A Multidimensional Relevance Matrix Method to Refine Text Web Content

N.P.V. Kumara, N.U. Jayasinghe, M.F.M. Aflal, A.M.A. Ali, A.S.M. Naufal
(vishva8kumara@gmail.com , nirmaljayasinghe@yahoo.com , aflal_777@yahoo.com
anverali@gmail.com , naufal_salaam@yahoo.co.uk)

Supervisor : Ms. Nipunika Vithana (nipunika.v@sliit.lk)

Sri Lanka Institute of Information Technology
Metropolitan Campus, BoC Merchant Towers, Colombo 03 (www.sliit.lk)

Abstract : Modern search engines lack the ability to return results grouped into a predefined subject field. Moreover, they return a number of *links* to information, not directly *the* information. This paper presents a new method of refining text information, by directly looking at the information and deciding how relevant it is to a certain subject context. The aim is to provide a quick and efficient method for Internet users to pinpoint and access the information they are searching for within a predefined subject field, avoiding the requirement to select pages from a lengthy search list and visiting a number of Web pages. The information would be retrieved from the search results links directly and compiled into a document, which will give the user an idea about the most relevant information available on the Internet on their search criteria.

Keywords: Multidimensional Relevance Matrix (MRM), subject relevance

1 Introduction

As the size of modern multimedia databases such as the Internet grow at a titanic rate [1], methods to efficiently and effectively retrieve required information out of them has come to be of extreme importance. These retrieval methods greatly increase the value and usability of such data collections, whose value depend on the factor of how easy it is to pinpoint a piece of required information from them. The Internet, for example, contains a vast amount of information. As the amount grows, search methods to locate required data have become a hotbed of research. The basic problem is to pick out a required piece of information out of a large collection with the least possible effort and time.

Modern search engines lack a major detail in this process: to determine how relevant a piece of information is (say, a text paragraph) to one subject field. To accomplish this, the search tool literally has to 'read' the data. But computers, unlike humans, cannot perform such a task just being supplied with some piece of information. Instead, existing search methods tend to look at a document as a whole, and try to give an overall rating to the information they contain [2], and thereby to decide how relevant a document would be in comparison with a search term.

In a nutshell, modern search methods are not exactly efficient at deciding how relevant a piece of information is in a certain subject area. Nor there is provision to opt in which broad subject field the user would be expecting results when a search is conducted. For example, let's take an Internet search on the term 'car'. The results would include pages about cars with emphasis on a number of fields (i.e. engineering, scientific, commercial etc.). If the user only needs results relevant in engineering aspect of cars, there should be a method to separate the results relevant to engineering from the main results set for 'car' and present them to the user.

Another problem faced by search techniques available today is the large amount of information found on the Internet. Search engines can reliably index only a fraction of this and still even a simple Internet search would return a results list with a large number of results links. Users have to visit a number of links to get to the information they need, in the process encountering spurious results pages, irrelevant results, redundant information, advertisements and other distracting and potentially misleading information.

In this research, we present a solution to pinpoint field-specific information from a traditional Internet search. The solution software will act as an interface between a search engine and a user and have provision for a user to specify a search term as well as to specify in which subject field they are looking for results. The software would link up with a traditional search engine to get a results listing, visit a number of links and retrieve information relevant to the subject field specified by the user, compile a convenient document out of them and present it to the user. In this way, the users can get information relevant to a subject field they prefer, and would be spared the trouble of visiting a number of Internet links in order to get to the information.

2 Methodology

The technique we propose basically uses emulation of the human ability to look at a piece of text information and decide how relevant it is to a certain subject category. When initiating a search, the user is given the option to select in which predefined subject category they would like the results filtered in. The software uses an inbuilt database which maps basic language terms to the predefined set of subject fields, allowing it to decide how relevant a piece of text information is to a certain subject field by looking at the words it is composed of.

After the user specifies a subject category and search keywords, the software would get a results listing for those keywords from an existing search engine. Then it would visit the first ten results links and retrieve the text from them in an automated process, strip the text content down into words and rank the sentences according to the relevance of the component words, and reorganize the sentences to present the user with a concise and structured document which will give them a snapshot view of the most relevant information available online for their particular search term/subject category combination.

2.1 Technology

The major technical challenge that had to be tackled in this project was to emulate the human ability to filter out information relevant to a certain subject field from among a large amount of irrelevant information. To handle this, the best method was to drop down to one of the smallest meaningful units of language, the *word.* At a very basic level, the word-relevance mapping multi-dimensional relevance database determines how much a certain term is relevant in a certain subject field. This, along with the other techniques and technologies applied, will be described in detail in the following subsections.

2.1.1 The Multidimensional Relevance Matrix

The Multidimensional Relevance Matrix, or the database which determines how much a certain term is relevant to a given subject field, is the main component of this application. It enables the software to determine, in the highest possible *natural* manner, whether a certain piece of text information is relevant or not in a given subject category. The term *Multidimensional Relevance Matrix* stems from the idea that subjects can be treated as axes (or dimensions) in a three-dimensional visualization of the database structure[1], and terms appearing along various points along each axis, according to the terms' relevance in each subject. A simplified version of this view is shown in Figure 1.

[1] It should be noted although this simplified view may suggest it, it is not the case that a single term is represented by a single point in the multi-dimensional space of subject fields. The relationship between relevancies in different subjects for a single term can be fuzzy.

