2008

Personalized Web Content with Fuzzy System

Bassam Almogahed

San Jose State University

Recommended Citation

Almogahed, Bassam, "Personalized Web Content with Fuzzy System" (2008). *Master's Projects*. 77.
DOI: https://doi.org/10.31979/etd.2rye-9fya
https://scholarworks.sjsu.edu/etd_projects/77

Personalized Web Content with Fuzzy System

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Bassam Almogahed
bassamdo@gmail.com

May 08

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Chris Tseng

Professor of Computer Science, San José State University

Dr. Robert Chun

Professor of Computer Science, San José State University

Dr. Michael Beeson

Professor of Computer Science, San José State University

APPROVED FOR THE UNIVERSITY

List of Tables and Figures

Figures

Figure 3.1: System Overview 16
Figure 3.2: Pre-Fuzzy Stage Design Flow 19
Figure 3.3: Membership functions (a) trapezoidal (b) triangular, and (c) Gaussian 21
Figure 3.4: System Membership functions 23
Figure 4.2: System Web Interface 28
Figure 4.2: Membership Function of the frequency Factor 33
Figure 4.3: Rules Activation 34
Figure 4.4: Values Substitution 34
Figure 4.5: Perform Logic Operation 35
Figure 4.6: Final Score Calculation 35
Figure 4.7: Initial Result Page 37
Figure 4.8: Alternative Result Page 38
Figure 4.9: Sample Image Output 39
Figure 4.10: Another Sample Image Output 39
Figure 4.31: “Blue Prints” Online System Architecture 40

Tables

Table 1. Fuzzy Rules Table 25
Table 2. Score Table 25
Table 3. Candidates Terms 31
Table 4. Calculated Terms’ Scores for each Factor 32
Table 5. Final Scores for all Candidates 36
Table 6. Results Analysis using Several Categories 42
Table 7. Results Analysis using News from Several online Leading sites 43
Table 8. Results Analysis using Data from eopinion.com 43
Table 9. Results Analysis using Data from NASA/FIPS 44
Table 10. Overall Performance of YTE, Google AdWords, and Extractor 44
Table 11. Consistency and Efficiency of YTE, Google AdWords, and Extractor 44

Table of Contents

1. **Abstract**………………………………………………………………………………………………….. 6

2. **Project Overview**

2.1 Introduction... 7

2.2 Current State of the Art in keywords Extraction Based on Text... 10

2.3 Goals and Description.. 11

. 2.4 Technical Background... 12

3. **Design**

3.1 Design Overview... 15

3.2 Pre-Fuzzy Stage.. 17

3.3 Fuzzy System Stage... 20

3.4 Post Fuzzy Stage.. 28

4. **Implementation**.. 28

5. **Results** ……………………………………………………………………………………………... 40

5.1 Input material.. 41

5.2 Extractor Results... 41

6. **Comparison and Discussion**……………………………………………………………………………. 42

6.1 Comparators.. 42

6.2 Comparison... 43

6.3 Discussion... 44

7. **Conclusion and Future Enhancements**……………………………………………………………….. 45

7.1 Conclusion.. 45

7.2 Future Enhancement... 46

8. **References** ………………………………………………………………………………………………… 47

1. Abstract

In this paper, we will develop a new computational intelligence methodology to automatically analyze and summarize web content during a user surfing sessions. The output of this process is meaningful keywords or phrases which will be used to bring the user other contents such as images that closely relate to the web pages that he or she is currently surfing.

ACKNOWLEGEMENT

I would like to give a special thank to Dr. Tseng for his suggestions and guidance thought out the whole writing project phases. I would also like to thank Dr. Chun and Dr. Beeson for being my committee members and help me through this process.

2. Project Overview

2.1 Introduction

The importance of the Internet has grown to the point that it has become a basic necessity for many people. For some, the ability to access the Internet at all times is so essential that we find ourselves signing on not only from home and work, but also in our leisure time. People have begun signing on at cafés and hotels, and even at the bus stop. Research companies have been quick to realize the potential benefits springing from this overwhelming consumer need, and have produced advertisements accordingly. By fostering user interest and delivering demographic- appropriate advertisements, organizations such as Google and Yahoo have gained immense popularity not only for the obvious commercial benefits of their products, but for their underlying, strategic marketing techniques as well. The purpose of this project is to enhance the current web experience and generate more personalized web content that not only interests the user, but spans multiple forms.

There are a variety of materials that could be presented to the user, such as text, images, audio files, video files, and web content. In general, this includes any information that could be managed in an electronic format.

The goal of this project is to develop a system to suggest such content which correspond with users' interest during a surfing session. There are two stages to achieve the aforementioned goal. The first and foremost focus of this project is to analyze and summarize the user activity automatically while browsing a certain web page. The second stage is to use the information from the first stage to find and present matching content, mainly images.

2.1.1 Project Motivation:

Most successful attempts to bring users the images and videos that relate to their search queries or to the content of the web pages that they visit have been solely text-based systems. These systems rely heavily upon the text that is associated with the content, such as file name, html title, hyperlinks, tags, etc. However, many believe that content-based analysis of images or video files will be the next groundbreaking revolution in this field. For example, if an image of a basketball is stored somewhere in a computer and the user has forgotten where it was stored, he will either try to open all possible folders that it could be saved in, or will try to search for it using keywords. If the image was labeled with a meaningful keyword that is representative of its content, then a text-based search will find it. In many cases, though, the situation is more complicated. This is where the content-based analysis system comes in.

On the other hand, even if we were able to successfully achieve such a working system, it would not mean the elimination of today's text-based technology. They will, however, be joined together in order to achieve better and more accurate results. Therefore, working on improving text-based techniques has significant potential; especially considering that the other approach has is far from completion.

2.1.2 Statement of Purpose

Automotive term extraction or the selection of important, meaningful phrases in the body of a document is the key success to this project. The phrase can be a word or few words which should capture the main topic that is discussed in a given document. Since the number of text document online still growing everyday, our main focus of this project is to automatically extract keyphrases from a text document and use these keyphrases to bring other forms of related content to the user.

This project attempts to get closer to the users by analyzing the content of a web page they are visiting during surfing sessions and using some existing technology such as fuzzy logic, XML, PHP and JavaScript. The system is able to compile all information gathered, perform analysis upon it and then generate content that relates to users interests automatically without disturbing the users; they don't have to search for an image or a video, they don't have to type, it will be generated automatically to them in form of a suggestion tool and they have the choice to view the suggested content or continue in their browsing activity. This, in turn, will increase user productivity and efficiency during internet surfing sessions.

2.2 Current State of the Art in keywords Extraction Based on Text

With the rapid growth of the internet today and the extremely large number of online document that is in there, it became very difficult to retrieve the right information even when the user is providing the search keywords. Many search engines such as Google and Yahoo still perform their search process based on a text matching process which leads in many cases to irrelevant information in the returned results. What is more difficult is automatically extracting these keywords from a page that a user browsing and then using these keywords to retrieve additional information that might be of interest to the user.

In the past 2 decades there were many papers and research done that discussed topics about data mining [1] and information retrieval. Since we are focusing specifically in terms and keyphrases extraction, we will highlight some early interesting techniques that covered closely this topic.

Krulwich and Burkey (1996) used a heuristic approach based on syntactic clues, such as the existence in the title, headers, bolded and italics words, use of acronyms, etc. this technique resulted in a low precision large number of keyphrases [2]. Another approach in the same year by Muñoz used Neural Network technology where he uses unsupervised learning algorithm to produce two-word keyphrases. In additional to low precision of the results set, it wasn't applicable to keyphrases with different length but two-words [3].

Another approach which was based the statistics between a compound noun and its component where each candidate gets several scores based on statistics methods such as tf·idf. This approach gives a promising results however it is domain based therefore, all the analysis and the process of filtering out "useless phrases" is heavily relay on the domain [4]. The statistics approach on automatic term recognition proved its effectiveness in this area and in some degree or another in played a part of the extracting process [5][6][7]. However, it was obvious that there is still a room of improvement that can be made and thus many researches used other techniques such as genetic algorithms, technical dictionary hierocracy, Neural Network approaches to further enhance the statistical results [8][9]. Similarly, we used the statistical approach further enhance the results, however, in our approach we incorporated new factors and we design a fuzzy logic to achieve our desired goals.

2.3 Goals and Description

Generally, the extraction of terms and phrases from a document can be used for several reasons such as page indexing, tags for a journal/ articles, or simply as a form of summarization. Search engines such as Google and Yahoo uses some sort of term extraction techniques in the back end which they use for bringing advertisement to the users as they visit a related page.

In this project, we aim to obtain more meaningful phrases and keyword which represent closely and accurately the main focus of the page and use them to bring to the user other form of online content which they might be interested on without having to leave their browser. We choose to return images to the user; however, this can used to return other web format such as videos and text.

Our solution is to achieve the previous goals in real time, therefore, we needed to find a simple, more efficient way, yet, getting results than truly represent the web page that is been tested. The system first grabs the content of the URL visited, obtains some phrases and keywords, and goes through a simple elimination and validation process to choose the possible candidates that will go through testing and further calculation. These candidates go through several tests and calculation to determine the ranking of each keyword/phrase reflecting its representation of the content that was extracted from the tested web page.

2.4 Technical Background

There are several techniques, new technologies, and open source software that are used on the process of developing this software. A general description of each is presented along with some of their drawbacks and advantages.

A. Yahoo Term Extraction API (Yahoo API):

Description:

This is a content analysis web service (version 1) that yahoo network developers created, and it returns a list of significant keywords and phrases which are extracted from a larger content. This process works by submitting web content to yahoo API with an option of submitting a query that should help with the extraction process. The returned output comes in XML format and it claims that the terms are returned in order of importance [10].

Advantages:

Using this API, on the other hand, is interesting in the fact that while we don't really know how this API works, the fact of the matter is that many of the terms returning this API are very useful. Regardless of the length of the documents involved, by merely reading these terms, we will, in almost all cases, be able to guess what the page content is about without reading or even looking at the web page itself.

Drawbacks:

Although yahoo claims that these terms are returned in the order of their importance in the content of the page, the fact is they are returned in a completely random order. However, the major drawback of this API is the existence of terms which are not related to the content such as some html tags or non suggestive words that exist as part of the output. Looking at these terms, we can easily determine those that are meaningful. The difficult part, however, is to achieve this automatically.

B. Google Images Search Engine:

Description:

It was intended to build an image database and compare the results to leading search engine results. However, due to the unfair comparison where we will populate the database ourselves whereas search engine will give their results according to the portion of the internet they are covering. Therefore, our system will use Google image search engine to return the images that correspond the winner terms from the extraction stage [11].

Advantages:

Using Google images API for instance will return to us the same top results as if we went to Google images page and typed a specific keyword or a phrase so in that sense we saving the user the trouble of opening a new page and typing some text.

Drawbacks:

The final output of the system comes from another source, Goggle images API. Therefore, the accuracy of our results is depending in the accuracy of Google images which we don't have control over it.

C. Classifier4J – ISummariser

Description:

Classifier4J is an open source Java library designed for text classification [12] .It has an implementation of a Bayesianclassifier which is based on Bayes' theorem [13], and Vectorclassifier which uses the vector space search algorithm [14]. This Java library has some other features such as, ISummariser, which works with text where the user enters text content and number of desired sentences, and as an out, it gets a summary of this content.

