Implementation of Web Page Summarization System

Hsu Yatanar, Daw Gilmour Hole
University of Computer Studies, Yangon
hsuyadanar@gmail.com

Abstract

World Wide Web is a growing sea of information accessible to different kind of user. One of the problems that caused by the rapid growth of the Web, which is called information overloading because WWW becomes to grow into a large million of web pages, users of WWW face with information overload. Users cannot read every web page posted in a web site because of the high volumes nature of a web site. Web Page Summarization is one of the effective ways to alleviate the information overload problem. Web page summary can help users get an idea of browsing time .This paper develops an automatic web page summarization system using text summarization theory found in information retrieval. Web page summarization has become an important application recently due to the increasing amount of information available on the Internet. However, to generate summaries as coherent as human authored summaries are a great challenge. Web document summarization techniques are derived from traditional text summarization techniques.

Keywords: Text Summarization, Stemming, Lexicon

1. Introduction

As the amount of on-line information increases, more and more effort is dedicated to creating Web Page Summarization systems. Web Page Summarization System is based on Text Summarization. Text Summarization is an active field of research in both the IR and NLP communities. Summarization is important for IR since it is a means to provide access to large repositories of data in an efficient way. High quality summarization requires sophisticated NLP techniques in addition, which are normally not studied in IR. In particular, for domains in which the aspects of interest can be pre-specified, summarization looks very much like Information extraction. Summarization is therefore a good challenge problem to bring together techniques from different areas.

Text summarization (TS) is the process of identifying the most salient information in a document and conveying it in less space (typically by a factor of five to ten) than the original text. In principle, TS is possible because of the naturally occurring redundancy in text and because important (salient) information is spread unevenly in textual documents. Identifying the redundancy is a challenge that hasn't been fully resolved yet.

Web page summarization system is developed from traditional text summarization. Unlike traditional documents with well-structured discourse, Web documents are often semi-structured, and have more diverse contents than narrative text, such as bullets, short sentences, emphasized text and anchor text associated with hyperlinks. Consequently, Web Page summarization is a non-trivial extension of the plain document summarization task due to the greater variety of possible feature sets. The identification of narrative text for summary generation is a key component of Web page Summarization system.

In this system, Section 1 represents introduction, section 2 describes related work, section 3 is theory background, and section 4 shows web page summarization system. Section 5 describes system implementation and its result and section 6 is conclusion.

2. Related Work

Many approaches to web summarization have been reported in the literature. Web Document Summarization using Hyperlinks presents web summarization method that uses the hyperlink structure of a web page, used a context algorithm and nature of incoming link to the web page [3]. Temporal Web Page Summarization describes web page summarization based on trend and variance analysis [1]. A Web Trained Extraction Summarization System uses extraction as a method for web page summarization, they trained web document using bigram and extract summary by using similarity based approach [5]. Document summarization using Wikipedia presents document summarization and use Wikipedia to map sentences of document to concept in Wikipedia, and select sentences for the summary based on the frequency of the mapped-to concept [4].A Text Summarizer for Swedish used statistical and linguistic methods as well as heuristic methods, and it contains a dictionary for scoring [2].

3. Theory Background

3.1 Text Summarization

Text Summarization is the technique where computer automatically creates an abstract, or summary, of one or more texts. The initial interest in automatic shortening of texts was spawned during the sixties in American research libraries. A large amount of scientific papers and books were to be digitally stored and made searchable. However, the storage capacity was very limited and full papers and books could not be fit into database those days. Therefore summaries were stored, indexed and made searchable. Thus, the technique has been developed for many years and in recent years, with the increased use of the internet.

Summary is a reductive transformation of source text to summary text through content reduction by selection and/or generalization on what is important on the source.

The process of text summarization can be broken down into following stages.

- Interpretation of source text content to arrive at a source text representation.
- Transformation of source text representation into summary representation.
- Generation of summary from summary representation.

During the first stage, the source text is interpreted, first 'locally' at the level of individual sentences before being interpreted 'globally' to give the source text representation. During the second stage of summarization process, the source text representation is transformed into the summary text representation. The final stage is generating text summary from the summary representation.

