

2008

# Fuzzy Content Mining for Targeted Advertisement

Yezhou Wang  
*San Jose State University*

Follow this and additional works at: [https://scholarworks.sjsu.edu/etd\\_projects](https://scholarworks.sjsu.edu/etd_projects)

Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Wang, Yezhou, "Fuzzy Content Mining for Targeted Advertisement" (2008). *Master's Projects*. 117.  
DOI: <https://doi.org/10.31979/etd.vdxh-j74b>  
[https://scholarworks.sjsu.edu/etd\\_projects/117](https://scholarworks.sjsu.edu/etd_projects/117)

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact [scholarworks@sjsu.edu](mailto:scholarworks@sjsu.edu).

Fuzzy Content Mining for Targeted Advertisement

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

Yezhou Wang

May 2008

© 2008

Yezhou Wang

ALL RIGHTS RESERVED

The final report of the project Fuzzy Web Mining for Targeted  
Advertisement by Yezhou Wang at  
[http://www.cs.sjsu.edu/private/mscs/reports/wang\\_yezhou.pdf](http://www.cs.sjsu.edu/private/mscs/reports/wang_yezhou.pdf) has been  
approved.

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

---

Professor Chris Tseng

Date

---

Professor Ron Mak

Date

---

Professor Soon Tee Teoh

Date

## **Abstract**

### Fuzzy Content Mining for Targeted Advertisement

By Yezhou Wang

Content-targeted advertising system is becoming an increasingly important part of the funding source of free web services. Highly efficient content analysis is the pivotal key of such a system. This project aims to establish a content analysis engine involving fuzzy logic that is able to automatically analyze real user-posted Web documents such as blog entries. Based on the analysis result, the system matches and retrieves the most appropriate Web advertisements.

The focus and complexity is on how to better estimate and acquire the keywords that represent a given Web document. Fuzzy Web mining concept will be applied to synthetically consider multiple factors of Web content. A Fuzzy Ranking System is established based on certain fuzzy (and some crisp) rules, fuzzy sets, and membership functions to get the best candidate keywords. Once it has obtained the keywords, the system will retrieve corresponding advertisements from certain providers through Web services as matched advertisements, similarly to retrieving a products list from Amazon.com. In 87% of the cases, the results of this system can match the accuracy of the Google Adwords system. Furthermore, this expandable system will also be a solid base for further research and development on this topic.

## Table of Contents

1. Introduction.....	6
2. Drupal CMS Design.....	8
3. Motivation and Fuzzy Logic.....	10
4. Content-targeted Analysis.....	14
4.1 Content gathering and filtering.....	16
4.2 Candidate Select.....	17
4.2.1 Select mode.....	17
4.2.2 Yahoo Term Extraction implementation .....	18
4.2.3 Post-processing .....	21
4.3 Attributes of Candidate .....	21
4.3.1 Term Frequency .....	21
4.3.2 Title overlap .....	24
4.3.3 META information and location.....	24
4.4 Fuzzy Inference System Process.....	26
4.4.1 Fuzzy Rules.....	26
4.4.2 Implication and Aggregation .....	28
4.5 Ads Retrieval .....	33
5. System Demo and Evaluation.....	34
5.1 Step-by-Step Ads Generation .....	35
5.2 Performance Evaluation.....	37
6. Reference .....	38

## List of Figure and Tables

Figure 1.1: Google AdSense and its delay.....	7
Figure 2.1: Drupal application web architecture.....	9
Figure 2.2: Front page of a Drupal-driven CMS.....	9
Figure 3.1: Crisp Set vs. Fuzzy Set.....	11
Figure 3.2: A membership function.....	12
Figure 3.3: Gaussian Membership Function.....	12
Figure 3.4: FIS process.....	14
Figure 4.1: System architecture model.....	15
Figure 4.2: Gaussian curve of TF.....	27
Figure 4.3: Gaussian curve of overlap.....	28
Figure 4.4: Ads generated.....	34
Figure 5.1: System main page.....	35
Figure 5.2: New content submission.....	36
Figure 5.3: Newly posted content with generated ads.....	37
Figure 5.4: Google Adwords.....	37
Table 4.1: Term frequency sample.....	23
Table 4.2: Title Overlap.....	24
Table 4.3: Candidates weight table.....	32

## **Keywords**

**Content-targeted advertising:** Programs automatically find relevant keywords on a web page, and then display advertisements based on those keywords [3].

**Keyword-targeted (Search-targeted) advertising:** Keywords extracted from a search query are matched against keywords associated with ads provided by advertisers. [2]

**Fuzzy logic** is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem (Klir 1997).

## **1. Introduction**

Since the first Web advertisement appeared on 1994, online advertising has been greatly improved to better match potential customer's personal interests. An approach widely accepted is to make online advertisements displayed to the online users relate to the Web content that is browsed to maximize the advertising effect. Therefore, how to accurately analyze Web-based content (such as an HTML format document) and then bring out the most appropriate ads has become a crucial topic. For example, Google (who call themselves an advertising company rather than a computer science company) has released a competitive content-targeted Web-based application Google AdSense to analyze Web content and then retrieve the relevant advertisements for the user who is reading the Web page.



Home » Blogs » My blog

## Microsoft Board Meets Over Yahoo Bid

[View](#) [Edit](#)

Submitted by [admin](#) on Thu, 05/01/2008 - 08:16.

Microsoft board met on Wednesday to decide how to proceed in the company's bid to acquire Yahoo, although no final decision was reached, The Wall Street Journal reported.

The major stumbling block in the negotiations has been the price, which Microsoft is willing to increase to up to US\$33 per share, but not to the \$35 to \$37 range that major Yahoo shareholders, management and board members want, the Journal said, citing anonymous sources. Microsoft's original cash-and-stock offer, made on Feb. 1 and valued at \$44.6 billion at the time, stood at \$29.12 as of Tuesday's market close, the paper said.

One observer suspected that Microsoft leaked the information about its latest deliberations to help push its agenda.

"This leak is obviously a calculated attempt to dangle another few dollars in front of Yahoo shareholders in hopes that they will put pressure on Yahoo to strike a deal," wrote Silicon Alley Insider's Henry Blodget in a blog post. "We suspect Ballmer and the board may now wait and see what impact this leak has before making their final decision."

There was no official announcement from either company by the close of the U.S. business day Wednesday. A spokeswoman for Microsoft said the company does not comment on board meetings, citing company policy.

An announcement from Microsoft is now expected later in the week, the Journal said.

Microsoft's next move is something that all parties with a stake in the deal have been waiting for since Yahoo failed to agree to a deal by Saturday, the deadline Microsoft had set three weeks earlier.

Many observers had expected a reaction from Microsoft first thing Monday morning, but as the silence stretched into Wednesday afternoon, the media speculation mill has gone into overdrive.

The major stumbling block in the negotiations has been the price, which Microsoft is willing to increase to up to US\$33 per share, but not to the

Ads by Google

[Drupal Tutorial Video](#)  
Learn how to use Drupal to build your own professional website.  
[Drupal Tutorials.org](#)

[Drupal Hosting](#)  
Web Hosting for Drupal sites, Free Drupal installation, tutorial  
[SiteGround.com](#)

[Free Tutorial](#)  
A Dedicated Website To Free Tutorial  
[Tutorial.Seetful.net](#)

[PHP MySQL site in minutes](#)  
Create PHP code for any MySQL database. No programming.  
[www.XLineSoft.com/phprunner](#)

Figure 1.1: Google AdSense

Figure 2.1 displays a typical UI of Google AdSense on a blog site. Circled on the right side, the ads displayed in the AdSense block are related to the Web content that the user is browsing, making these ads more effective. But this figure also shows some drawbacks of Google AdSense: the ads displayed at that moment made no sense to the article. It is because the article was newly posted and AdSense was not able to scan and analyze it in real time, causing a response delay.