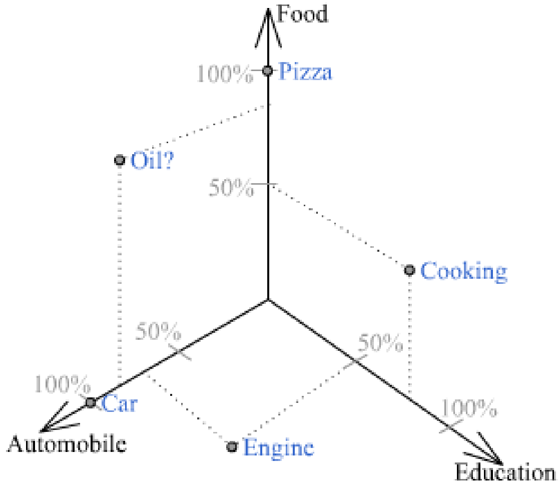

Figure 1: Simplified 3-D graphic visualization of the Multidimensional Relevance Matrix

In the simplest possible view, this database maps terms and relevancies. A term's relevance in a certain subject category is given as a percentage value, between 0 and 100, usually in a multiple of 10. Attempting to achieve a bigger resolution would have been inappropriate and probably impossible at an early stage of development. The database is preprogrammed with a number of such relevance values. As the database grows bigger, it would be possible to go into more intimate levels of relevance value.

2.1.2 Fetching and Filtering the Information

The software's user interface provides a textbox input for search term entrance and a checkbox selection of the subject field from a list. After the user enters the search term/subject field combination and presses the 'search' button, the software couples up with a normal Web search engine to get a traditional results listing for the term the user has entered. From the returned page, the software filters out the first ten main search result links. The hypertext markup (HTML) structure of this search page varies from one search engine to the other [3], and separate techniques would be required to extract the links. In this instance, we have only included provision for link extraction from Google's search result pages. It would easily be possible to add extensions so other search engines may be used as well.

After filtering out the main ten links of the results page, the software would sent ten different, simultaneous HyperText Transfer Protocol (HTTP) requests to the ten link locations. Sending a raw HTTP request rather than using .NET's[2] inbuilt Internet Explorer component has many advantages, including speed of request and blocking of distracting pop-ups and advertisements, thereby achieving a major objective of building this software.

While the ten pages load in parallel, the word-relevance database or the Multidimensional Relevance Matrix would be loaded out of hard disk to the computer's main memory. This step invariably results in faster queries and results, and helps to build the final document within an acceptable timeframe. After all ten links are loaded, they are place in a queue and the human-readable text information is extracted and separated into sentences. There is a set timeout period to go into this stage from page loading stage; without this a slow-loading page or a broken search link can slow down or stall the software's performance altogether.

In parallel to this process, all the words in the sentences are listed with their occurring frequency, in the descending order of the frequency. Words that are too trivial to be considered as significant words are ignored. The sentences are again analyzed to separate the words. If those words have postfixes to make it past tense, plural form, etc., (i.e. -s, -ing, -ed) those are removed and checked in the word-relevance database's set of words if they exist. If the words are found in the database, the rel-

[2] Microsoft Visual Basic 2005 technology was used to develop the software.

evance value is read and the sentence is ranked accordingly.

The most relevant 50% of sentences (or optionally, a given number of sentences) are retained, and the rest is discarded. The 15 words with highest occurring frequencies are selected as the significant words[3]. Later the user can change this selection. The selected (retained) set of top relevant sentences is again analyzed to check if they consist a significant word. If so, the sentence is tagged with the significant word.

Finally the sentences are grouped in to paragraphs according to their significant words. A picture is searched through Google's Image Search and the first image found is included on the top of the document[4]. A menu of paragraphs is created and placed to the left of the image, for the sake of easier navigation and comprehension. If the search is initiated from Quick Access Menu[5], a new instance of the system default Internet browser is opened to view the page. The document is archived (by default setting) in the current user's *My Documents* folder with the option of the archiving location being selected at the user's discretion.

3 Results and Future Work

This method was implemented with the major purpose of helping an Internet user to access the information they are searching for without visiting a number of links in a lengthy results page. During the tests conducted, it was proven that the system achieved this purpose successfully. Through its innovative information access and filtering process, the system allows a user to get a snapshot view of the latest and most relevant information available in the World Wide Web at the moment, for their particular search term/subject field combination.

A number of development options were recognized. This includes adding a complex self-learning process to the word-relevance database so it would be able to learn new terms on itself once in operation. A technology demonstrator for this technology is already included in the system. Further, this learning process can be decentralized, using an XML-based central server to which all newly learned terms from the clients would accumulate. Yet another requirement is to standardize the relevance ranking process for terms, which is currently done in an ad-hoc manner. This would require work on a much broader perspective than this project was conducted.

4 Conclusion

The Internet has grown in great leaps and bounds since its humble beginnings, and has become a vast repository of generations of human knowledge. As it continues to grow in size and variety, better methods to retrieve information efficiently from large linked databases become of increasing importance. Tomorrow's search methods will have to be lighter and leaner, and most importantly, *smarter*, than today's.

This research was primarily driven by that need. Today's available search methods are too mechanical for the emerging needs, where we need intelligent search and filtering methods which can virtually *read and understand* the information and classify it, without relying only on metadata. The relevance-based filtering method presented in this paper has shown promise in attaining these objectives.

References

[1] K. G. Coffman and A. M. Odlyzko. *The Size and Growth Rate of The Internet.* AT&T Labs-Research, Revised version, October 2, 1998.

[2] L. Page. *U.S. Patent No. 6285999: Method for Node Ranking in a Linked Database.* U.S. Patent Office, New York, 1998.

[3] T. Powell. *Web Design: Complete Reference.* McGraw-Hill, New York, 2002.

[3] *SigWords*

[4] Image is not analyzed for relevance

[5] The application can be minimized to a background process, where its functionality is controlled by a Quick Access Menu obtained via a system tray icon.