Advantages:

A summary by definition is a shortened version of the original. The main purpose of such a simplification is to highlight the major points from the genuine (much longer) subject. If a phrase or a keyword reflects the content of the original text, we expect it to be part of the summary and thus will have a higher weight during the calculation stage later in the overall process.

Drawbacks:

Given that the aim is use this software in any website during the user surfing session, some web pages will not have enough text or a formal article structure format and thus for these pages, the summarizer will not return as good of results comparing to online articles and news pages which end to be more organized and follows the standard writing format in the English language.

3. Design

3.1 Design Overview

Extractor should be able to use the web page *url* to get the content of the page, prepares all the data that will be used in the fuzzy system, and finally returns the keywords and images that are expected as an output. Therefore, and for simplicity we can theoretically divide the system into three stages: Pre-Fuzzy Stage, Fuzzy Stage, and Post-Fuzzy. Figure 4.1 shows the flow of the overall process.

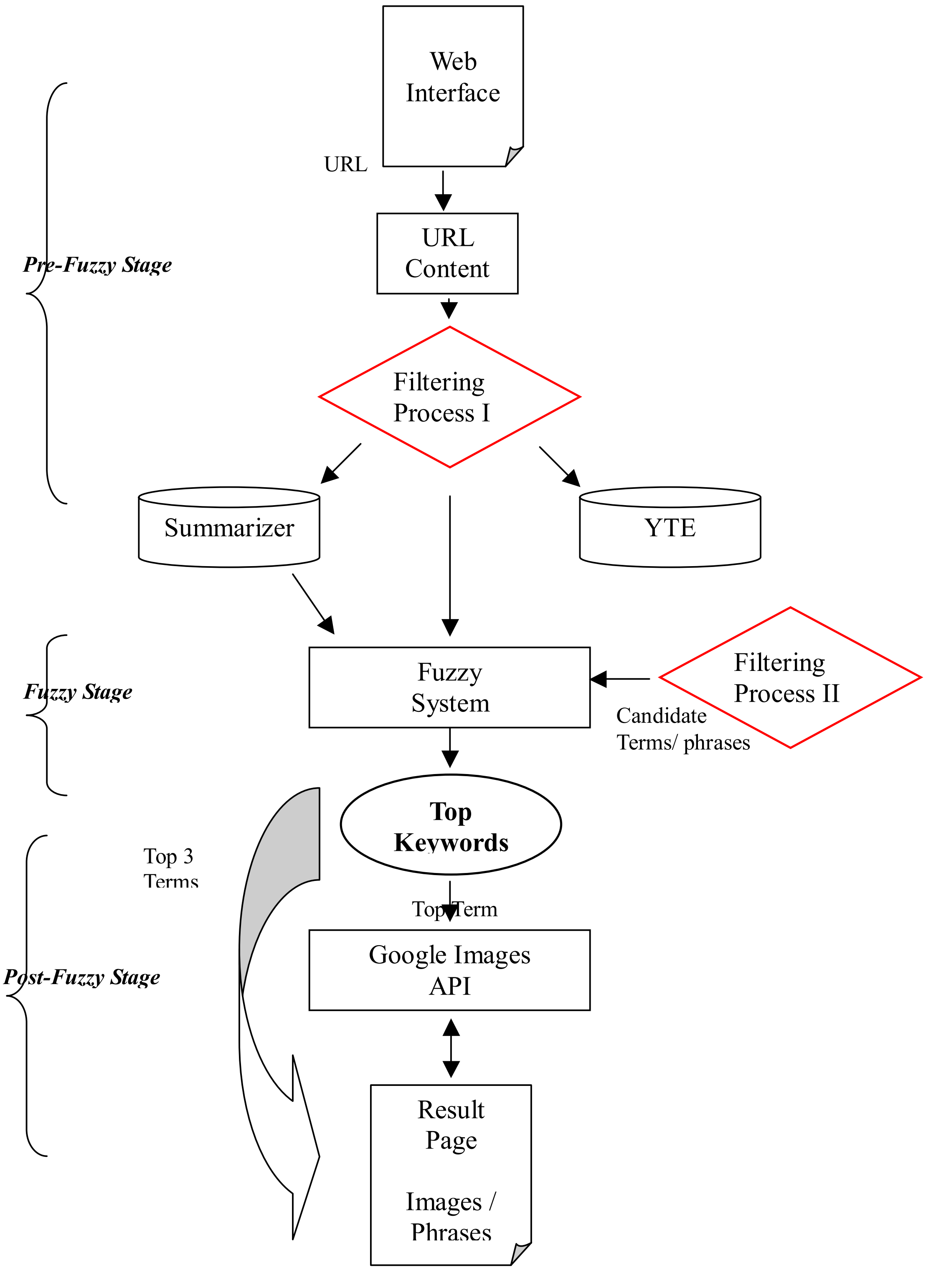


Figure 3.1 Overall Overview

3.2 Pre-Fuzzy Stage

The first operation that the Extractor does is getting the content of the web page that the user is surfing. This content comes in the form html files which includes many useless information beside what appears in the actual web page, such information is: html tags, scripts that developers use in the back end, etc. Next, the content goes through *Filtering Process I*, where most of this unnecessary information gets eliminated, and the remaining is mostly the actual text that appears in the page. The system sends this filtered content to Yahoo Term Extraction, which returns back list of keywords and phrases that it is claimed to represent the document. It is important to mention that among the returned list, there are few keywords that truly reflect the content, however, this list is broad and vague and it is not ordered from most to least relevant as claimed in YTE documentation [10]. Moreover, some returned terms are not actual words or words that do not exist in the original documentation. These words gets eliminated in the *Filtering Process II* where the system checks whether the term exists in the original document and whether it exists in the *stop words list* which is a collection of words that are very common in English language to be a stand alone keyword /phrase such as:

> 'a', 'an', 'the', 'and', 'of', 'i', 'to', 'is', 'in', 'with', 'for', 'as', 'that', 'on', 'at', 'this', 'my', 'was', 'our', 'it', 'you', 'we', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '10', 'about', 'after', 'all', 'almost', 'along', 'also', 'amp', 'another', 'any', 'are', 'area', 'around', 'available', 'back', 'be', 'because', 'been', 'being', 'best', 'better', 'big', 'bit', 'both', 'but', 'by', 'c', 'came', 'can', 'capable', 'control', 'could', 'course', 'd', 'dan', 'day', 'decided', 'did', 'didn', 'different', 'div', 'do', 'doesn', 'don', 'down', 'drive', 'e', 'each', 'easily', 'easy', 'edition', 'end', 'enough', 'even', 'every', 'example', 'few', 'find', 'first', 'found', 'from', 'get', 'go', 'going', 'good', 'got', 'gt', 'had', 'hard', 'has', 'have', 'he', 'her', 'here', 'how', 'if', 'into', 'isn', 'just', 'know', 'last', 'left', 'li', 'like', 'little', 'll', 'long', 'look', 'lot', 'lt', 'm', 'made', 'make', 'many', 'mb', 'me', 'menu', 'might', 'mm', 'more', 'most', 'much', 'name', 'nbsp', 'need', 'new', 'no', 'not', 'now', 'number', 'off', 'old', 'one', 'only', 'or', 'original', 'other', 'out', 'over', 'part', 'place', 'point', 'pretty', 'probably', 'problem', 'put', 'quite', 'quot', 'r', 're', 'really', 'results', 'right', 's', 'same', 'saw', 'see', 'set', 'several', 'she', 'sherree',

'should', 'since', 'size', 'small', 'so', 'some', 'something', 'special', 'still', 'stuff', 'such', 'sure', 'system', 't', 'take', 'than', 'their', 'them', …ect.

The terms that survive this elimination stage are the candidates that are further considered for the remaining of the computations.

In parallel, the system uses the open source, ISummariser, to obtain a summary of the document using the same strategy as a human will follow to write a summary, for instance, first and last sentence of the first paragraph, topic sentence, the first sentence of the second and third paragraph, etc. In addition, the system also extract the title of the document as it appears within the html tags due to the high likelihood of containing the come of the terms that reflects the topic of the web content.

The last operation in the pre-fuzzy stage that the system does is calculating the frequency of each candidate term in the original document, and checks whether or not it exists in summary and the title.

At the end of this stage, the system has the following data:

1) candidate terms/ key phrases
2) summary of the original text content
3) title of the document if any
4) frequency count for each candidate term
5) whether or not each candidate exists in summary and the title

This data are all passed to the fuzzy system for further calculation in order to rank the candidates from most to least relevant. See figure 4.2 for an overview of this stage.

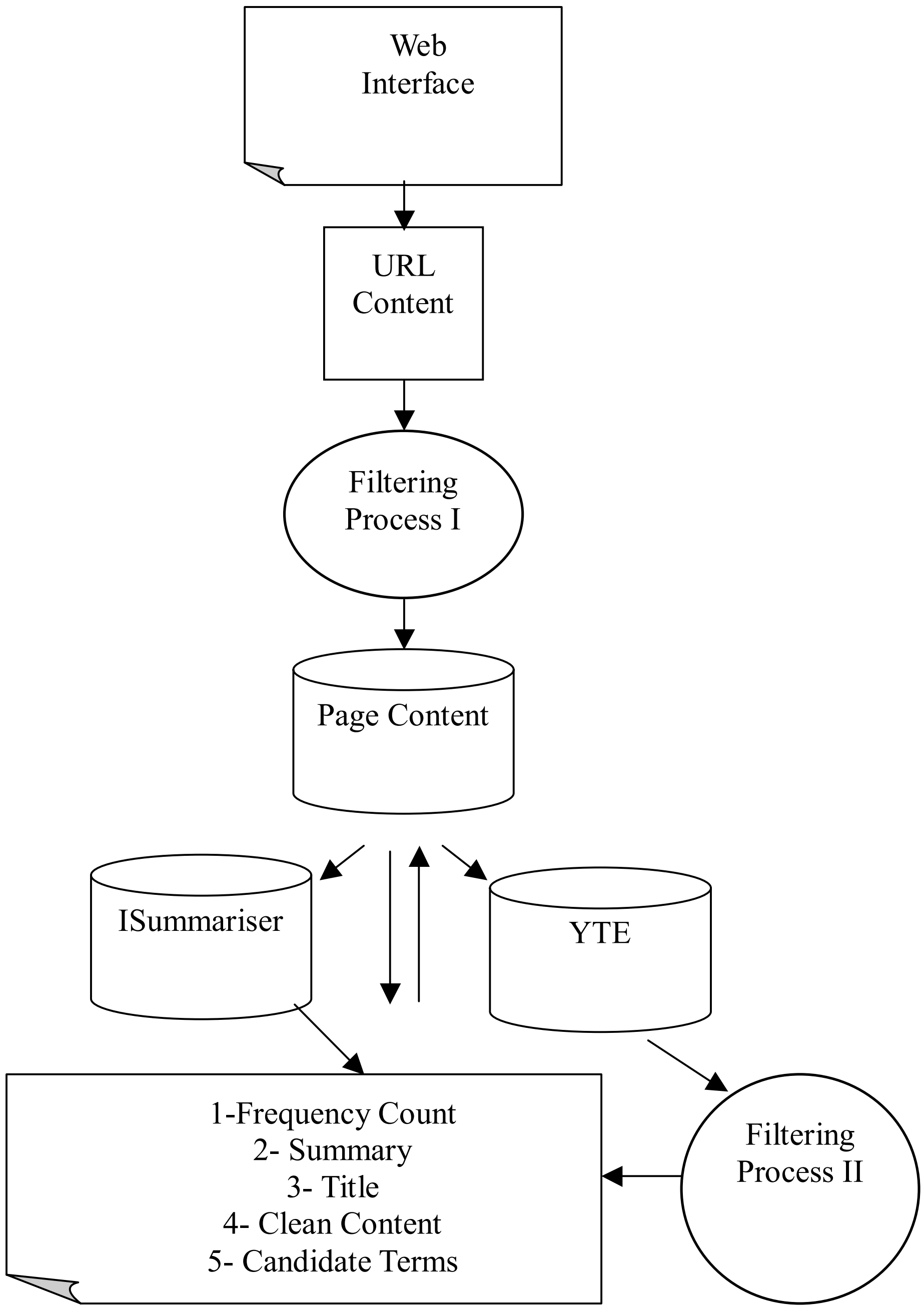


Figure 3.2 Pre-Fuzzy Stage Design Flow

3.3 Fuzzy System Stage

"Fuzzy Logic is basically a multi-valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like rather warm or pretty cold can be formulated mathematically and processed by computers" [Peter Bauer].