3.2 Types of Summarization

Text summaries have been classified into e***xtracts, abstracts, informative summaries, indicative summaries, generic summaries and query-based*** summaries [7].

- **Extract**: An ***extract*** consists of a set of sentences picked from the document to be summarized. These sentences are presented in the order in which they occur in the original document. To have a more meaningful summary with a proper flow, sentences may have to be re-ordered. Further, sentences selected for the extract do not undergo any semantic and syntactic changes.
- **Abstract**: An ***abstract*** of a document is a re-phrased shorter version of the original document conveying most of the information present in the original document in a concise way. Both extracts and abstracts can be classified as informative/indicative summaries.
- **Informative Summary**: An ***informative summary*** is one that serves as the surrogate to the original document. This means, if we read the informative summary we need not read the original document as all that we need to know is present in this type of summary.
- **Indicative Summary:** An ***indicative summary*** is one that gives us a general idea as to what the original document is all about.
- **Generic Summary:** Building a ***generic*** text summarization system which understands what is important to a user is a difficult task. Things can be simplified if the summarization system has a hint as to what the user of the system feels important.
- **Query-based Summary:** A summarization system that takes a query vector as an input and generates a summary based on this query is said to produce a ***query-based*** summary.

3.3 Method use for summarization

Most research on summary generation techniques still relies on extraction of important sentences from the original document to form a summary. There are several methods for measuring the importance of a sentence. Some algorithms calculate a weight for each sentence, taking into account the position of the sentence and word frequencies [6] while other algorithms use semantic information in order to find the hierarchy of concepts. There are also different methods for summary generation from a single document and from multiple documents.

This paper focuses on extraction methods from a single document and calculate the score of sentence by using TF*ISF (Term Frequency * Inverse Sentence Frequency) Scheme. TF*ISF weighted scheme is a basic equation to identify term frequency in document.

TF*ISF Weighted Scheme

IF*ISF = Term Frequency * Inverse Sentence Frequency

IF = Term Frequency

$$ISF = \log\left(\frac{\text{no of sentence in document}}{\text{no of sentence that contain term}}\right)$$

Also the format of texts is limited: Web Page summarization system considers that the input formal is formed as static page and News Web Page. We can produce HTML page of any domain, but it is better to summary News Web Page because we use Lexicon to identify Nouns and Proper noun for scoring system.

4. Web Page Summarization system

Web page summarization system implement for the following objectives:

- To study World Wide Web and their nature
- To study text summarization technique
- To study scoring method used in text summarization system
- To apply text summarization technique in web page summarization
- To explore methods for summarization web page
- To implement a web page summarization system

5. Proposed System

This system implement of Web Page Summarization System using Text Summarization technology found in Information Retrieval. It is window based system that help user to summary the input web page. Our system is an extraction based summarization system.

5.2 System Overview

The input of this system is html web page. Unlike the text document, web page contains many others JavaScript code and CSS (Cascading Style Sheet) for formatting; these elements are not concerned in summarization, so they are removed. The HTML page is preprocessed for summarization. Preprocessing includes removing html comment, style sheet, JavaScript code and other html tags that are not needed for summarization. We also need to identify words that contain in title sentence and header sentence because we need to calculate sentence score using these words. And then paragraph are indentified. Various statistics about the web page are collected at the same time such as bold word, italic word, and underline word. For each paragraph, sentences are indentified. Then the system tokenized each sentence, perform stopword removing, stemming. Each sentence is scored by the scoring criteria. Finally the sentences with highest score are selected for summary. The resulting summary is prepared and outputted with a HTML page. The resulting page is then presented to the user.

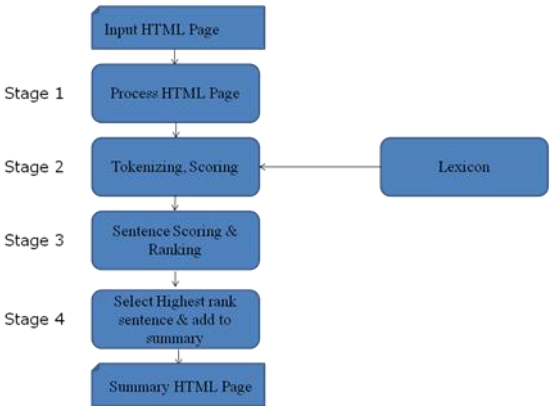

Figure 1: Overview Design of the System

5.2.1 Process HTML Page

Processing the HTML page is used for preprocessing the input HTML page. We implement processing the HTML page in Stage 1 of this system. Preprocessing includes removing html tag, code, identifying paragraphs using and tag, splitting sentence in each paragraph and construct Sentence, In HTML parser, regular expression is used for tokenize the sentence. HTML pages contain more than valid input text. They contain other formatting tags such as <b>, <u> tags.

