.1.1 Application of TFIDF on a Web Site

TFIDF is aimed towards extracting keywords from a single document rather than a whole document collection. Thus in order for TFIDF to generate a keyword list for an entire Web site, the output keywords from all pages should be combined somehow. 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 the number f_i of pages in which keyword i appears. 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 $A_i = W_i/f_i$.
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 used in the summarization framework [21]. Preliminary tests show that in terms of acceptable percentage (see 4.2.1), f_i is the best feature.
4. Replace each word stem by its original form which appears most frequently in the collection (e.g., "engin" ("engineering": 8, "engineer": 2) → engineering).

3.2 KEA Method

KEA [18] 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 learning model 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 number of words in the document. Those candidates that happen to be human-authored key phrases are positive examples in the KEA model construction. In the extraction stage, KEA uses the model 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 calculates their two feature values as above. Then each candidate is assigned a weight, which is the overall probability that this candidate is a key phrase.

3.2.1 KEA Training

KEA is a domain-independent method [18], which means a KEA model trained on one domain (e.g., computer science) performs well on another domain (e.g., biology). The training set bundled with the Java-based KEA package (Version 2.0)[10] 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[11]. 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 [18] shows that a training set of 25 or more documents can achieve good performance. We apply the obtained CSTR model to extract key phrases from all plain text and from only the narrative text of new Web pages, respectively.

3.2.2 Application of KEA on a Web Site

Similar to the application of TFIDF, to apply KEA on an entire Web site, the following is performed:

1. Unite the 5 key phrases from each Web page to obtain a single list. Each key phrase i has a weight, $w_{i,j}$, in page j, which is an 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 show that W_i is the best feature for KEA 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.

3.3 Keyterm Method

The Keyterm method is the application of the C-value/NC-value [6] method to the extraction of key terms from a Web document corpus.

3.3.1 Automatic Term Extraction

The Keyterm 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 method 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.

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

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 method, an extension to the C-value method, incorporates information of context words into term extraction. Context words are those that appear in the vicinity of

[10] `http://www.nzdl.org/Kea`

[11] `http://www.nzdl.org`

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:

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

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.

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

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 [6].

Experiments in [6, 11] 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 one 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 the open linguistic filter to extract terms from a Web site.

3.3.2 Keyterm Identification

From the candidate term list C (ranked by *NC-value*) of a given Web site, we choose the top 25 terms as the key phrases for the given Web site.

4. EXPERIMENTS AND EVALUATION

In this section, we first describe our user study and then present evaluation results.

4.1 Experimental Methodology

In our work, 20 DMOZ Web sites are randomly selected from four DMOZ subdirectories. The Web sites are of varying size and focus. The URLs of these test Web sites are listed in Table 2.

For a given Web site, each of the three key phrase extraction methods is used to extract the top 25 key phrases from all plain text and from the narrative text only. This leads to a total of six key phrase lists for the target Web site. Table 3 presents two key phrase lists generated by KEA from the Software Engineering Institute (SEI) Web site[12].

As shown in Table 3, key phrases such as "sei" and "software engineering" are captured from both text sources. However, there are 7 meaningless key phrases (marked in boldface) from all plain text while only one from the narrative text. In particular, for all plain text there are 2, 4, and 6 meaningless key phrases in the top 5, 10, and 15 key phrases,

[12] `http://www.sei.cmu.edu`

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

from all plain text	from the narrative text
sei	sei
o-blank	software
navigation buttons	software engineering
software engineering insti-	software engineering insti-
tute	tute
software	development
software engineering	systems
publications	architecture
white space	engineering
transparent	process
architecture	software architecture
systems	technology
information	product
engineering	improvement
contact	payment
sq-blank	program
page	**transparent**
product	information
software architecture	research
program	model
development	process improvement
member	publications
services	acquisition
model	member
technology	services
process	product line

Table 3: Key phrases extracted by KEA from all plain text and from only the narrative text of the Software Engineering Institute Web site.

respectively. Such key phrases as "o-blank" has nothing to do with the essential contents of the SEI Web site and obviously they should be discarded. This indicates that extraction of key phrases from the narrative text only can obtain the significant key phrases while eliminating most of the meaningless key phrases at the same time.

Moreover, it appears that most of the meaningless key phrases are those quite specific to Web. It is of interest to design a simple method, which is able to seek and remove such Web-specific phrases. Such a method can be used as a baseline and compared to the more sophisticated approach of identifying narrative text first.

