Spring 5-7-2011

Computing with Granular Words

Hailong Hou

COMPUTING WITH GRANULAR WORDS

by

HAILONG HOU

Under the Direction of Dr. Yanqing Zhang

ABSTRACT

Computational linguistics is a sub-field of artificial intelligence; it is an interdisciplinary field dealing with statistical and/or rule-based modeling of natural language from a computational perspective. Traditionally, fuzzy logic is used to deal with fuzziness among single linguistic terms in documents. However, linguistic terms may be related to other types of uncertainty. For instance, different users search 'cheap hotel' in a search engine, they may need distinct pieces of relevant hidden information such as shopping, transportation, weather, etc. Therefore, this research work focuses on studying granular words and developing new algorithms to process them to deal with uncertainty globally. To precisely describe the granular words, a new structure called Granular Information Hyper Tree (GIHT) is constructed. Furthermore, several technologies are developed to cooperate with computing with granular words in spam filtering and query recommendation. Based on simulation results, the GIHT-Bayesian algorithm can get more accurate spam filtering rate than conventional method Naive

Bayesian and SVM; computing with granular word also generates better recommendation results based on users' assessment when applied it to search engine.

INDEX WORDS: Computing with word, Granular word, Granular information hyper tree, Spam filtering, Recommendation system, Collaborative intelligence.

COMPUTING WITH GRANULAR WORDS

by

HAILGON HOU

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2011

COMPUTING WITH GRANULAR WORDS

by

HAILONG HOU

Committee Chair: Dr. Yanqing Zhang

Committee: Dr. Yingshu Li

Dr. Rajshekhar Sunderraman

Dr. Vijay Vaishnavi

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2011

To my parents, my wife, and my daughter.

ACKNOWLEDGEMENTS

First of all, I thank my advisor, Dr. Yanqing Zhang, for his guidance and support throughout my research and study at Georgia State University. His professional and broad insight in computational intelligence area inspired and guided me in this research. I really appreciate the research methodology, knowledge, and attitude of life learned from this great professor.

Secondly, I would like to thank the committee members, Dr. Rajshekhar Sunderraman, Dr. Yingshu Li, and Dr. Vijay Vaishnavi for their valuable suggestions and help to improve my thesis.

I would also like to thank all the people who helped me in my life and study.

Finally, I would like to thank my parents for their strong, persistent encouragement, understanding, and their continuing support as I have pursued my educational goals. Special gratitude goes to my wife for the love and continuous support to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Computing With Granular Word and Its Applications

Computing with words (CWW) plays a crucial role in natural language processing. In recent years, it has been applied to multiple areas such as E-mail filtering, semantic web, search engines, etc.

The content-based filters are useful to filter spam nowadays. For example, the Bayesian anti-spam algorithm evaluates words or phrases in each individual message to filter unsolicited bulk E-mail (UBE) and unsolicited commercial E-mail (UCE). However, those content-based algorithms cannot effectively filter uncertain messages (such as fuzzy E-mail) in a real dynamic environment. A fuzzy E-mail is useful for one person but not useful for others. For instance, an E-mail about refrigerator advertisements is considered as a fuzzy E-mail since it may be useful for most families, but useless for Eskimos. The current content-based anti-spam algorithms cannot identify such fuzzy E-mails.

On the other hand, CWW is a general methodology that can compute and reason by using linguistic terms. Fuzzy logic is a major technology dealing with fuzziness among linguistic terms. However, linguistic terms may be related to other types of uncertainty. For example, different users search 'cheap hotel' in a search engine, the users may need distinct pieces of relevant hidden information such as shopping, transportation, weather, etc. The reason is that multiple words and/or word clusters are related to ‘cheap hotel’.

Human thinking based on a natural language is too complex to be represented by a computer language. CWW created by Zadeh in [2] and [31] is a novel methodology for applications in machine intelligence and natural language processing. Fuzzy logic is a basic technology for CWW.

Every word in the nature language is not meaningful when it exists alone, one word must have relevant concepts to define itself globally. For example, when user searches a hotel with a search engine, he or she usually expects to obtain more relevant information such as environment, price and location weather. Furthermore, real time information such as current reservation status is also important information related to “hotel”. Obtaining information globally helps a person make right decisions efficiently. Therefore, such dynamic information is useful for special words like travel or airport. The keyword with the granular information is called ‘Granular Word’; it reveals the relations among words. Computing with Granular Word (CWGW) is proposed in this research to deal with linguistic data analysis.

The CWGW can process word clusters. In the method, a new structure GIHT is used to describe the relations among the words and obtain values of the weights. Then, the granular information can be retrieved from the GIHT by a keyword.

With the structure of GIHT, granular computing is used in spam filtering and the recommendation system. Several research algorithms like GIHT based on Bayesian model for spam filtering, computing with granular word and CWGW based collaborative filtering algorithm are proposed in this thesis.

1.2 Related Work

1.2.1 Computing With Word

Fuzzy logic, originally proposed by Zadeh in 1965 [33], was applied to CWW field in 1996 [34] and [60]. Fuzzy IF-THEN rules and the compositional rule of inference were used. Since 1996, a lot of CWW methods have been developed based on fuzzy logic. A symbolic generalization of fuzzy logic admits self reference [35]. It entails the randomization of declarative knowledge, which yields procedural knowledge. In [36], a computational theory of linguistic dynamic systems for computing with words was made by fusing procedures and concepts from several different areas including Kosko's geometric interpretation of fuzzy sets , HSU' cell-to-cell mappings in nonlinear analysis, equi-distribution lattices in number theory, and dynamic programming in optimal control theory. The proposed framework enables us to conduct a global dynamic analysis, system design and synthesis for dynamic linguistic systems that use words or linguistic terms in computation, based on concepts and methods used in conventional dynamic systems.

Some new systems and methods have been proposed for CWW recently. The CWW architecture for making subjective judgments was designed [37][38][39][40]. Then a new CWW engine called Perceptual Reasoning (PR) was proposed. It also uses fuzzy IF-THEN rules. However, unlike a traditional Mamdani or TSK model, in which fired rules are combined by using the union, or addition, a LWA (linguistic weighted averages) in PR is used to combine the fired rules. Ying proposed a different model of "computing with words" within the framework of computing theory [41]. Since classical models of computation aim at describing numerical calculation, their inputs are precise data rather than vague data. In other words, "computing with words" was considered as a computational procedure with vague inputs. Words are explained as fuzzy sets of input alphabets, i.e., possibility distributions over the input alphabet. In [42], Wang and Qiu developed the formal aspect of computing with words via the fuzzy Turing machine and the fuzzy grammar with strings of words. Very recently, Qiu and Wang [43] has developed a probabilistic system of computing with words based on the model of computing with words proposed by Ying [41]. In [42], Qiu and Wang studied probability distributions over the input alphabet rather than possibility distributions over the input alphabet.

Moreover, researchers also studied word clusters with uncertainty. Rough set theory introduced by Zdzislaw Pawlak in the early 1980s [44, 45] can handle vagueness and uncertainty. A new post processing strategy called word suggestion was proposed based on a multiple-word trigger-pair language model for Chinese character recognizers [46]. Rough set theory was used in the study to discover negatively correlated relationships between words in order to avoid wrong words in the process of word suggestion. In [47], rough set analysis was used as a methodology to identify the relative importance of variables for individuals who interact with various computers and other communication systems aboard such as Airborne Warning and Control Systems (AWACS). Specially, the rough set was used to help to retrieve information and rank links [49-51]. Furthermore, some other methods were developed to deal with word clusters. In [52], a genetic word clustering algorithm was used to classify words in the phrases of a linguistic corpus. The underlying goal of word classification is to build a good probabilistic model of the language defined by the phrases in the corpus. In [53], a linguistic knowledge acquisition model makes use of data types, infinite memory and an inferential mechanism for inducing new information from known data. In [54], an approach to text categorization combines distributional clustering of words and a Support Vector Machine (SVM) classifier. A new language model, the Multi-Class Composite N-gram, is applied to avoid a data sparseness problem for a spoken language in that it is difficult to collect training data [55][56]. A new approach was created for clustering words in a given vocabulary [57][58]. The method is based on a paradigm first formulated in the context of information retrieval, called latent semantic analysis. A text-mining algorithm was used to the text sources of bilingual corpora [59]. However, many existing algorithms mainly focus on classifying and clustering the words, but don't study how to compute clusters of relevant words.

1.2.2 Spam Filtering

E-mail spam, also known as "bulk E-mail" or "junk E-mail”, has existed since the beginning of the Internet. The basic idea is that nearly identical messages are sent to numerous recipients by E-mails [32]. Spam can be described as an unsolicited bulk E-mail (UBE) [31]. Unsolicited commercial E-mail (UCE) is the most common type of spam. UCE seeks to engage a potential consumer in order to exchange goods or services for money. Spam becomes a significant problem because there are about 90 billion messages per day. Symantec reported that phishing attempts increased by a 30% from Jan 2006 to the end of the year [29]. Statistics from the Distributed Checksum Clearinghouse (DCC) project showed that 57% of the E-mail messages checked by the DCC network until September in 2008 were likely to be bulk E-mails [13]. About 85.65% of the threats checked by MX Logic came from spammers in 2008 [11].

Researchers have invented methods to filter spam during the past dozens of years.

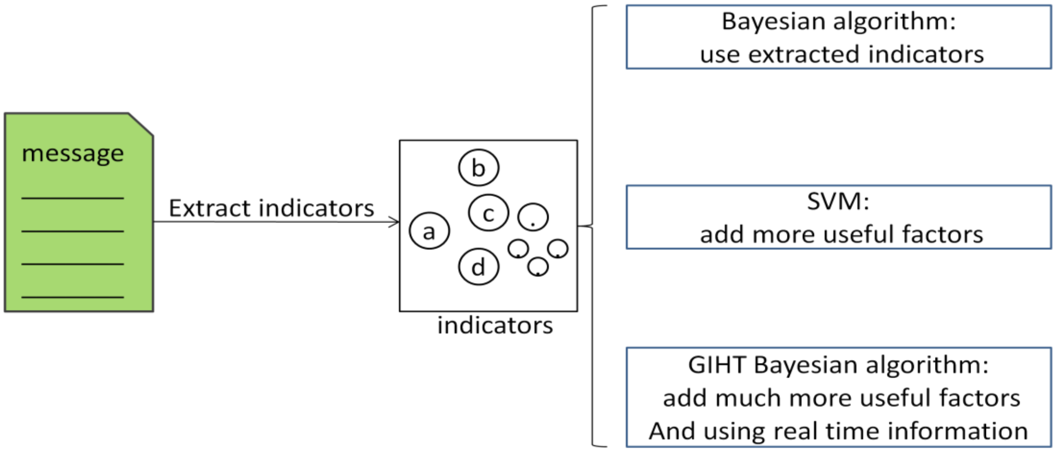


Figure 1.1 Compare spam filtering algorithms

Those methods deal with spam in different ways. The methods fall into three categories: a) list-based filters attempt to stop spam by categorizing senders as spammers or trusted users, block or allow their messages accordingly. The blacklist, real-time blackhole list, whitelist and greylist fall into this category. b) Rather than enforcing across-the-board policies for all messages from a particular email or IP address, content-based filters evaluate words or phrases found in each individual message to determine whether an email is spam or not. Relevant methods include word-based filters, heuristic filters, Bayesian filters and SVM filters. c) Additionally, some other methods include challenge/response system, collaborative filters, and DNS lookup system. The collaborative methods are more efficient than a single algorithm.

Content-based filters which evaluate words or phrases found in each individual message to determine whether an E-mail is spam or legitimate. The two main content-based filtering methods are the Bayesian based algorithm and the supervised algorithms based algorithm.

Bayesian filtering is one of the most widely used content-based algorithms to identify spam E-mails. The methods [28, 30] were widely used. Accordingly, the Naïve Bayesian algorithm has been integrated in many E-mail clients [1, 4]. The algorithms based on Bayes theorem extract keywords and other indicators from E-mail messages, and then determine whether the messages are spam using statistical or heuristic schemes. The Naïve Bayesian filter was superior to a keyword-based anti-spam filter which was included in a widely used E-mail reader [2]. The Naïve Bayesian could filter

99.5% spam with only 0.03% mis-classification [19]. In [25], the incremental training of personalized spam filters was evaluated in terms of the entire tradeoff between true positives and true negatives. Four filtering techniques for a Naïve Bayesian filter were assessed based on cost-sensitive measures. A new technique was also proposed to make a positive contribution as the first pass filter.

The machine learning algorithms were also used for content-based methods to filter spam. A learning approach was proposed to spam sender detection based on features extracted from social networks constructed from E-mail exchange logs [23]. The approach extracts several features from E-mail social networks for each sender. Based on these features, a supervised model is used to learn the behaviors of spammers and legitimate senders, and then assign a legitimacy score to each sender. Scores are made available in a database where online mitigation methods can query for the score of a particular sender [1]. In [24], the method called Spam Filtering Model Based on Support Vector Machine (SVM) was proposed. A SVM has some attractive features such as eliminating the need for feature selections for efficient spam classification. In [6, 2, 4, 8, 21], more different SVM based algorithms are proposed to filter spam.

However, neither the Bayesian based algorithm nor the machine learning method considers words' dynamic factors in the real environment. The diversity of factors indubitably alters E-mail's property so that a useful E-mail may become useless and verse vice. Moreover, spammers developed sophisticated techniques to trick content based filters by clever manipulation of the spam content [22, 12].

1.2.3 Visual Recommendation System

Search engines have acted a more and more important role in human life. However, popular search engines such as Google [62], Microsoft Bing [63], Yahoo [61], Ask [64], AOL [65] only utilize key words on a literal interface There are only few graphical interface based web search engines like Kartoo [67] and Bing visual search [63]. Kartoo is a special search engine with an interface which displays all relative contents in a graphical way; however it still needs a lot of inputs. Microsoft Bing provides a visual search based on their new technology called silverlight, however it works similarly as dmoz.com which only gives the result for one key word, and just replaces the input words by pictures.

As described above, most of current search engines do not provide 1) a convenient way to input relative concepts, 2) a proper graphical interface for disabilities and mobile device users, and 3) the detailed recommendation which provides complete information related to user inputs. Thus, it is necessary to develop a new visual recommendation system to satisfy the three requirements.

Researchers have already developed various interfaces of search engines. In [69], a new human computer interface for a Web search engine was introduced. Despite the noteworthy improvement provided in the Web search engine, the user interface still uses a textual sorted list. A prototype search engine with 3D models was created to investigate the design and implementation issues [70]. The search engine has three main components: (1) acquisition: 3D models are collected from the web, (2) analysis: they are analyzed for later matching, and (3) query processing and matching: an online system has to match user queries and the collected 3D models. In [71], the FOX-MIIRE 3D-Model Search Engine was proposed based on Adaptive Views Clustering (AVC) algorithm. The AVC method uses statistical model distribution scores to select the optimal number of views to characterize a 3D-model. The search engine also uses a probabilistic Bayesian method to retrieve 3D-models visually similar to a query 3D-model, photos or sketches. They present results on the Princeton 3D Shape Benchmark database (1814 3D-models). The 3D-model search engine is available on-line to test and assess the results. The problem of efficient query processing in scalable geographic search engines was studied [72]. Query processing is a major bottleneck in standard web search engines. Differently, geographic search engine query processing requires a combination of text and spatial data processing techniques. Several algorithms for efficient query processing in geographic search engines were integrated into an existing web search query processor. Also, a lot of research has been done in recommendation system. An agent-based personalized recommendation method called Content Recommendation System based on private Dynamic User Profile (CRESDUP) [73]. The system collects private data of users at the client side, mines them, and updates private Dynamic User Profile (DUP) at the client side. The system fetches preferred messages from the content server according to DUP. In [74], a semantic recommender system was proposed for e-learning (learners could find and choose the right learning materials suitable to their fields of interest). The proposed web based recommendation system contains ontology and web ontology language (OWL) rules. The rule filtering was used as a recommendation technique. The recommendation system architecture consisted of two subsystems: Semantic Based System and Rule Based System. Modules for the subsystems include Observer, Learner profile, Recommendation storage and User interface. The personalized recommendation system was designed based on multi agent systems [75].