This project aims to design a system that provides a content-targeted real-time advertising system. This system is not only usable for normal web pages but also works properly for a variety of systems such as the popular Drupal-driven content management system. This system is able to retrieve and analyze Web content such as blog entries, and then extract the best keywords representing the article subject. When a user is browsing a blog article, system will display certain ads based on the keywords of that article.

Amazon Web Service (AWS) is the ads provider for this project.

This project consists of three development phases. The first phase builds a Drupal-driven blog system with appropriate function modules on load. The second phase uses fuzzy Web mining to analyze content from the blog system to obtain the best keyword(s) that represent the content. The third phase loads a specific interface for sending keywords to AWS to retrieve corresponding advertisements.

## **2. Drupal CMS Design**

Drupal is a popular open source modular content management system (CMS). Like many other modern CMSs, it allows the system users and the administrator to post, customize, and manage the content and display of the Web site in an efficient manner. Developers world-wide have benefited from its open source strategy and created thousands of official or unofficial Drupal modules that tremendously increase the functionality of this CMS. Another advantage of using the Drupal system in this project is that Drupal is a PHP-based system. As an efficient server-side scripting language, PHP is widely used and was also chosen to be the scripting language to implement content-targeted analysis.

For best compatibility and generality, Apache 2.2.8 and MySQL 5.0 were chosen to be the Web server and database. The entire system is a typical 3-tier Web application:

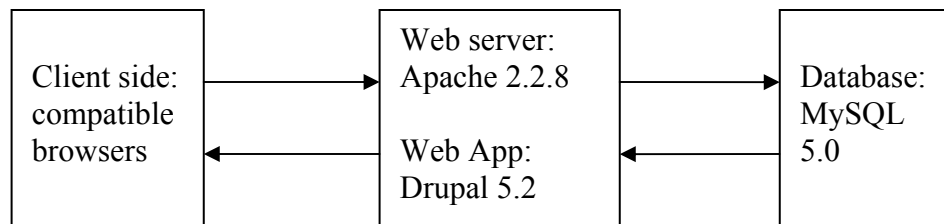


Figure 2.1: Drupal application web architecture

Based on the basic Drupal 5.2 system, several Drupal-compatible modules have been loaded to increase the functionalities of the system, such as Google AdSense module mentioned previously to provide the ability to compare results.

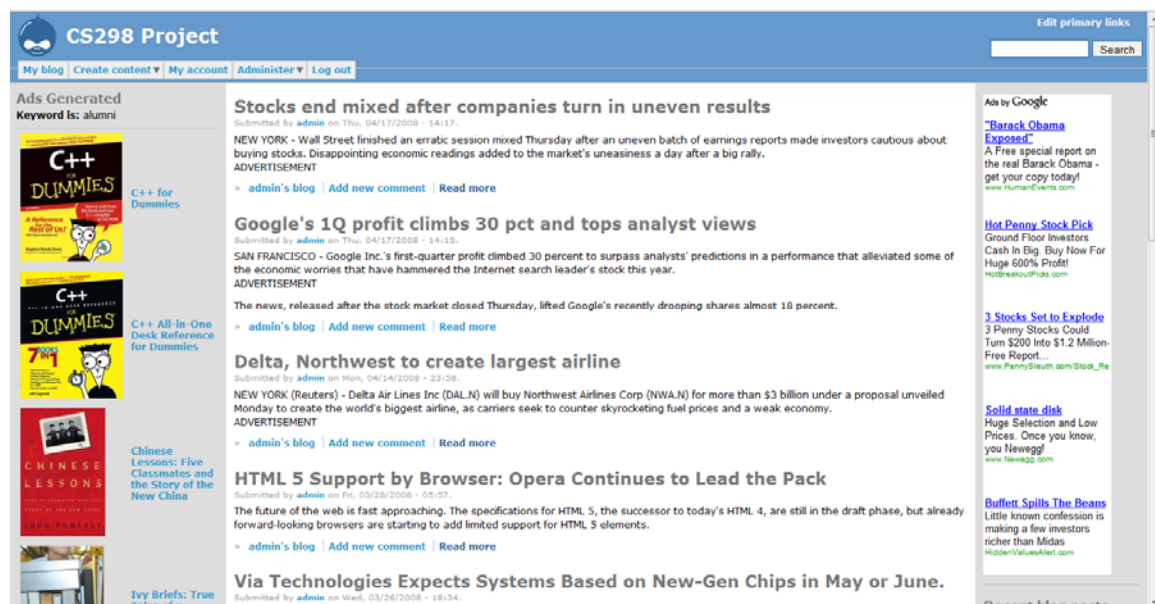


Figure 2.2: Front page of a Drupal-driven CMS

### 3. Motivation and Fuzzy Logic

The uncertainty of meaning with natural language is a major focus for content analysis. The content-targeted advertising business model raises a big challenge on that issue. Traditional ways such as using classic TF-IDF (Term Frequency – Inverse Document Frequency) model combined with many other rules are mostly based on pure numerical analysis. However, linguistic meanings are usually vague. In many cases, it is fairly difficult to use purely crisp words, numbers, or functions to represent vague semantic meaning. Instead, fuzzy logic provides the benefits of the Boolean operation (using operators such as “AND”, “OR” or “NOT”) while overcoming its drawbacks. Documents, queries and their characteristics could easily be viewed as fuzzy granular classes of objects with un-sharp boundaries and fuzzy memberships in many concept areas. [7]

Fuzzy set theory was invented by Dr. Lotfi A. Zadeh at UC Berkeley in 1965. Different from a normal set, whose elements have clear (crisp) values, a fuzzy set has elements with “bivalent” values, meaning that whether each element belongs to this set or not is represented by a membership value (degree of truth). A high membership value means that that element highly belongs to that set, and vice versa. In other words, fuzzy sets are sets of objects with un-sharp boundaries in which membership is a matter of degree. [6] The left graph of Figure 3.1 is a normal set representing season change. Each season has a crisp value (0 or 1) representing its existence and there is a clear time point to switch from the previous season to the next one. After that point, a new season suddenly comes into existence. In contrast, the fuzzy set represented by the right graph

shows a totally different story. Seasons here do not suddenly come or go but gradually show their influences. For example, spring gets its peak value around mid April but gradually fades after that. At the same time, summer heat is getting higher. Obviously, the representation of a fuzzy set shows a more realistic situation compared to crisp values in this case.

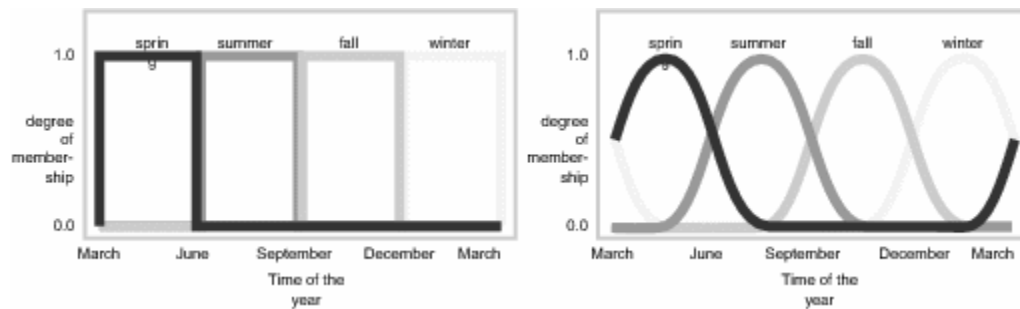


Figure 3.1: Crisp Set vs. Fuzzy Set

Membership values are generated by membership function (MF) which defines how each input value is mapped to a membership value between 0 and 1. For each element  $e$  that belongs to a fuzzy set FS, there is a membership function  $M$  such that  $0 \leq M(e) \leq 1$ . Therefore, a fuzzy set can be defined as  $FS = \{e, M(e) \mid e \in E\}$ , where  $E$  is the universe of discourse whose elements are denoted by  $e$ . For example, a membership function calculating GPA can be as simple as  $\text{gpa}(g) = \{0, \text{ if } g < 2; (g - 2), \text{ if } 2 \leq g < 3; 1, \text{ if } 3 \leq g\}$  which can be represented by the graph below.