In the past few years, the fuzzy logic methodology has been used in many applications as is a powerful problem-solving strategy because of the ability of getting a reasonable conclusion from vague and imprecise information.

In this project, we needed to use the fuzzy system due to answer a major question: what is the degree of association of a given term to the content that is been tested?

We will briefly describe some fuzzy terminology as we discuss our fuzzy system. However, For more in depth history and usage of fuzzy logic and fuzzy system, see [15][16][17][18].

We will divide our fuzzy system into five steps process and we will give a brief overview of each step as part of the analysis which should be efficient to understand how the system works. The three factors that will be used in the fuzzy system and are obtained from the pre-fuzzy stage:

A) Term Frequency Count

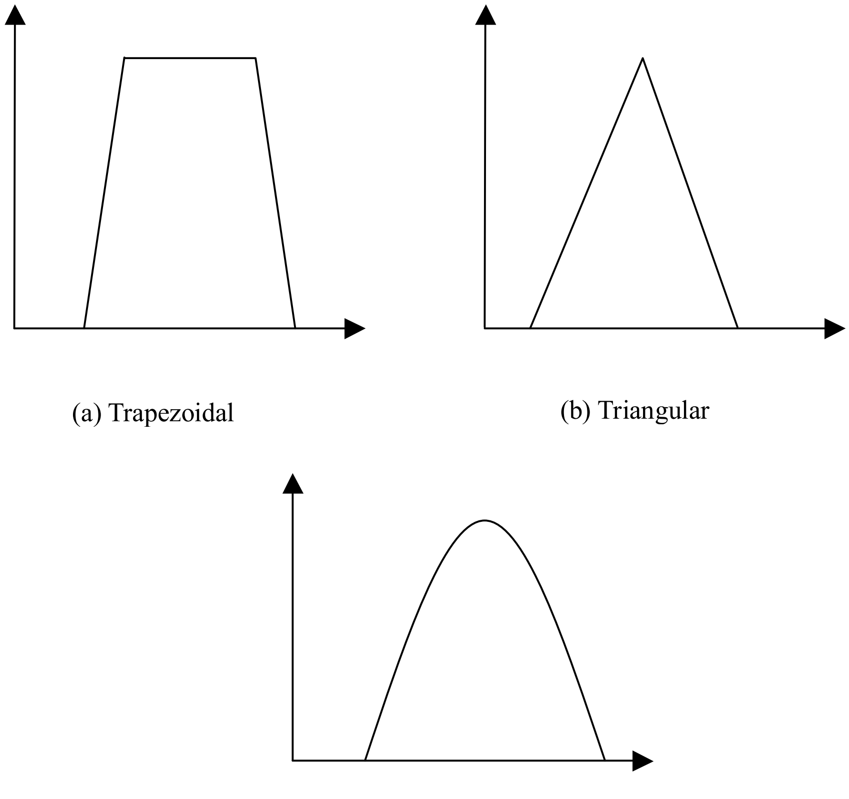
B) Existence in Title or not

C) Existence in the Summary

Along with these factors, the fuzzy system also obtains the original content of the URL and candidate terms.

Step I: Fuzzification

In this step, the crisp values are transformed into fuzzy variable. On other words, each crisp value gets a grade of membership for each linguistic term of the fuzzy set. A membership function is used in this transformation process. There are many membership functions that can be used: Gaussian, trapezoidal and triangular each has a certain shape. The choice of a membership function should reflect the designer knowledge of the subject and it's driven by the application [19].



(a) Trapezoidal

(b) Triangular

(c) Gaussian

Figure 3.3 Membership functions (a) trapezoidal (b) triangular, and (c) Gaussian

In Extractor, there are three factors that the fuzzy system considered to determine the final score for a certain term, however, only the frequency count, can be fuzzified.

Frequency Count	Range Value
High	Input Dependent [0,1]
Medium	Input Dependent [0,1]
Low	Input Dependent [0, 1]

Title	Range Value
Exists	1
Does not Exist	0

Summary	Range Value
Exists	1
Does not Exist	0

Both the existence of the term in the title and summary get either a value of 0 or 1. On the other hand, the frequency factor is divided into three fuzzy sets: low, medium and high. We decided to use Gaussian membership function (gaussmf) to give this linguistic terms mathematical meaning and the main reasons of choosing gaussmf are [20]:

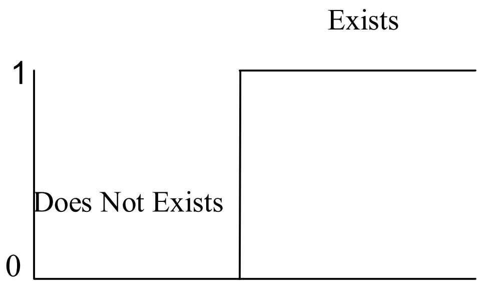
- It provides a smooth transition between member and nonmembers of a fuzzy set.
- It is simple to calculate, and efficient (needed for real time calculations).
- It provides better flexibility and performance comparing to the linear functions.

$$\text{Gaussmf} = Y = ae^{-(G/F)} \text{ where } G = (x-b)^2, \text{ and } F = 2c^2 \rightarrow (1)$$

The parameters:

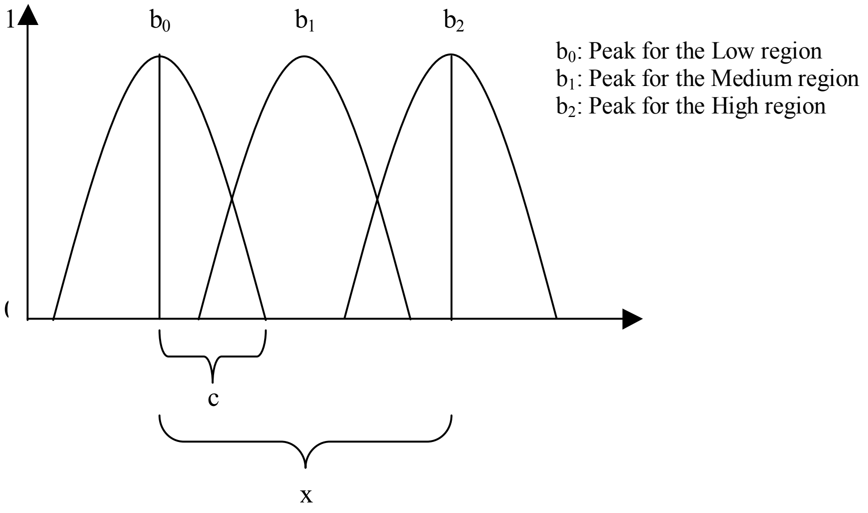
a: the height of the peak (always 1 in our calculation)
x: the frequency count
c: the range (standard deviation)
b: peak position

At the end of the Fuzzification step, each term gets a membership score in respect to each fuzzy set. Therefore at this time of the overall process, each term will have 5 scores as shown in the figure 3.4 (a) and 3.4 (b) below:



$\mu_{title} = \{0, 1\}$ $\mu_{summary} = \{0, 1\}$

(a) Membership Function for Title and Summary


1
b_0
b_1
b_2
b_0: Peak for the Low region
b_1: Peak for the Medium region
b_2: Peak for the High region
(
c
x

$\mu_{freq} = [\mu_{low}, \mu_{medium}, \mu_{high}]$

(b) Membership Function for the frequency

Figure 3.4: System Membership functions

Step II: Rules

In this Stage, the relationship between the inputs and the outputs is constructed and it reflects the designers' decisions and their vision of the problem. This relationship, fuzzy rules, is the heart of the system and it's adjustable to achieve the desired result. These rules are tabulated as fuzzy words and usually are in the form of if/then statements. Extractor fuzzy system includes 12 rules which reflects all possible combinations for the 3 used factors. These rules are as follows:

1. *Frequency High + Title is 1 + Relevance is 1 → Rank is Top High*
2. *Frequency High + Title is 1 + Relevance is 0 → Rank is Top Medium*
3. *Frequency High + Title is 0 + Relevance is 1 → Rank is Top Low*
4. *Frequency High + Title is 0 + Relevance is 0 → Rank is Middle Low*
5. *Frequency medium + Title is 1 + Relevance is 1 → Rank is Top Medium*
6. *Frequency medium + Title is 1 + Relevance is 0 → Rank is Middle High*
7. *Frequency medium + Title is 0 + Relevance is 1 → Rank is Middle Low*
8. *Frequency medium + Title is 0 + Relevance is 0 → Rank is Bottom*
9. *Frequency Low + Title is 1 + Relevance is 1 → Rank is Top Low*
10. *Frequency Low + Title is 1 + Relevance is 0 → Rank is Middle Low*
11. *Frequency Low + Title is 0 + Relevance is 1 → Rank is Bottom*
12. *Frequency Low + Title is 0 + Relevance is 0 → Rank is Bottom*

When one of the rules is activated, the term will be placed in the corresponding position in a ranking table. This table is divided into 6 regions: Top High, Top Medium, Top Low, Medium High, Medium Low, and Bottom. Since we are interested in the terms that will have top scores, we tuned our system so that it focus on the top portion of the table and a clear distinguishing can be made among all candidate terms.

We can view the system rule function as:

$$F = \{L, ML, MH, TL, TM, TH\}$$

Frequency

↓

Table 1. Fuzzy Rules Table

Title →

	Low	Medium	High	
1	Top Low	Top medium	Top High	1
	Middle Low	Middle High	Top Medium	0
0	Bottom	Middle Low	Top Low	1
	Bottom	Bottom	Middle Low	0

Relevance ←

Since we have 6 hierarchy levels in the ranking table, each region is assigned a score between 1- 6 where 1 is the bottom level and 6 is the top high level. These scores will come into play in the last step where the final score for each term is calculated.

Frequency

Table 2. Score Table

Title →

	Low	Medium	High	
1	4	5	6	1
	2	3	5	0
0	1	2	4	1
	1	1	2	0

Relevance ←

Step III: Calculation

The processing of the fuzzy rules begins in this stage where of each tested candidate will go through some calculation to determine its membership to each of the three factors. Therefore, possible final results for a term that for instance a medium high frequency and appears in the title and not the summary will look like:

$\mu_{title} = \{\text{Exists, Doesn't Exist}\} = \{1\}$
$\mu_{summary} = \{\text{Exists, Doesn't Exist}\} = \{0\}$
$\mu_{freq} = [\mu_{low}, \mu_{medium}, \mu_{high}] = [0, 0.3, 0.7]$

Step IV: Rules Activation and Decision Function

At this stage, the rules are activated and calculation of the decision function is computed. If the rules contain more than one decision factor then an operation such as OR, AND, MAX, etc is performed. At the end of this stage, each candidate will have a value between 0 and 1 that represent its membership scores for each of the 6 regions in the ranking score – the closer the value to 1, the high is the membership.