Neither research in the graphical interface of a search engine nor research in a text based recommendation system can completely solve the issue, especially for the mobile devices' user and disabilities. Thus, a new visual recommendation system will be proposed. The new algorithm combines granular words with collaborative intelligence and the personal information to make a convenient way for a user to input a personalized query.

1.3 Motivation and Objective

With more and more web data, traditional natural language processing methods are no longer effective. For example, real time information is very useful for a person, however it is difficult to be organized and analyzed because of uncertainty with dynamic features. E-mail filtering usually utilizes the group history because processing personal message is difficult.All the issues just described inspire us to propose a new method to explore words in natural language. We will redefine the word with more relatives, make a new structure which can organize all those words, and develop a new computation method.

1.4 Dissertation Organization

The remaining chapters are organized as follows shown in Figure 1.2. Chapter 2 proposes the concept 'granular word’. Chapter 3 presents the algorithm 'computing with granular word'. Chapter 4 introduces the structure 'granular information hyper tree' which organizes all the granular words. Chapter 5 and chapter 6 present two applications, the GIHT based Bayesian algorithm for spam filtering and a visual recommendation system for search engine, respectively. Chapter 7 concludes this dissertation research and presents future research directions.

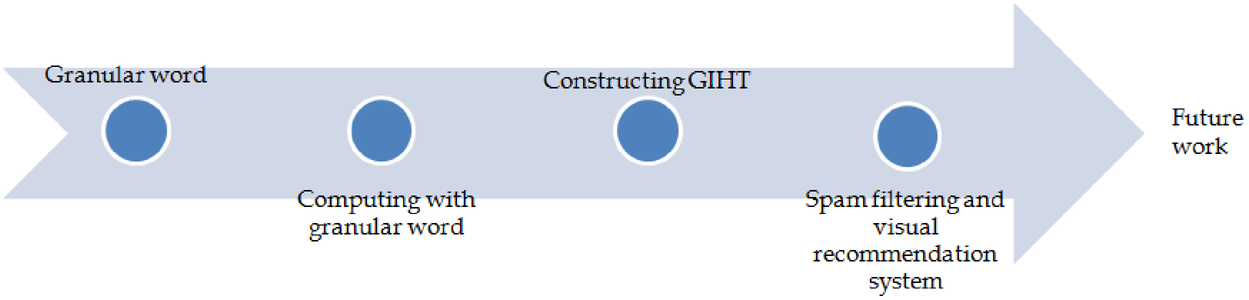

Granular word
Computing with granular word
Constructing GIHT
Spam filtering and visual recommendation system
Future work

Figure 1.2 Flowchart of thesis

CHAPTER 2

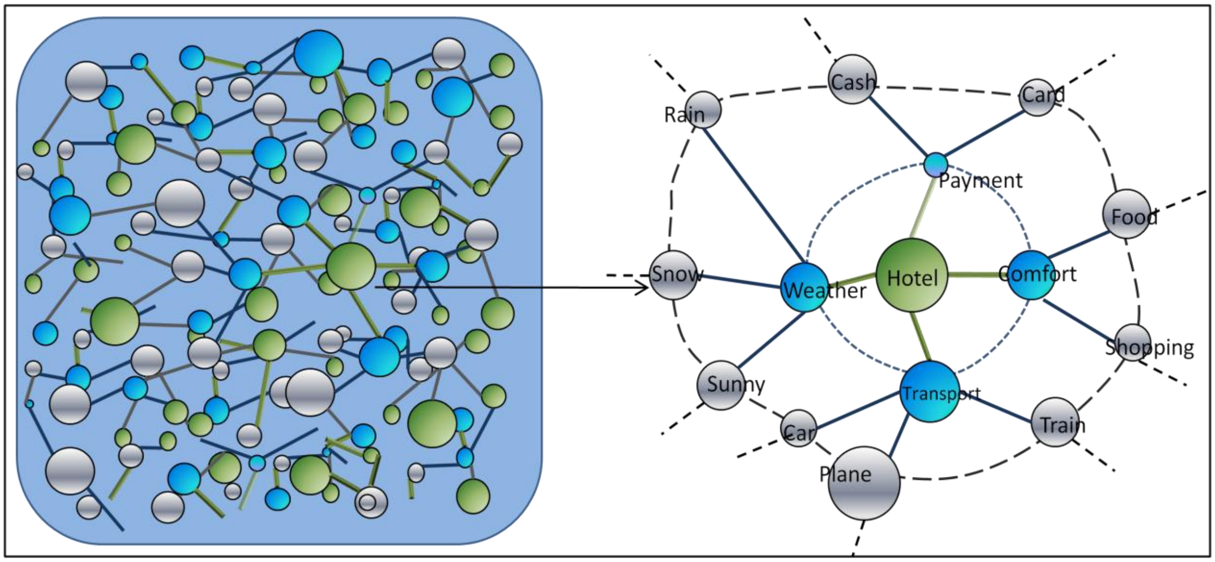
GRANULAR WORD

2.1 Overview

To obtain and organize all the granular words, the granular information hyper tree is proposed. An algorithm is presented to construct the GIHT by combining the word relationships in ODP and WordNet. The method to populate the dynamic weights of relationships is stated at the end.

2.2 Words' Relations in Nature Language

Considering the complex relations among concepts in nature language, a multi-



(a) Nature Languate (b) Granular Word 'hotel'

Figure 2.1 Granular Word

dimension structure of words with mutual relations is involved to solve the problem. Figure 2.1 shows the similar structure of the relative words, the nodes in Figure 2.1 (a) depict the words, and lines describe relationships in the natural language. Every word has several associative words. Theoretically, the more associative concepts are considered with the word, the more information can be obtained.

2.3 Simplification of Words

To organize the information of the words and the relations, the words in Figure 2.1 (a) are simplified at first. They are mapped into a tree as shown in Figure 2.1 (b). The solid lines in the tree denote the relationships, and the node in the center is the root word, there are several sub trees connected to the root node, all the nodes in the sub tree are the relative concepts; the relative concepts and relationships form one Granular Word 'hotel'. Moreover, in order to realize computing with Granular Words, all the weights of relationships shown in the tree will be normalized to [0, 1] by using fuzzy logic. The values denote the degrees of the dependencies among the words.

2.4 Schemas to Confine the Granular Word

It's hard to describe the concept 'granular word' without any limitation since the sub tree is connected to the root node may contain too many descendants. Large sub trees may cause complicate computation. To solve this problem, the *levels schema* and the *chains schema* are introduced to confine the Granular Words. In other words, the Granular Words can be restricted by the *level schema* and the *chain schema*.

With the *level scheme*, the root word lies in the center, the first level around the root word contains the words related to the root word, and the second level contains the words related to the words in the first level, and so on. For example, a person who will travel to New York is searching a hotel there, the information he wants to obtain is not only the hotel, but also other information related to his travel as shown in Figure 2.1 (b). By the level schema, the relative words may be limited within 2 levels which are: Payment, Comfort, Transport, and Weather; the words in the second level are: cash, card, food, shopping, car, plane, train, rain, snow, and sunny. If the user only provides information like 'cheap hotel in New York', the level schema restricts it with 2 levels, i.e. only the relative words on the first level and the second level should be considered.

For a root word with ***n* levels** of the associated words, the whole information set $\boldsymbol{S_L}$ can be expressed by the level schema such that:

$$S_L = \bigcup_{i=1}^{m} w_i \bigcup R --- (1)$$

In formula (1), g denotes the granular word, R means the root word, and w_i is a set of all the word in level i, the intact information of the Granular Word is the union of root and *m* levels of words (*m*<=*n*).

The chain schema counts the number of the sub trees primarily, and then it uses the number as the number of the chains (each chain is one sub tree connected to the root node). In the word computation, some chains may never be used under certain conditions. Therefore, the information set $\boldsymbol{S_C}$ of the Granular Word with ***t chains*** can be also expressed by the chain schema:

$$S_C = \bigcup_{j=1}^{k} w_j \bigcup R --- (2)$$

In formula (2), the intact information of Granular Word is the union of root and the *k* chains of words (*k*<=*t*).

2.5 Definition of Granular Word

With the simplification by mapping granular words into tree, the level schema and

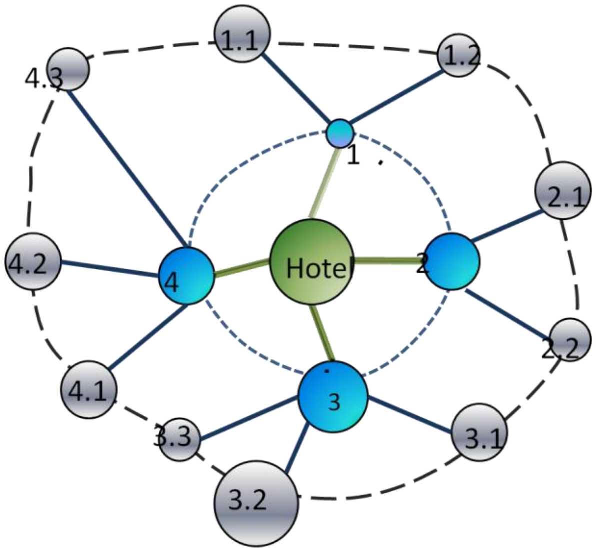


Figure 2.2 IDs assigned in granular word

the chain schema are introduced to confine granular word. To describe every word in the tree, IDs are assigned to each sub tree as shown in Figure 2.2. Then the granular word can be defined as in definition 1:

Definition 1:

The Granular Word (GW) is a cluster of words which includes the root word (R) and all other associative words (R^{rw}). The GW can be represented by a tree structure, the weights can be normalized to [0, 1]. Mathematically, granular word can be defined as:

$$G = \bigcup_{r=1}^{n} w_r \bigcup R --- (3)$$

The formula denotes that a granular word G is a word set contains root word and n relative word w_r with its ID and weights.

CHAPTER 3

COMPUTING WITH GRANULAR WORD

3.1 Normalizing the Weights of Granular Word

The weights of relationships in granular word can be obtained from online services (the method will be discussed in chapter 4). Since the online information may be presented in different types, such as number, linguistic terms and images, etc, the fuzzy logic is utilized to normalize all of the real time values to [0, 1].

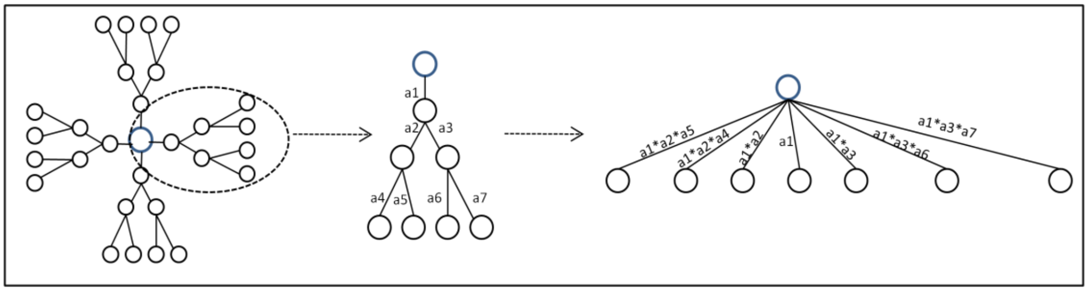
3.2 Compressing Paths Algorithm

To further simplify the structure of the Granular Words in order to be computed conveniently, compressing paths is considered. The compressing paths algorithm is derived from the discrete set. It will make every node in the Granular Word re-connect to the root node directly, and every new weight is changed to the product of all the old weights within one path. The algorithm has 3 functions:

MAKE-SET(*R*): construct a new set, the only member is *R*. Since all the sets are discrete, *R* is not a member in other sets.

UNION(R, y): union sets which contain root R and y, respectively (such as SR and Sy) into one new set.

FIND-SET(x): return a pointer which point to the set which only contains x.



(a) A Granular Word *(b) a chain before compressing path* *(c) the chain after compressing path*

Figure 3.1 compressing the paths of Granular Word

The complete compressing Algorithm is given below:

Algorithm 3

```
MAKE-SET(R)

    p[R] ← R

    rank[R] ← 0


UNION(R, y)

    LINK (R, FIND-SET(y))


LINK(x, y)

 if  rank[x] = rank[y]

    then

            {

            rank[y] ←rank[y]+1

            Link(x, y)
```

```
                }
 elseif   rank[$x$]> rank[$y$]
     then    {
                p[$y$] $\leftarrow x$
                 $t$ = rank($x$)-rank($y$)
                $Wx = \prod_{i=1}^{i=t} W_i$
                }
  else        {
                p[$x$] $\leftarrow y$
                 $t$ = rank($y$)-rank($x$)
                $Wy = \prod_{i=1}^{i=t} W_i$
                }

FIND-SET($x$)
  if    $x \neq$ p[$x$]
     then
                {
                p[$x$] $\leftarrow$ FIND-SET (p[$x$])
                return   p[$x$]
```

}

The LINK(x, y) is a sub function of UNION(R, y). Figure 3.1 shows the procedure of compressing the paths of a Granular Word. After compressing, a Granular Word can be represented with one level model as: G(R, S, x), where R is the root of the Granular Word, S is the set of all the surrounding words, x is the set of all the new weights of the surrounding words.

3.3 Granular Word Computation Model

The compressing paths algorithm simplifies the structure of the Granular Word to one level model. All the related concepts around the root word are moved to the first level. The computation of the one level Granular Word G(R, S, x) has two ways: single Granular Word computing and multiple Granular Words computing.

3.3.1 Single Granular Word Computation

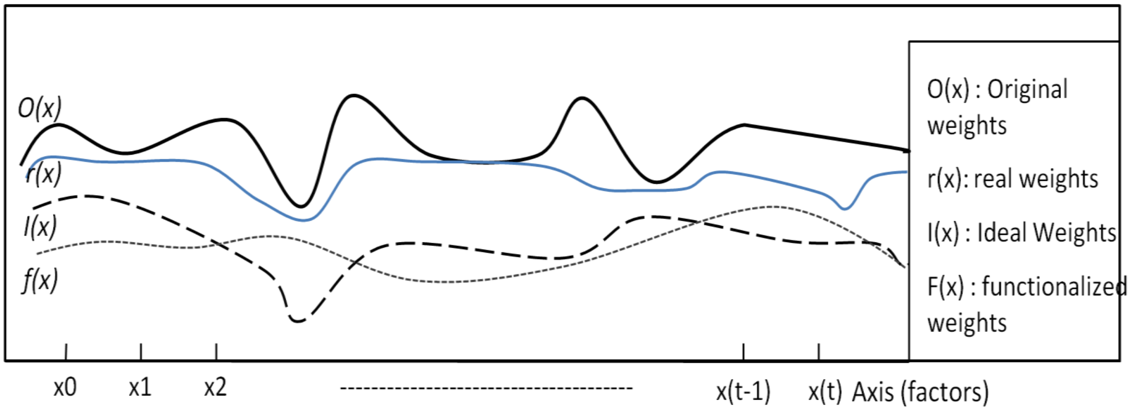


Figure 3.2 Functions of computing with granular word

For a compressed Granular Word g(R, S, x), a general model is defined to perform the computation: g(R, S, x, f(x)) where f(x) is a predefined function to compute the weights of related concepts.