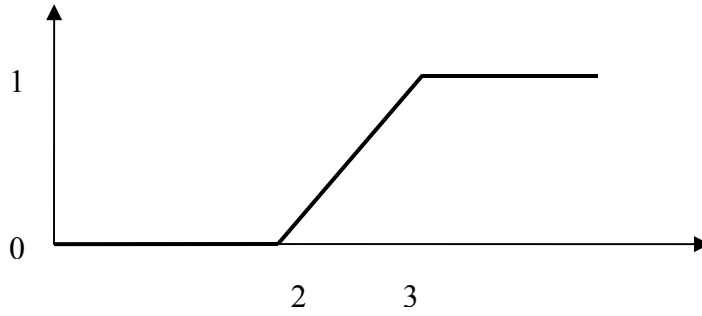


Figure 3.2: A membership function

There are several popular membership functions including triangular (trimf), trapezoidal (trapmf) and Gaussian. Because of their smoothness and concise notation, Gaussian is chosen to be the membership function in this project. Gaussian function is widely used to process input variables and represent the corresponding relevance values.

$$\text{Gaussian function: } f(x) = ae^{[-(x-\mu)^2]/\sigma^2}$$

Amplitude  $a$  is the height of the peak, mean  $\mu$  is the position of the center of the peak, and standard deviation  $\sigma$  controls the width of the peak. *gaussamp*( $x, \mu, \sigma, a$ ) will be used in the rest of this report to easier demonstrate a Gaussian function.

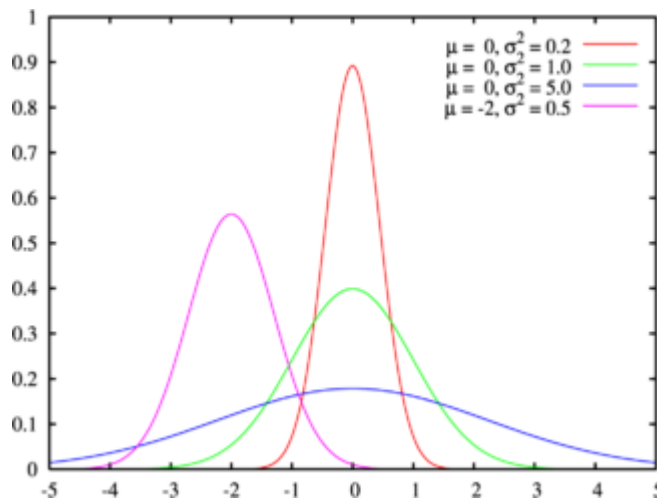


Figure 3.3: Gaussian curves

Multiple membership functions can be combined into a more complex membership function using fuzzy set operators such as union, intersection, complement and/or many others derived from these basic operators. For example, using union operator, two side-by-side intersecting single Gaussian “peaks” can be combined to a bimodal McDonald “M” looking curve.

Fuzzy logic is derived from fuzzy set theory to handle those “approximate” factors rather than crisp ones. As a logical system, it provides a more semantically sensible Boolean method (such as “mostly true but a little false”) to represent membership values. It is also easier to understand compared to complex numerical computations. It allows for setting membership values to a range, for instance, between 0 and 1.

Generally, fuzzy logic rules use if/then logic. Depending on different member (input) values, linguistically, fuzzy logic rules can use words from “poor” to “fair” to “good”. For example, some simple fuzzy logic rules can be

If service is poor or food is rancid, then tip is cheap

If service is good, then tip is average

If service is excellent or food is delicious, then tip is generous

More rules can be defined as necessary, such as using words like “very good” or “very poor”.

The entire process architecture of mapping the member inputs to outputs using fuzzy logic is the fuzzy inference system (FIS). The fuzzy inference process comprises of five parts: 1.) fuzzification of the input variables, 2.) application of the fuzzy operator in the

antecedent, 3.) implication from the antecedent to the consequent, 4.) aggregation of the consequents across the rules, and 5.) defuzzification. [MathWorks] It takes member inputs from all fuzzy sets matters into consideration. Compared to a serial numerical process, fuzzy logic can handle all inputs in parallel and then better consider them together. This is exactly what this project focus on. In chapter 4.3.3, this FIS process will be applied and discussed in detail.

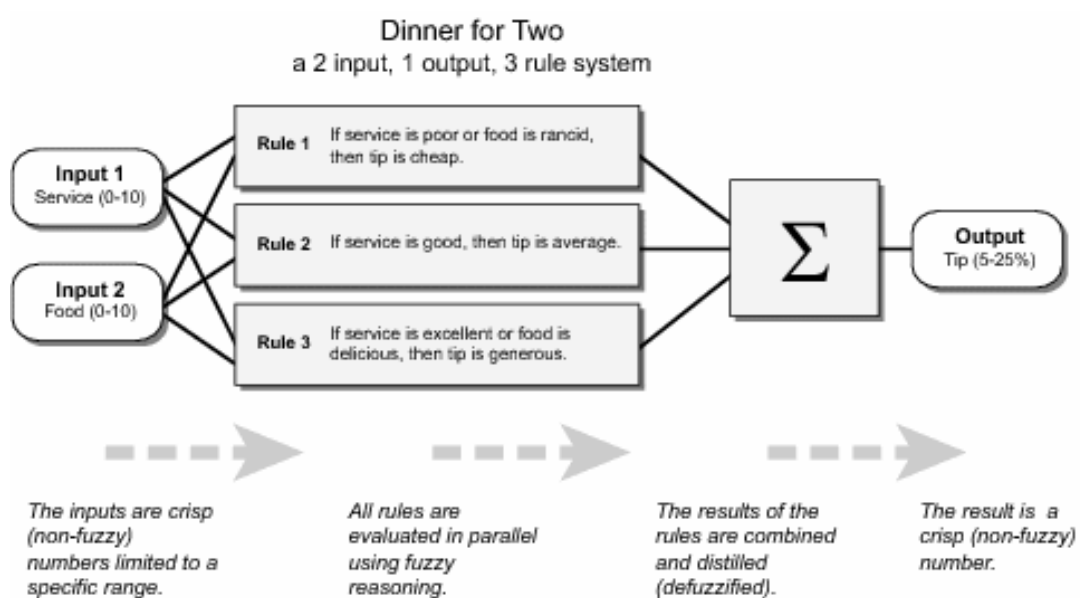


Figure 3.4: FIS process

## 4. Content-targeted Analysis

Different from a keyword-targeted advertising system, which is based on specific keywords directly provided (searched) by users, a content-targeted system relies on matching ads and its associated keywords to the text of a Web page [1]. The goal of a content-targeted system is to extract the best keywords that represent the main subject of a blog article, news page, or email page, for instance. These keywords will then be



transferred to an advertising system to match appropriate ads and display them at a certain place of the Web page. As viewed by the content readers, the ads are related to the Web content they read (and interested in usually). In a typical commercial model, any click-through of these ads will bring revenue to the web content providers. By providing the most relevant ads on their Web page, Web content providers can squeeze more revenue from their Web content.

The entire content analysis and ads matching process consists of three major phases.