Due to the only possibilities for two of the factors are 0 and 1, one rule from the rule table will be activated. First, membership values will be substituted for each factor and then a logic operation will be performed to combine the scores into one score for the corresponding activated region in the ranking table. Since we are looking for the centered average point among the three factors, we will use the following equations as it is the most robust in the average [21][22]:

$$f_v\,(a, b, c) = (a + b + c) \rightarrow (2)$$

Then to scale the results down to a value between [0,1] we use a scale function, $f_{s,}$,as follows:

$$f_s = f_v\,/3 \rightarrow (3)$$

At the end of this step the rule function F , will be populated with these values, each value represent the membership of a particular term to the corresponding region in the ranking table for instance:

$$F = \{L, ML, MH, TL, TM, TH\}$$

$$F = \{0, 0, 0, 0.16, 0.64, 0.32\}$$

Step V: Defuzzification

There are few Defuzzification methods that can be used to calculate of the final decision of a fuzzy system. The two common used methods are: the maximum method and centroid method where in the first one, the candidate final score is the same as highest membership score in the rule function, on the other hand, the final score using the centroid method will represent the center of gravity of the membership scores [23].

The final decision, FD, is calculated using the second method since it takes into account the calculated values for all the regions in the previous step and most importantly, since each region has a different score value in the score table which is essential to get the desired expected final score and can't be ignored.

$FD = T/S = \sum\mu.D / \sum\mu$ where:

T is the sum of the 3 weighted Y (Gaussian function) scores.

S is the sum of the 3 raw Y scores.

D is the equivalent value from the table score.

On other words, T is the weighted values of each output member function are multiplied by their respective output membership function from the table score, the center points. Then, they are summed together, whereas, S, is the sum of the weighted member function values.

At the end of this step each term is outputted in decreasing order where the term with the highest score placed first and it is the term that the system is considered most relevant to the page visited. Finally, the system passes this ordered list to the post-fuzzy stage.

3.4 Post Fuzzy Stage

In this stage, the top keywords are used to bring additional web content to the user such as images, videos, and certainly other text files. The accuracy of the results that this stage brings to the user relays on the performance of the fuzzy system. If the fuzzy system returned poor results, the output of this stage will also be poor as consequence and vice versa.

In extractor, this is the simplest part of the whole process, where the top ranked keyword is sent to Google Images API and the results are displayed to the user along with the top 3 keywords. The decision of choosing to use Google images API is due to enormous time and data that will be needed to fill a local database with enough labeled images that represent all possible web pages that can be tested. In addition to the availability and the flexibility of using Google image search API to retrieve the desired images. The system allow the user to see images for all three keywords, however, the user gets the images for the first keywords as a default.

4- Implementation:

We have used several technologies and open source software to build extractor. In this section, we will go through a sample execution of the software and will explain the implementations and results for each stage individually.

The sample article below is taking from http://www.sciencenews.com and its content will be used as an input in our example. The full article can be found at http://www.sciencenewsforkids.org/articles/20070613/Feature1.asp

Where Have All the Bees Gone? [24]

“Entomologists—scientists who study insects—have a real mystery on their hands. All across the country, honeybees are leaving their hives and never returning.

It doesn't take long before a hive is nearly empty. Researchers call this phenomenon colony-collapse disorder. According to surveys of beekeepers across the country, 25 to 40 percent of the honeybees in the United States have vanished from their hives since last fall. So far, no one can explain why.

Colony collapse is a serious concern because bees play an important role in the production of about one-third of the foods we eat, including apples, watermelons, and almonds. As they feed, honeybees spread pollen from flower to flower. Without this process, called pollination, a plant can't produce seeds or fruits.

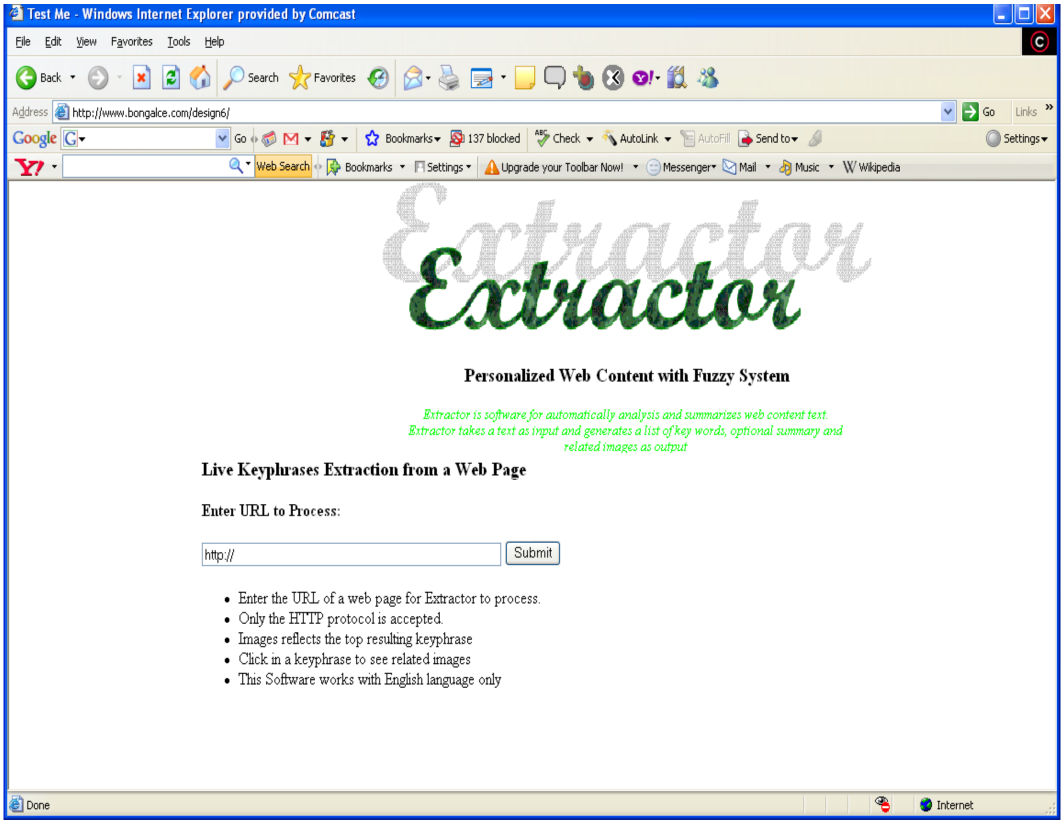
Now, a group of scientists and beekeepers has teamed up to try to figure out what's causing the alarming collapse of so many colonies. By sharing their expertise in honeybee behavior, health, and nutrition, team members hope to find out what's contributing to the decline and to prevent bee disappearances in the future”.

Sick bees

“It could be that disease is causing the disappearance of the bees. To explore that possibility, Jay Evans, a research entomologist at the United States Department of Agriculture (USDA) Bee Research Laboratory, examines bees taken from colonies that are collapsing. "We know what a healthy bee should look like on the inside, and we can look for physical signs of disease," he says. And bees from collapsing colonies don't look very healthy. "Their stomachs are worn down, compared to the stomachs of healthy bees," Evans says. It may be that a parasite is damaging the bees' digestive organs. The bees' inability to ward off such parasites suggests that their immune systems may not be working as they should”. [24]

Stage 1: Content Extraction, YTE Terms

Extractor is hosted in a public server at http://www.bongalce.com/Extractor/main/ where system requires the user to enter a valid URL in either HTTP/HTTPS to be processed. A screen shot of extractor web interface, home page, is shown below in figure 6.1 below.


Test Me - Windows Internet Explorer provided by Comcast
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.bongalce.com/design6/
Go Links
Google Go Bookmarks 137 blocked Check AutoLink AutoFill Send to Settings
Y! Web Search Bookmarks Settings Upgrade your Toolbar Now! Messenger Mail Music Wikipedia
Extractor
Done
Internet

Personalized Web Content with Fuzzy System

Extractor is software for automatically analysis and summarizes web content text. Extractor takes a text as input and generates a list of key words, optional summary and related images as output

Live Keyphrases Extraction from a Web Page

Enter URL to Process:

http:// Submit

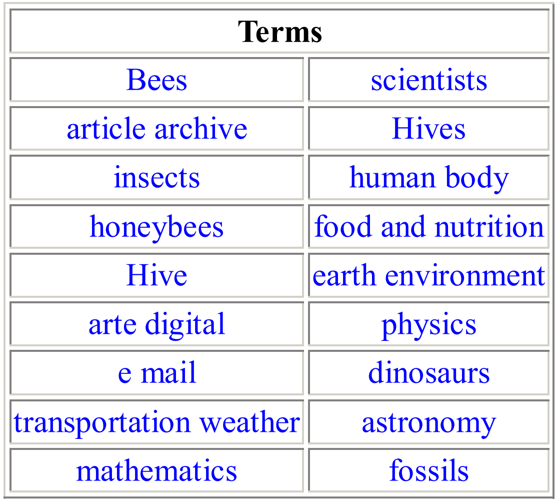
- Enter the URL of a web page for Extractor to process.
- Only the HTTP protocol is accepted.
- Images reflects the top resulting keyphrase
- Click in a keyphrase to see related images
- This Software works with English language only

Figure 4.1: System Web Interface

The software starts out with obtaining the content of web page a user is visiting. Using PHP technology and *curl* which is a free software distributed under the MIT License and used for transferring files with URL syntax, Extractor sends the page content to the first filtering process where content such as html tags , and JavaScript code gets eliminated. Then, it sends filtered content to yahoo term extraction and collects the terms that yahoo sends back. It is important to mention that some the terms and phrases returned may not appear in the article but its part of the web content of the tested page. However, YTE sometimes returns some words that are not actual words or doesn’t even exist in the content. These words get eliminated in the second filtering process. Below are the results of our input after the end of this stage:

Ignored term: phenomen

Table 3. Candidates Terms



Terms	
Bees	scientists
article archive	Hives
insects	human body
honeybees	food and nutrition
Hive	earth environment
arte digital	physics
e mail	dinosaurs
transportation weather	astronomy
mathematics	fossils

These terms in table 3 are the candidates that will be sent to the fuzzy system for further testing and evaluation.