4.1.1 User Study Design

Research in [14, 18] evaluates key phrase extraction methods by matching automatically extracted key phrases with human authored ones instead of precision and recall. In [16], Turney defines *acceptable* key phrases as good and fair key phrases, as ranked by human subjects.

In this work, we conduct a user study where human subjects read and rank key phrases of a given Web site based on their understanding of how these key phrases relate to the essential topics of the target Web site. This means human subjects rank the key phrases against a hypothetical gold standard of their own. The study makes sense in that these key phrases are mainly created for the purpose of facilitating Web users in navigation and understanding of Web sites. Similar studies where human subjects rank documents or phrases have been reported in [7, 9, 11, 16].

In our study, we focus on the "source of text" factor. We understand that other factors such as "subject" (inter-rater reliability) and "Web site" (e.g., academic vs. commercial) might play an important role in this learning task. Investigation of these factors is a direction of future research. For each given Web site, subjects are asked to do the following:

- 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 main 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".

- Read each of the six key phrase lists (2 text sources by 3 methods) of the target Web site. Based on the *relatedness*, which is defined as *the extent to which a key phrase is related to the most essential topic*, rank key phrases 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 particular type of key phrases. One way to prevent this is to randomize the order in which the six key phrase lists of a Web site are presented to subjects.

We also observe that users' background might create bias, i.e., the users would not be able to judge well key phrases of topics they are not familiar with. However, this should not have a big impact on our study because: 1) we focus on the *source of text* factor instead of the *topic* (Web site) factor; 2) the 20 Web sites we choose are all in the information technology area and computer science graduate students should be able to do reasonable ranking.

4.1.2 Study Recruitment

A related research reported in [4] asked 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 [7], 37 subjects were asked to rank 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 were reported in these two studies.

We chose a size of 10 subjects in our study. Each subject was asked to review 10 out of 20 Web sites such that each Web site is covered by exactly 5 subjects. This means that for each key phrase list, we have a sample size of 100 with replication. Participants were graduate students in computer science with strong reading comprehension and Web browsing experience.

4.2 Evaluation Results

In this subsection, we explain how to compare the quality of key phrases obtained from all plain text and from only the narrative text via a statistical analysis of both *acceptable percentage* and *quality values*, which are calculated based on the ranking data collected in the user study. Our main objective is to learn whether narrative text classification can make a significant difference in the key phrase extraction task.

For each of the six key phrase lists, we have a sample size of 100 with replication. Let n_1, n_2, n_3, n_4, and n_5 be the number of key phrases that receive a score of 1, 2, 3, 4, and 5, respectively. Hence $\sum_{i=1}^{5} n_i$ will be 25.

4.2.1 Analysis of Acceptable Percentage

We are interested in the extent to which these key phrases are acceptable to human readers. Related research in [16] defines *acceptable* key phrases as those that are ranked good or fair by human subjects. In our work, acceptable key phrases are those that receive a score of 3, 4, or 5. Hence, the percentage, p, is formally defined as:

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

Table 4 summarizes the mean and variance of acceptable percentage over 100 replications achieved by three methods from two text sources. For example, on average only 38.2% (9 out of 25) key phrases extracted by the TFIDF method from the plain text are acceptable to human readers, leading to a variance of 0.086. In contrast, the same method can achieve as high as 49.4% (12 out of 25) key phrases from the narrative text.

As we can see in Table 4, all three methods achieve a better set of key phrases from only the narrative text than from all plain text. It is also clear that Keyterm outperforms KEA, which further outperforms TFIDF.

We apply the pairwise ANOVA with fully repeated measures on the acceptable percentage data and a significant difference at the 5% level is found between the application of all three methods on all plain text and on the narrative text only, which is summarized in Table 5. For example,

Text/Method	TFIDF	KEA	Keyterm
All plain text	.382/.086	.555/.072	.640/.061
Narrative text	.494/.082	.635/.074	.719/.055

Table 4: The mean and variance of acceptable percentage over 100 replications achieved by the three methods from all plain text and from the narrative text only, respectively.

when the KEA method is applied, key phrases extracted from the narrative text only are significantly better than those extracted from all plain text.

Text/Method	TFIDF	KEA	Keyterm
all plain text vs. **narrative text**	<<	<	<

Table 5: ANOVA results for comparisons of key phrase extraction from all plain text and from the narrative text only, using the measure of acceptable percentage of key phrases. << indicates $P_{value} \leq 0.01$; < means $P_{value} \in (0.01, 0.05]$. In both cases, a significant difference at the 5% level is found.

