# Personality Trait Generation using Web Search Query Log

Deeshen Shah Shreya Bastikar Dhruven Vora Bhagyashree Deokar

## ABSTRACT

The involvement of Internet technology into the day-to-day life of every individual is very much evident from the fact that the number of users accessing World Wide Web's massive source of information has been increasing exponentially. Access to such enormous amount of information is prominently facilitated with the help of Search queries which define the user's way of querying the massive source of information. Our paper aims at finding the personality traits of the user by analyzing his/her pattern of web access via search queries stored in web search query logs. User's pattern of access present in the web search query log contains user's trend of accessing information (majorly search queries) and his order and preferences of navigating through the Web. In our system, we use this information to create a user profile, which will then be refined and processed using Probabilistic approach whose results will infer the personality traits of the user. Each of these traits comprises of a set of categories that are assigned to the user based on the analysis and processing of count of these categories, associated with the user's search query log.

## General Terms

Data Mining, Artificial Intelligence.

## Keywords

Bayesian Classification, Data Mining, Trait Realization System, User Classification.

## 1. INTRODUCTION

Personality traits can be used to define the attitude of a human. Understanding their traits can help us to discover their mindset such as technical minded, business oriented, bibliophile, shopaholic, spiritual, etc. Many organizations would like to have such kind of information for a particular category of web users for different reasons viz. understanding the users' mindset for providing personalized interface/navigation, customized information content, special deals to valued online customers, to provide "Human Face" to the web, etc. Our paper explains one such approach of finding the personal traits of users by means of various data mining techniques. The elaboration of the entire flow and working of the system has been presented in this paper. Algorithm required for the same has also been described.

## 2. EXISTING SYSTEMS

### 2.1 Trust-Aware Recommender Model Based on Profile Similarity

Jingyu SunP, Xueli YuP, Xianhua LiP and Zhili Wu mentioned in [1] have taken into account, web history of two different users. These two different types of users are common users and expert users. Common users have little search experiences and domain knowledge. Expert users are apt at using proper search keyword for gaining required information because of their domain knowledge. Hence, their model relies on the search keywords used by expert users. Web pages visited by the expert users from the result of search are also considered as more trustworthy as compared to the pages visited by common users. Based on the similarity of search keyword used by expert users and common users, web pages visited by expert users are recommended to the common users because that provides more appropriate results. It is implemented by using Trust aware recommendation model and Profile Similarity Computation Algorithm. Trust aware recommendation model performs two main tasks.

- To find expert users from the community
- Based on the similarity of keywords used by the expert and common user, relevant web pages visited by expert user are recommended to the common user.

Similarity between expert user's profile and common user's profile is determined with the help of proposed Profile Similarity Computation Algorithm. In our case, similarity between keyword used by the new user and users in Training Set and Knowledge Base is observed. Depending on the similarity, categories are assigned and by computing the values based on categories, traits are identified for the user.

### 2.2 Paper Classification for Recommendation on Research Support System Papits

Tadachika Ozono and Toramatsu Shintani in paper [2] categorized research paper by extracting suitable keyword, hence to make it easy for user to search research paper based on categories. Research papers are classified into two different categories viz. manually classified and automatically classified. Manually classified research paper acts as training set. Accuracy of automatically classified paper by classification system is improved by using manually classified research paper and user's feedback. Automatic classification system extracts most appropriate keywords by using feature selection technique. It also depends on Information gain matrix to pluck the words. Similarity between previously manually classified research paper and new research paper is calculated for category assignment. It is implemented by using K-nearest neighbor algorithm. In this, similarity is measured by using cosine formula which is given in that paper and in our paper; similarity between categories is computed by ontology concept. Feature selection method specified in above mentioned paper is best for identifying most suitable words, based on which classification of user into different categories and consequently into different traits will be done.

