Abstracting Keywords from Hypertext Documents

BEN CHOI & BAOLIN LI

Computer Science, College of Engineering and Science
Louisiana Tech University, Ruston, LA 71272, USA
pro@BenChoi.org

Abstract. This paper presents a process for abstracting keywords from hypertext or text documents. The abstracted keywords, like keywords listed in a paper, identify the contents of a document. Our proposed process can be used, for example, to identify the contents of HTML documents returned from a search engine, to allow users to quickly find their needed information. The proposed process not only considers the occurrent frequency of a word in a document, like other related works, but also considers the occurrent frequency of its synonyms. It also considers key phrases consisting of two or three words. To increase the accuracy of the frequency count of words, a stemming algorithm is used to remove suffixes. Our tests show that the stemming algorithm consumed on average 56.7% of the total computation time, and that the proposed process can on average abstract 52% of the keywords provided by the authors of the tested documents.

Keywords: web mining, keyword extraction, information retrieval, and hypertext

1. Introduction

Keywords in a technical paper highlight the contents of the paper. They are in general the most compact, although usually incomplete, ways to summarize the contents. A reader can simply scan the keywords to get a general idea of the subject matters covered in the paper. We propose to use keywords to summarize the results of Web pages returned from a search engine. The purpose is to allow users to quickly scan the search results to find the information they needed in shorter time. However, Web pages or hypertext documents usually do not include a keyword list. We propose a process to automatically abstract keywords from hypertext or text documents.

Our proposed process not only considers the occurrent frequency of a word in a document, like other related works, but also considers the occurrent frequency of its synonyms. It also considers key phrases consisting of two or three words. To increase the accuracy of the frequency count of words, a stemming algorithm is used to remove suffixes.

Ben Choi and Baolin Li, "Abstracting Keywords from Hypertext Documents," International Conference on Information and Knowledge Engineering, pp. 137-142, 2002.

Our proposed process, in fact, is relatively simple comparing to many other related researches. Researchers have used various complex techniques for extracting keywords form documents, such as, Machine Learning [1], Fuzzy system [2], Neural Networks [3][4], and Self Organization Map [5]. For example, Zhang [3] used neural networks for automatic identification of keywords. The networks are trained to recognize keywords on the basis of their relationships to seed words that are manually selected to indicate a subject area. After training, the networks can be used to extract keywords automatically from other documents and assign subject area to the documents.

Our proposed method incorporated several key features found in relatively simpler statistical techniques. For instance, Sheth et al. [6] compared computational linguistics methods and statistical weighting methods for generating keywords. They found that computational linguistics analysis is expensive to implement but does not perform as well. Thus, they proposed to use weighting methods of simple word count for extracting keyword from a document. Hulth et al. [7] argued that frequency analysis of the terms found in the document text might be the primary source of knowledge about the document. They also proposed to include a hierarchically organized domain specific thesaurus as a second source of knowledge. Jenkins et al. [8] also used frequency analysis of terms but the weight associated with a term depends on where it was found within the document.

2. Keyword Generation Process

Our proposed keyword generation process is based on the following hypotheses:

- A keyword is not a stop word. A stop word means a word that is commonly used and has no topic meanings when treated as a single word.
- Different words having the same stem have similar or related topic meanings. For instance, car and cars express the same topic. A simple and effective way of keyword generation can be based on the morphological transformation of language. The most common morphological method is stemming. Stemming process removes the suffixes and retains the stem that affects the meaning of the word.
- Frequently used words represent the topic of a document in greater degree than less frequently used words. Usually an author will repeatedly use keywords to explain the subject matters.
- The earlier the word appears in a document, the more important it is in representing the topic of the document. Usually the main topic of a document will appear in the early part of the document.
- A word and its synonyms represent related concepts. To make an article more readable, writers often use