Figure 3.2 shows 4 functions: O(x) presents the original weights of the factors, r(x) expresses the real time values of the weights which retrieved from online services, I(x) illustrate the ideal personalized situation which shows the ideal weights, and f(x) denotes the predefined function. In the algorithm, both I(x) and f(x) should be theoretically included by the r(x) because r(x) is the real time function from the GIHT. The general algorithm will involve f(x) and r(x). Thus, the standard deviation method will be used to perform the computation and the result of trust variable p is used to evaluate each sample (p is an uncertainty value between 0 and ∞, denotes how close f(x) to user's requirements). Now assume there are l candidate samples associated with the Granular Word, equation (4) can be used to compute p.

$$p = \frac{\sqrt{\sum_{i=1}^{l}(f(x\ ) - S(x_i))^2}}{\sqrt{\sum_{i=1}^{l}(f(x\ ) - r(x_i))^2}} --- (4)$$

Here, S(x) denotes the values of the sample's weights. In the formula, the p is smaller, the better result is better, which means the sample is closer to user requirements; and verse vice.

3.3.2 Multiple Granular Words Computation

For the multiple Granular Words computation, the words are related to each other; otherwise, each one can be handled as a single Granular Word. To apply the formula (4) to the multiple Granular Words computation, the simple way is to convert these words into one word. A root election algorithm is proposed to perform the operation. With the algorithm, a formed Granular Word after election keeps the strongest relationship, which means the sum of the weights of the formed Granular Word with an elected root word must be bigger than the sum of the weights of any other structures of this Granular Word.

Algorithm 4:

ELECTING-ROOT

```
for ($x$=w(1) → w($n$))
    { $x$ is root word, connect other words.
      Call Algorithm 1;
      $Tmp$ = Sum of relationships
      if ($E$ < $Tmp$)
          { $E$=$Tmp$ ;
            $R$=$x$;
          }
    }
```

In algorithm 4, Tmp is a temporary variable; E is a variable which holds the sum of weights. The algorithm traversals all the words and sets each as the root word temporarily, then computes the Tmp of the weights, and finally compares it with E. If E is less than Tmp, set Tmp’s value to E, and set current word as the root word, otherwise, continue.

3.3.3 Personalized Computation

The general algorithm has been discussed previously that a predefined function f(x) is used to compute the result. However, users generally like personalized results more than general results. In Figure 3.2, I(x) depict the ideal function which is the users’ personal ideal status. But, since the final result associates with a set of values of the weights, the predefined function f(x) cannot be set exactly as same as I(x) to get the perfect result. Nevertheless, in the real world, it is possible to get it by investigating user's requirement. Assume that the function I(x) was given. Then the formula to compute granular word can be represented as:

$$p = \frac{\sqrt{\sum_{i=1}^{l}(I(x\ ) - S(x_i))^2}}{\sqrt{\sum_{i=1}^{l}(I(x\ ) - r(x_i))^2}} --- (5)$$

In formula (5), if p is close to 0, S(x) is close to I(x), meaning the sample is optimal; otherwise, the sample is close to r(x) (meaning the sample is poor). Finally, the optimal results will be returned to the user.

3.4 Experiment and Evaluation

In this section, an example of the computing with Granular Word is presented. The assumed problem is described as following:

Question: there are 3 types of heaters, the information of groups and the average values for the features of the links are shown in table IV; a person who is living in New York, and 65 years old need 'cheap heater' on March 1st 2009, the temperature was between 23-34°F at the day. Please find 2 optional heaters for the person.

Four main steps include 1) Defining basic concepts, 2) Defining function, 3) Result computation, and 4) Personalization.

3.4.1 Defining Basic Concepts

3.4.1.1 Concepts for the CWGW

As mentioned, all the weights in granular word were normalized into [0, 1] before computing. Figure 3.2 shows the structure of the granular word 'heater' with 2 levels and 5 chains. By definition 1, the Granular Word can be presented as:

$$G_{heater} = \bigcup_{r=1}^{n} w_r \bigcup R$$

Base on the structure, the dynamic information on March 1st 2009 as shown in the second column in Table 3.1 (if there is no real time value for a weight, the predefined value will be taken).

3.4.1.2 Linguistic Variables for the CWW

To compare the algorithms, the TSK model is selected to perform the computation.

The Linguistic membership functions of the age are defined as following:

$$\mu_{\text{age}} = \begin{cases} young: y = -\dfrac{1}{45} * x + 1 \\ midage: y = \dfrac{1}{45} * x \ldots (x \le 45) \\ \qquad\qquad y = -\dfrac{1}{45} * (x - 45) + 1 \ldots (45 < x \le 90) \\ oldage: y = \dfrac{1}{45} * x - 1 \end{cases}$$

The Linguistic membership functions of the temperature are defined as:

$$\mu_{\text{temp}} = \begin{cases} low: y = -\dfrac{1}{33} * (x - 20) + 1 \\ medium: y = \dfrac{1}{33} * (x - 20) \cdots (x \le 56) \\ \qquad\qquad y = -\dfrac{1}{32} * (x - 53) + 1 \ldots (x > 56) \\ high: y = \dfrac{1}{33} * (x - 53) \end{cases}$$

The Linguistic membership functions of the price are defined as:

$$\mu_{\text{price}} = \begin{cases} cheap: y = -\dfrac{1}{3490} * (x - 20) \\ medium: y = \dfrac{1}{3490} * (x - 20) \ldots (20 \le x \le 3490) \\ \qquad\qquad y = -\dfrac{1}{3490} * (x - 20) - 1 \ldots (3490 \le x \le 7000) \\ \exp ensive: y = \dfrac{1}{3490} * (x - 3490) \end{cases}$$

All of the functions are depicted in Figure 3.2 (a), (b) and (c).

3.4.2 Basic Functions

Table 3.1 Computing with granular word

Related concepts	Real time values	f(x)	Normalized	S(x)	S(x)	S(x)	I(x)
			f(x)	heater A	heater B	heater C	
Region	North latitude 36	North latitude 36	0.08	0.095	0.092	0.07	0.1
History	very need	very need	0.05	0.04	0.053	0.045	0.06
Weather	23F-34F	23F-34F	0.08-0.3	0.07-0.28	0.062-0.39	0.07-0.36	0.04-0.37
Mechanical	43 categories	44 categories	0.1	0.063	0.051	0.082	0.05
Access, duct work	7	7	0.001	0.0008	0.001	0.0015	0.001
Air distributor	119	119	0.013	0.011	0.0105	0.012	0.01
Building services	29	29	0.003	0.004	0.003	0.0029	0.002
Plumbing	228	228	0.025	0.017	0.016	0.024	0.018
Ventilating, AC	397	397	0.043	0.033	0.03	0.042	0.029
Decoration	35	35	0.004	0.0021	0.002	0.004	0.001
Type	27	28	0.2	0.215	0.25	0.21	0.3
Boilers	73	73	0.044	0.047	0.059	0.041	0.06
Radiant Heating	97	97	0.018	0.022	0.024	0.015	0.025
Water Heaters	31	31	0.004	0.005	0.005	0.005	0.006
Heating elements	8	8	0.06	0.076	0.078	0.062	0.08
Furnaces, incinerators, Kilns	126	126	0.08	0.087	0.098	0.085	0.1
Price	\$20-\$7000	\$50-\$100	0.0071-0.03	0.007-0.025	0.006-0.024	0.009-0.02	0.008-0.024
Popular	2000	600	0.3	0.23	0.27	0.29	0.2
Watts	1500-25000	1000-3000	0.04-0.12	0.08-0.11	0.1-0.12	0.05-0.09	0.1-0.15
Age	65	65	0.813	0.9	0.78	0.82	0.8

3.4.2.1 Function of the CWGW

After obtaining the real time values from online services, f(x) is defined, which means compute with Granular Word 'heater' related to conditional word 'cheap' with the function f(x). Theoretically, any function can be predefined; however the greedy algorithm is selected to help define f(x) to get the most benefit for the user. The word 'cheap' may have multiple meanings such as low price, not popular, reasonable watts; therefore the values of these three weights are smaller than others which may keep the biggest real time values. The final f(x) can be presented as:

$$F\left(x\right) \ = \{x \mid low\ \left(price,\ popular,\ watts\right) \bigcup high\ \left(other\ factors\right)\}$$

The function *low* defines the conditions of 'cheap' and the function *high* realizes the greedy algorithm, f(x) and normalized f(x) are shown columns 3 and 4 in Table 3.1, respectively.

Table 3.2 the Fuzzy Rules

Age\Temperature	low	medium	high
young	0.3x+0.5y+20	0.3x+0.3y+10	0.3x+0.2y+5
Mid age	0.2x+0.5y+10	0.2x+0.2y+10	0.2x+0.1y+5
old	0.5x+0.5y+30	0.5x+0.5y+15	0.5x+0.5y+10

3.4.2.2 Fuzzy Rule Base of the CWW

The CWGW only needs one function, but the CWW needs 9 fuzzy rules. The nine fuzzy rules are listed in Table 3.2.

3.4.3 Computation and Results

3.4.3.1 Computation of the CWGW

There are 2 steps in the computing with Granular Word: weights compressing and the general result computation. In the first step, Algorithm 4 compresses the weights of Granular Word; the result is shown in Figure 3.3. Every word around the root word has a new computed weight.

The general algorithm is used for the case that there is no personalized computation. With equation (4), S(x) is the attribution of certain sample, if P's value is close to 0, the sample is close to user's requirement. Since there are three groups of links which

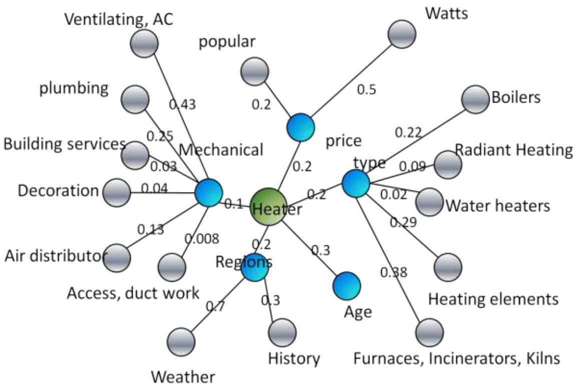


Figure 3.3 real time weighted Granular Word

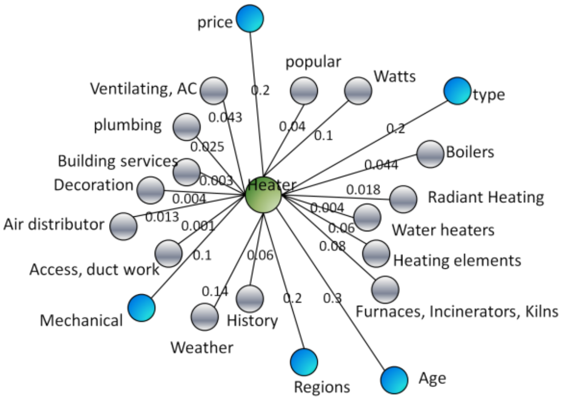


Figure 3.4 Relations compressed Granular Word

contain 'heating' products with related features shown in Table 3.1, if the range of the satisfaction of P is set to [0, 0.2], then heaters A and B are the products suitable for the person.

3.4.3.2 Example of the CWW

Fuzzy reasoning is shown in Figure 3.5, the user's age and the temperature fire 4 fuzzy rules to match the cheap price: rule 1=[mid age, low temperature], rule 2=[mid age, medium temperature], rule 3=[old age, low temperature], rule 4=[old age, medium temperature]. Thus, the result range of the TSK model is:

$$p_{\text{low}} = \frac{\mu_1 * rule1 + \mu_2 * rule2 + \mu_3 * rule3 + \mu_4 * rule4}{\mu_1 + \mu_2 + \mu_3 + \mu_4} \approx 48$$

$$p_{\text{high}} = \frac{\mu_1 * rule1 + \mu_2 * rule2 + \mu_5 * rule3 + \mu_6 * rule4}{\mu_1 + \mu_2 + \mu_5 + \mu_6} \approx 52$$

The normalized range is [0.0069, 0.0074], which means all of the 3 heaters are satisfied by the person. It is obvious that CWW doesn't answer the question.

3.4.4 Personalization

Suppose the age of the user is bigger than the average level of the given age, and the user's age is 85.

3.4.4.1 Personalized Result of the CWGW

When strategies are employed to obtain the user's personalization information, another function I(x) is defined based on the user's personal information as shown in Table 3.1. Then equation (5) can be used to perform the computation, heaters B and C will be returned to the person finally. Compared with the result of equation (4), the personalized result is much better.

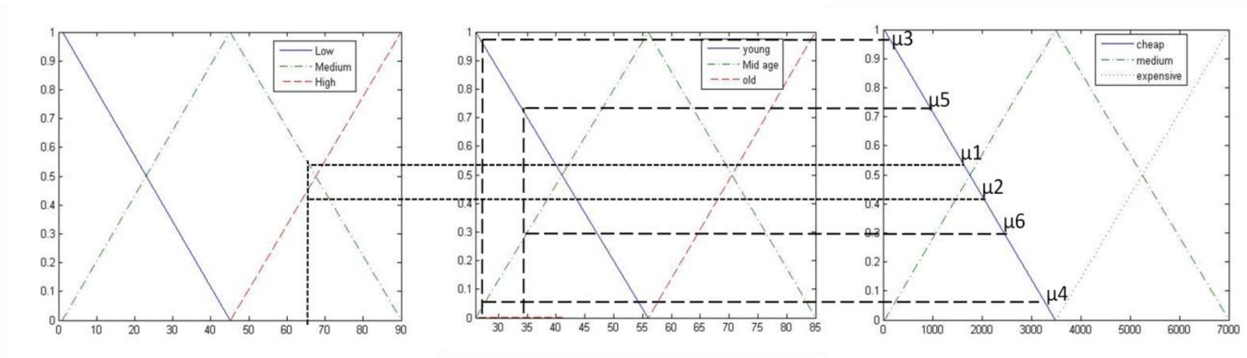
3.4.4.2 Personalized Result of the CWW

The age update means that the value of the variable is redefined by the user. Then the result is:

$p_{\text{low}} \approx 52.4$

$p_{\text{high}} \approx 56.2$

After normalization, all of the 3 heaters satisfy the demand of the user.