## 2.3 Analyzing and Classifying Personality Traits with Smartphone

Gokul Chittaranjan, Jan Blom, and Daniel Gatica-Perez state that personality traits are concluded for the person based on the usage of mobile data in paper [3]. Applications, call log, message log and Bluetooth in the user's smart phone are the main input parameters for the system they have proposed. In our case, input parameters are web history of the user. Personality traits are inclined towards the psychological aspect of the person. Personality traits derived by them are Extraversion, Agreeableness, Conscientiousness, Emotional Stability and Openness to Experience. Based on the majority of use of message, application, calls and Bluetooth, personal traits are identified. The manner in which they are used is also a crucial factor. For example, person who spends maximum time on application, messages and has very low count in received call log, person is categorized as not an extrovert. While the user who has very low count of missed call in call log is more extrovert. Users who do not pick up calls in public or social function are considered as more conscious. Personality traits proposed by us will be more towards the interest of the user in various fields like astrologist, technical, etc. Personality traits are allocated based on the standard regression coefficient(β) and t-statistic. Only after crossing thresholds for the above mentioned parameter, appropriate trait will be concluded about the user. In our case, user need to satisfy some threshold condition for the value computed based on probabilistic approach (considering parameters related to category) to fall into particular personality trait. In both cases, user can be mapped to multi-valued category and hence to multiple traits. They are based on one-to-many relationships.

# 3. PROPOSED SYSTEM

Our proposed system is rightly named as User Trait Generation system as it assigns personality traits to any user by analyzing their web search query log. The framework of this system comprises of the following three main subsystems: Training Set, Web Search Query Log to User History (WSQLtoUH), and Trait Realization System as shown in figure 1.

## 3.1 TRAINING SET FORMATION

Training Set consists of a list of users who are well known for their traits. Training set is used to define the threshold that decides which users are to be included in the Knowledge Base (which is later used for Naïve Bayesian Classifier Algorithm to assign traits to new users). A threshold value for each trait is determined by considering users in the training set. A set of "n" users for each of the "m" traits (included in our system) is taken into consideration.

Terms involved for determining threshold value are:

$C_i$: It is the count of number of search queries for category *i*.
$C_t$: It is summation of all $C_i$ where $i=1$ to $n$.

$W_i$: It is the weight defined as $\frac{C_i}{C_t}$ for category *i*.

$R_j$: It is the $\sum W_i$ ; where *i* represent; the categories which are belonging to trait j.
$TH_j$: It is threshold value for trait $T_j$.
$T_j$: It is the trait where $j=1 \ldots m$.
$U_k$: It is the user where $k=1 \ldots p$

For each trait, we follow the following steps to calculate threshold value:

1. Find the value of $(C_i)_k$ for each category of the user $U_k$.
2. Find the value of $(C_t)_k$ for the user $U_k$.
3. Calculate $W_i$, *i* is the category. where $W_i = \frac{C_i}{C_j}$
4. Add $W_i$ of all categories which belongs to trait $T_j$ that will give $R_j$.
5. Above 4 steps should be carried out for all users belonging to trait $T_j$ whose threshold needs to be calculated.
6. After calculating $R_j$ value for all users who belong to trait $T_j$, find average of $R_j$ values which will serve as threshold $TH_j$ for trait $T_j$.

The threshold value of each trait generated in this step is the important aspect in the process of Knowledge Base generation (which is explained in the later section) which takes place at the time of analyzing and assigning traits to a new user.