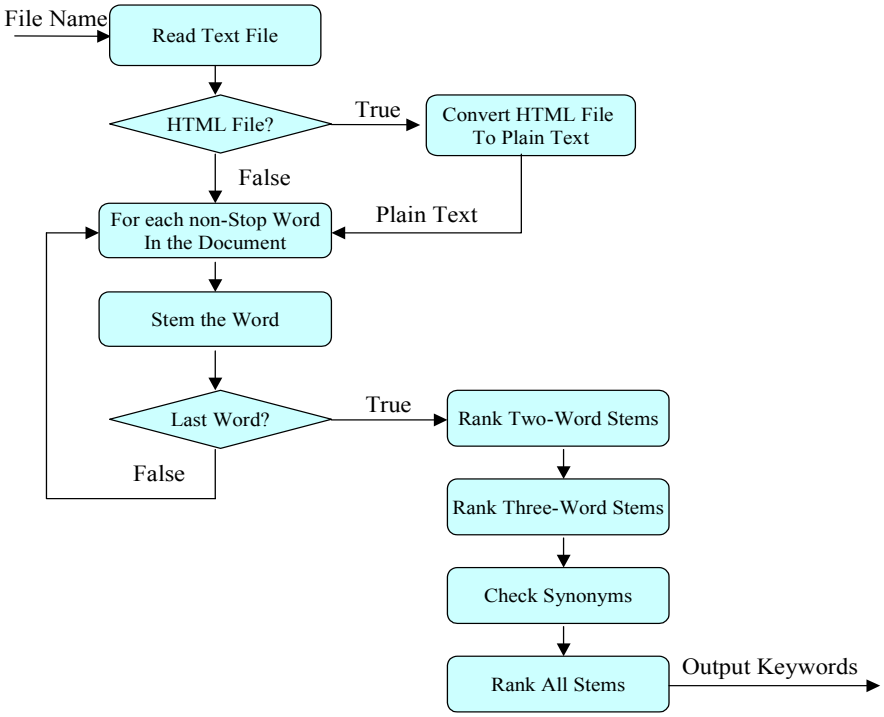

Figure 1. Keywords generation process

synonyms to avoid high repetition of words.

- Keyword can be a single, a sequence of two, or more non-stop words. However, keyword consisting more than three words are not as common. Thus, the number of words in a keyword or phrase will be limited to three in our keyword generation process.
- A keyword consisting of a single word is usually not an adjective or adverb. An adjective or adverb cannot by itself represent a keyword since adjectives or adverbs are words used to modify other words. However, when an adjective is combined with other words, it can form a two-word or three-word keyword.

The proposed process is illustrated by a flowchart (Figure 1) and outlined as follow. To abstract keywords from an HTML document, contents of the document are first abstracted. HTML tags and scripting codes are removed. Stop words are then removed. Synonyms of each of the remaining non-stop words are listed. Stemming algorithm is then used to remove suffixes of the remaining non-stop words. The resulting stems are ranked based on the occurrent frequency of the stems and their synonyms. When ranking the stems, single word stems, two consecutive stems, and three consecutive stems will be used to form

the final keyword stems list. The process is also applicable to plaint text documents. In this case, the conversion step from HTML document to plain text format will be ignored and the program will directly go to the stemming step (see Figure 1). The process is described in more detail in the following.

Stemming Words: There are two widely used stemming algorithms: Lovins algorithm and Porter's algorithm. Lovins algorithm specifies 260 suffix patterns and uses an iterative heuristic approach. The Porter's algorithm is a simpler version than Lovins algorithm. It uses 60 rules that are organized into sets. Conflicts within a set of rules are resolved before applying the next set. The rules are also separated into five distinct phases numbered 1 to 5. They are applied to the words in a document from phase 1 moving on to phase 5. Each phase will remove a type of suffix of the word. After the five stages, the stem of the word will be left. Since Porter's algorithm is simpler and faster, we employed Porter's algorithm in our implementation. During the process, all stop words and words having less than three letters will also be dropped, and any upper case characters will be changed to lower case.

Ranking Single Word Stems: Frequency count will be used for ranking single word stems. A position factor will be used to boost the scores for those stems that appear in the early part of a document. If a stem appears in the first third of the document, a factor of 1.3 will be given; if the stem is first used in the second third of the document, a factor of 1 will be assigned to it; and if the stem appears in the last third, a factor of 0.8 will be designated to it. Since a single-word keyword is not likely to be an adjective or adverb, the score for an adjective or adverb will be set to zero. The total score for a stem is equal to the sum of the frequency counts multiplying by the position factor. The stems will be ranked according to their total scores.

Ranking Two-Word and Three-Word Stems: After single word stems are obtained, the stems are check for two or three words phrases. For two consecutive stems, if the first word is an adjective, the two-word phrase qualifies as keyword. If the second word is an adjective or adverb, then the score of phrase is set to zero. Similarly, if the second or the third word of a three-word phrase is an adjective, then the score for the phrase is set to zero.

Checking for Synonyms: Our proposed process checks synonyms for each of the non-stop words. The ranking of a word is increased based on the number of its synonyms occurring within the document. Since a synonym is less important than the word itself, a factor having a value less than one, in our implementation 0.8, is used for each of the synonyms. The synonyms are taken from [9].

Ranking All Stems: All of the stems, including single word stems, two-word stems, and three-word stems, will be ordered according to their total scores. The words or phrases having top scores will be outputted as keywords of the document. In our implementation, users can specify the number of keywords to