(a)Age (b) Temperature (c) Price

Figure 3.5 Computing with word by fuzzy logic

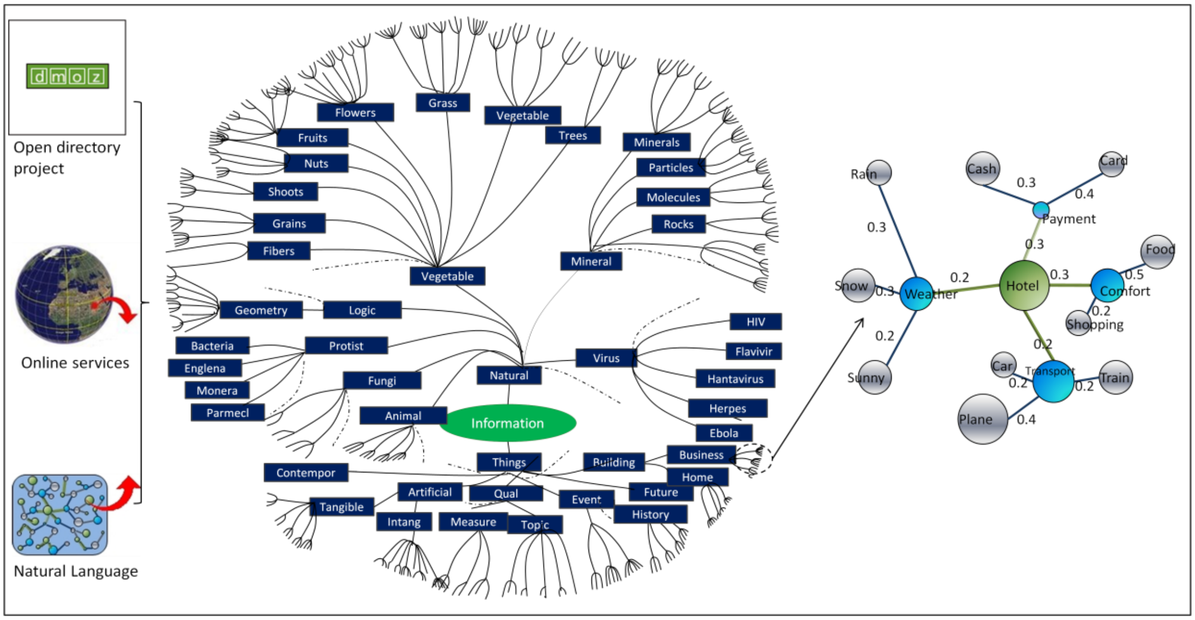
CHAPTER 4

GRANULAR INFORMATION HYPER TREE

4.1 Granular Information Hyper Tree

In [18], the Factor Hyperbolic Tree (FHT) was proposed to obtain the words' information from online services to filter spam. A similar structure named Global Information Hyper Tree (GIHT) is newly created to present the Granular Words. It is shown in Figure 4.1(b).

Algorithm 4.1 is used to form the GIHT. In the algorithm, IMPROVEMENT(ODP) is the function to transverse the whole tree to add more relations among concepts. For



(a) Combining (b) Global information hyperbolic tree (c) a Granular Word from GIHT

Figure 4.1 Structure of Granular Word

each object in ODP, the algorithm searches the relative concepts in WordNet, then compares them with the current neighbors in ODP to find intersections, and finally add new relations by calling function ADD_RELATION(node). FIND_WORDNET in the algorithm stands for searching the relative concepts for a certain word which is passed from ADD_RELATION. Then the GIHT is formed as in Figure 4.1(a), the information node is the root and it organizes all the sub trees; the nodes in sub trees present the concepts, the neighbors surround a word are considered as the related factors for the word. For example, the neighbors of word 'air condition' are 'regional', 'health' and 'business', so the three concepts will be used when computing the E-mail which contains 'air condition'. Figure 4.1 (b) shows the relationship between words. The object is represented by a factor set in the GIHT, and the values of the factors in the set will be updated corresponding to changeable environmental factors. The Granular Words in the GIHT are represented by word clusters, and the values of relations in the clusters are updated corresponding to the changing of environment. Figure 4.1 (c) is an example after the relationships are updated to a specific condition, and the values are obtained from the condition and then normalized to [0, 1].

ALGORITHM 4.1 GIHT_CONSTRUCTING:

```
IMPROVEMENT (ODP) Class
Start (node)
        {
        while haschild (node)
                {
```

```
                node = node.i (n≥i≥1)
                visit (node)
                ADD_RELATION (node)
                }
        if (haschild (node))
                Start (node)
        else
                {
                while node.parent ≠ null and node = node.parent.n
                        node = node.parent
                }
}

ADD_RELATION (node)
    for each element in FIND_WORDNET(node)
        if (element not neighbored to node)
            if(find(element in ODP)==TURE)
                relation.add(element->node)

FIND_WORDNET (node)
{
        Find(node)
        Return(relative nodes)
}
```

4.2 GIHT and its Application Areas

The GIHT is constructed by extracting relationships among words from ODP and WordNet. The cloud computing can be utilized to provide semantic values or numerical values for the weights in GIHT, and each value is normalized into [0, 1].

The GIHT can be applied to very broad areas. (1) In nature language processing, it can be used to process ambiguity, synonym, etc. (2) Further, computing with granular words satisfies the requirement of people living in the era of information blast. (3) Granular word can also be used in the security fields.

CHAPTER 5

GIHT BASED BAYESIAN ALGORITHM FOR SPAM FILTERING

5.1 Introduction

A new content-based filtering algorithm called the GIHT based Bayesian (FHT-Bayesian) algorithm was proposed in this chapter, it computed words and phrases by considering various relevant factors to perform spam filtering in a dynamic environment. In the model, the Ranked Term Frequency (RTF) algorithm was used to extract indicators from fuzzy E-mails related to environmental factors. Type-1 and Type-2 fuzzy logic systems were used to evaluate the indicators to determine whether an E-mail was spam based on the environmental factors. Additionally, weights of factors in a GIHT database were updated according to dynamic conditional factors in the real-time environment. In the new algorithm, the Bayesian algorithm primarily classified E-mails into 3 categories: absolute spam, absolute legitimate E-mails, and fuzzy E-mails. Then the GIHT classifier was used to re-classify the fuzzy Emails into either spam or legitimate E-mails. Simulation results showed that the GIHT-Bayesian algorithm was more precisely than the Naïve Bayesian (NB) and the Support Vector Machines (SVMs).

5.2 Using GIHT for Decision Making

A decision is normally made by human based on relevant factors. For example, whether air conditioning (AC) is useful depends on two factors: temperature and zone. The distance and the terrain may help people determine what kind of vehicle is needed for an expedition. Thus, any object in common life is influenced by diversity of factors. To express the relationship among factors and objects, granular words are used to describe every object of interest with related factors; the values of factors can be constantly updated. Thus, every related concept in a granular word is considered to be a factor to the root word.

GIHT consists of words and the relationships. The records in the GIHT include products, weather, human, society, country, culture, zone, etc. In order to clarify the concept, a simple example is illustrated as following:

What factors are related to a heater?

Weather *In summer, a heater may be useless; in winter, it is possibly very useful.*

Zone *In Florida, a heater is practically useless; in New York, it is useful.*

People *For old people, a heater may be needed; for young people, it may not be as necessary.*

Popularity degree *If people would like to buy a heater in the winter instead of AC, it has increased popularity relative to an AC; otherwise, it does not.*

In the example, each explanation of the factor describes how it works to the heater's usefulness.

5.3 Classification of Factors

In a decision making system, these factors are classified into two categories in: normal and abnormal.

Normal factors: factors in common life belong to this category, these factors change with time naturally or periodically, such as weather, age, etc.

Abnormal factors: factors usually happen suddenly and affect the object immediately, such as disasters, earthquake, etc.

To compute the object with factors, these factors are normalized to a number between 0 and 1 by fuzzy logic. To achieve a more accurate expression, a type-2 fuzzy set which is an extension of type-1 fuzzy sets is considered. A type-2 fuzzy set is used to normalize factors because: (1) the variation of normal factors obeys natural rules, which means the value of normal factors at certain time must fluctuate in a range; and (2) a range of values will be generated if periodic values of normal factors are selected.

Figure 5.1(a) shows the temperature curve of Atlanta in 2008 [15]. The value of every month fluctuates in a range (the blue dots indicate lowest temperatures, orange dots indicate highest temperatures, and the red dots indicate average values), the type-2

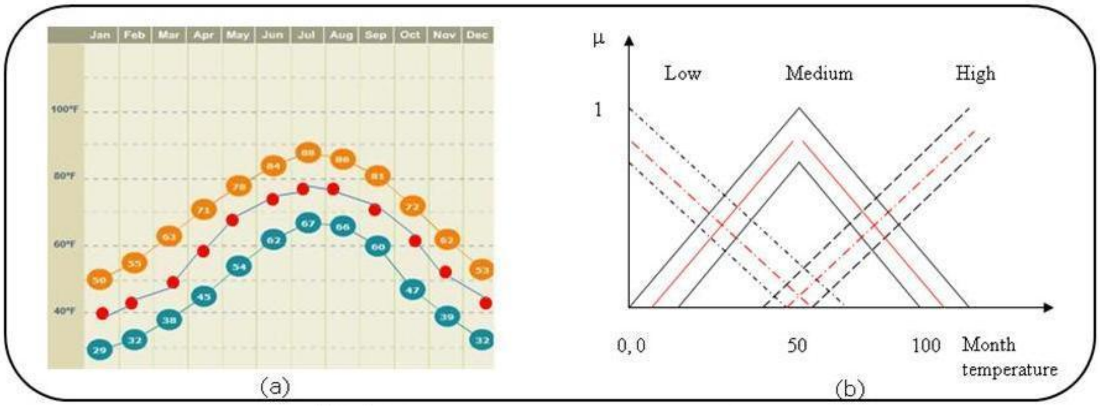


Figure 5.1 type-2 fuzzy sets

membership function of a specific month's temperature is shown in Figure 3.1(b), where uncertainty of a temperature value is represented by an additional dimension (the red line indicates the defuzzied membership function). All initial weights between factors and objects are set to 0; and type-2 fuzzy logic is used to compute the weights between normal factors and objects w_{n-i} (i is the index normal factor).

5.4 Ranked Term Frequency (RTF) and Fuzzy E-mail

Key words are extracted from E-mail as the features. RTF performs extracting key words or phrases from messages.

The Term Frequency Inverse Document Frequency (TF-IDF) weight is an algorithm often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is for a document in a collection or corpus. A RTF algorithm was proposed to mine nouns in the E-mail, while the IDF concept is ignored because more accurate features are expected to be obtained from the single E-mail.

The RTF in the given document is simply the ranked number of times that a given term appears in that document. This count is usually normalized to prevent bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j.

$$\Gamma_{(i,j)} = \frac{n_{(i,j)}}{\sum_k n_{(k,j)}} --- (6)$$

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.

The RTF result will be obtained after all $\Gamma_{(i,j)}$ have been ranked. A high weight in RTF is reached by a high term frequency (in the given document); hence the weights tend to filter out important noun terms in d_j. To catch the main meaning of documents, the first k (1≤k≤5) nouns are selected (for short message, we can only get less than 3 terms; for a normal E-mail, we can get 3-5 terms; for long message, it will be more, however, long messages are rarely used), when the first k nouns with high frequency are extended into granular words, the granular words cover more than 90% of the main terms in an E-mail.

5.5 GIHT-Bayesian Algorithm

Real spam is sent to a mailing list or newsgroup for E-mail advertising for a product or service. In addition to wasting people's time with unwanted E-mails, spam also consumes a lot of network bandwidth. The fuzzy E-mail which defined in this research is an uncertain E-mail that may or may not be useful for particular users. The proposed method of distinguishing spam from legitimate E-mail in fuzzy E-mail is to analyze the conditional factors to be obtained by searching in the GIHT. The prerequisite of processing fuzzy E-mails is to classify the fuzzy E-mails from other E-mails.

5.6 E-mail Classification by the Bayesian Algorithm

In this section, we primarily define the set of the fuzzy E-mails N_F : the fuzzy E-mail is the E-mail that may be spam for a person but not for another one. Thus there are 3 classes in the new classification of E-mail: absolute spam, fuzzy E-mail and absolute legitimate E-mail. In the new algorithm, the Bayesian algorithm is used to perform the classification. The traditional Bayesian anti-spam algorithm maintains two hash tables,

Algorithm 5.1: GIHT Bayesian Algorithm

```
Begin
Step 1. Classify E-mail into 3 categories: absolute spam, absolute legitimate E-mail, and fuzzy E-mail by using the fuzzy Bayesian algorithm.
Step 2. Extract $k$ nouns in fuzzy E-mail by using RTF and input to the FHT.
Step 3. Search related factors to each noun in the FHT and provide factors to the Fuzzy Logic Unit in two types: normal factors and abnormal factors.
Step 4. Fuzzy Logic Unit will compute normal factors to obtain an incomplete result $r_1$ ; then abnormal factors will be involved to compute another incomplete result $r_2$ .
Step 5. Weighted Computation Unit will calculate two incomplete results $r_1$ and $r_2$ with predefined weights to obtain a result $t_r$ for each token.
Step 6. All the $t_n$ of words are involved to calculate the final result.
End
```

which pertain to a spam set and legitimate E-mail set, respectively. The new Bayesian algorithm maintains 3 hash tables, where the additional table pertains to the fuzzy E-mail set. If one E-mail is spam for some persons but is legitimate E-mail for others, it is a fuzzy E-mail, and the token will be saved in the fuzzy hash table. From the Bayesian theorem and the theorem of total probability, given the vector $x = \langle x_1, x_2 ......x_n \rangle$ of message d, the probability of message d belonging to category C is shown in formula (4), where $k \in \{spam, legitimate, fuzzy\}$. The probabilities $P(X \mid C)$ are practically impossible to estimate directly because X is too large. The Naïve Bayesian assumes that the attributes $X_1, X_2, ..., X_n$ are conditionally independent given in the category C.

$$P(C = c \mid X = x) = \frac{P(C = c) * \prod_{i=1}^{n} P(X_i = x_i \mid C = k)}{\sum_k P(C = k) * \prod_{i=1}^{n} P(X_i = x_i \mid C = k)} --- (7)$$

5.6.1 Processing Fuzzy E-mails

In order to re-filter fuzzy E-mails by the conditional factors, original E-mails must be processed by the following steps: (1) extract features of fuzzy E-mails by the RTF algorithm; (2) search factors related to the features in GIHT; and (3) represent E-mails by features with associated factors.

5.6.2 GIHT-Bayesian Algorithm

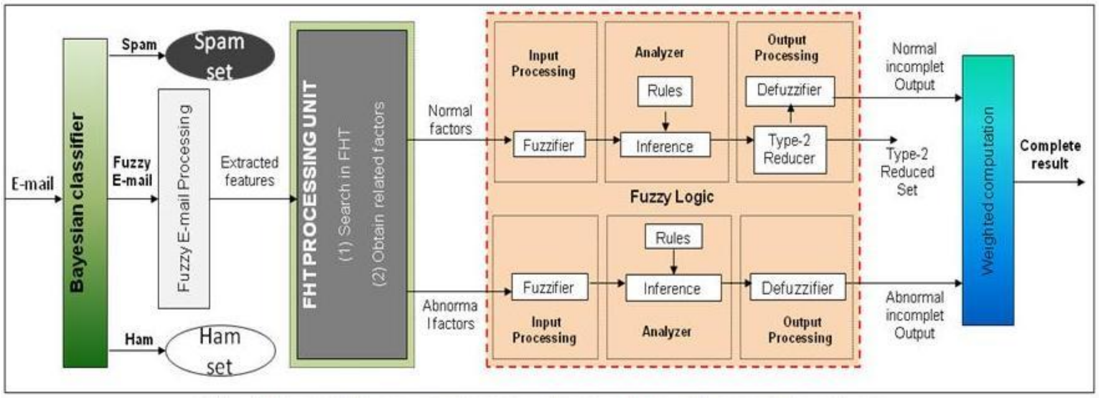


Figure 5.2 Computation model of GIHT-Bayesian algorithm

The GIHT-Bayesian algorithm is a computation model based on GIHT, Bayesian methods and fuzzy logic as shown in Figure 5.2. E-mails are classified into three categories by Bayesian classifier at first; consequently fuzzy E-mail's indicators are extracted by RTF algorithm; then the indicators will be searched in GIHT structure to obtain related factors, the information and user's personal information (considered as normal factors) are forwarded to the factors computation unit; once the factors computation unit is triggered, it will compute the normal factors and abnormal factors, respectively, and generate the final result.