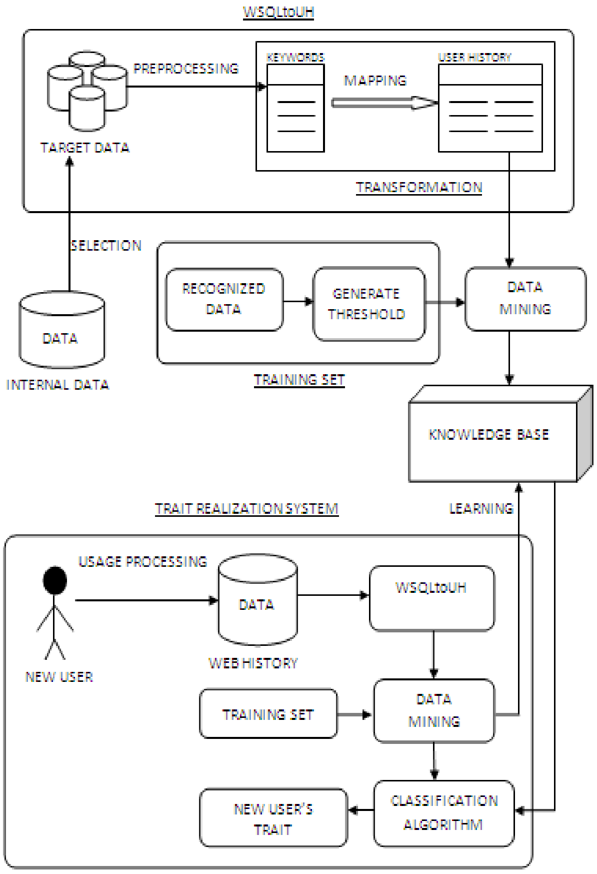
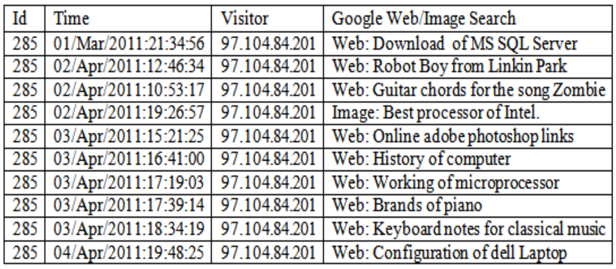


**Fig 1: Proposed (User Trait Generation) System.**

## 3.2 WSQLtoUH

**Internal data:** Internal data consists of Web Search Query Logs. Web Search Query Log (WSQL) is a log that stores the list of queries fired by the user to the search engine to search for the required information. These search queries form the basis of our system for assigning suitable traits to the user.

Queries are stored in an unstructured format. Figure 2 shows an example for WSQL which is extracted by log extractor to represent it in suitable format for analysis. WSQLtoUH deals with mapping of unstructured format (queries entered by user) into structured format suitable for generation of the Knowledge Base. The structured format consists of representation of data in the form of table's viz. CATEGORIES, TRAITS, USER_LIST, USER_TRAIT, and USER_HISTORY. Structured format of tables are shown in the figure 3.



| Id | Time | Visitor | Google Web/Image Search |
|---|---|---|---|
| 285 | 01/Mar/2011:21:34:56 | 97.104.84.201 | Web: Download of MS SQL Server |
| 285 | 02/Apr/2011:12:46:34 | 97.104.84.201 | Web: Robot Boy from Linkin Park |
| 285 | 02/Apr/2011:10:53:17 | 97.104.84.201 | Web: Guitar chords for the song Zombie |
| 285 | 02/Apr/2011:19:26:57 | 97.104.84.201 | Image: Best processor of Intel. |
| 285 | 03/Apr/2011:15:21:25 | 97.104.84.201 | Web: Online adobe photoshop links |
| 285 | 03/Apr/2011:16:41:00 | 97.104.84.201 | Web: History of computer |
| 285 | 03/Apr/2011:17:19:03 | 97.104.84.201 | Web: Working of microprocessor |
| 285 | 03/Apr/2011:17:39:14 | 97.104.84.201 | Web: Brands of piano |
| 285 | 03/Apr/2011:18:34:19 | 97.104.84.201 | Web: Keyboard notes for classical music |
| 285 | 04/Apr/2011:19:48:25 | 97.104.84.201 | Web: Configuration of dell Laptop |

**Fig 2: Internal Data (Web Search Query Log [4]) displayed by log extractor**

### 3.2.1 SELECTION

As shown in the figure 2, the Internal Data block mainly consists of the Web Search Query Log of the user. According to the requirements of the system, the search queries of each user is extracted from WSQL and stored in a format suitable for its further processing. For example, the list of search queries of a user that is extracted is shown in the Table 1.