HTML pages also include JavaScript code and CSS (Cascading Style Sheet) for formatting; these elements are not concerned in summarization, so they are removed. The purpose of preprocessing is to clean the cluttered html page suitable for summarization. Other important tag suitable for summarization are marked and collected such as title tag, head tag, bold and underline tags etc.

Region for paragraph are identified. For each identified paragraph each line is identified. The result from this stage is clean paragraphs consisting of sentences.

5.2.2 Stemming

The performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the

suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

The words that appear in web pages often have many morphological variants. Thus, pairs of terms such as "computing" and "computation" will not be recognized as equivalent without some form of natural language processing (NLP). In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. For this reason, a number of so-called *stemming Algorithms*, or *stemmers*, have been developed, which attempt to reduce a word to its *stem* or root form. This paper uses porter stemming algorithm to stem words.

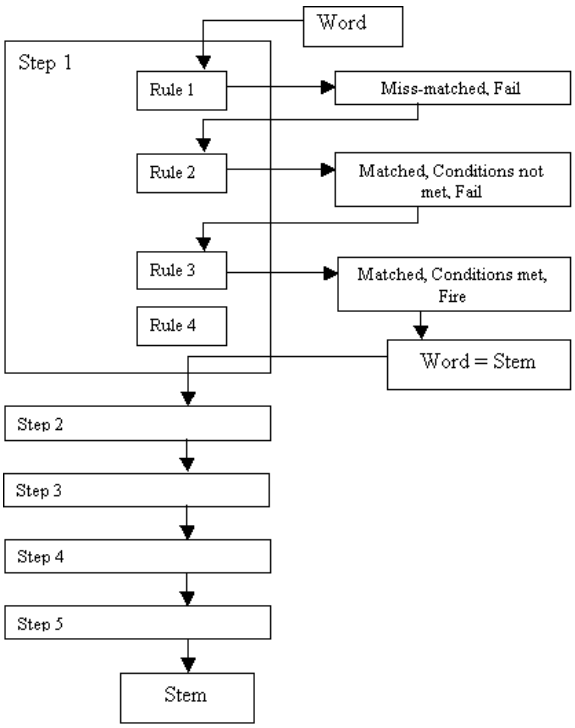

Figure 2: Porter Stemmer

The Porter Stemmer is a conflation Stemmer Developed by Martin Porter at the University of Cambridge in 1980. The Stemmer is based on the idea that the suffixes in the English language (approximately 1200) are mostly made up of a combination of smaller and simpler suffixes. This Stemmer is a linear step Stemmer. Specifically it has five steps applying rules within each step. Within each step, if a suffix rule matched to a word, then the conditions attached to that rule are tested on what would be the resulting stem, if that suffix was removed, in the way defined by the rule. For example such a condition may be, the number of vowel characters, which are followed be a consonant character in the stem (Measure), must be greater than one for the rule to be applied.

Once a Rule passes its conditions and is accepted the rule fires and the suffix is removed and control moves to the next step. If the rule is not accepted then the next rule in the step is tested, until either a rule from that step fires and control passes to the next step or there are no more rules in that step whence control moves to the next step. This process continues for all five steps, the resultant stem being returned by the Stemmer after control has been passed from step five. See figure 2. In this system, Stemming is used in Stage 2.

5.2.3 Lexicon

Lexicon is used for scoring words. Our Web Page Summarization is proposed to summarize News Web Page. So, we use a dictionary as a lexicon because we identify search word is noun or proper noun. If search word is not noun or proper noun, we can add more score. If search word is human name, it doesn't contain in lexicon. So, this sentence may contain more important facts. We use lexicon Stage 2 of this system.