1. Content gathering and filtering
2. Ads candidates (keywords) choosing
3. Implementation of fuzzy ranking
4. Ads Retrieval

## ■ Architecture Model

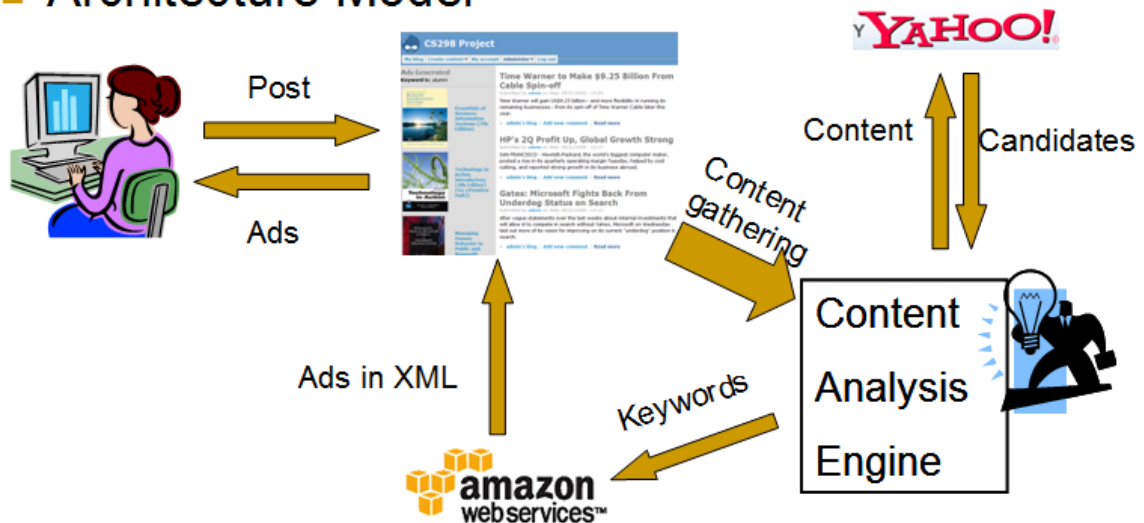


Figure 4.1 System Architecture Model

## 4.1 *Content gathering and filtering*

Content gathering and filtering is the first step of the content-targeted analysis process. The main purpose of this operation is to collect all useful information from target Web page for further processing. The script language PHP is a good solution for this purpose. A PHP application can take input from a file or data stream containing targeted content, implement data processes, and finally output a data stream.

The plain text data directly grabbed from a Web page usually contains everything in the page including effective content and non-content information such as HTML tags and Javascript code. Some proper filtering methods should be implemented to filter out non-content code but keep the rest intact. Based on different targets, many different methods have been introduced to help filter Web-based content; some are famous such as Bayesian email filtering. For Web page filtering purpose in this project, a reasonable way is to use regular expression which can be implemented by PHP to identify different parts of target data. The simple script below as an example replaces all tags represented by variable `$tag` with a blank (" ").

```
$tags = "<.[?p.?>|<.[?br.?/?>|\\(|\\)|,|:|\\.|;|'|"|{|}|&|-|\\*|".  
"  
"[:space:]](to|in|of|on|up|is|by|or|we|me|us|it|at|as|an|and|but|not|the|for|i  
ts|was|has|have|were|been|when|than|this|that|they|what|their|which|where) [:sp  
ace:]]";  
  
if (eregi($tags, $text)) {  
    $text = strtolower(trim(eregi_replace($tags, ' ', $text)));  
}
```

In addition to the regular expression method, more system-specified filtering schemes can be used to optimize the filtering effect. With this in mind, we can be

relatively more effective and intuitive to filter out useless code by understanding the structure of the targeted system. For instance, Drupal CMS has its specific HTML division (<div> in HTML) distribution style which is helpful for locating the main content.

Here below is the major process to get the page content and start filtering.

```
/*1.) grabbing page content from blog page*/
$oriContent = file_get_contents("http://127.0.0.1/drupal/?q=node/$nid");
... ..

/*2.) getting target content division*/
$arrContent = split('<div class="content">', $arrContent[1]);
$arrContent = split('</div>', $arrContent[1]);
$text = $arrContent[0];
... ..

/*3.) split content into word level and save into an array*/
$arrContent = split(' ', $oriContent);
... ..

/*4.) create regular expression template and do further filtering*/
... ..
```

## **4.2 Candidate Select**

### **4.2.1 Select mode**

Many previous published approaches estimated the ad relevance based on co-occurrence of the same words or phrases within the page and within the ads (or keywords

binding to the ads). [4] Comparing to strategies that only count on individual words, considering multiple words (as a phrase) has significant advantage on accuracy. For example, a page talking about new operation system “Windows Vista” may have high term frequency (TF) on the word “vista”. If the analysis is based on individual word, the ads triggered by this “vista” may include travel or vista point information. In contrast, if the word “vista” is considered to be part of “windows vista”, there will be a much lower chance to make such a mistake when matching ads.

For the sake of bringing in the most relevant advertisements, there is no need to find out or consider every possible phrase in an article. Instead, the focus can be put on those with high TF values. Start with each high-TF individual words, combine it with the word before it and then the word after it distinctively to see if there is a phrase match (including partly match) compared to a certain “phrase pool”. If the result is positive, the system can do a further try that extend one more word toward the direction of the previous match. A high bound can be set up such as five words at most. With longer phrase matches, higher accuracy is expected in most cases. As a result, the strategy is to treat phrases instead of individual words as raw advertisement candidates and perform further analysis on the local server to decide the best candidates.

#### **4.2.2 Yahoo Term Extraction implementation**

Yahoo Term Extraction (YTE) is a Web application that can perform similarly to what was described in the end of the previous section. YTE accepts an entire text and does a real time analysis and then returns mostly meaningful phrases, although a small number will not make sense or even be correct words. Based on the inverse checking,

most phrases returned by YTE contain at least one word having a top-ten TF value. Like many other popular Web services, YTE accepts REST (Representational State Transfer) technology as a way to submit the original document content and then returned the result.

There are five parameters in a YTE REST query. When constructing an YTE query, we need to provide at least two parameters: the appID and the context (document). The YTE web service will then send the response (a list of extracted words) in one of three formats: XML, json or Serialized PHP. In this project, we use PHP “curl” to post the request and receive the result in the Serialized PHP format. (“curl” is a library that allows you to connect and communicate to many different types of servers with many different types of protocols. It currently supports the http, https, ftp, gopher, telnet, dict, file, and ldap protocols.)

```
/*get document content from content array*/
$fcontext = implode(" ", $arrWord);

/*Yahoo term extraction web service*/
$url = "http://search.yahooapis.com/ContentAnalysisService/V1/termExtraction";
$appID = 'cCPf5tfV34EABQtj85ZJeHFT.5UXibp0Vs1CIo33DQ_UsDQayaUVUzVU5SFTePkm_nc-';

/*curl session*/
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,
"appid=$appID&output=php&context=$fcontext");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); //RETURN THE CONTENTS OF THE CALL
$response = curl_exec($ch);
```

```
curl_close ($ch);
```

According to the YTE specification, adding a “query” which consists of the keywords representing the “main idea” of a document will be helpful. Unlike many other Web pages each of which has a META tag containing several keywords representing the main topic, a Drupal blog entry has no META tags in the HTML script. The only place outside the main text that may contain the main idea of the document is the article title, which will be used as the YTE query parameter. Once getting the result returned by Yahoo, the system converts it to an internal array. Here below displays a sample array converted from Serialized PHP format.

```
Array ( [0] => ford escape hybrid [1] => hillary rodham clinton [2] => barack  
obama [3] => chrysler 300c [4] => ford executive [5] => bill ford [6] =>  
presidential hopefuls [7] => apparently [8] => presidential candidate [9] =>  
chairman bill [10] => executive chairman [11] => automakers [12] => early june  
[13] => hemi [14] => hypocritical [15] => john edwards )
```

Although YTE returns keywords extracted from an article, it neither provides information of how it generates the keyword list nor the order of the keywords importance (rank). Without a preview of the document, it is still quite difficult to accurately summarize the main idea of this document only using the keywords listed above, let alone matching the most appropriate ads. However, considering the semantic meanings a YTE result brings in, we can consider YTE phrases as raw candidates for ads retrieval.