The details of the model are given in Algorithm 5.1. In step 4, factorial analysis is applied to set up a rule base. The factorial analysis algorithm can be used to obtain the affection degree of normal factors by analyzing historical data. The values of these degrees are represented by $1 * n$ matrix: $Degree = \{d1, d2\ldots dn\}(0< di <1)$.

Then the rules are computed by equation (8) (α is the parameter of the membership function). Moreover, for the abnormal factors, the type-1 fuzzy system is applied to compute an incomplete output. Further, equation (9) is used in step 5 to calculate t_{kr} for each token (*w*1 and *w*2 are predefined weights). Finally, formula (10) computes the final result λ.

$$Rl_i = 1 - \prod_{1}^{n} (1 - a_n d_n) --- (8)$$

$$t_{kr} = w_1 * r_1 + w_2 * r_2 --- (9)$$

$$\lambda = 1 - \prod_{1}^{k} (1 - t_{kr}) --- (10)$$

Additionally, to identify the spam E-mail in the fuzzy E-mail by the GIHT-Bayesian algorithm, the GIHT-Bayesian algorithm automatically learns from prior cases and gets the criterion value by averaging λs. Then E-mail is identified by the GIHT algorithm - if the output is smaller than λ, it is spam; otherwise, it is not.

5.7 Experiment and Evaluation

5.7.1 FHT Bayesian Algorithm

The E-mail filtering result of the GIHT-Bayesian depends on two factors: the number of factors and the weights of the normal and abnormal factors. Moreover, the different ratios of the weights of normal factors and abnormal factors yield different crisp outputs λ. The ratio is set to a fixed value 1 for simulations.

The GIHT and the Bayesian algorithm are supervised algorithms. The online E-mail dataset Trec05 with 92,189 messages is used, 4932 messages are randomly selected as the training set. In order to obtain real data, 10 terminal users with different backgrounds were invited to help artificially classify E-mails into spam, legitimate E-mail, and fuzzy E-mail at first (we obtained the set of fuzzy E-mail by finding out the E-mail that would be ham (resp. spam) for certain persons but be spam (resp. ham) for others). Finally, there were 2317 spam, 1943 legitimate E-mails and 672 fuzzy E-mails. The 3 categories of E-mail were processed by the Bayesian classifier to form 3 hash tables primarily; subsequently, the fuzzy E-mails were added into the GIHT part to obtain the criterion λ = 0.55 by averaging all outputs λs. If the final result of other fuzzy E-mail which was computed by the GIHT computational model was greater than 0.55, it would be legitimate E-mail; otherwise, it was spam.

5.7.2 Evaluation

The two popular datasets Enron spam [10] and Spam Assassin [14] are selected to evaluate the algorithm. The Enron Spam data set was introduced in 2006 [1]. The preprocessed version, which contains only the subject and the body of the message, is used in the experiment. The 10 invited users were asked to pick out 200 fuzzy E-mails, 400 absolute junk E-mails and 400 absolute legitimate E-mails from the Enron3 data set which contains 4012 legitimate messages and 1500 junk E-mails. The newly created data set had total of 1000 messages resulting in 40% spam, 20% fuzzy E-mail, and 40% legitimate E-mail. The Spam Assassin corpus contained 6047 E-mails with 31% spam ratio. Specially, invited users picked out 157 fuzzy E-mails directly from a “hard ham” set which contains 250 E-mails and 43 fuzzy E-mails from general spam. Moreover, 400 absolute spams and 400 absolute legitimate E-mails were selected from the original ham set and the original spam set. Finally, the formed test set of Spam Assassin also resulted in 40%, 40% and 20% (in order to obtain accurate result, the HTML tags were removed from the E-mails in the selected set).

5.7.3 Performance Benchmarks

The following performance measures are considered in the experiment: Accuracy (Acc) which measures the percentage of correctly classified messages, Spam Recall (SR) (resp. Ham Recall (HR)) which is the percentage of spam (resp. legitimate) messages assigned to the correct category and Spam precision (SP) (resp. Ham precision (HP)) which is the percentage of messages classified as spam (resp. legitimate) that are indeed spam (resp. legitimate). By denoting $n_{x \to y}$ the messages of class x that are classified to class y, N_x is the number of E-mails in particular class x. S, L and U indicate the spam, the legitimate and the fuzzy class, respectively. FL and FS indicate the legitimate fuzzy Email and the spam, respectively. The following formulas are used to handle the fuzzy E-mail.

$$Acc = \frac{n_{S \to S} + n_{FS \to S} + n_{FL \to L} + n_{L \to L}}{N_F + N_S + N_L} --- (11)$$

$$SR = \frac{n_{S \to S} + n_{FS \to S}}{N_S + N_{FS}} --- (12)$$

$$HR = \frac{n_{L\to L} + n_{FL\to L}}{N_L + N_{FL}} --- (13)$$

$$SP = \frac{n_{S\to S} + n_{FS\to S}}{n_{S\to S} + n_{FS\to S} + n_{FL\to S} + n_{L\to S}} --- (14)$$

$$HP = \frac{n_{L\to L} + n_{FL\to L}}{n_{L\to L} + n_{FL\to L} + n_{FS\to L} + n_{S\to L}} --- (15)$$

5.7.4 Results

The experiments were performed by using 10-sub datasets. Both test sets of Enron spam and Spam Assassin were randomly divided into 10 partitions with the original ratio of the set.

VS.net 2008 was used to develop the Naïve Bayesian algorithm and the GIHT-Bayesian algorithm. Particularly, the GIHT was derived from ODP [16] and WordNet [17]. SVM approaches performed the text classification by using the cosine kernel of LIBSVM [5].

The goal was to compare the GIHT- Naïve Bayesian with the SVMs on the spam filtering task. The results with the best configuration for each classifier on each dataset were shown in Tables 5.1 and 5.2. Tables 5.1 and 5.2 summarize the average values of those results over 10 subsets on Spam Assassin and Enron3, respectively.

These results show that the GIHT-Bayesian achieves higher performance than the Naïve Bayesian and the SVMs algorithm overall. But it gets the same results as the Naïve Bayesian and worse results than the SVMs when handling only absolute spam and absolute legitimate E-mails. The reason of this result was that the fuzzy E-mail was

Table 5.1 Bayesian vs. SVM vs. F-B on Spam assassin

E-mail Set	Performance Benchmarks	Methods		
		Bayesian	SVM	FHT-Bayesian
absolute spam and absolute legitimate	Accuracy	97.00%	97.38%	97.00%
	Spam Recall	95.50%	95.75%	95.50%
	Spam Precision	98.45%	98.97%	98.45%
	Ham Recall	98.50%	99.00%	98.50%
	Ham Precision	95.63%	95.88%	95.63%
fuzzy E-mail	Accuracy	87.00%	87.50%	95.00%
	Spam Recall	59.62%	65.38%	84.62%
	Spam Precision	38.75%	42.50%	68.75%
	Ham Recall	66.89%	68.92%	86.49%
	Ham Precision	82.50%	85.00%	94.12%
Overall	Accuracy	95.00%	95.40%	96.60%
	Spam Recall	88.32%	89.68%	93.32%
	Spam Precision	86.51%	87.68%	92.51%
	Ham Recall	92.18%	92.98%	96.10%
	Ham Precision	93.00%	93.70%	95.33%

used in spam filtering processing, and the GIHT-Bayesian obtained quite better score than the other two methods for the fuzzy E-mails. Theoretically, the GIHT-Bayesian dynamically classifies the fuzzy E-mails according to each person, but the Naïve Bayesian and the SVM treat fixed E-mails as spam or as legitimate E-mail to every person. Actually, because all the fuzzy E-mails were picked by 10 different persons, the sets of spam and legitimate E-mails in the set of fuzzy E-mails are different for each person, the intersections may exist. Statistically, 200 fuzzy E-mails were used in each E-mail set, there were averaged 148 legitimate E-mails to each user in the selected fuzzy Email set of Spam assassin; and there are averaged 26 legitimate E-mails to each user in the selected fuzzy E-mail set of Enron3; so by using equation: $(\tau*10-200)/10$ (τ is the number of average legitimate E-mails), there are averagely 128 and 6 intersect legitimate E-mails regarding to Spam assassin and Enron3, respectively. Additionally, the reasons of the more legitimate E-mails were in Spam assassin was that the invited user selected 157 fuzzy E-mails from 'hard Ham' set. The overall accuracy of the fuzzy E-mail classification of the Spam assassin was higher than that of Enron3.

Table 5.2 Bayesian vs. SVM vs. F-B on Enron3

E-mail Set	**Performance Benchmarks**	**Methods**		
		Bayesian	**SVM**	**FHT-Bayesian**
absolute spam and absolute legitimate	Accuracy	97.00%	97.38%	97.00%
	Spam Recall	95.50%	95.75%	95.50%
	Spam Precision	98.45%	98.97%	98.45%
	Ham Recall	98.50%	99.00%	98.50%
	Ham Precision	95.63%	95.88%	95.63%
fuzzy E-mail	Accuracy	65.00%	68.00%	86.00%
	Spam Recall	59.62%	65.38%	84.62%
	Spam Precision	38.75%	42.50%	68.75%
	Ham Recall	66.89%	68.92%	86.49%
	Ham Precision	82.50%	85.00%	94.12%
Overall	Accuracy	90.60%	91.50%	94.80%
	Spam Recall	88.32%	89.68%	93.32%
	Spam Precision	86.51%	87.68%	92.51%
	Ham Recall	92.18%	92.98%	96.10%
	Ham Precision	93.00%	93.70%	95.33%

CHAPTER 6

VISUAL RECOMMENDATION SYSTEM FOR SEARCH ENGINE

6.1 Introduction

Most of current search engines lead a user to a page that only allows the user to input key words. However, the blast of information on the Internet yields new requirements that people expect to search broad personalized information quickly and conveniently. For example, a mobile device user wants to easily input a cluster of words rather than input key words one by one on a small screen. The disabilities are eager to search unabridged information by using simplest Graphical User Interface (GUI). Thus, we propose a novel visual recommendation system for a new search engine. With the new GUI, the user can just click select or unselect element on the graph to form his or her personal query for searching. The new recommendation system involves several concepts: granular word, collaborative intelligence, and personal information. We have developed a framework which implements the new recommendation algorithm. A real demonstration was also created to show the efficiency of the system.

6.2 The Design of the Recommendation System

Figure 6.1 shows the details of the recommendation system. The system consists of users, GIHT, Collaborative system, and the recommendation GUI. The user provides granular words and personal information. The granular words are forwarded into the GIHT to perform term expansion to match historical queries. The personal information is forwarded into the collaborative system; the collaborative system will find out which group the user belongs to and match the granular words with historical queries. All the results will be forwarded into the recommendation GUI. The recommendation GUI will compute the input and present the result as a recommend graph.

6.2.1 Collaborative Intelligence System

Collaborative Intelligence has the ability to create, contribute to, and harness the power within networks of people; it enables participants to coordinate their actions The collaborative sub system plays a crucial role as shown in Figure 6.2, it realizes three functionalities: (1) collecting the information of user behavior (Which web link the user clicked? How long the user stays on one link? What actions user has taken? etc.) and

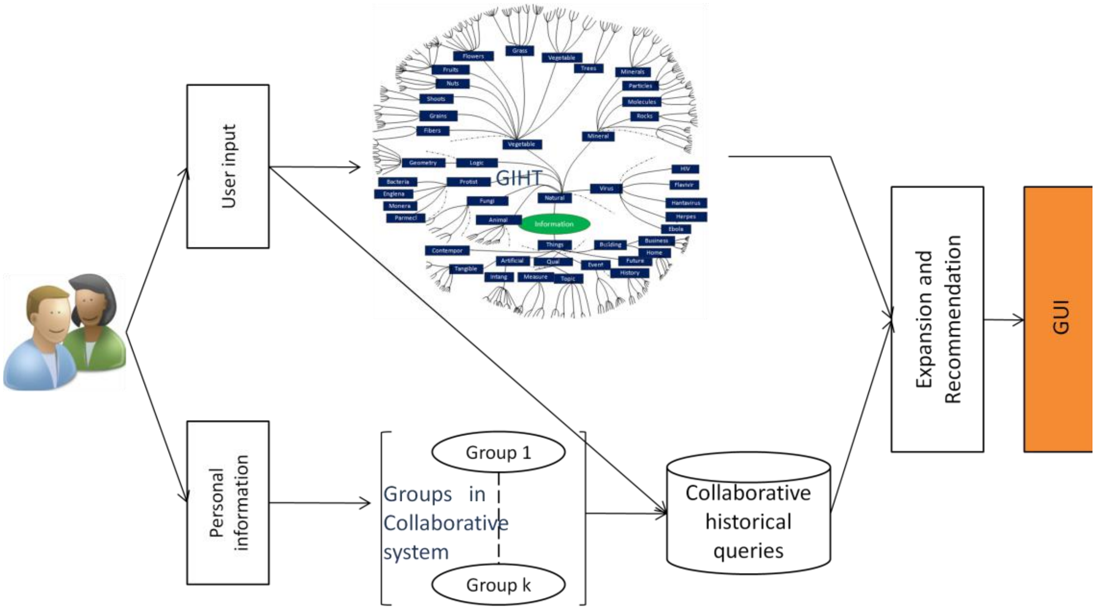


Figure 6.1 Graphical Recommendation Systems

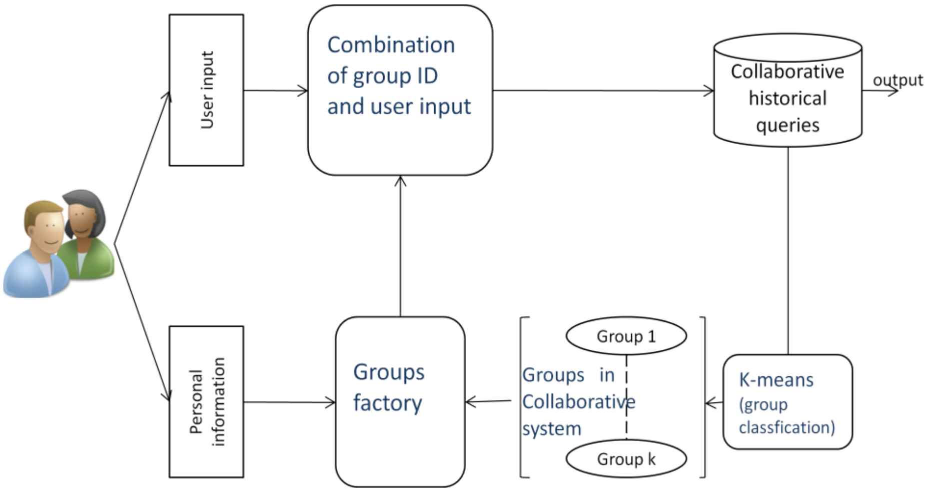

User input
Combination of group ID and user input
Collaborative historical queries
output
Personal information
Groups factory
Group 1
Groups in Collaborative system
Group k
K-means (group classfication)

Figure 6.2 Collaborative Intelligence Systems

personal information, (2) Classify users into different groups based on users' information and their historical behavior, also perform group matching for certain user, and (3) Search the queries for the granular words.