We load lexicon from text file to array list when running our program. If we want to change our system domain, examples News Web Page to Technical Web Page, we change the lexicon from dictionary to other Technical domain keywords. So, we can score words that contain in Technical domain keywords.

5.2.4 Tokenization and Scoring

In tokenization stage, we need to do the following stages.

- Removes all new line characters "\n" in the original text.
- Finds the word boundaries by searching for the characters such as "**.**", "**,**", "**!**", "**?**", "<", ">", "**:**", " " ", " (" ,")", spaces, tabs and new lines.
- Removing stopwords from each sentence.

The out put from tokenization are sentence that does not contain stopword. We implement tokenization and scoring in Stage 2 of this system.

Scoring scheme for each sentence is following:

1. **Title**: The words included in the title along with the following sentences get a high score.
2. **Header**: The words included in the title along with the following sentences get a high score.
3. **Term Frequency**: Consider the frequency of each term include in each sentence (without stopword)
4. **Position score**: There is a theory that certain types of documents have their key meaning in certain parts of it. For example in the newspaper text, the first four paragraphs are the most important, while in technical papers the conclusion section is the most important part.
5. **Sentence Length:** The score given to a sentence reflects the number of words in a sentence, normalized by the length of the longest text in the passage.
6. **Proper Name:** Sentences which contain proper nouns get a higher scoring. Numerical Data: Whenever a number is identified in a text, the line that includes it gets one additional point. To identify proper noun, we use lexicon that contain dictionary.
7. **First Sentence**: It should always be included in the summary. This is done by assigning it a very high score.
8. **Special HTML tags such as <i>, <b>, <u>:** The important word are expressed as bold, italic and underline format.

5.2.5 Sentence Scoring and Ranking

Score for each sentence is calculated by using the score for each token in the sentences. Each sentence score is calculated by using the following equation:

WordScore = (word frequency) * (additional score)

Sentence Score = ∑ WordScore

Word frequency is calculates by **TF*ISF** weighted scheme.

There is a risk that long sentence will be ranked higher, in order to avoid such phenomenon the sentence score is multiplied by the Average Sentence Length and later divided by number of word in the sentence for normalization.

$$\text{Average Sentence Length} = \frac{\text{Word count}}{\text{Line coint}}$$

$$\text{Final Sentence Score} = \frac{(\text{ASL} * \text{Sentence Score})}{\text{number of words in sentence}}$$

The following scoring criteria are assumed to calculate word score for this web page summarization system.

Bold Word = word score*0.5

Italic Word= word score*0.5

Underlined Word= word score*0.5

Proper Noun Word= word score*0.5

Header One= word score*0.5

Header Two= word score*0.4

Header Three= word score*0.3

Header Four= word score*0.2

Header Five= word score*0.1

Header Six= word score*0.05

Title= word score*0.5

Position Score = (1/ line no) * 10% of Final Score

We use this scoring and ranking in Stage 3 of this system.

5.2.6 Sentence Extraction

The highest score sentences from web page are extracted and showed ordered by sentence position. And then showed as html page and we can stored as html page. We extract highest score sentence in Stage 4 of this system. The extracts generated are in size shorter than the original texts. However, the number of sentences that are summarized by Web Page Summarization System can be limited. User of Web Page Summarization System can choose the percentage of summary based on the number of sentence of original web page. We can summarize 20%, 30%, 40% and 50% of original web page. But, the summary between 30% and 50% may be the best summary of original web page.

5.2.7 System Implementation

This system is implemented as window based system using Microsoft Visual Studio 2008, Framework 3.5. This system is extraction based summarization and uses TF*ISF weighting scheme to identify frequency of words. The input of this system must be standard HTML page and must be static page. Many News Home page always contain link to other detail page. If we summarize this page, we can't get effective summary. First, the input web page is preprocessed in Stage 1 to

identify paragraphs, title, heading and other words format such as bold, italic and underline words. And then, each sentence from paragraph is tokenized, stemmed in Stage 2. In this Stage, lexicon and other word score criteria to score words. In Stage 3, we calculate sentences score by using the result of word score from Stage2 and other sentence scoring criteria. In Stage 4, we extract highest score sentence according to percentage of summary option and produce summary. Following figure shows how user input web page.