### **4.2.3 Post-processing**

As mentioned previously, some YTE phrases are not returned in the correct form. For instance, duplicated meanings is a relatively major problem, such as returning “google” and “google inc” at same time or “affordable health care” and “quality affordable health care” at same time. Some other problems include incomplete phrases such as “s market”, or the wrong word such as “inroad”, etc. The function of post-processing is to eliminate or correct these kinds of problems and errors, and to release the candidates in a more correct and clearer formation for further processing to discover which candidates have the highest ranking.

## **4.3 *Attributes of Candidate***

In order to rank these generated ads candidates (YTE phrases), the system performs a fuzzy ranking analysis involving several “characteristic” factors of these candidates. Major factors considered include term frequency, title overlap and optional META information, which are also the major parameters of fuzzy rules.

### **4.3.1 Term Frequency**

TF-IDF is a weight often used in information retrieval and text mining fields. It is a statistical measure used to evaluate how important a word is to a document in a document collection. The TF value of a word simply denotes the number of times that word appears in a given document. Usually, a high TF value shows a more important position the word holds in a document (since the word shows more frequently). The IDF value is to measure a general importance of a term in an entire document corpus by checking its

“rarity” compared to other words in the entire documents corpus. If a word is rare, the word will have high IDF value and vice versa. Therefore, if a keyword appears frequently in a given document (means higher TF value) but appears infrequently in the overall document collection (means higher IDF value), this keyword owns high overall TF-IDF weight.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad tfidf_{i,j} = tf_{i,j} \cdot idf_i$$

Figure 4.1: TF-IDF

Figure 4.1 display the TF-IDF function where  $n_{i,j}$  is the number of occurrences of the considered term in document  $d_j$ , and the denominator is the number of occurrences of all terms in document  $d_j$ .  $|D|$  is the total number of documents in the corpus. And  $|\{d_j : t_i \in d_j\}|$  is the number of documents where the term  $t_i$  appears (that is  $n_{i,j} \neq 0$ ).

Up to the end of CS298, the IDF value is not being used. The major reason is that it is very difficult to get an appropriate (large and typical enough) document corpus. The ideal corpus for IDF should be all the documents on the WWW. For major search engines like Yahoo or Google, since huge amount of Web pages have been crawled and cached, they may have a document corpus relatively close (but still not equal) to the ideal definition. This can hardly be done in this project. If the document collection is too small, it will not be representative to show how unique a certain keyword is, making its IDF value inaccurate and harmful rather than helpful.



An alternative design for using IDF can be implemented in recent future is to define all posted documents on an entire blog system as the document corpus. A more unique meaning a word has in the blog system, a higher IDF weight it has within the restricted domain. As more and more blog entries are posted and the document corpus gets larger and larger, the accuracy of the IDF value of a particular word keeps improving.

In contrast, TF can be attained easily by using a PHP script to collect all words within a given document and then calculate the frequency of each distinctive one by using a loop operation.

Term	TF	Percentage
clinton	17	3.26%
she	16	3.07%
said	16	3.07%
obama	16	3.07%
her	10	1.92%
health	8	1.54%
debate	7	1.34%

... ..

Table 4.1: Term frequency sample

All the distinctive words acquired will be stored into a backend database with their TF values for future process and usage.

### 4.3.2 Title overlap

The definition of title overlap is how many words in a candidate phrase appears in the article title. If more words of a candidate phrase appear in the article title, the title overlap value of the candidate phrase is higher and vice versa.

Obama and Clinton clash in testy debate

YTE Phrase	InTitle Overlap
hillary clinton	0.50
barack obama	0.50
health insurance	0.00
affordable health care	0.00

Table 4.2: Title Overlap

For the article “Obama and Clinton clash in testy debate”, the candidate phrases “hillary clinton” and “barack obama” both have an overlap value of 0.5, meaning that half of the phrase appears in the article title. In contrast, another phrase such as “health insurance” scores zero (0.00) because neither “health” nor “insurance” appears in the title.

### 4.3.3 META information and location

Besides two major attributes discussed above, there are some other optional attributes that can be considered if the targeted objects are located on a platform other than Drupal being used in this project.

Keyword META is a common HTML tag which holds the information about a Web page in some non-Drupal system. It may contain the keywords representing the main

subject of a Web document (as defined by a page builder). For instance, the keyword META of a database provider looks like

```
<META name="keywords" content="enterprise, applications, software, database,
middleware, fusions, business, Oracle">
```

Under such a case, a new attribute of candidate “meta overlap” can be built to measure the overlap situation of candidate to the MEAT keywords. It is pretty similar to the definition of title overlap described in last section.

Similarly, a “heading overlap” can also be defined if there is an overlap situation happens between the candidates and the headings of an article. The headings are shorter than a sentence and usually each one occupies an entire line to indicate the meaning of next section. For example, the “NEW MECHANISMS” in the following news is a heading:

Russian Foreign Minister Sergei Lavrov did not specify what security guarantees might be offered. But he said a combination of negotiations and incentives could defuse the stalemate over Iranian enrichment and wider conflict in the Middle East.

#### NEW MECHANISMS

"I think the 'Six' could make the following step: directly put concrete offers on the negotiating table, give Iran security guarantees and ensure a more distinguished place in negotiations on the situation in the Middle East," he said.

There are still some challenges blocking using heading information to regular candidate attribute. Two major problems are the uncertainty of a heading content (a heading is sometimes is a slogan or a quoted people speech) and the uncertainty of the location of a heading (how to make sure some words shorter than a sentence is a heading

but not, for example, an explanation to a figure). This is a research and implementation direction in future to further improve the attribute design of candidates.

## **4.4 Fuzzy Inference System Process**

The function of a fuzzy ranking system can be represented as a fuzzy inference system introduced in Chapter 3. It takes all input fuzzy sets into consideration using certain fuzzy rules to determine the final relevance degrees. The fuzzy sets are those attributes of an article including TF, title overlap and others maybe considered mentioned in Section 4.3.3.

### **4.4.1 Fuzzy Rules**

The first step to build a fuzzy ranking system is to build fuzzy logic rules. Since fuzzy logic handles reasoning and results in a more approximate way rather than traditional crisp logic (“predicate logic”), fuzzy rules also looks more at the approximate side. Four rules handle possible situations in our content analysis process.

- If TF is high, the relevance is high. If combined TF weight of the individual words of an YTE phrase is high, then the relevance of this phrase is high.
- If TF is low, the relevance is low. If combined TF weight of the individual words of an YTE phrase is low, then the relevance of this phrase is low.
- If overlap is high, the relevance is high. If more words contained in a phrase appear in an article, then the relevance of this phrase is high.
- If overlap is low, the relevance is low. If fewer or no words contained in a phrase appear in an article, then the relevance of this phrase is low.

The first two fuzzy rules are implemented on every distinctive word within an article, storing every word with its TF weight in as described in the previous process. According to the scale of each TF weight, all TF values are scaled into a (0, 10] range with the highest TF defined as 10. For example, if the highest TF value 16 is scaled to 10, then a TF value 8 will be 5. These scaled values are the fuzzy set member inputs and ready to be operated by membership function. The TF membership functions are defined using Gaussian function and each of which represents a corresponding fuzzy rule.

- High set: `gaussamp(x, 10, 2, 1)`; factor: 10

This function represents the fuzzy rule “if TF is high, the relevance is high”.