6.2.1.1 Group Classification in Collaborative Intelligence System

In the collaborative intelligence sub system, groups support the personalization of the user's query. The system collects both the users' information and the historical behavior, all the groups were classified based on the information. Moreover, the k-means method was used to classify the users. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters. We defined k centroids according to numbers of different types of information collected by the system. This algorithm aims at minimizing an objective function as given by formula (16):

$$J = \sum_{j=1}^{k} (\sum_{i=1}^{n} [x_i^{(j)} - c_j]^2) --- (16)$$

Where $[x_i^{(j)} - c_j]^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j.

The algorithm is composed of the following steps:

(1)Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.

(2)Assign each object to the group that has the closest centroid.

(3)When all objects have been assigned, recalculate the positions of the K centroids.

(4)Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm is used in group classification in the collaborative intelligence system.

6.2.1.2 Group Matching and Query Matching

The group factory (The abstract factory design pattern is used to generate different groups) is responsible for the group matching as shown in Figure 6.2. The matrix matching was used in group factory, which means the group factory creates a matrix depending on the user's personal information; each element in the matrix is one feature of user. Because the groups has been classified and historical queries has been saved in different tables according to each group, a user just need to be found which group it belongs to, then the recommendation query can be extracted from the proper historical query table. To mach a user to one group, the standard deviation is used to compute a result s_i for each group, it is illustrated as:

$$\left(\begin{bmatrix} u[1,1] & \cdots & u[1,k] \\ \vdots & \ddots & \vdots \\ u[k,1] & \cdots & u[k,k] \end{bmatrix} - \begin{bmatrix} g[1,1] & \cdots & g[1,k \\ \vdots & \ddots & \vdots \\ g[k,1] & \cdots & g[k,k] \end{bmatrix} \right)^2 = S_i --- (17)$$

As a set of values <s_1, s_2, s_3, ... s_n> calculated by equation (17), the minimum s_i is used to find the group ID. The queries table of this group ID will be searched; and all the matched queries are returned and forwarded to the recommendation GUI.

6.3 Query Expansion and Recommendation

Both outputs from GIHT and collaborative intelligence system are used for the query expansion and the recommendation unit.

To combine the result from GIHT and the result from historical database, the frequency is used to measure the recommendation ranks and weights of the related terms. For example, for a user input word 'cancer', the GIHT expands it to a granular word g{cancer; breast,.....eye}, the historical database outputs {"breast cancer", "bone cancer"}, and then the recommendation algorithm computes the frequency, and finally the order is recommended as {breast, bone, ...eye} related to 'cancer'.

6.4 Ranking algorithm

Since the output of the recommendation system is a personalized granular word, the ranking algorithm seeks results to match the personalized granular word.

Currently, a simple ranking algorithm is proposed to handle the searching results. At first, for every result, a score is computed by equation (18),

$$\delta 1=(1-\frac{1}{V_t})^{\left(n_s/n_m\right)}---(18)$$

where n_s is the total number of nodes in personal granular word, n_m is the number of the matching nodes in a result, V_t is the visiting time of the result link. The bigger the V_t is, the bigger δ1 is. The more matching nodes, the bigger δ1 is. After calculating the scores for all the returned links, the arbitrary sorting algorithm is used to sort the scores.

6.5 Experiments and Evaluation

6.5.1 Software Environment

According to the requirement of the graphical model, the interface was realized by the new technology released by Microsoft named 'silverlight' [13], associated with ASP.NET. The 'silverlight' currently can work cross windows, Linux [14], Mac [15] and smart phone OSs [16] [17], etc. Furthermore, My-SQL was used to store the GIHT. Moreover, the application only focuses on health domain.

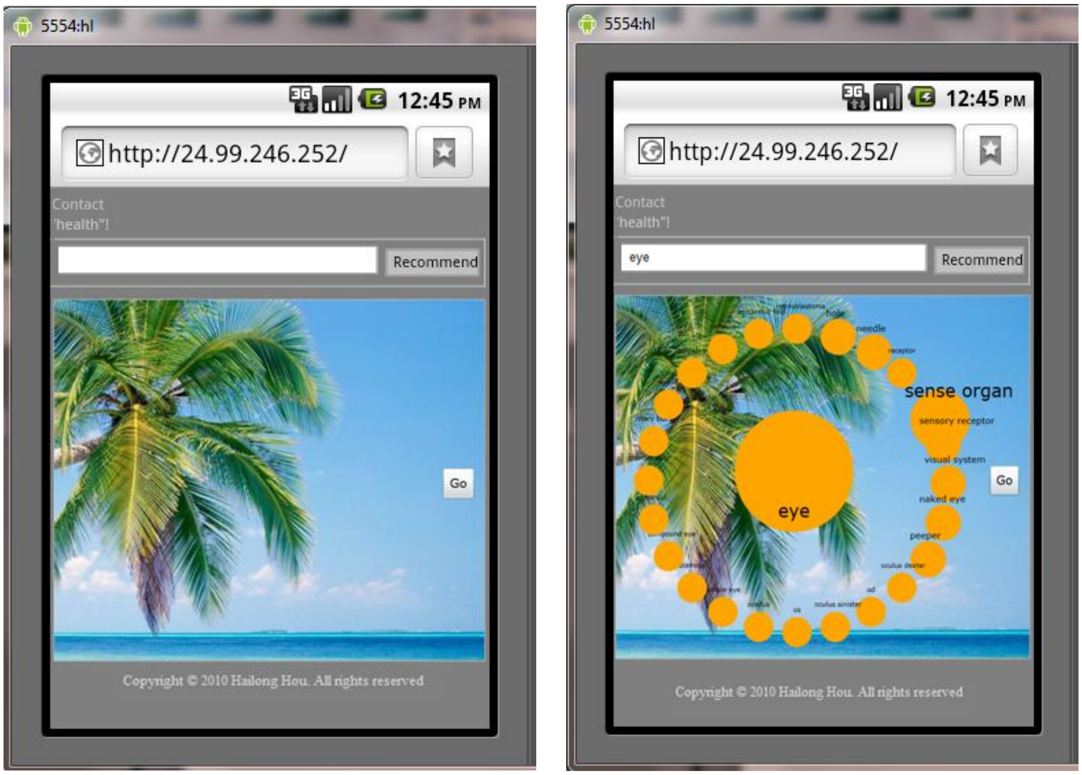
Android emulator associates with MonoDroid (MonoDroid is a development stack for using C# and core .NET APIs, and silverlight to develop Android-based applications) were used to test the application.

Google mobile and Bing mobile were selected to compare to the application in query recommendation part and search result part respectively.

6.5.2 Interface

The convenience of input is shown through the GUI. Figure 5.5.2 shows the GUI, there are two steps for the input, 1) a user inputs granular words and clicks on 'Recommend' button to obtain a query recommendation; 2) the user clicks on the 'Go' button to get the search result.

The application is designed especially for the mobile devices' users and disabilities.



(a) Graphical Interface

(b) One word input

Figure 6.3 Visual recommendation system Interface

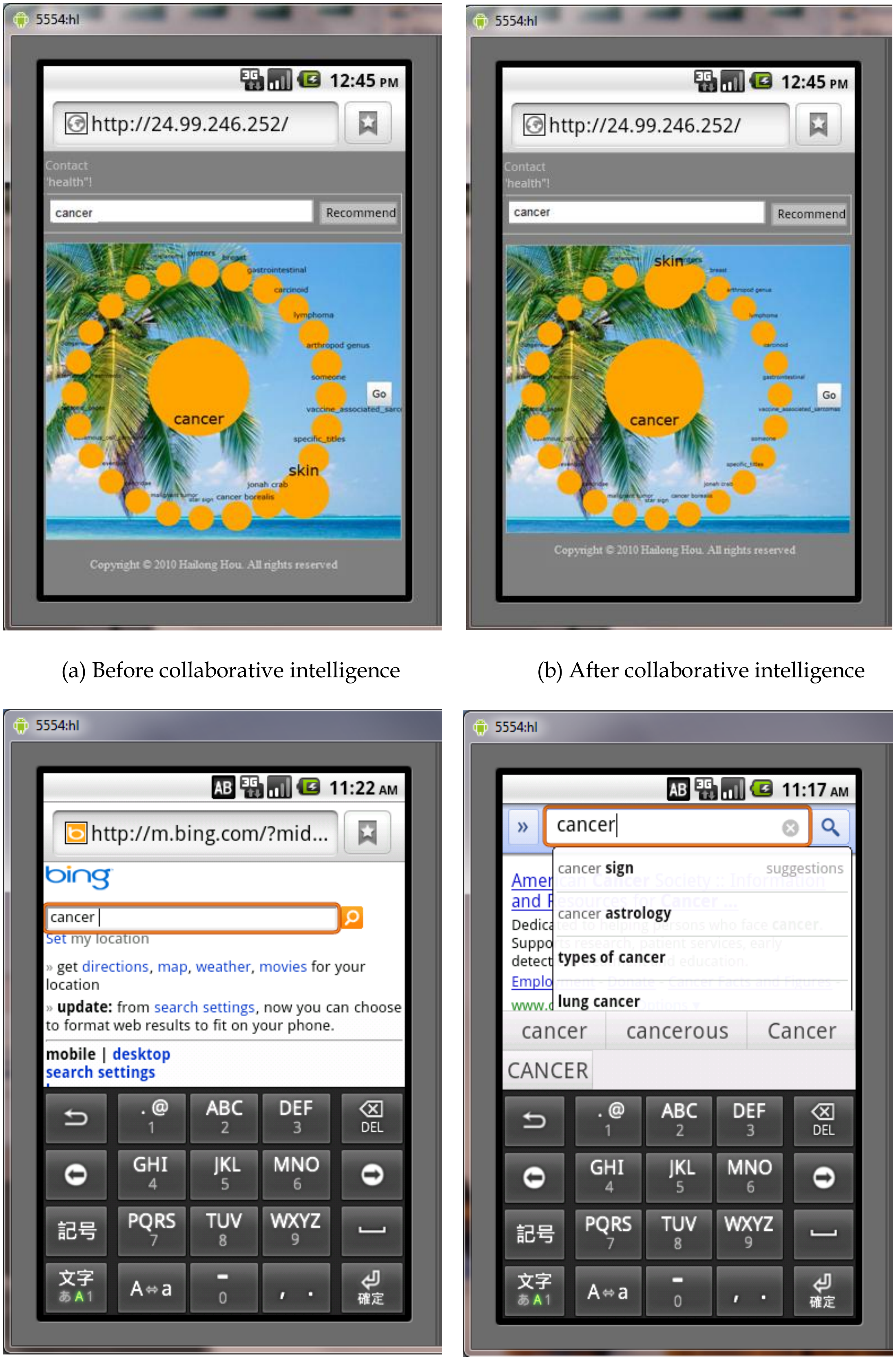
The application displays a graph of granular words. And when the mouse moves over on the related concept, it will be enlarged for the user to easily select a personalized granular word as shown in the Figure 6.3(b).

6.5.3 Recommendations

6.5.3.1 Single Word Recommendation

Single word recommendation is most frequently used by a mobile device user. Figures 6.5.3.1 (a) and (b) shows the recommendation result of query 'cancer'. Figure 6.5.3.1 (a) shows the recommendation without collaborative system, the graph presents the original granular word and the user selected relative 'skin'. Figure 6.5.3.1 (b) shows the recommendation with collaborative system, the relative 'skin' is moved to the starting position clock wisely since other members in the same group has made their selections of word 'skin' previously.

The recommendation results of Bing mobile and Google mobile are depicted in Figures 6.5.3.1 (c) and (d). Bing mobile does not provide any recommendations, and Google mobile provides recommendations. However, there are only four recommendations available, and the user can only select one suggestion each time. In the visual recommendation system, a user can select more granular words. Thus, the visual recommendation system is more convenient than Bing mobile and Google mobile in query recommendation for single words.



Set my location

» get directions, map, weather, movies for your location

» **update:** from search settings, now you can choose to format web results to fit on your phone.

mobile | **desktop**

search settings

cancer **sign**

cancer **astrology**

types of cancer

lung cancer

(a) Before collaborative intelligence

(b) After collaborative intelligence

(c) Bing

(d) Google

Figure 6.5.3.1 Single word recommendation

6.5.3.2 Multiple Words Recommendation

As shown in Figures 6.5.3.2 (a) and (b), the input granular words are drawn on the interface separately in GIHT. For example, the granular words 'breast' and 'cancer' are drawn on the interface of Figure 6.5.3.2 (a) because both of them are granular words, and the interface displays them with different colors. Figure 6.5.3.2 (b) shows the recommendation with collaborative intelligence, each word shows the relatives based on the historical query in this group.

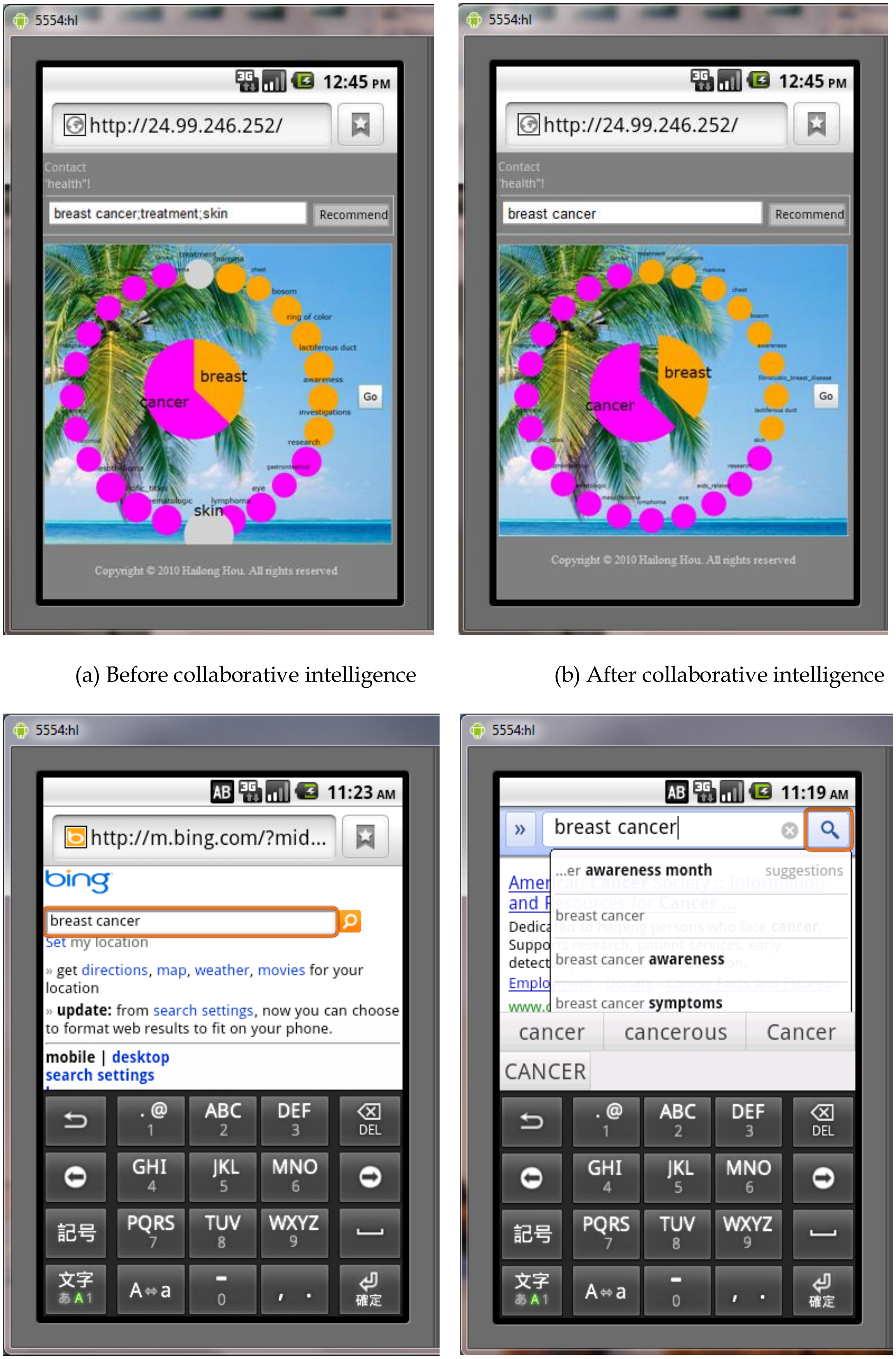
Bing mobile still does not recommend anything, and Google mobile only recommends relatives to the whole user input.