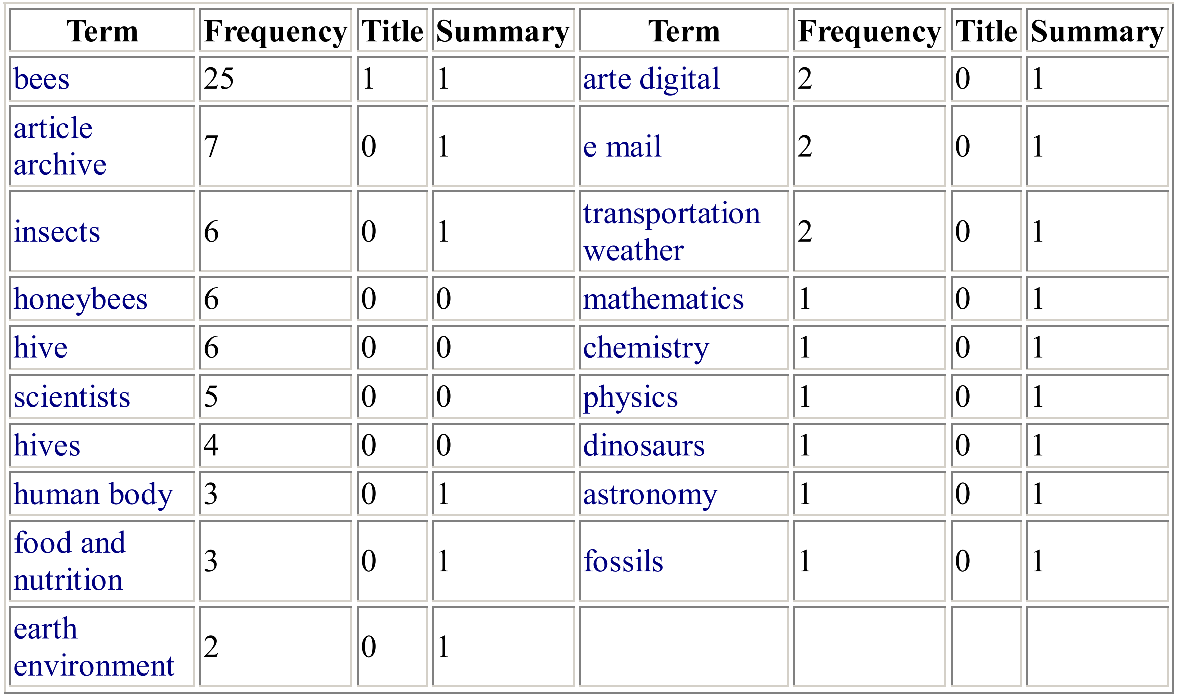
Stage 2: Title and Summary Extraction

Most articles and web pages have a title that can be extracted from the html tags. To obtain a summary for the article we used an open source java library (classifier4j) which contains a summary feature, ISummariser. Since the library is written in java and we using a PHP technology, we used curl and a servlet to interface the two. The summarizerservlet returns the output in XML instead of outputting an html page and only the summary is consists of three sentences which usually captures the key points of the article.

Stage 3: Pre-Fuzzy Calculations

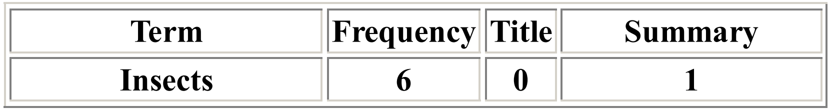
Now, that we have the web content, candidate terms, summary and the title, Extractor, gives a score of a 1 or a 0 for each candidate term indicating whether it exists in the title and the summary and calculates the frequency of these terms in the original document. At then end of this stage a we obtain a similar table as the one below:

Table 4.Calculated Terms' Scores for each Factor



Term	Frequency	Title	Summary	Term	Frequency	Title	Summary
bees	25	1	1	arte digital	2	0	1
article archive	7	0	1	e mail	2	0	1
insects	6	0	1	transportation weather	2	0	1
honeybees	6	0	0	mathematics	1	0	1
hive	6	0	0	chemistry	1	0	1
scientists	5	0	0	physics	1	0	1
hives	4	0	0	dinosaurs	1	0	1
human body	3	0	1	astronomy	1	0	1
food and nutrition	3	0	1	fossils	1	0	1
earth environment	2	0	1				

As we notice from the table above the calculated data is getting larger and larger, therefore, we will focus in one term in the remaining stages of the process. We will use the candidate insects since it is not an extreme case.



Term	Frequency	Title	Summary
Insects	**6**	**0**	**1**

Stage 4: Fuzzy Stage

We wrote our fuzzy engine in PHP using Gaussian's function as our choice of the desired membership function. As we already went through all details of the fuzzy stage and the reasoning of our decision within each stage in the fuzzy system, we will go through a sample computation process using the candidate term, insects, as an input.

Term	Frequency	In the title	In the summary	Y1	Y2	Y3
insects	6	0	1	0.71	0.51	0.00

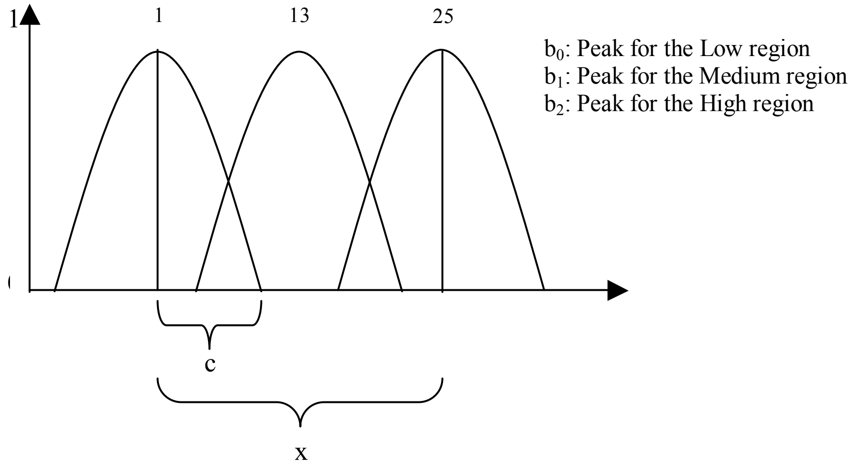

1
1
13
25
b_0: Peak for the Low region
b_1: Peak for the Medium region
b_2: Peak for the High region
c
x

Figure 4.2. Membership Function of the Frequency Factor

At the beginning of this stage, the frequency factor will be fuzzified for each term and Gaussians' membership function will be calculated 3 times to give a degree of membership for the term to each fuzzy set: Low, Medium and High.

$Y = e^{-(G/F)}$ where $G = (x-b)^2$, and $F = 2c^2$
Where:

- b will be recalculated 3 times
- b1 = 1 (Lowest), b2 = 12 (Median), b3 = 25(Highest)
- c = 6 (Range)
- X = 6

Membership Degrees for the term "insects" are:

$$\mu_{\text{freq}} = [\mu_{\text{low}}, \mu_{\text{medium}}, \mu_{\text{high}}]$$

$$\mu_{\text{freq}} = [0.71, 0.51, 0.00]$$

Fuzzy Rules Activation:

since the term insects get a score of 0 for the factor "title" and a score of 1 for the factor "summary", then three rules will be activated as we see in the table below:

			Frequency			
		Low	***Medium***	***High***		
Title	***1***	Top Low	Top medium	Top High	*1*	Summary
		Middle Low	Middle High	Top Medium	*0*	
	0	Bottom	Middle Low	Top Low	*1*	
		Bottom	Bottom	Middle Low	*0*	

Figure 4.3: Rules Activation

Substitute the Values:

			Frequency			
		0.71	***0.51***	***0.00***		
Title	***1***	Top Low	Top medium	Top High	*1*	Summary
		Middle Low	Middle High	Top Medium	*0*	
	0	Bottom	Middle Low	Top Low	*1*	
		Bottom	Bottom	Middle Low	*0*	

Figure 4.4: Values Substitution

Combine the Three Factors (logic operation):

Frequency

	0.71	***0.51***	***0***	
1	Top Low	Top medium	Top High	*1*
	Middle Low	Middle High	Top Medium	*0*
0	0.57	0.50	0.33	*1*
	Bottom	Bottom	Middle Low	*0*

Title

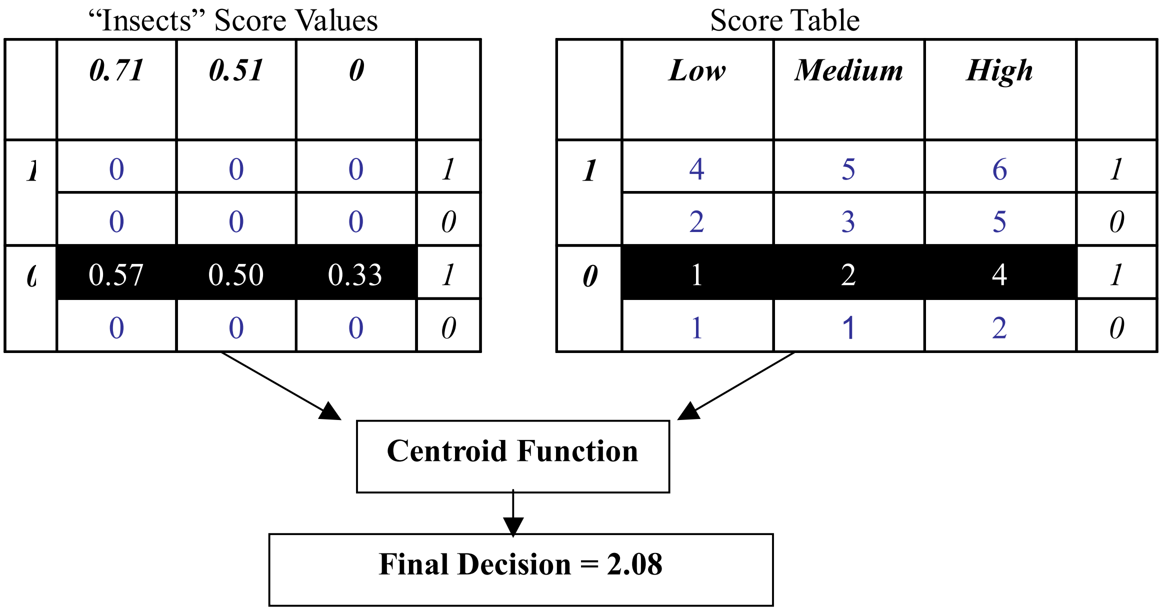
Summary

Figure 4.5: Perform Logic Operation

$$F = \{L, ML, MH, TL, TM, TH\}$$

$$F = \{0.57, 0.50, 0, 0.33, 0, 0\}$$

Final Scaled Fuzzified Decision (Centroid Method):



"Insects" Score Values

	0.71	***0.51***	***0***	
1	0	0	0	*1*
	0	0	0	*0*
0	0.57	0.50	0.33	*1*
	0	0	0	*0*

Score Table

	Low	***Medium***	***High***	
1	4	5	6	*1*
	2	3	5	*0*
0	1	2	4	*1*
	1	1	2	*0*

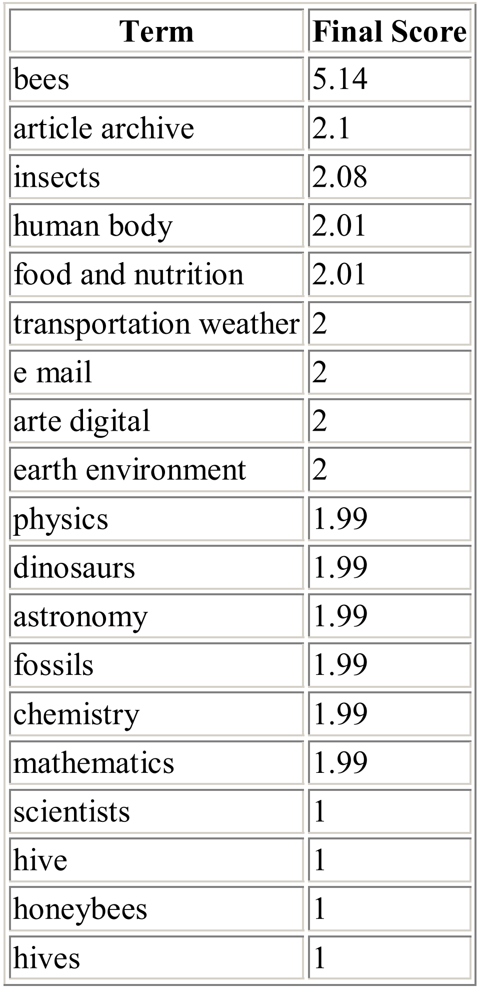
Centroid Function

Final Decision = 2.08

Figure 4.6: Final Score Calculation

Each term gets a final value between 1 and 6 represent the membership degree of this term to the document where the close the value to 6, the higher the membership degree. The three terms with the highest scores are sent to the post fuzzy stage where the retrieval of images takes place. Below are the final scores for all the candidates from our sample article and they are ordered in from highest to lowest. The top 3 terms: bees, article archive, and insects will be used in the next stage.