We also compare the three methods when they are applied on either all plain text or only the narrative text. A significant difference between any two methods is found in both cases, which is summarized in Table 6. For example, when applied on the narrative text only, Keyterm significantly outperforms KEA, which further significantly outperforms TFIDF. This is not surprising, as KEA uses both tf-idf and distance features; and Keyterm goes one step further, where contextual information and linguistic knowledge is incorporated in key phrase extraction.

Method/Text	all plain text	narrative text
TFIDF vs. KEA	<<	<<
TFIDF vs. Keyterm	<<	<<
KEA vs. Keyterm	<	<

Table 6: ANOVA results for comparisons of key phrase extraction methods working on either all plain text or only the narrative text, using the acceptable percentage measure.

4.2.2 Analysis of Quality Values

In addition to the acceptable percentage measure, we also aim to compare the key phrase lists using the quality value measure. The quality value, q, of 25 key phrases in a list is defined as follows:

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

The difference between the acceptable percentage measure and the quality value measure is that the former gives equal weight to (a summation of) the number of key phrases with scores 3, 4, and 5, while the latter gives different weight to key phrases with different scores (number of such elements times the score they receive).

The mean and variance of quality values of key phrases extracted by the three methods from two different text sources are summarized in Table 7.

Text/Method	TFIDF	KEA	Keyterm
All plain text	2.27/.815	2.99/.236	3.19/.193
Narrative text	2.76/.745	3.15/.259	3.34/.222

Table 7: The mean and variance of quality values (out of a possible 5) over 100 replications achieved by the three methods from all plain text and from the narrative text only, respectively.

We also apply ANOVA on the quality values data. We obtain the same result as using the acceptable percentage measure, i.e., key phrase extraction from only the narrative text significantly outperforms extraction from all plain text. We also find that when using the quality value measure, Keyterm significantly outperforms KEA at the << level and KEA significantly outperforms TFIDF at the << level.

The above observations can be explained by the intrinsic relationship of the acceptable percentage measure and the quality value measure, as they are both based on users' rankings.

5. CONCLUSION AND DISCUSSION

In this paper, we study the significance of narrative text classification in the task of automatic key phrase extraction in Web document corpora. We benchmark three methods, TFIDF, KEA, and Keyterm, used to extract key phrases from all plain text and from only the narrative text of 20 test Web sites, respectively. We demonstrate that narrative text classification can significantly improve the performance of key phrase extraction from Web document corpora.

Future research involves several directions: 1) A comparative study of classification methods such as Support Vector Machines in the narrative text classification task; 2) Investigation of incorporating Web-specific features to improve key phrase extraction in Web document corpora; 3) Key phrase extraction from Web pages returned by a search engine for better query formation; 4) Estimation, via a user study, of the optimal number of key phrases to be presented to Web users.

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 are grateful to the reviewers for their constructive comments.

6. REFERENCES

[1] J. Allan and H. Raghavan. Using Part-of-speech Patterns to Reduce Query Ambiguity. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, Tampere, Finland, August 11–15, 2002.

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

[6] 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.

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

[8] 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.

[9] W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node Similarity in Networked Information Spaces. Technical Report CS-2001-03, Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada, September 26, 2001.

[10] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, June 18 1999.

[11] 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.

[12] D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic Question Answering on the Web. In *Proceedings of the Eleventh International World Wide Web Conference*, pages 408–419, Honolulu, Hawaii, USA, May 7–11, 2002.

[13] M. Song, I. Song, and X. Hu. KPSpotter: A Flexible Information Gain-based Keyphrase Extraction System. In *Proceedings of the Fifth ACM International Workshop on Web Information and Data Management*, pages 50–53, 2003.

[14] 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.

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

[16] 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.

[17] I. Witten. Browsing around a Digital Library. In *Proceedings of the Australasian Computer Science Conference*, pages 1–14, 1999.

[18] 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.

[19] Y. Zhang, E. Milios, and N. Zincir-Heywood. A Comparison of Keyword- and Keyterm-Based Methods for Automatic Web Site Summarization. In *Technical Report WS-04-01, Papers from the AAAI'04 Workshop on Adaptive Text Extraction and Mining*, pages 15–20, San Jose, CA, USA, July 26, 2004.

[20] 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.

[21] 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.