6.5.3.3 Personalized Query

Google mobile suggests historical queries by ranking all of them. However, oure system presents ranking queries according to groups classified by collaborative intelligence, and provides the options for a user to form a personalized query.

6.5.3.4 Performance Evaluation

3 groups of unknown users (10 medical researchers, 10 patients and 10 general users) were invited to generate query set_1 which contains 150 queries in health domain; query set_2 contains 150 queries generated by the same users. These two sets were input into the visual recommendation system, Bing mobile and Google mobile. The results are shown in Table 6.1, Table 6.2, and Table 6.3 for research group, patient group,



breast cancer;treatment;skin

breast cancer

breast cancer

Set my location

» get directions, map, weather, movies for your location

» update: from search settings, now you can choose to format web results to fit on your phone.

mobile | desktop

search settings

breast cancer

Copyright © 2010 Hailong Hou. All rights reserved

(a) Before collaborative intelligence

(b) After collaborative intelligence

(c) Bing

(d) Google

Figure 6.5.3.2 Multiple words recommendation

and general group respectively. For each group, the visual recommendation system targeting more queries than Bing mobile and Google mobile; and the query targeting obviously increased after set_1 inputted into collaborative intelligence system. The average Table 6.4 shows the visual recommendation system targets 72.7% of the queries, Bing mobile has no recommendations, and Google mobile matches 52.7% of the queries. For query set_2, the visual recommendation system targets 83.3% of the queries, Bing mobile still has no recommendations, and Google mobile targeted 50.7% queries. Thus, the visual recommendation system is better than Bing mobile and Google mobile.

Table 6.1 Comparison of Different application of researcher group

R_Set\Application	Visual recommendation	Bing mobile	Google mobile
Researcher Set_1	68.0%	0.0%	50.0%
Researcher Set_2	78.0%	0.0%	52.0 %

Table 6.2 Comparison of Different application of patient group

P_Set\Application	Visual recommendation	Bing mobile	Google mobile
Patient Set_1	76.0%	0.0%	56.0%
Patient Set_2	88.0%	0.0%	64.0%

Table 6.3 Comparison of Different application of general group

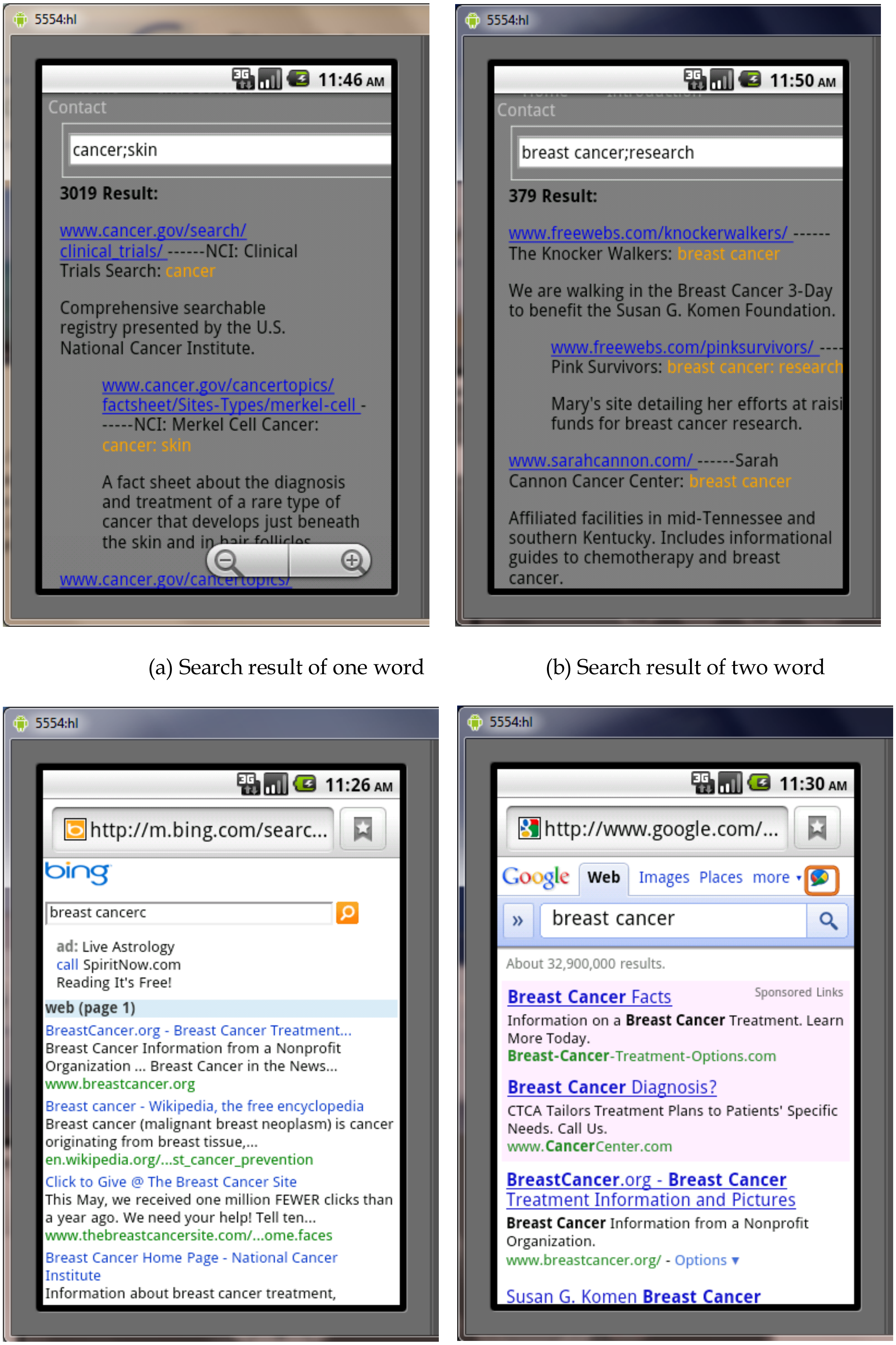
G_Set\Application	Visual recommendation	Bing mobile	Google mobile
General Set_1	74%	0.0%	52.0%
General Set_2	84.0%	0.0%	48.0%

Table 6.4 Comparison of Different application on average value

Set\Application	Visual recommendation	Bing mobile	Google mobile
Set_1	72.7%	0.0%	52.7%
Set_2	83.3%	0.0%	50.7%

5.5.4 Search Result

The ranking algorithm (18) was used to process the returned results. In the application, each result was returned as a cluster; the first link in a cluster was related to the root word, other links corresponded to the interests the user selected. Figures 6.4.1 (a) and (b) show the screen shot of the result regard to granular word 'Cancer' with selected relative 'skin' and multiple granular word 'breast cancer' with selected relatives 'research'.



5554:hl

11:46 AM

Contact

cancer;skin

3019 Result:

www.cancer.gov/search/clinical_trials/ ------NCI: Clinical Trials Search: cancer

Comprehensive searchable registry presented by the U.S. National Cancer Institute.

www.cancer.gov/cancertopics/factsheet/Sites-Types/merkel-cell ------NCI: Merkel Cell Cancer: cancer: skin

A fact sheet about the diagnosis and treatment of a rare type of cancer that develops just beneath the skin and in hair follicles

www.cancer.gov/cancertopics/

(a) Search result of one word

5554:hl

11:50 AM

Contact

breast cancer;research

379 Result:

www.freewebs.com/knockerwalkers/ ------The Knocker Walkers: breast cancer

We are walking in the Breast Cancer 3-Day to benefit the Susan G. Komen Foundation.

www.freewebs.com/pinksurvivors/ ---- Pink Survivors: breast cancer: research

Mary's site detailing her efforts at raisi funds for breast cancer research.

www.sarahcannon.com/ ------Sarah Cannon Cancer Center: breast cancer

Affiliated facilities in mid-Tennessee and southern Kentucky. Includes informational guides to chemotherapy and breast cancer.

(b) Search result of two word

5554:hl

11:26 AM

http://m.bing.com/searc...

bing

breast cancerc

ad: Live Astrology call SpiritNow.com Reading It's Free!

web (page 1)

BreastCancer.org - Breast Cancer Treatment...
Breast Cancer Information from a Nonprofit Organization ... Breast Cancer in the News...
www.breastcancer.org

Breast cancer - Wikipedia, the free encyclopedia
Breast cancer (malignant breast neoplasm) is cancer originating from breast tissue,...
en.wikipedia.org/...st_cancer_prevention

Click to Give @ The Breast Cancer Site
This May, we received one million FEWER clicks than a year ago. We need your help! Tell ten...
www.thebreastcancersite.com/...ome.faces

Breast Cancer Home Page - National Cancer Institute
Information about breast cancer treatment,

(c) Bing search result

5554:hl

11:30 AM

http://www.google.com/...

Google **Web** Images Places more ▾

» breast cancer

About 32,900,000 results.

Sponsored Links

Breast Cancer Facts
Information on a **Breast Cancer** Treatment. Learn More Today.
Breast-Cancer-Treatment-Options.com

Breast Cancer Diagnosis?
CTCA Tailors Treatment Plans to Patients' Specific Needs. Call Us.
www.**Cancer**Center.com

BreastCancer.org - **Breast Cancer** Treatment Information and Pictures
Breast Cancer Information from a Nonprofit Organization.
www.breastcancer.org/ - Options ▾

Susan G. Komen **Breast Cancer**

(d) Google search result

t

Figure 6.5.4.1 Search result

cancer', and Google mobile tries to list the result with relatives which our application has done in recommendation part.

6.5.4.1 Search Result Evaluation

The searching results of Set_1 and Set_2 are shown in Table 6.5, Table 6.6, and Table 6.7 for each group, the result of visual recommendation system is better than Bing mobile and Google mobile, the result of set_2 for each group is better in visual recommendation, the reason also is that there is collaborative intelligence system. On average showing in Table 6.8, for query set_1, the users can find 65.3% interests in the first 10 results of the visual search application; the users can find 12.0 % by Bing mobile and 36.0% by Google mobile in the first 10 results. For query set_2, the users can find 73.3% interests from top 10 results of the visual search application, 16.0% by Bing mobile and 38.3% by Google mobile. Therefore, the visual recommendation system is also better than Bing mobile and Google mobile.

Table 6.5 Search Result of research group

R_Set\Application	Visual recommendation	Bing mobile	Google mobile
Research Set_1	62.0%	12.0%	32.0%
Research Set_2	74.0%	14.0%	38.0%

Table 6.6 Search Result of patient group

P_Set\Application	Visual recommendation	Bing mobile	Google mobile
Patient Set_1	66.0%	10.0%	36.0%
Patient Set_2	70.0%	16.0%	34.0%

Table 6.7 Search Result of general group

G_Set\Application	Visual recommendation	Bing mobile	Google mobile
General Set_1	68.0%	14.0%	40.0%
General Set_2	76.0%	18.0%	44.0%

Table 6.8 Search Result on average

Set\Application	Visual recommendation	Bing mobile	Google mobile
Set_1	65.3%	12.0%	36.0%
Set_2	73.3%	16.0%	38.3%

6.5.5 Analysis

The visual recommendation system has two advantages: (1) it is more suitable for a mobile device holder and a disabled person since it allows the user to input granular words and interact with the GUI to form a personal search, and (2) the user can easily obtain global interested information from limited clusters.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

The new concept 'Granular Word' is defined, and a tree structure GIHT is proposed to organize all the granular words. Based on the granular word, a GIHT-Bayesian algorithm is created to filter E-mails by considering normal factors and abnormal factors. In the algorithm, the fuzzy E-mail is introduced to describe the uncertain E-mail that is valuable for a person but useless for another. The GIHT-Bayesian algorithm extractes words from E-mails primarily, then computes them with relative concepts extracted from GIHT to obtain a score λ of the E-mail, and finally uses the score to perform E-mail filtering. The results of the experiments indicate that the new technique can effectively classify fuzzy E-mails into either spam or ham in a personalized way. Furthermore, the algorithm of computing with granular word is developed. The granular information is retrieved from the GIHT for a word. The discrete set method is used to compress the weights of Granular Word into one level model. A predefined function f(x) combined with the standard deviation approach is used to compute the result. The experiments illustrate that the algorithm is an effective method to compute the Granular Word. Finally, the granular word is applied to the visual recommendation system merging the CWGW with collaborative intelligence and user's personal information to make the recommendation in a GUI. The mobile device holder or a disabled person can quickly form the personalized query, and then obtain the clustered result. The experiments show that computing with granular word is useful.

7.2 Future work

In the future, I will try to use cloud computing to populate the weights of GIHT; I will continue exploring user behavior analysis, collaborative intelligence and query recommendation, and will elaborate the ranking algorithm to build a general web search.

7.2.1 Mining Granular Words

Obtaining online information corresponding to the concepts is very challenging because the GIHT is huge. The Internet is a good way to obtain information and populate the database. Since ODP has already provided links for every concept, words' information can be extracted from these websites to populate the database. For instance, weather information can be obtained from links with word 'weather', such as www.weather.org; information about the law can be mined from government websites related to term 'law'; sports information can be obtained from TNT or ESPN websites which related to term 'sport', etc.

The cloud computing is becoming more and more popular to process huge data, it provides powerful distributed computing of information. As shown in Figure 7.1, the cloud computing can be used to obtain information from online services, and then feedback information to GIHT.