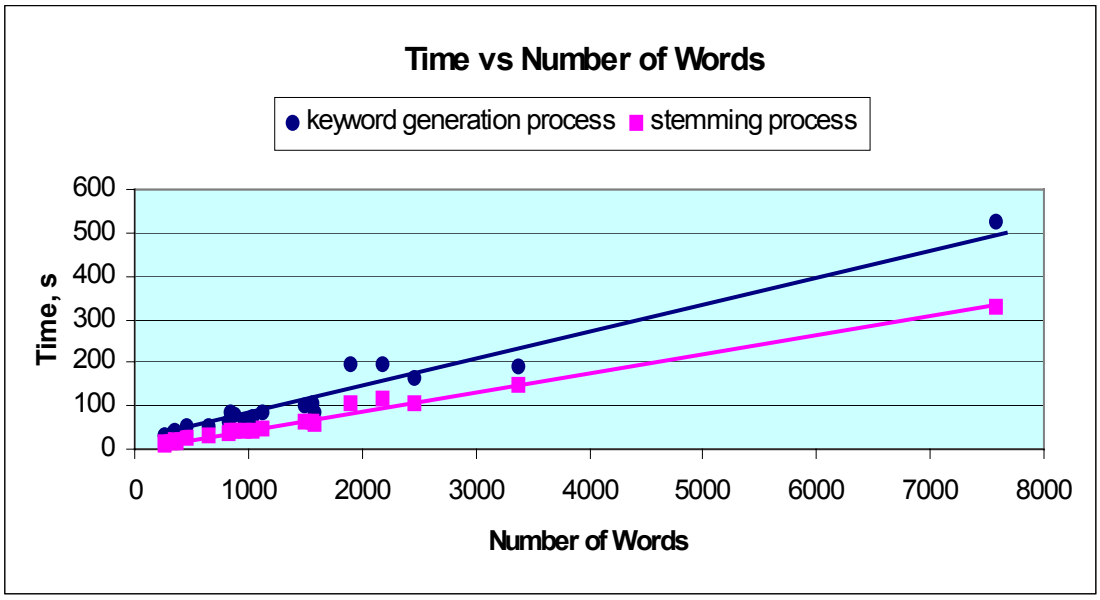

Figure 2. Relationship between processing time and number of words

be outputted; otherwise a default number of keywords will be provided.

3. Test Results

The proposed process was implemented and tested. Twenty HTML and text documents, ranging from several hundred words to several thousand words, were used in our tests. The test results show that computation time is linearly proportional to the number of words in the documents as denoted in Figure 2. Our tests also indicated that the computation time for stemming takes on average 56.7% of the total time.

To determine the accuracy of the proposed process, technical papers (taking from [10]) that contain keyword lists are used. Our test results show that on average the process can generate 52% of the keywords listed in the papers.

4. Conclusion and Future Research

A process for abstracting keywords from hypertext or text documents is described. The process takes into account the synonyms and the position of the words in the documents. We proposed to use keywords to summarize the results of Web pages returned from a search engine. The purpose is to allow users to quickly scan the search results to find the information they needed in shorter time.

For a system to process an increasing large number of Web pages returned from a search engine and to provide a reasonable response time, a relative simple process is chosen in this investigation. However, our test results show that even for such a simple process both the processing time and the accuracy require further improvements. Thus, we concluded that it is necessary

to perform pre-processing by abstracting and storing keywords in a database instead of pro-processing by generating keywords after a search is performed. This work is part of the increasing popular research in Web mining and much future research remains to be done.

Reference

[1] Mladenic, D., Grobelnik, M., "Assigning keywords to documents using machine learning," in *Proceedings of the 10th International Conference on Information and Intelligent Systems IIS-99*, Varazdin, Croatia, September 1999.

[2] Rowena Chau and Chung-Hsing Yeh, "Explorative multilingual text retrieval based on fuzzy multilingual keyword classification," in *Proceedings of the fifth international workshop on Information retrieval with Asian languages*, November 2000.

[3] Shaomin Zhang, Heather Powell, and Dominic Palmer-Brown, "Keyword Extraction using An Artificial Neural Network" http://www.uilots.let.uu.nl/~Paola.Monachesi/personal/abstr/zhang.html

[4] Yi-Ming Chung, William M. Pottenger, and Bruce R. Schatz, "Automatic subject indexing using an associative neural network," in *Proceedings of the third ACM Conference on Digital libraries*, May 1998.

[5] Azcarraga, A & Yap, Teddy Jr., "Comparing Keyword Extraction Techniques for WEBSOM Text Archives", in *The 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2001)*, Dallas, Texas, USA, November 7-9, 2001.

[6] Shyam Sheth and Debbie Yau, SmartScape: Intelligent Keyword Generation, http://sydewww.uwaterloo.ca/UnderGrad/workshop/1999-2000/smartscape.html

[7] Anette Hulth, Jussi Karlgren, Anna Jonsson, Henrik Boström, and Lars Asker, "Automatic Keyword Extraction Using Domain Knowledge," in *Proceedings of Second International Conference, CICLing 2001*, Mexico-City, Mexico, February 18-24, 2001.

[8] Charlotte Jenkins, Mike Jackson, Peter Burden, and Jon Wallis, "Automatic RDF Metadata Generation for Resource Discovery," in *The proceedings of the 8th International WWW Conference*, Toronto, Canada, May 1999.
(also in http://www.scit.wlv.ac.uk/~ex1253/rdf_paper/).

[9] Thesaurus.com http://www.thesaurus.com

[10] SC96 Technical Paper Abstracts, http://www.supercomp.org/sc96/proceedings/sc96proc/tabst.htm