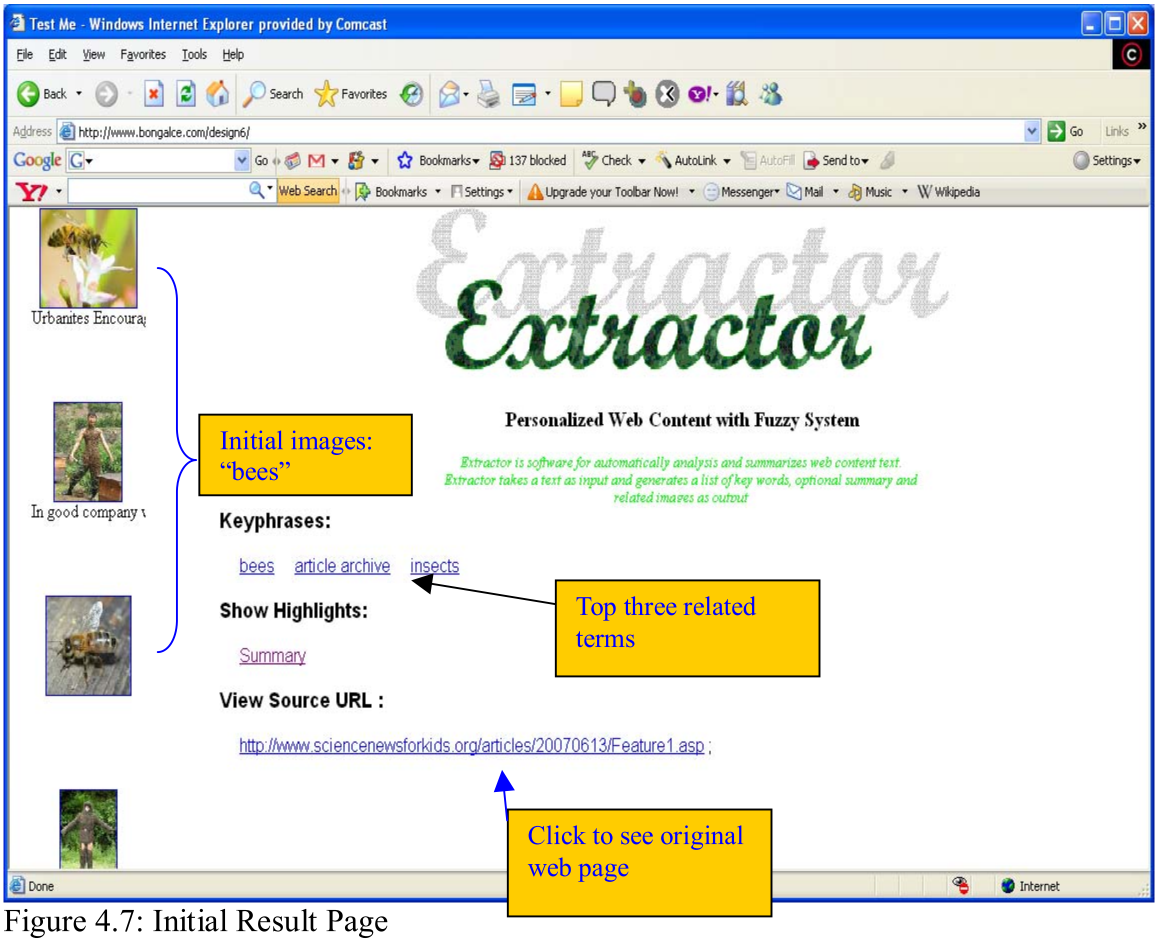
Table 5 Final Scores for all Candidates



Term	Final Score
bees	5.14
article archive	2.1
insects	2.08
human body	2.01
food and nutrition	2.01
transportation weather	2
e mail	2
arte digital	2
earth environment	2
physics	1.99
dinosaurs	1.99
astronomy	1.99
fossils	1.99
chemistry	1.99
mathematics	1.99
scientists	1
hive	1
honeybees	1
hives	1

Stage 5: Image Retrieval

This is the stage where the system uses the most related terms to bring additional web content to the user: other web pages, images or videos. Extractor is designed to return the closely related images to the web page from Google images API. Initially, only the images that represent the term with the absolute highest score displayed to the user, however, the user has the choice of seeing additional images that represent the second and the third highest term. Figure and figure show the results of the web page we tested in this section with the initial images returned and the additional optional images requested by the user [11].



Personalized Web Content with Fuzzy System

Extractor is software for automatically analysis and summarizes web content text.
Extractor takes a text as input and generates a list of key words, optional summary and related images as output

Keyphrases:

bees article archive insects

Show Highlights:

Summary

View Source URL :

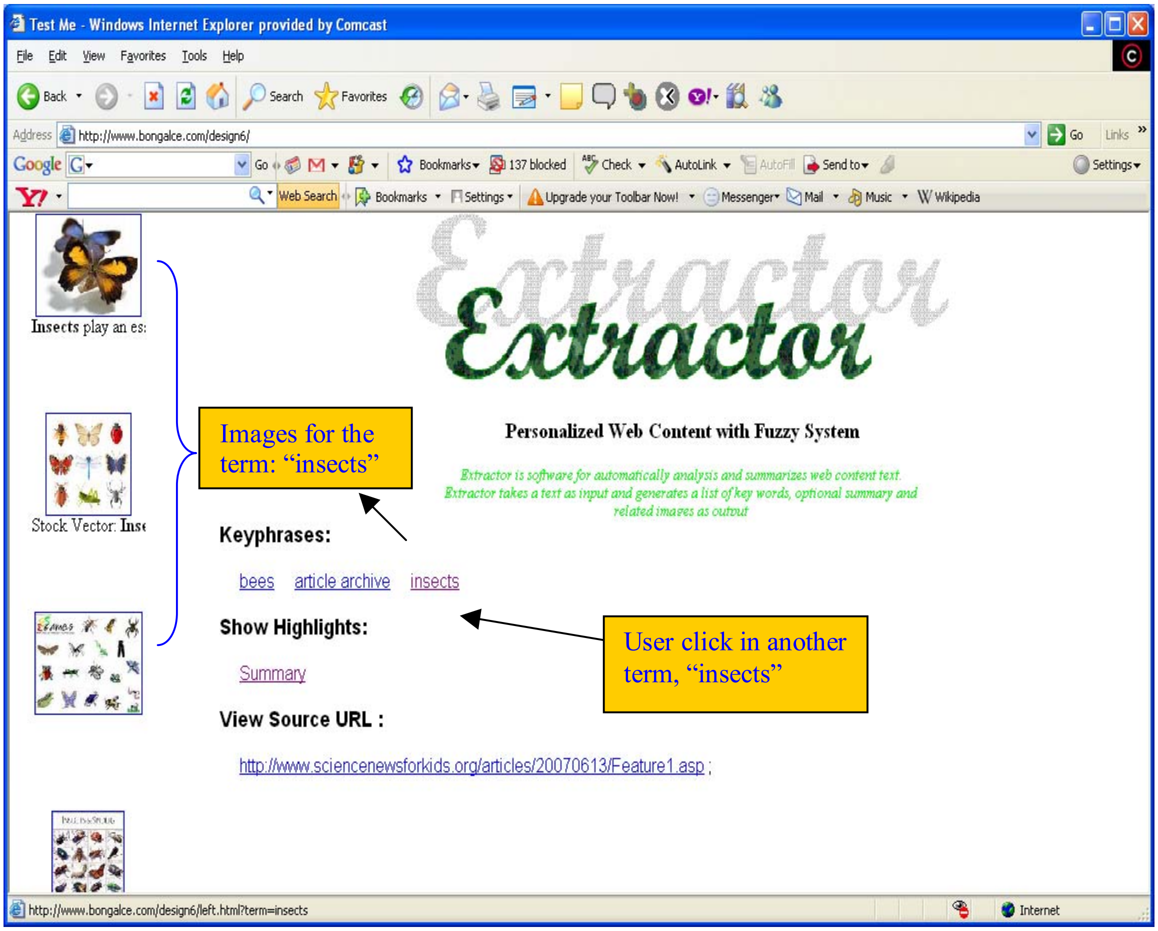
http://www.sciencenewsforkids.org/articles/20070613/Feature1.asp ;

Initial images: "bees"

Top three related terms

Click to see original web page

Figure 4.7: Initial Result Page


Test Me - Windows Internet Explorer provided by Comcast
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.bongalce.com/design6/
Go Links
Google Go Bookmarks 137 blocked Check AutoLink AutoFill Send to Settings
Web Search Bookmarks Settings Upgrade your Toolbar Now! Messenger Mail Music Wikipedia
Insects play an es:
Stock Vector: Inse
http://www.bongalce.com/design6/left.html?term=insects
Internet

Extractor

Personalized Web Content with Fuzzy System

Extractor is software for automatically analysis and summarizes web content text. Extractor takes a text as input and generates a list of key words, optional summary and related images as output

Images for the term: “insects”

Keyphrases:

bees article archive insects

Show Highlights:

Summary

View Source URL :

http://www.sciencenewsforkids.org/articles/20070613/Feature1.asp ;

User click in another term, “insects”

Figure 4.8. Alternative Result Page

If the user chooses to view any of these images, the image appears in a new window so that the user doesn’t get redirected from the original output page. For instance, in the figure above, if the user clicked in the top image from the “insects” results list, a new window pops up contains the requested image. As we notice above, the original URL that contains the picture is in the address bar and the picture can be printed, saved, or send to another user. Of course all these images are copy righted to the original web page that contains them

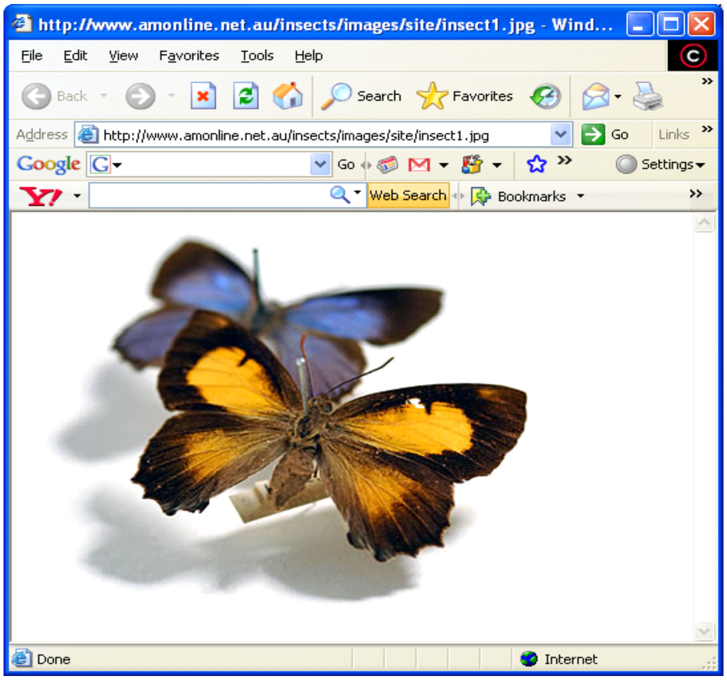

http://www.amonline.net.au/insects/images/site/insect1.jpg - Wind...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://www.amonline.net.au/insects/images/site/insect1.jpg
Go
Links
Google
Go
Settings
Web Search
Bookmarks
Done
Internet