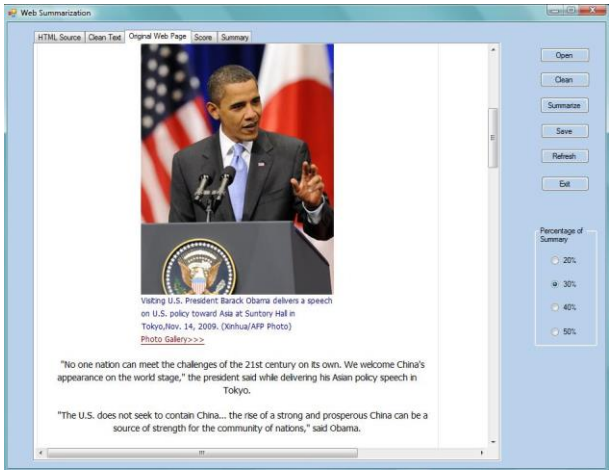

Figure 3: Implementation of user input form

Following figure shows 30% summary output of above web page from our Web Page Summarization System.

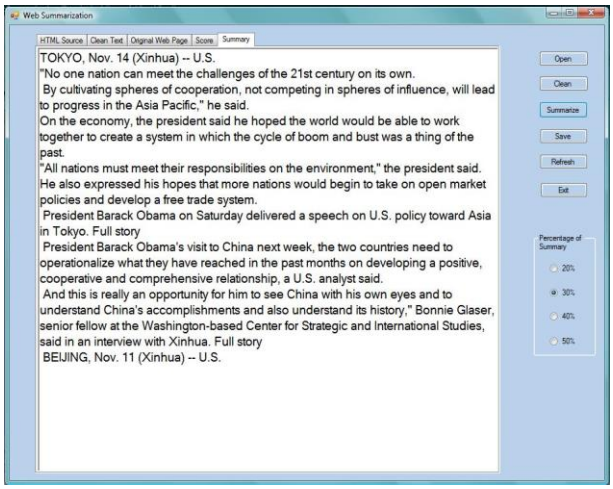

Figure 4: Out Summary of input Web Page

6. Conclusion

This paper implements a web page summarization system that can automatically summarize web page. The user gives a web page to summarize into the system, and the system can give a summarization page to the user. The summarization uses TF*ISF weighting schema and scoring method to summarize the web page**.** Our system is an extraction based summarization system. So, the result summary from this system is not meaningful as abstraction based summarization system.

An Extraction based summarization system is advantageous because very often statistical techniques such as frequency and correlation correspond to the relevance or importance. Here we don't need any background knowledge of NLP, and we could automate the process easily and efficiently. On the other hand, the results can be lack of coherence because of dangling anaphor such as pronoun and conjunction. According to our testing experiences, our system can produce important sentences from web page.

7. References

1. A. Jatowt, and Mitsuru Ishizuka, Temporal Web Page Summarization, Techincal Report, University of Tokyo ,Japan.

2. A. Jatowt, and Mitsuru Ishizuka, Temporal Web Page Summarization, Techincal Report, University of Tokyo ,Japan.

2. H. Dalianis, Swesum- A Text Summarizer for Swedish Master Thesis.

3. J-Y Delort , B.Bouchon-Meunier, M.Rifqi, Enhanced, Web Document Summarization using Hyperlinks, ACM Conference on Hypertext and Hypermedia , 20035. Hercules Dalianis, Swesum- A Text Summarizer for Swedish ,Master Thesis.

4. Krishnan Ramanathan, Yogesh Sankarasubramaniam, Nidi Mathur, Ajay Gupta, Document summarization using Wikipedia, HP Laboratories, 2009.

5. Liang Zhou and Eduard Hovy, A Web Trained Extraction Summarization System, Proceeding of the HLT-NACCL ,2003.

6.Dalianis, H., M. Hassel, J. Wedekind, D. Haltrup, K. de Smedt and T.C. Lech. 2003.
Automatic text summarization for the Scandinavian languages.

7. G.Ravindra, Information Theoretic Approach to Extractive Text Summarization, PhD Thesis2006.