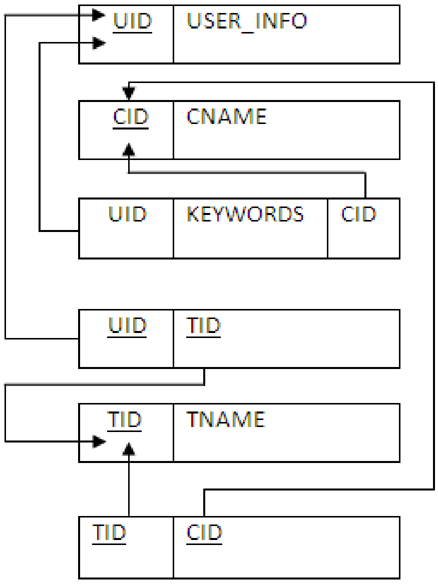


**Fig 3: Structured Format**

### 3.2.2 PREPROCESSING

In Preprocessing, extraction of keywords takes place. Different approaches like Simple Statistics Approach, Linguistics Approach, and Machine Learning have been already implemented for facilitating the process of keyword extraction [5]. These keywords can be used to understand the traits of the user. For instance, on the basis of the example given in Table 1, a small set of keyword-search queries association is shown in Table 2.

### 3.2.3 TRANSFORMATION

Transformation is a process of mapping the keywords into "categories". Categories are based on the genre of information. The information can be categorized based on various aspects such as which branch of science do they belong to, etc. This categorized information is very much crucial in deciding the user's behavioral aspect. Categories can be software, programming, computer, processor, hardware, musical instrument, music composition, genre or anything else to name a few. This kind of categorization is done using ontology (refer figure 4 for mapping keywords for the Software category). Ontology, a branch of Artificial Intelligence, is a form of knowledge representation technique which classifies the concepts according to the domain to which they belong and establishes a relationship between them. For mapping keywords to category, Linguistic Based Matcher can be used, which is specified in paper [6]. For example, Keywords like "JDBC", "MS SQL Server", etc. get mapped to category "software" as shown in figure 4. Further, category "software" will be mapped into trait "technical". Categories are identified using ontology, and are mapped into structured format as shown in Table 3.

**Table 1: Search queries of the user with UID 1**

| UID | Search Queries |
|---|---|
| 1 | Download of MS SQL Server |
| 1 | Robot Boy from Linkin Park |
| 1 | Guitar chords for the song Zombie |
| 1 | Importance of Meditation |
| 1 | Best processor of Intel. |
| 1 | Tour India in 31 days travel package. |
| 1 | Cost of guitar |
| 1 | MP3 Downloads of Nickelback. |
| 1 | Online adobe Photoshop links |
| 1 | History of computer |
| 1 | How to implement JDBC |
| 1 | Know members in Indian Parliament |
| 1 | Famous album of A. R. Rahman |
| 1 | Working of microprocessor |
| 1 | Brands of piano |
| 1 | Who is founder of facebook |
| 1 | Important parts of printer |
| 1 | Keyboard notes for classical music |
| 1 | Application of virtual reality |
| 1 | Configuration of dell Laptop |

**Table 2: Extraction of Keywords from Search Queries**

| UID | Search Queries | Keywords |
|---|---|---|
| 1 | Download of MS SQL Server | MS SQL Server |
| 1 | Robot Boy from Linkin Park | Linkin Park |
| 1 | Guitar chords for the song Zombie | Zombie |
| 1 | Importance of Meditation | Meditation |
| 1 | Best processor of Intel. | Intel |
| 1 | Tour India in 31 days travel package. | travel |
| 1 | Cost of guitar | guitar |

### 3.2.4 DATA MINING METHOD

Data Mining Method is the most important part of the system as it aids the process of assignment of trait to the user under consideration. A set of "m" traits is defined to be assigned to the user. After the mapping of keywords onto their respective categories, we need to follow the following steps:

- **STEP 1:**

Each trait consists of a set of categories associated with it. For example: Consider a Trait “Technical”, it contains various Categories associated with it, such as:

*Technical= {Software, Programming, Processors, Computer, Hardware}*

- **STEP 2:**

Now, the weight of each trait is to be calculated. Suppose we consider the trait "technical", we have to sum up the count of all the categories of the user that are associated with the trait "Technical". This sum is represented as "*X*". Also, we have to sum up the count of all the categories of that user which is denoted by "*Y*". Using the values "*X*" and "*Y*", we can calculate the weight $Z_{Technical}$ of that user using the following formula:

$$Z_{technical} = \frac{X}{Y}$$

Similarly, we can calculate the weight for every trait such as: $Z_{Tj} = X/Y$ where: $j = 1, 2, \ldots, m$ traits where,

$X = \sum Count$ of User’s Categories of $T_j$

$Y = \sum Count$ of all Categories of the User

- **STEP 3:**

After the weight of each trait is calculated, this weight is compared with the threshold value $TH_j$ (obtained from Training Set) of the respective trait. If it is equal to or exceeds $TH_j$ then that particular trait $T_i$ is assigned to the user and an entry of the same is made into the Knowledge Base. Knowledge Base is a training set which is used in Naïve Bayesian Classifier Algorithm to assign traits to new users. Now in this example, we consider the trait “Technical”, if $TH_{Technical}$ is equal to or exceeds the threshold $Z_T$, then trait “technical” is assigned to the user and is inserted into the Knowledge Base. Similarly, for the other values of $Z_{Tj}$, if they exceed their respective thresholds, those many traits are assigned to the user before inserting the result into the Knowledge Base.

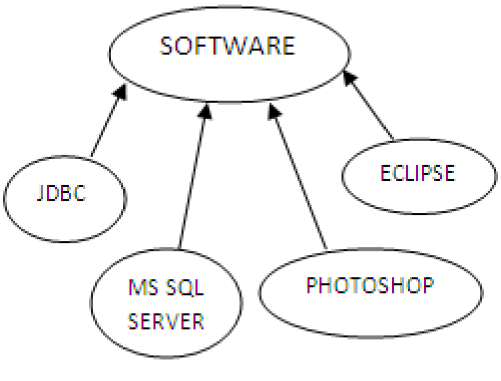


**Fig 4: Ontology diagram for mapping Keyword to Category**

## 3.3 TRAIT REALIZATION SYSTEM

Trait Realization System is used for assigning traits to the new user based on his search queries and Knowledge Base. Trait Realization System comprises of following mentioned entities:

**Table 3: User_History Mapping**

| CID | Category |
|---|---|
| 1 | SOFTWARE |
| 2 | COMPUTER |
| 3 | HARDWARE |

| UID | Keywords | CID |
|---|---|---|
| 1 | MS SQL SERVER | 1 |
| 1 | Photoshop | 1 |
| 1 | JDBC | 1 |

### 3.3.1 USAGE PROCESSING

Search queries of new users are extracted from the web search query log. These search queries are passed to the WSQLtoUH.

### 3.3.2 WSQLtoUH

As explained in the section 3.2, the search queries of the user are processed and the keywords are extracted from them. These keywords are mapped on to the defined categories. This information (categories) is then processed using the data mining method (discussed later in the section 3.3.4), and Training Set for the assignment of trait to the user.

### 3.3.3 TRAINING SET

Training set, as described in Training Set Formation (section 3.1), is used to calculate the threshold. This threshold is used to differentiate the new users to decide whether he has to be included in the Knowledge Base or not.

### 3.3.4 DATA MINING METHOD

The weight of the categories (every trait comprises a set of categories) obtained from WSQLtoUH is compared with threshold of that existing trait in the training set. If it is equal to or exceeds the threshold, then the user under consideration is assigned that trait and information related to that user is entered into the Knowledge Base. But, if it is less than the threshold, then the user is classified based on the information present in the Knowledge Base. For this, Naïve Bayesian Classifier algorithm is applied on the classifying attributes i.e. the categories of that user. After applying classification algorithm, the appropriate trait is assigned to the user.