Figure 4.9 Sample Image Output

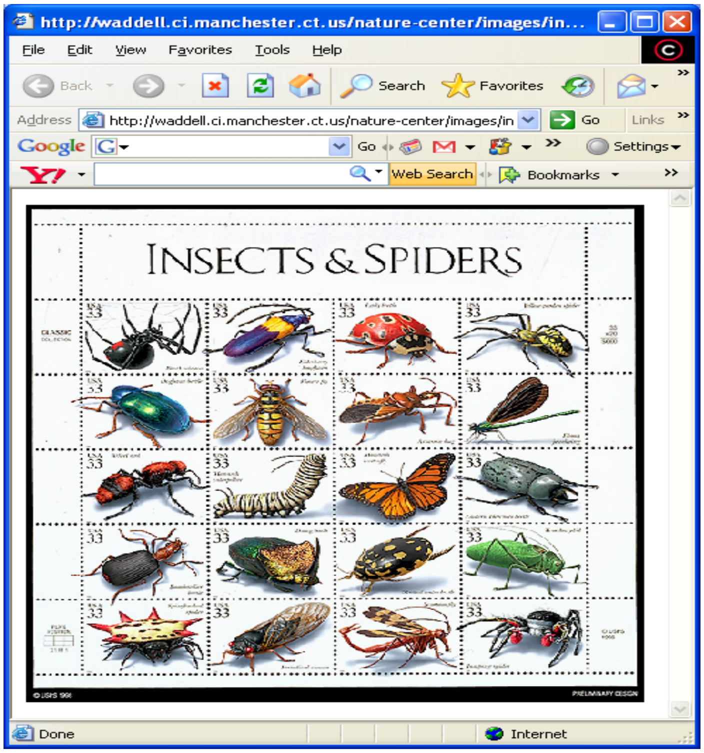

http://waddell.ci.manchester.ct.us/nature-center/images/in...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://waddell.ci.manchester.ct.us/nature-center/images/in
Go
Links
Google
Go
Settings
Web Search
Bookmarks
INSECTS & SPIDERS
Done
Internet

Figure 4.10 Another Sample Image Output

The “blue prints” of Extractor begging from the home page till the displaying of a particular image is shown in the figure below:

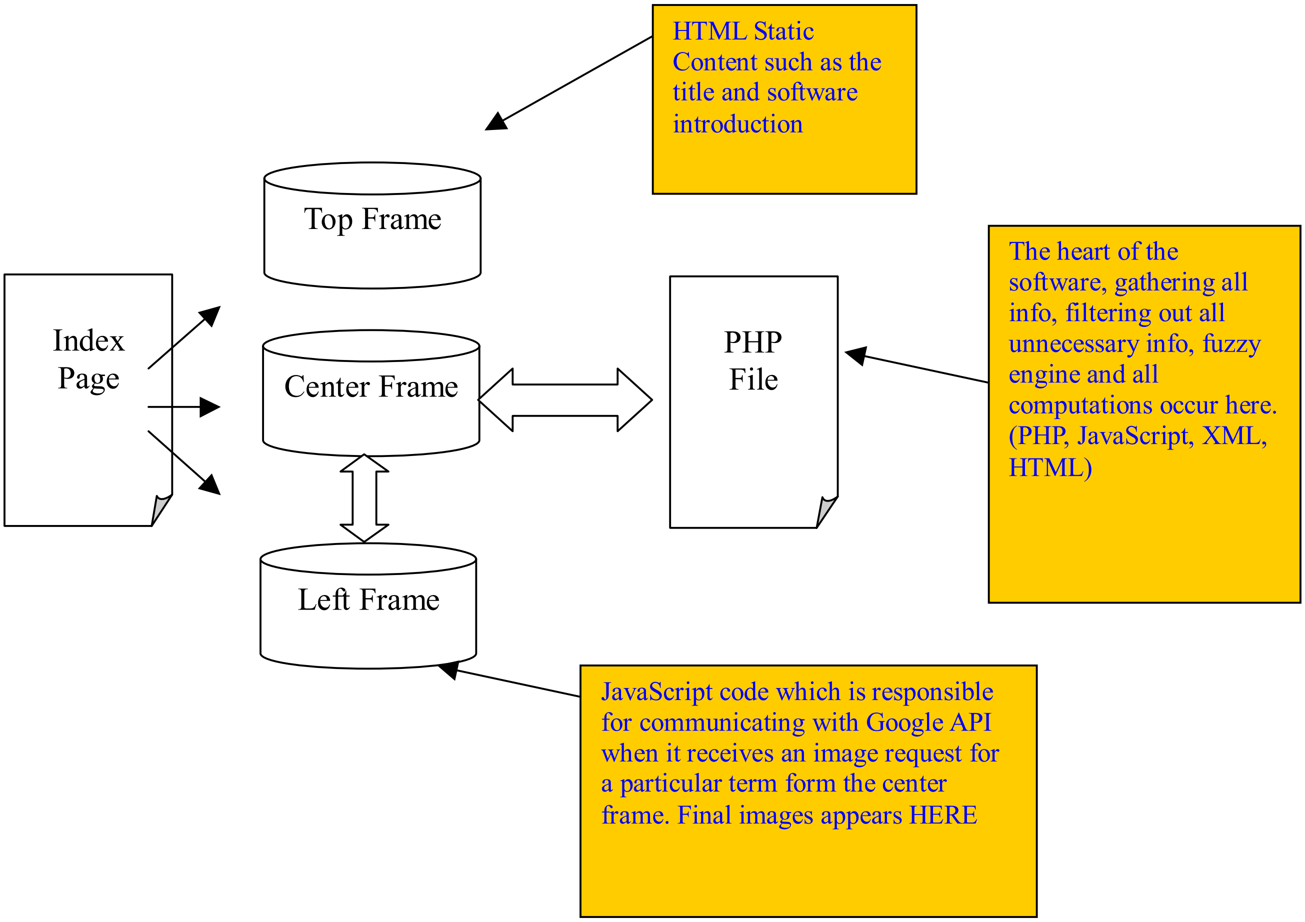

HTML Static Content such as the title and software introduction
Top Frame
Index Page
Center Frame
PHP File
The heart of the software, gathering all info, filtering out all unnecessary info, fuzzy engine and all computations occur here. (PHP, JavaScript, XML, HTML)
Left Frame
JavaScript code which is responsible for communicating with Google API when it receives an image request for a particular term form the center frame. Final images appears HERE

Figure 4.21 “Blue Prints” of the Online System Architecture

5. Results:

Extractor’s takes web pages as an input, analyze their contents and summarize them into several meaningful terms or phrases. In this section, we will evaluate the performance of the software and since there are thousands and perhaps millions of web pages online, we will divide this section into two parts: input material and software results.

5.1 Input material:

The input to Extractor is a url link and since there are tremendous amount of web pages and in order and to follow non-bios approach for evaluating the software, we decided to use domains and input categories as similar scientific papers which covered the same topic yet used different computing methodology [25][26][27].

We divided our testing inputs into three sub categories based on these three papers: News, blogs and reviews and articles from NASA, FIPS (Federal Information Processing Standards Publications). The news category was mainly taken from cnn.com, abcnews.com, yahoo news, Google news, and other top online news pages. These news included variety of sub domains such as politics, sports, medicine and health, entertainment and sports. The second category was taken was taken mainly from eopinion.com where all the input data for [26] was taken. It includes review and blogs about all kinds of products and services that are offered online (books, music, movies, travels, etc). Last, list of links was taken from NASA and FIPS which are two of the main input sources that were used in [27].

5.2 Extractor Results:

For each page tested, we evaluated the following:

1- Efficiency of the software (in second)

2- Consistency

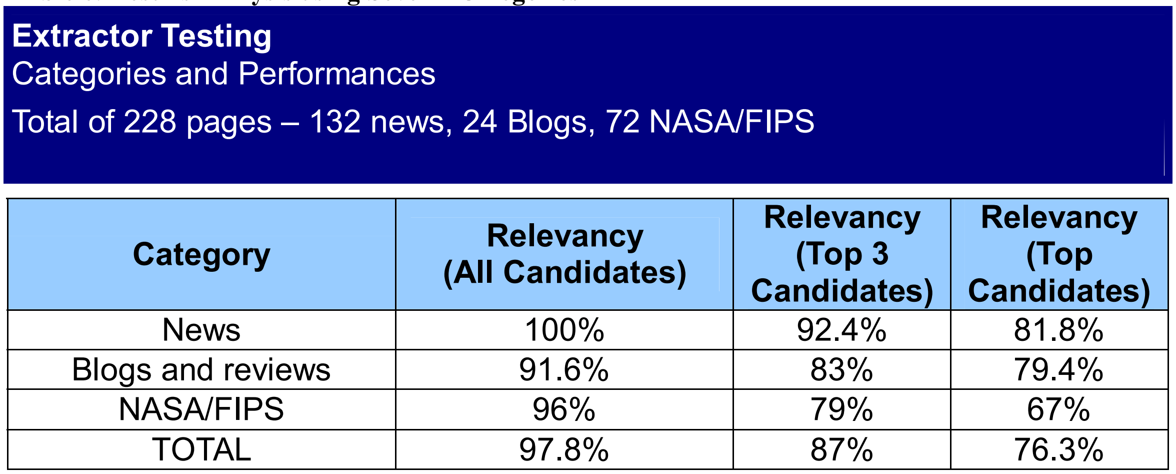
3- Relevancy (Do the terms reflect the content of the page?)

   a. Among all candidates used

   b. Top three candidates

   c. Top candidate

Table 6. Results Analysis using Several Categories



Extractor Testing

Categories and Performances

Total of 228 pages – 132 news, 24 Blogs, 72 NASA/FIPS

Category	Relevancy (All Candidates)	Relevancy (Top 3 Candidates)	Relevancy (Top Candidates)
News	100%	92.4%	81.8%
Blogs and reviews	91.6%	83%	79.4%
NASA/FIPS	96%	79%	67%
TOTAL	97.8%	87%	76.3%

As we see from the table above, extractor best performance is when testing news domains which its content tends to be organized and focused. On the other hand, blogs, reviews, and FIPS documents which tend to be either not very well structured and/or less focused are resulting in a lower performance percentage by the system.

The system is consistent in which it returns the same results for the same page as long as the content of the page has not changes. The computation time for each page is between 2-5 seconds which we consider an important factor since the software is intended for online use.