- Low set: `gaussamp(x, 0, 2.35, 1)`; factor: 1

This function represents the fuzzy rule “if TF is low, the relevance is low”.

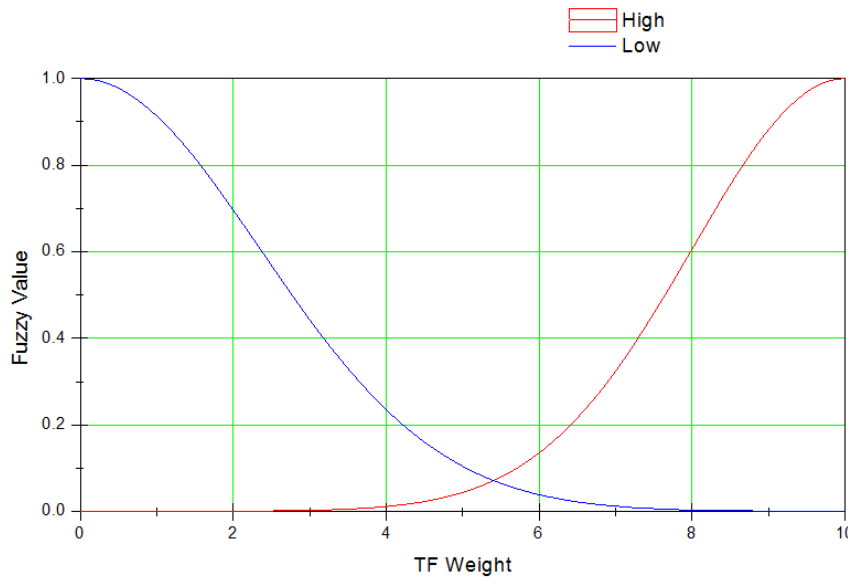


Figure 4.2: Gaussian curve of TF

For title overlap, the Gaussian function are defined as

- High set: `gaussamp(x, 10, 0.24, 1)`; factor: 1

This function represents the fuzzy rule “if overlap is high, the relevance is high”.

- Low set: `gaussamp(x, 0, 0.27, 1)`; factor: 0.1

This function represents the fuzzy rule “if overlap is low, the relevance is low”.

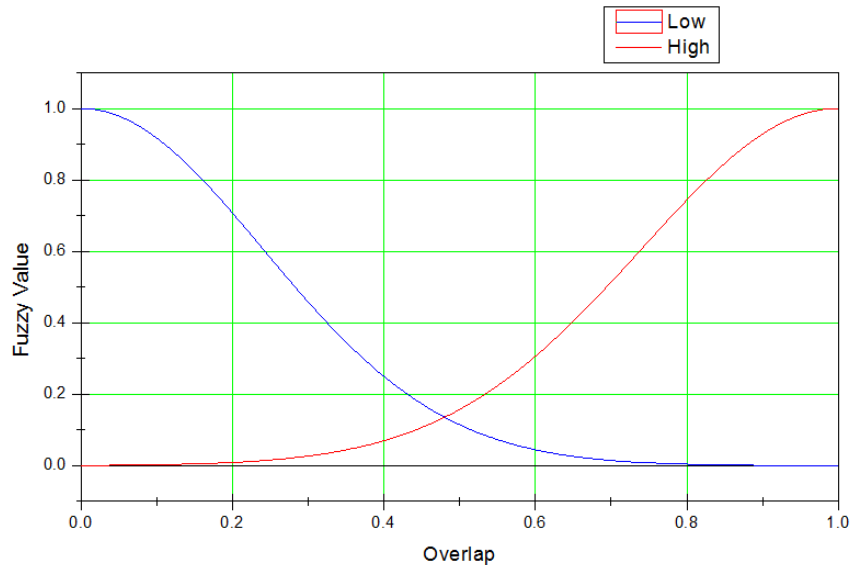


Figure 4.3: Gaussian curve of overlap

The factor parameter controls the weight (or say, “importance”) of different rule and involved fuzzy set. For instance, “High TF” is 10 times important than “High Overlap” because the former’s factor (10) is 10 times larger than the latter’s (1). This is discussed in next two sections. Besides, if there are other overlaps involved such as heading overlap, it should also be defined at this stage.

#### 4.4.2 Implication and Aggregation

While some researches tend to use graph tools (such as Fuzzy Logic Toolbox™ from MathWorks™) to handle the fuzzy operator, implication method, and aggregation

process, this project uses a more straightforward numerical way to combine multiple fuzzy rule outputs and get the final results.

The output of any fuzzy rule is the product of the output of the membership function and the factor. For example, inputting a scaled TF value 5.882 into the “High TF” membership function will generate a value 0.12 as the result of fuzzification. At the same time, the “Low TF” membership function will generate a value 0.044 as the result of fuzzification. Since “High TF” has a set factor 10, the fuzzified output 0.12 will be enlarged for 10 times to 1.2; since the “Low TF” has a set factor 1, the output 0.044 keeps the value. Similar operation is also applied to the title overlap rules.

Once having the high and low TF membership value, a fuzzy TF value is able to be generated by combining the high TF set with the low TF with factor involved:

$$\text{Fuzzy TF value} = \frac{(\text{Low TF} \times \text{Low TF factor}) + (\text{High TF} \times \text{High TF factor})}{\text{Low TF} + \text{High TF}}$$

Using this function, both high set and low set rules are considered and they influence the fuzzy output value according to their factors. For the example above, the fuzzy TF value will be

$$(0.044 \times 1 + 0.12 \times 10) / (0.044 + 0.12) = 7.585$$

A glimpse on PHP code (briefed):

```
/* 1.) Normalize all TF into a (0, 10] range and then
2.) calculate TF HIGH and TF LOW membership function and then
3.) calculate the fuzzy value of TF*/

for ($i=0; $i<count($arrFreq); $i++) {
```

```

/* some code here*/

/* normalize TF values into (0, 10] range*/
$arrFreq[$i]['normalized'] = ($arrFreq[$i]['num']/$max) * 10;

/* for each word, get its TF LOW and TF HIGH membership (Gaussian) function
output*/
$arrFreq[$i]['low'] = exp(-0.5*(pow($arrFreq[$i]['normalized']/2.35, 2)));
$arrFreq[$i]['high'] = exp(-0.5*(pow(($arrFreq[$i]['normalized']-10)/2,
2)));

/* calculate the fuzzy value of TF*/
$arrFreq[$i]['fuzzified'] = ($arrFreq[$i]['low'] * 1 + $arrFreq[$i]['high']
* 10) / ($arrFreq[$i]['low'] + $arrFreq[$i]['high']);

/* rest code here*/
}

```

The title overlap fuzzy value can be obtained in a similar way and it is not necessary to do normalization (because the input value is already within a range [0, 1]). The modification can be done on the Gaussian function side to make it fit this overlap input range, making the title overlap membership functions have much smaller “width” setting. The code below only displays the different part of title overlap from the TF side.

```

/* Calculate Overlap fuzzy value*/
$arrKeyRanked[$i]['low'] = exp(-0.5*(pow($arrKeyRanked[$i]['inTitle']/0.24,
2)));

$arrKeyRanked[$i]['high'] = exp(-0.5*(pow(($arrKeyRanked[$i]['inTitle']-
1)/0.26, 2)));

$arrKeyRanked[$i]['inTitlefuzz'] = ($arrKeyRanked[$i]['low'] * 0.1 +
$arrKeyRanked[$i]['high'] * 1) / ($arrKeyRanked[$i]['low'] +
$arrKeyRanked[$i]['high']);

```



Here the factor for title overlap is designed to be 1/10 weight of TF. This value is designed mostly empirically. The first reason is that the title overlap attribute is not as important as TF attribute since the latter is a more “panorama” view to the entire article. Another reason is that the overlap value will be used as a factor in a future multiplication so that it should not be set too high. With the 1-high and 0.1-low setting, the highest value will be 1 (means 100% increase when used as a multiplication factor).