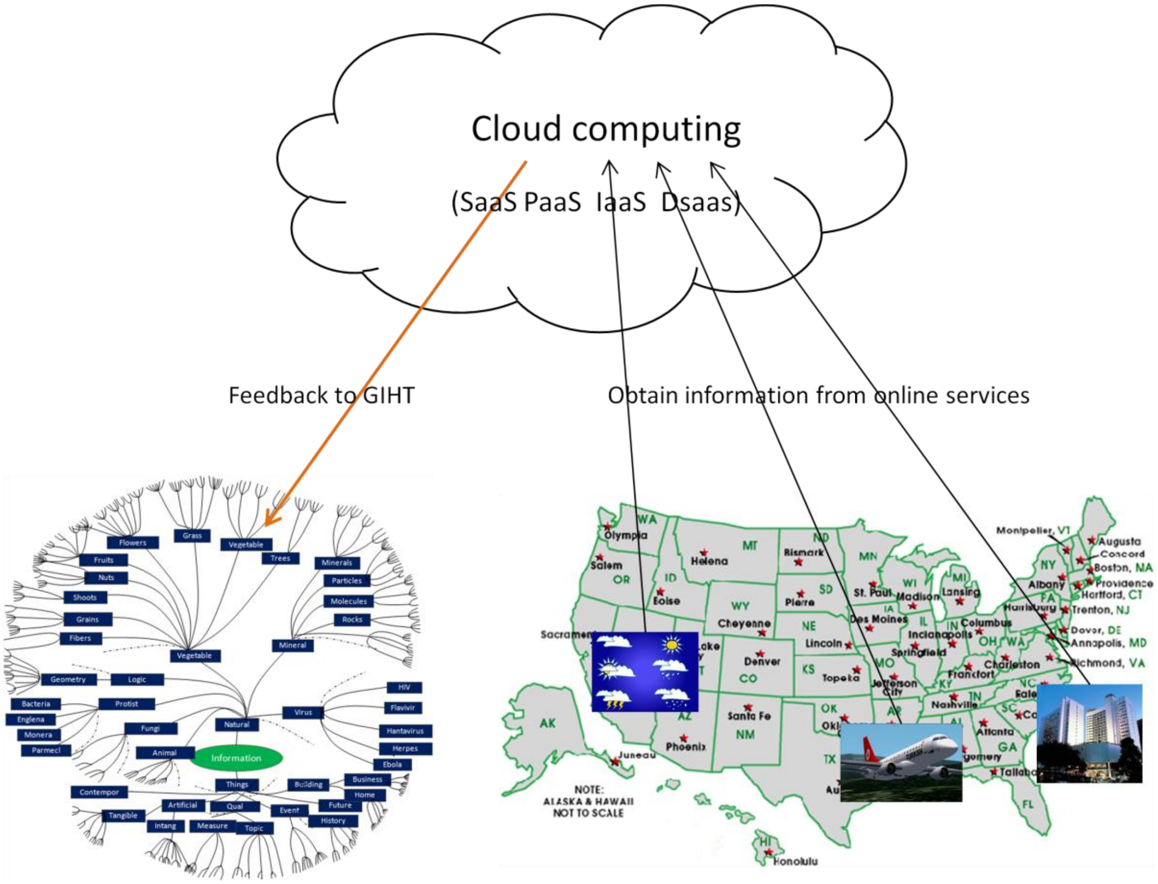


Figure 7.1 Obtain real time information by using cloud computing services

7.2.2 Search engine

The user behavior is very important for search engine, like what kind of content the user interest? Which link the user clicked? What is the personal information about the user? Base on the information, the groups can be classified to help query recommendation and result ranking. The collaborative intelligence definitely can improve the user experience of searching, I interest in its quality improvement: (1)

Group moderation and facilitation. (2)Adherence to a small set of fundamental rules related to member interaction. (3) No limits to thinking; or the promotion of creative thinking. (4) Strong group membership feedback. (5) Quality control. Ideas need to be nurtured, but the solutions should be upheld after a critical peer review. Finally, I want to keep doing some research on query recommendation on mobile devices, because mobile devices is becoming more and more popular currently, the search tool need well designed to fit the small screen and small keyboard, the query recommendation is one core part of this requirement.

CHAPTER 8

REFERENCES

[1] Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?". *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006)*, Mountain View, CA, USA, 2006.

[2] I. Androutsopoulos, J. Koutsias, and K. V. Chandrinos, "Machine Learning for Spam Detection References", *Proceedings of 2000 Workshop on Machine Learning in the New Information Age, Barcelona*, pp.9-17, May 2005.

[3] Zhan Chuan, Lu Xianliang, Hou Mengshu, Zhou Xu," A LVQ-based neural network anti-spam email approach", *ACM SIGOPS Operating Systems Review*, Vol. 39 Issue 1, pp. January 2005.

[4] J.Clark, I.Koprinska and J.Poon, "A neural network based approach to automated e-mail classification", *Proceedings of the 2003 IEEE/WIC international conference on web intelligence*, Halifax, pp. 702-705, October 2003.

[5] W.W.Cohen, Learning rules that classify E-mail, *Proceedings of 1996 AAAI Spring Symposium on Machine Learning in Information Access*, Palo Alto, pp.18-25, April 1996.

[6] L.F. Cranor, and B.A. LaMacchia, Spam, *Communications of ACM*, Vol.41, No.8, pp. 74-83, August 1998.

[7] Sculley D, Gabriel M. Wachman," *Relaxed Online SVMs for Spam Filtering", SIGIR'07*, Amsterdam, The Netherlands, July 23–27, 2007.

[8] H.Drucker, D.Wu and V.N.Vapnik, "Support vector machines for spam categorization", *IEEE Transactions on Neural networks*,Vol.10,No.5,pp.1048-1054, May 1999.

[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[10] http://www.iit.demokritos.gr/skel/i-config

[11] http://www.mxlogic.com/threat_center/

[12] http://www.paulgraham.com/better.html

[13] http://www.rhyolite.com/anti-spam/dcc/graphs/?BIG=1&resol=2y#graph1

[14] http://www.spamassassin.org/publiccorpus

[15] http://www.weather.com.

[16] http://www.we-globe.net/WebLab/Download/DmozRdf2MySQL.html

[17] http://wordnet.princeton.edu/obtain

[18] H.L. Hou, Y. Chen, R. A. Beyah and Y.-Q. Zhang, "Filtering Spam by Using Factors Hyper Trees," *Proc. of IEEE Globecom 2008 Computer and Communications Network Security Symposium (GC'08 CCNS)*, Nov. 30- Dec. 4, 2008.

[19] Androutsopoulos Ion, John Koutsias, Konstantinos V. Chandrinos, Constantine D. Spyropoulos, "An Experimental Comparison of Naïve Bayesian and Keyword-Based Anti-Spam filtering with Personal E-mail Messages." *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval 2000,* Athens, Greece July 24 - 28, 2000.

[20] R. Islam, and W. Zhou, "An adaptive Model for spamfiltering using machine learning algorithms", *7thInternational Conference on Algorithms and Architecturesfor Parallel Processing (ICA3PP)*, Hangzhou, China, June 11-14, 2007.

[21] Md. Rafiqul Islam, Wanlei Zhou and Morshed U. Choudhury, "Dynamic Feature Selection for Spam Filtering Using Support Vector Machine", *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*

[22] Graham-Cumming.J, "The spammers' compendium," *Jun.2006. Retrieved: Jun. 2006* http://www.jgc.org/tsc/.

[23] HoYu Lam, DitYan Yeung, "a learning approach to spam sender detection based on social networks", *CEAS 2007 Fourth Conference on E-mail and AntiSpam*, August 23, 2007.

[24] Islam,Md.R.;Chowdhury,M.U.;Wanlei Zhou. "An Innovative Spam Filtering Model Based on Support Vector Machine", *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05).*

[25] Vangelis Metsis, Ion Androutsopoulos, Georgios Paliouras, "Spam Filtering with Naïve Bayes- Which Naïve Bayes?", *CEAS 2006 Third Conference on Email and AntiSpam*, Mountain View, California USA, July 2728, 2006.

[26] Vikas P. Deshpande, Robert F. Erbacher, and ChrisHarris. " An Evaluation of Naïve Bayesian Anti-Spam Filtering Techniques". *Proceedings of the 2007 IEEE, Workshop on Information Assurance*, United States Military Academy, West Point, NY 20-22 June 2007.

[27] Graham, Paul (2002). "A Plan for Spam." *http:// www.paulgraham.com/spam.html*, August 2002. 8.

[28] Patrick Pantel and Dekang Lin. "SpamCop-- A Spam Classification & Organization Program." *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*.

[29] ZuCFikar Ramzan, Candid W¨uest, "Phishing Attacks: Analyzing Trends in 2006", *CEAS 2007 Fourth Conference on E-mail and AntiSpam*, August 23, 2007.

[30] Sahami, S. Dumais, D. Heckerman, E. Horvitz (1998). "A Bayesian approach to filtering junk E-mail." *AAAI'98 Workshop on Learning for Text Categorization.*

[31] Spamhaus. *The definition of spam,* Jan. 2007. Retrieved: Jan 2007 http://www.spamhaus.org/definition.html.

[32] Wikipedia. *Spam (electronic)*, Jan. 2007. Retrieved: Jan. 2007 http://en.wikipedia.org/wiki/E-mail_spam.

[33] Zadeh, L.A. (1965). "Fuzzy sets". *Information and Control* 8 (3): 338-353.

[34] Zadeh, L.A., "Fuzzy logic = computing with words", *Fuzzy Systems, IEEE Transactions on Volume 4, Issue 2*, May 1996 Page(s):103 - 111Digital Object Identifier 10.1109/91.493904.

[35] Stuart H. Rubin, "Computing with Words," *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, Vol. 29, No, 4, August 1999.

[36] Fei-Yue Wang, Yuetong Lin, and James B. Pu, "Linguistic dynamic systems and computing with words for complex systems". *Systems, Man, and Cybernetics, 2000 IEEE International Conference on Volume 4*, 8-11 Oct. 2000 Page(s):2399 - 2404 vol.4 Digital Object Identifier 10.1109/ICSMC.2000.884350

[37] Mendel, J.M., " The perceptual computer: an architecture for computing with words," *Fuzzy Systems, 2001. The 10th IEEE International Conference on Volume 1*, 2-5 Dec. 2001 Page(s):35 - 38 Digital Object Identifier 10.1109/FUZZ.2001.1007239.

[38] Mendel, J.M., "Perceptual Reasoning: A New Computing with Words Engine," *Granular Computing, 2007. GRC 2007. IEEE International Conference* on 2-4 Nov. 2007 Page(s):446 - 446 Digital Object Identifier 10.1109/GrC.2007.55.

[39] Mendel, J.M.; Wu, D, " Perceptual Reasoning for Perceptual Computing", *Fuzzy Systems, IEEE Transactions on: Accepted for future publication Volume PP, Forthcoming,* 2003 Page(s):1 - 1 Digital Object Identifier 10.1109/TFUZZ.2008.2005691.

[40] Mendel, J.M.; "Computing with Words: Zadeh, Turing, Popper and Occam," *Computational Intelligence Magazine, IEEE Volume 2*, Issue 4, Nov. 2007 Page(s):10 - 17 Digital Object Identifier 10.1109/MCI.2007.9066897.

[41] M. Ying, "A Formal Model of Computing with Words", *IEEE Trans. Fuzzy Syst.*, Vol. 10, pp.640-652, Oct. 2002.

[42] D. Qiu and H. Wang, "A probabilistic model of computing with words", *J. Comp. Syst. Sci.*, Vlo.70, pp.176-200, 2005.

[43] H. Wang and D. Qiu, "Computing With Words via Turing Machines: A Formal Approach", *IEEE Trans. Fuzzy Syst.*, Vol. 11, pp.742-753, Dec. 2003.

[44] Zdzislaw Pawlak, "Rough Sets: Theoretical Aspects of Reasoning about Data", *Kluwer Academic Publishers*, Norwell, MA, 1992.

[45] Yong Chen, Kwok-Ping Chan, "Using data mining techniques and rough set theory for language modeling", April 2007, *Transactions on Asian Language Information Processing (TALIP)* , Volume 6 Issue 1

[46] Michael D. Coovert, Dawn Riddle, Linda R. Elliot, Samuel G. Schiflett, "Using rough sets to determine construct importance in a dynamic HCI environment", April 2000, CHI '00: *CHI '00 extended abstracts on Human factors in computing systems*.

[47] Grzymala-Busse, J. W. Knowledge acquisition under uncertainty--A rough set approach.J, *lntel. Rob. Syst.* 1, 1 (1988), 3-16.

[48] Songbo Tan, Yuefen Wang, Xueqi Cheng, "Text Feature Ranking Based on Rough-set Theory", November 2007,WI '07: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence.*

[49] Chi Lang Ngo, Hung Son Nguyen," A Method of Web Search Result Clustering Based on Rough Sets", September 2005, WI '05: *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence.*

[50] P. Das-Gupta," Rough sets and information retrieval", May 1988, SIGIR '88: *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval.*

[51] Hernandez, G. Bobadilla, L. Sanchez, O. "A genetic word clustering algorithm", *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. On page(s): 1075 - 1080 Vol. 2

[52] Paolo Allegrini, Simonetta Montemagni, Vito Pirrelli, "Learning word clusters from data types", July 2000, *Proceedings of the 18th conference on Computational linguistics - Volume 1.*

[53] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, Yoad Winter, "Distributional word clusters vs. words for text categorization", March 2003, *The Journal of Machine Learning Research* , Volume 3.

[54] Hirofumi Yamamoto, Shuntaro Isogai, Yoshinori Sagisaka, "Multi-Class Composite N-gram language model for spoken language processing using multiple word clusters", July 2001, *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*.

[55] Wen Wang; Vergyri, D. "The Use of Word N-Grams and Parts of Speech for Hierarchical Cluster Language Modeling", Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.

[56] Bellegarda, J.R. "Latent semantic mapping [information retrieval]", *Signal Processing Magazine, IEEE Volume 22,* Issue 5, Sept. 2005 Page(s): 70 - 80

[57] Bellegarda, J.R.; Butzberger, J.W.; Yen-Lu Chow; Coccaro, N.B.; Naik, D. "A novel word clustering algorithm based on latent semantic analysis", *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings.*, 1996 IEEE International Conference on Volume 1, 7-10 May 1996 Page(s):172 - 175 vol. 1

[58] Chung-Hong Lee; Hsin-Chang Yang, "Text mining of multilingual corpora via computing semantic relatedness", S*ystems, Man and Cybernetics, 2002 IEEE International Conference on* Volume 5, 6-9 Oct. 2002 Page(s):5 pp. vol.5

[59] John Lamping, Ramana Rao and Peter Pirolli, "A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies". *Proc. ACM Conf. Human Factors in Computing Systems*, CHI: 401–408, ACM.

[60] Lotfi A. Zadeh, "From Computing with Numbers to Computing with Words - From Manipulation of Measurements to Manipulation of Perceptions", *Int. J. Appl. Math. Comput. Sci.*, 2002, Vol.12, No.3, 307–324.

[61] Yahoo Inc. www.yahoo.com.

[62] Google. www.google.com.

[63] Microsoft Bing. www.bing.com

[64] www.ask.com

[65] AOL LLC. www.aol.com

[66] www.silverligh.net

[67] www.kartoo.com

[68] Http://www.pcworld.com/article/152399/microsoft_pushes_for_silverli ght_on_g 1_and_iphone.html

[69] Amato, A. Di Lecce, V. Piuri, V. "A New Graphical Interface For Web Search Engine", *Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2007. VECIMS 2007. IEEE Symposium on.

[70] Patrick Min, John A. Halderman, Michael Kazhdan, Thomas A. Funkhouser, "Early Experiences with a 3D Model Search Engine", *Web3D '03: Proceedings of the eighth international conference on 3D Web technology*.

[71] Tarik Filali Ansary, Jean-Phillipe Vandeborre, Mohamed Daoudi, “3D-Model Search Engine from Photos”, *Proceedings of the 6th ACM international conference on Image and video retrieval* .

[72] Amato, A. Di Lecce, V. Piuri, V. “A New Graphical Interface For Web Search Engine”, *Virtual Environments, Human-Computer Interfaces and Measurement Systems,* 2007. VECIMS 2007. IEEE Symposium on.

[73] Ting Chen ; Wei-Li Han ; Hai-Dong Wang ; Yi-Xun Zhou ; Bin Xu ; Bin-Yu Zang ; "Content Recommendation System Based on Private Dynamic User Profile",*Machine Learning and Cybernetics*, 2007 International Conference on Date:19-22 Aug. 2007

[74] Shishehchi, S. ; Banihashem, S.Y. ; Zin, N.A.M. ; " A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system", *Information Technology (ITSim)*, 2010 International Symposium in Date:15-17 June 2010.

[75] Wu Bing ; Wu Fei ; Ye Chunming ;" Personalized recommendation system based on multi_agent and rough set", *Education Technology and Computer (ICETC)*, 2010 2nd International Conference on 22-24 June 2010.