6. Comparison and Discussion:

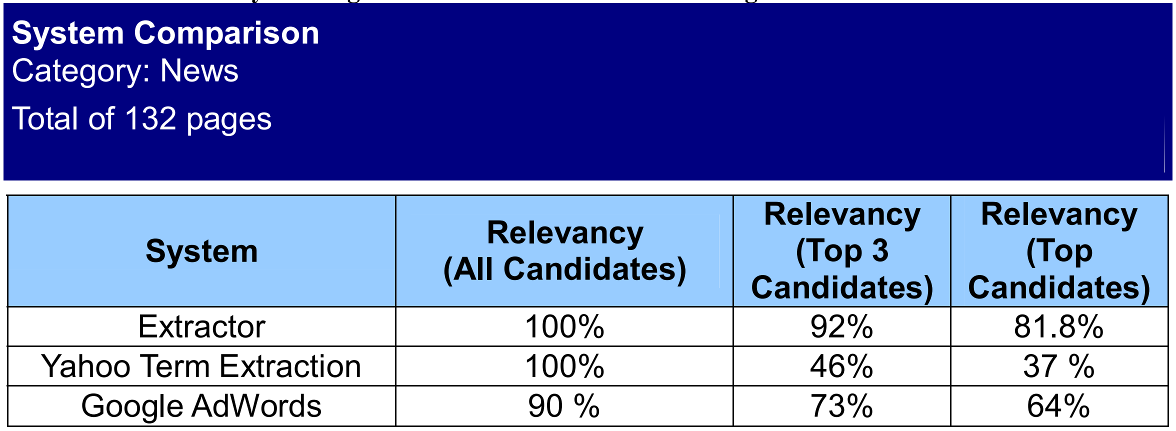
6.1 Comparators

We going to compare our system performance to another two how are build by two of the giant companies: Yahoo and Google. Yahoo made the API, YTE, which we used in the earlier stage of our system. Although the full list of candidates in Extractor is overlapped with YTE, the main comparison will be in the order which both systems claimed it is outputted where most important terms appears first[10].

The second API, AdWords, is made by Google. It outputs many keywords and phrases, 50 and maybe more per link test, however, it groups them to few keywords listed by relevancy [28].

6.2 Comparison

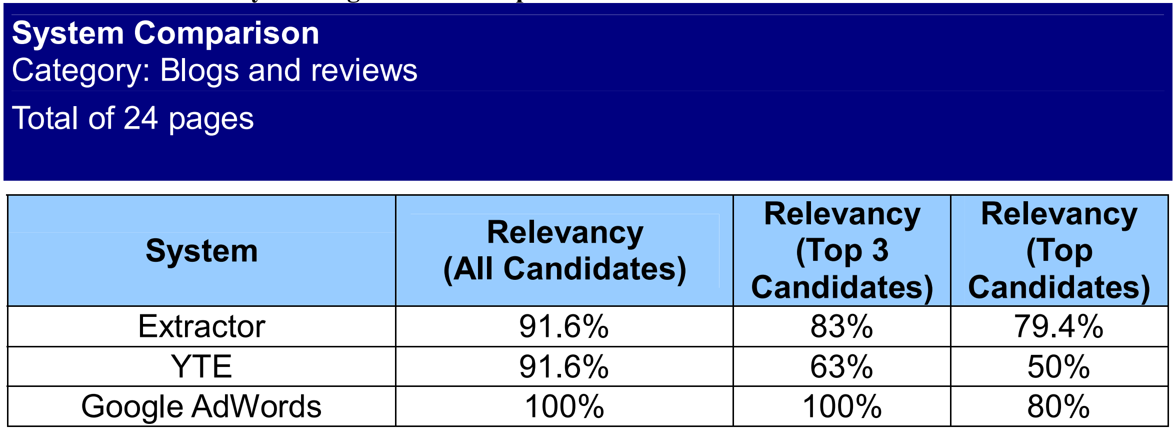
Table 7. Results Analysis using News from Several online Leading sites



System Comparison
Category: News
Total of 132 pages

System	Relevancy (All Candidates)	Relevancy (Top 3 Candidates)	Relevancy (Top Candidates)
Extractor	100%	92%	81.8%
Yahoo Term Extraction	100%	46%	37 %
Google AdWords	90 %	73%	64%

Table 8. Results Analysis using Data from eopinion.com



System Comparison
Category: Blogs and reviews
Total of 24 pages

System	Relevancy (All Candidates)	Relevancy (Top 3 Candidates)	Relevancy (Top Candidates)
Extractor	91.6%	83%	79.4%
YTE	91.6%	63%	50%
Google AdWords	100%	100%	80%

Table 9. Results Analysis using Data from NASA/FIPS

System Comparison
Category: NASA/FIPS
Total of 72 pages

Category	Relevancy (All Candidates)	Relevancy (Top 3 Candidates)	Relevancy (Top Candidates)
Extractor	96%	79%	67%
YTE	96 %	58%	38%
Google AdWords	92%	71%	54%

Table 10. Overall Performances of YTE, Google AdWords, and Extractor

System Comparison
All Categories – News, Blogs and Reviews, FIPS and NASA
Total of 228 pages

System	Relevancy (All Candidates)	Relevancy (Top 3 Candidates)	Relevancy (Top Candidates)
Extractor	97.8%	87%	76.3%
YTE	97.8%	52%	39%
Google AdWords	92%	75%	63%

Table 11. Consistency and Efficiency of YTE, Google AdWords, and Extractor

System Comparison
Consistency and Efficiency

System	Consistency	Computing time
Extractor	Stable	2-5 seconds
YTE	Stable	2-5 seconds
Google AdWords	Non-Stable	Up to 1 minutes

6.3 Discussion

The accuracy and efficiency of the terms extraction process are the keys for this software to be used online. Both, AdWords and YTE have major problems to be used online in their current state. As mentioned in table 10 above, Google's AdWords is not consistence. In many occasions, it produces different keywords for the same page and also, it takes up to one minute to produce the results. Moreover, if the consistency problem is fixed, AdWords top 3 phrases are 75% accurate and about 67% for the top phrase.

On the other hand, Yahoo term Extraction returns the results in 2-5 seconds which is very reasonable for software that will be used online. However, its accuracy level in terms of the top phrase and the top 3 phrases are generally low where it scores 52% and 39% respectively. The list of all the terms and phrases are consistence and it represent the page with a high score of 97.8% which is the reason of choosing YTE outputted list as first part of the Extractor overall process.

Extractor main strength comes from the ability to produce the most relative terms better than the other two systems where it has a score of 87% and 76.3% for the top 3 outputted phrase and the top most phrase respectively. Extractor seems to produce the best results when test online news pages and this is due the fact that these articles are focused on a specific topic and well organized.

7. Conclusion and Future Enhancements

7.1 Conclusion

This project is designed toward an automotive system which can analyze and predict the user intentions and interest and based on that it suggest to them the appropriate results that matches with it. Many research projects can be built on top of this system due to imperfection of search engines especially in images, audio or video content which mainly still based on text such as tags, or title description which is still far from accuracy and more importantly they still inputted

7.2 Future Direction and Enhancements

There is always a room for improvements especially the accuracy of the terms returned which should be achieved in several ways such as dictionary hierarchy of the software used under specific domain, a training system so that useless term get eliminated over time, better summarizer, other factors that we can incorporate in the fuzzy system etc. Obviously, the image results still far from decent since it is still based on text tagged to the image, not the content of the image itself.

This software can be used to bring other forms of web content such as videos, other web links, perhaps audio too and for easy access a technical improvements will be to have this software as a button in the browser which can be clicked when additional information is needed.

8. References

[1] M. Caramiaa, G. Felici, & A. Pezzoli, “Improving search results with data mining in a thematic search engine”, Computers & Operations Research, 31, 2387-2404 (2004).

[2] Krulwich, B., and Burkey, C. (1996). Learning user information interests through the extraction of semantically significant phrases. In M. Hearst and H. Hirsh, editors, AAAI 1996 Spring Symposium on Machine Learning in Information Access. California: AAAI Press.

[3] Muñoz, A. (1996). Compound key word generation from document databases using a hierarchical clustering ART model. Intelligent Data Analysis, 1 (1), Amsterdam: Elsevier.

[4] Hiroshi Nakagawa, Tatsunori Mori A Simple but Powerful Automatic Term Extraction Method

[5]Bourigault, D. (1992) Surface grammatical analysis for the extraction of terminological noun phrases. In Proceedings of the 14th International Conference on Computational Linguistics, COLING’92 pp.977-981.

[6] Dagan, I. and K. Church. (1994) Termight: Identifying and terminology In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart/Germany, 1994. Association for Computational Linguistics.

[7] Frantzi, K.T. and S.Ananiadou (1999) The C-value/NC-value domain independent method for multi-word term extraction. Journal of Natural Language Processing, 6(3) pp. 145-180

[8] Jong-Hoon Oh, KyungSoon L, and Key-Sun C. Term Recognition Using Technical Dictionary Hierarchy.Proceedings of the 38th Annual Meeting on Association for Computational Linguistics 2000. Hong Kong.Pages: 496 – 503.

[9] Khosrow Kaikhah Automatic Text Summarization with Neural, Second IEEE international conference on intelligen systems, June 2004

[10] Yahoo Term Extraction. http://developer.yahoo.com/search/content/V1/termExtraction.html

[11] Google Search API. http://code.google.com/apis/ajaxsearch/

[12] classifier4j. http://classifier4j.sourceforge.net.).

[13] Baye's theory http://plato.stanford.edu/entries/bayes-theorem/

[14] Vector classifier http://www.perl.com/pub/a/2003/02/19/engine.html

[15] Mai Gerhke, Carol L. Walker, and Elbert A. Walker. Normal forms and truth tables for fuzzy logics. Fuzzy Sets And Systens, 138:25–51, 2003.

[16] L. A. Zadeh. Fuzzy sets. Inf. and control, 8:338–353, 1965.

[17] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. Information Sciences, 8:43–80, 1975.

[18] L. A. Zadeh. Fuzzy logic. IEEE Computer, 21(4):83–93, April 1988.

[19] Berthold Michael, Hand D. J. Intelligent Data Analysis: An Introduction.

[20] Rubens N.O. The Application of Fuzzy Logic to the Construction of the Ranking Function of Information Retrieval Systems. Computer Modelling and New Technologies, 2006, Vol.10, No.1, 20-27

[21] Hung T. Nguyen, Walker E. First course in fuzzy logic, CRC Press, Boca Raton, FL, 1999.

[22] Hung T. Nguyen, Vladik Kreinovich Which Fuzzy Logic Is the Best: Pragmatic Approach (and Its Theoretical Analysis) Elsevier Science, June 05

[23] Wen-Liang Hung and Jong-Wuu Wu, Correlation of intuitionistic fuzzy sets by centroid method, Elsevier Science Inc Volume 144, Issues 1-4, July 2002, Pages 219-225

[24] Where Have All the Bees Gone? http://www.sciencenewsforkids.org/articles/20070613/Feature1.asp

[25] Yu-Sheng Lai, Chung-Hsien Wu, Meaningful Term Extraction and Discriminative Term Selection in Text Categorization via Unknown-Word Methodology ACM Transactions on Asian Language Information Processing, Vol. 1, No. 1, March 2002.

[26] Cheng Junsheng, Liu Bing, Hu Minqing, Opinion Observer: Analyzing and Comparing Opinions on the Web, ACM 1-59593-046-9/05/0005

[27] Peter D. Turney, Learning Algorithms for Keyphrase Extraction, 1999 National Research Council Canada

[28] Google AdWords https://adwords.google.com/select/KeywordToolExternal