As an example, if half of a phrase appears in the article title, the “inTitle” input value is 0.5. Based on the Gaussian functions above, the Low overlap fuzzy value will be 0.044 and the High overlap fuzzy value will be 0.458. And the overall overlap fuzzy value is

$$(0.114 \times 0.1 + 0.18 \times 1) / (0.044 + 0.458) * 100\% = 65.074\%$$

Because decisions are based on the outputs of all fuzzy rules implemented in a FIS, these rules (and their outputs) must be aggregated in some manner in order to make a decision. The previous processes generated two fuzzy values, the TF fuzzy value and the title overlap value. The former denotes the membership degree of each individual word contained in a candidate phrase and the latter denotes the “degree of relationship” between a candidate phrase and the article title.

Since the title overlap is a more “phrase-level” attribute and the phrase itself is built by individual words, the aggregation method is designed as a 2-phase process. First to aggregate individual TF value together to get a base value for each phrase, and then adjust the baseline phrase value by aggregating the overlap value.

The first TF aggregation is designed as

$$\text{Phrase base value} = \sum \text{square}(\text{TF of each word}) / \sum (\text{TF of each word})$$

Using this function, every word within a phrase will be considered. Obviously, those words having high TF values will have stronger influence than those with low TF values because they are all squared. This could assure that a phrase containing high-TF word will be considered more important. On the other hand low-TF words would balance the high-TF. It is especially effective when only one word within a phrase has a high TF value but all others' are low. Suppose three words "health", "insurance" and "care" have TF membership value 9, 7, 2 respectively. They are contained in two phrases "health insurance" and "health care". Using the TF aggregation method, the base value of phrase "health insurance" will be  $(9^2 + 7^2) / (9 + 7) = 8.125$  but the base value of "health care" will be  $(9^2 + 2^2) / (9 + 2) = 7.72$ . Semantically saying, the "health case" is less important than "health insurance" at this process moment.

The next stage is to adjust the baseline value by aggregating with the title overlap value. The aggressive design idea here is to dramatically increase the weight of phrases having high number of word members which appears in the article title. Multiplication is the way to achieve this goal. The final phrase weight equals the phrase baseline weight multiplied by the title overlap value. For instance, assume the phrase "health insurance" has a TF weight 8.125 as described above and the word "insurance" appears in the article title (which results in a 65.074% title overlap value). Then the final crisp weight of "health insurance" is  $8.125 * 1.65074 = 13.412$  (rounded)

Every candidate phrase will get an aggregated final weight after the entire process. It may be represented in a table below:

Candidate	TF fuzzified	*	Title Overlap	Low(0.1)	High(1)	Overlap Fuzzified	*	Final Weight
yahoo shareholders	8.776		0.50	0.114	0.180	65.074%		14.48728
microsoft board	8.714		1.00	0.000	1.000	99.985%		17.42667
board members	1.575		0.50	0.114	0.180	65.074%		2.59923
board meetings	1.574		0.50	0.114	0.180	65.074%		2.59909
wall street journal	1.091		0.00	0.000	0.000	0.000%		1.09072
major stumbling block	1.005		0.00	0.000	0.000	0.000%		1.00518

Table 4.3: Candidates weight table

The “TF fuzzified” column denotes the aggregated baseline value of each candidate and the “Overlap fuzzified” column denotes the title overlap value. These two values are multiplied to get the final weight as a crisp value.

## 4.5 Ads Retrieval

The current scheme is to use the top two ads candidates as the final ads keywords. Their weight proportion decides the final ads distribution.

In the Drupal system, a new block called “Amazon Ads” is created to retrieve ads using keywords from Amazon Web Service (AWS). Similar to Yahoo Term Extraction, Amazon also accepts REST to receive the ads query and return the ads in XML format. Sample code to query Amazon ads follows. Request keywords are saved in array \$keyword.

```
$request=
    "http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService"
    . "&AWSAccessKeyId=" . Access_Key_ID&Operation=" . $Operation
    . "&Version=" . $Version. "&SearchIndex=" . "Books"
    . "&Keywords=" . $keywords[0] . "&ResponseGroup=" . $ResponseGroup;
```

For receiving the ads, the system needs to convert XML format to an object and display the advertisement.

```
echo "<table cellpadding=5>";

for ($i=0; $i<$proWeight_1; $i++) {

    $item = $parsed_xml_1->Items->Item[$i];

    echo "<tr><td><a href=\"". $item->DetailPageURL.\"\" target=\"_blank\"><img
src=\"". $item->MediumImage->URL.\"></a></td>";

    echo "<td><a href=\"". $item->DetailPageURL.\"\" target=\"_blank\">". $item-
>ItemAttributes->Title."</a></td></tr>";

    echo "</table>";
```

Figure 4.4: Ads generated

## 5. System Demo and Evaluation

The keyword analysis process discussed in the previous chapters can be seamlessly integrated into drupal CMS. Currently, there are two ways to implement the analysis process. One is a step-by-step manual operation to better understand the keyword

analysis and ads generation process, and the other is a one-click operation for faster and more convenient experiment.

## 5.1 Step-by-Step Ads Generation

Here below is the front page of the test system, which consists of three columns: ads analysis block at left, content block at middle and Google AdSense at right (as comparison). The default ads (for the user who has not selected a blog post and is on the main page or configuration pages) are related to the keywords “alumni”, which is the default keyword of this system. Once the user clicks a blog topic, the system will show the content-targeted ads based on the blog post user choose.

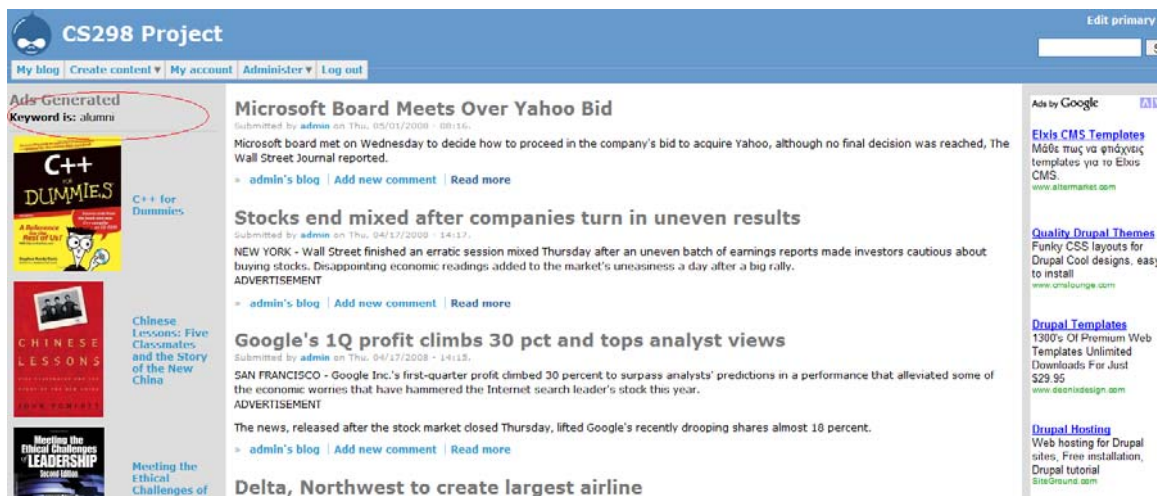


Figure 5.1: System main page

An ads analysis process starts from adding a new blog post. In the following case, a new post about AOL shutting down Netscape is posted. By modifying the Drupal core function (“node.module”), the keyword and ads analysis process can be integrated into the blog post submission process, which is a native function/operation of Drupal system.

