Automatic Keywords Extraction

Section 1

Summary of Assignments:

Yuanpei Zhang – Algorithm and its implementation in C Language; Connecting C code with JavaScript

Xin Li – Google Chrome® extension development with JavaScript

<table>
<thead>
<tr>
<th></th>
<th>Member 1 name</th>
<th>Member 2 name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of effort towards this assignment</td>
<td>50%</td>
<td>50%</td>
</tr>
</tbody>
</table>

Section 2

Our project is a Google Chrome extension application which can automatically extract the keywords of selected text segments on websites for its users. This project consists of two major parts: the algorithm and the development of Google Chrome® extension. However, since the core algorithm is intended to be programmed with C Language, and the Chrome® extension development tool is JavaScript, we need to consider the problem of connecting the fundamental functions programmed by C to the user interface programmed by JavaScript.

- Keywords extraction algorithm – TF-IDF

TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the
word in the corpus, which helps to control for the fact that some words are generally more common than others.

Our project utilizes the algorithm of TF-IDF. By calculating the offset of each word in the selected text segment from the corpus, the program can determine the most important words that feature the major content of the text.

The program implements TF-IDF as described above by C language. The free c code library Bag-Of-Words contains the necessary sources to implements the TF-IDF. Specifically, the library contains the source files providing the basic implementation of weight-setting and scoring, which are essential techniques in TF-IDF.

- Google Chrome® extension development

For our project, we will be utilizing the Google Chrome® as our operating platform. Chrome® extensions allow users to add functionality to Chrome® without diving deeply into native code. The extension can be developed with the common web development: HTML, CSS, and JavaScript.

Chrome® extensions are zipped bundle of files – HTML, CSS, JavaScript, images, etc. – that adds functionality to the Google Chrome® browser. Extensions are essentially web pages, and they can use all the APIs that the browser provides to web pages, from XMLHttpRequest to JSON to HTML5.

- Connecting C to JavaScript

The project we intend to do is an extension program that is able capture selected text on any website and automatically find out the keywords featuring the major content of the text. Therefore, one essential technique is to capture the selected text and send it to the core function programmed by C that extracts the keywords from the texts.

Since JavaScript runs at a completely different environment which is separated from the webserver by a HTTP connection, the best way to run the core function programmed with C Language in JavaScript is to fire a HTTP request to the server side on a specific URL which has the C code attached.
Section 3

1. Realistic constraints
   a) Economic
      The team managed to minimize to cost of the product. Basically the most
      expensive part is the software needed to program the application, which may
      cost hundreds of dollars depending on the type. However, once the
      programed is developed, this cost will not be a problem since it will be
      averaged out by the selling price.
   b) Environmental
      The product is environmentally friendly since the end product is a software
      application which does not contain any parts contaminating the environment
      or jeopardize human health.
   c) Health and safety
      As stated above, the product is unlikely to threaten human health.
   d) Manufacturability
      Since the end product is a downloadable supplication, it faces no problem of
      manufacturability. It may take the team some efforts to develop a Chrome®
      extension and to implement the automatic keywords extraction algorithm.
      However, when this product enters into business, the only problem is to
      sustain and to keep updating.
   e) Sustainability
      Since the end product is a software program which is non-destructible, we
      assume that the product can sustain a long life. However, the product need
      to be frequently updated as most software programs do to fix potential bugs
      and to improve its performance.

Section 4

- To create a Google Chrome® extension that is convenient to use for extracting
  keywords from selected text segments
- A functional and stable program for internet users to save time in browsing
  websites
- Being accurate and efficient
- Being easy to operate for users
- To minimize the negative effects on regular operation of other programs
- To bring innovation to a field in the midst of a technological standstill
Objective Tree

Program Performance

Automatic Keywords Extraction

Promote Sale

Functional and Stable

No Crash

The Extracted Words Can Correctly Feature the Major Contents of Texts

For General Articles: News, Novels, etc.

For Technical and Professional Articles

Having No Negative Effects

No Effects on Normal Operation of Other Programs

To Minimize the Created Junk Files and Data

Easy to Be Enabled/Disabled by Users

Low Cost

Easy to Download and Install

Reference


http://developer.chrome.com/extensions/index

http://stackoverflow.com/questions/2626859/chrome-extension-how-to-capture-selected-text-and-send-to-a-web-service

http://stackoverflow.com/questions/2796472/connecting-c-backend-to-javascript