A system under this configuration looks like figure below: the ads generator links disappears. The system directly displays the generated ads.

After inputting new blog post content, user simply clicks “Submit” button:

**Ads Generated**  
Keyword is: alumni

**Ivy Briefs: True Tales of a Neurotic Law Student**

**Chinese Lessons: Five Classmates and the Story of the New China**

**The Line of Beauty**

**Notre Dame Inspirations: The University's Most**

Home » Create content

**Submit Blog entry**

**Title: \***  
Mozilla Interfaces to Get 'Humanized.' Developer Says

**Body: \***  
Mozilla should soon be experimenting with some novel user-interface technologies for its browser and other products, according to a UI developer that joined Mozilla this week from startup company Humanized.  
ADVERTISEMENT  
Aza Raskin, who until Wednesday was president of Humanized, a five-person startup in Chicago, is now user experience lead for the Mozilla Labs team, he said Thursday. He will be working on technology to let people "do anything you want to do, anywhere, anytime on your computer," he said.  
Mozilla announced Wednesday that it had hired three of the principals from Humanized, which is known for its innovative work with the Enso project to create more intuitive user interfaces. Mozilla did not identify the people it hired, and Raskin was hesitant at first to say which of his colleagues had joined him at Mozilla. Eventually he acknowledged that Humanized's Jono DiCarlo and Atul Varma also have joined the company known for its Firefox browser.  
Enso user-interface software was designed to make it easier to perform daily tasks that require the use of multiple applications or functions. The software runs in the background and allows users to type in simple commands to access applications, instead of leaving the window or application they are in to use another one.

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <em> <strong> <code> <ul> <ol> <li> <dl> <dt> <dd>

- Lines and paragraphs break automatically.

[More information about formatting options](#)

[File attachments](#)

Figure 5.2 New content submission

The system will automatically go through the entire keyword analysis and ads retrieving process, and then display the corresponding ads based on the top keywords. In this case, the top two are “mozilla” and “raskin”.

**Ads Generated:**

**Top four Keywords are:**

- mozilla
- raskin
- user interfaces
- firefox-browser

**Dynamic HTML:  
The Definitive  
Reference  
(Dynamic Html)**

**Firefox For  
Dummies (For  
Dummies  
(Computer/Tech))**

**Programming  
Firefox: Building  
Rich Internet  
Applications with  
XUL  
(Programming)**

## Mozilla Interfaces to Get 'Humanized,' Developer Says

**View Edit**

Submitted by [tester](#) on Thu, 01/17/2008 - 21:10.

Mozilla should soon be experimenting with some novel user-interface technologies for its browser and other products, according to a UI developer that joined Mozilla this week from startup company Humanized.

ADVERTISEMENT

Aza Raskin, who until Wednesday was president of Humanized, a five-person startup in Chicago, is now user experience lead for the Mozilla Labs team, he said Thursday. He will be working on technology to let people "do anything you want to do, anywhere, anytime on your computer," he said.

Raskin announced Wednesday that it had hired three of the principals from Humanized, which is known for its innovative work with the Enso project to create more intuitive user interfaces. Mozilla did not identify the people it hired, and Raskin was hesitant at first to say which of his colleagues had joined him at Mozilla. Eventually he acknowledged that Humanized's Jono DiCarlo and Atul Varma also have joined the company known for its Firefox browser.

Enso user-interface software was designed to make it easier to perform daily tasks that require the use of multiple applications or functions. The software runs in the background and allows users to type in simple commands to access applications, instead of leaving the window or application they are in to use another one.

For example, if a user wants to open Firefox from the current screen, instead of having to find the Firefox icon or go to the Start menu in Windows, Firefox can be opened by pressing the Caps Lock key and typing "open firefox." Performing calculations and acquiring word definitions can be executed in a similar way from whichever window the user is in at the time.

"Those ideas need to be explored at Mozilla," Raskin said, though "it's unclear yet what form that will take."

He and his Humanized colleagues were attracted to Mozilla because the company "has a lot of vision" to extend the Web beyond the browser, he said. The Firefox browser remains the company's primary product for now, however, although it also offers the Thunderbird open-source e-mail client.

Raskin said that many advances on the Web, in terms of online services and mashups, have been designed with the developer in mind, and because of that they run the risk of making the browser a mere delivery vehicle for streaming applications-- much like what the desktop has become. He aims to use technology and ideas from the Enso project to add more human interaction not just to the browser, but to anything people do on the Web itself.

"I want the power of mashups not in the hands of the developer but in the hands of end-users-- in the hands of your grandmothers and your teen-aed son," he said. "So you really can be writing an e-mail and sav. 'Now I want a map in there.' Things like that-- a place where you

Figure 5.3: Newly posted content with generated ads

## 5.2 Performance Evaluation

The directly comparable system is Google Adwords. It provides the similar ability that analyzes the result of a given Web page and returns a keyword pool containing the keywords representing the subject.

How would you like to generate keyword ideas?

☐ Descriptive words or phrases  
(e.g. green tea)

☒ Website content  
(e.g. www.example.com/product?id=74893)

Enter a webpage URL to find keywords related to the content on the page. [?](#)

☐ Include other pages on my site linked from this URL

[► Or, enter your own text in the box below. \(optional\)](#)  
[► Filter my results](#)

Choose columns to display: [?](#)  
Show/hide columns

☒ Group keywords by common terms

Showing keywords grouped by these terms:  
[windows vista](#) (73), [microsoft windows](#) (7), [windows xp](#) (5), [stock market](#) (5), [window vista](#) (3), [windows](#) (31), [vista](#) (22), [stock](#) (28), [Miscellaneous keywords](#) (6)

Match Type: [?](#)

Figure 5.4: Google Adwords

After comparing 100 articles so far across technology, business, politics, entertainment and sport categories, there is an average 87% probability that all the top

four keywords generated by this CS298 project system match up the result from Google Adwords. Also, though it is subjective, many keywords showing up as top four but not in Adword result are actually strong keywords representing the major content of article. Another evaluation manner which will be held in near future is to organize real person with appropriate knowledge backgrounds to justify the accuracy of the keywords and ads generated. Their feedback will be classified into different level such as “great” or “mediocre”. Combined with the comparison result with Google Adwords, a better evaluation result can be obtained.

## 6. Reference

1. **Learning to advertise**, Anísio Lacerda, Marco Cristo, Marcos André Gonçalves; etc; Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 2006; SESSION: Web IR, Pages: 549 – 556
2. **An efficient algorithm for fuzzy Web-mining**, Rui Wu; Wansheng Tang; Ruiqing Zhao; Information Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on 8-10 Nov. 2004; Pages: 576 – 581,
3. **Finding advertising keywords on web pages**, Wentau Yih; Joshua Goodman; Vitor R. Carvalho; Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, 2006; SESSION: Mining the web, Pages: 213 – 222
4. **A semantic approach to contextual advertising**, Andrei Broder; Marcus Fontoura; Vanja Josifovski; Lance Riedel; Annual ACM Conference on Research and



Development in Information Retrieval, Amsterdam, The Netherlands, 2007;

SESSION: Web IR II, Pages: 559 - 566

5. **Semantic Web Content Analysis: A Study in Proximity-Based Collaborative Clustering**, Loia V. ; Pedrycz W. ; Senatore S. ; IEEE Transactions on Fuzzy Systems : Accepted for future publication, Volume PP, Issue 99, 2007; Page(s):1
6. **Computational Semantics**, Patrick Blackburn, INRIA; Johan Bos, U. of Edinburg;
7. **The application of fuzzy logic to the construction of the ranking function of IR systems**, N.O. Rubens; Computer Modeling and New Technologies, 2006, Vol.10, No.1, 20-27