### 3.3.5 CLASSIFICATION ALGORITHM

Users are classified using the Naïve Bayesian Classifier Algorithm, because of scarcity of information and sparseness of the weight of categories to which they belong. In order to apply Naïve Bayesian Classifier Algorithm, we require classifying attributes, a set of output classes, and training set. In our system, the classifier attributes that we have considered are categories of each user and the resulting classes of classification are the defined traits. Knowledge Base is the training set for this algorithm. The count of each category is obtained from the Data Mining Method. Now, Naïve Bayesian Classifier algorithm as specified in [7] is applied in order to deduce the trait for the new user. Now by taking each of the categories into consideration, a particular trait is assigned to the user based on the probability calculation as given below

$$P\left(\frac{T_j}{C_i}\right) = \prod P\left(\frac{C_i}{T_j}\right) * P(T_j)$$

*j=1, 2, …, m* traits.

$i = 1, 2, \ldots, n$ categories.
$C_i$ =selected category out of the set of categories of trait $T_j$
$T_j$ = trait under consideration

Similarly for remaining "*m-1*" traits, the above Naive Bayesian formula can be applied. The formula above calculates the product of the probabilities of count of the records which contain the category as "$C_i$" and belongs to the trait "$T_j$".

After calculating the probability of each trait, we assign top "*x*" traits to the new user, where *x* is average number of traits assigned to the existing user in knowledge base.

**For example:** Consider two traits out of m traits: Technical, Melomane (music-lover). Each Trait has the following categories:

**Technical**: Software, Programming, Computer, Processors, Hardware.

**Melomane**: Musical Instrument, genre, chart songs, composition.

According to the Naïve Bayesian Classification algorithm: We calculate the individual probabilities of each of the categories with respect to the trait to which it belongs. For that we shall calculate the weight of each category. Now, as we know the weight, in order to ensure accuracy, we shall calculate a range within which the weight belongs. This range will decide which users are to be considered from the knowledge base (which acts as the training set for this classification algorithm) for accuracy. The nearest neighbors are found using the minimum distance concept.

Distance: $| W_i' - W_i |$

Where $W_i'$ is weight of category *i* for user in Knowledge base. $W_i$ is the weight of category *i* for new user. This distance is calculated with all users in knowledge base who belongs to the category *i*. The distances which are having lowest value are selected as minimum distance. The users which are having minimum distance are selected as nearest neighbor.

After the nearest neighbors are found, we calculate the posterior probability $P\left(\frac{Ci}{Tj}\right)$. For calculating this probability, only those records are considered which satisfy the following conditions

- They belong to the trait $T_j$
- They contain the category $C_i$
- The weight value of this category is one from the set of nearest neighbors.

Where $i=1, 2, \ldots, n$ categories
$j=1, 2, \ldots, m$ traits.

So, if we consider a user with a record of categories represented as a set, **S = {Software, Computer, Hardware, Musical Instrument, Genre, Composition}** and we have to check if this person has a trait: Technical, then the following calculations is performed:

### 1. Calculate $P(C_i/T_j)$

P(Software/Technical)

For this, we have to calculate the range of values to which the weight of the "*Software*" belongs using the minimum distance concept. After you get the nearest neighbors of "*Software"*, we find out the count of records in the Knowledge base which satisfy:

- They belong to the trait: Technical
- They contain the category: Software
- The weight of this category is one from the set of nearest neighbors of that category.

If "*x*" is the number of records satisfying all 3 conditions, then

$$P\left(\frac{Software}{Technical}\right) = \frac{x}{(\text{total number of users who belong to trait "}Technical\text{")}}$$

Similarly, calculate $P\left(\frac{Computer}{Technical}\right)$, $P\left(\frac{Hardware}{Technical}\right)$, $P\left(\frac{Musical\_Instrument}{Technical}\right)$, $P\left(\frac{Genre}{Technical}\right)$, $P\left(\frac{Composition}{Technical}\right)$ following the same procedure.

### 2. Calculate $P(T_j)$

This is the ratio of number of records in the knowledge base with trait as "Technical" to the total number of records.

$$P(Technical) = Count\left(\frac{\text{number of records with trait "}Technical\text{" in KB}}{\text{total number of records in Knowledge Base}}\right)$$

### 3. Calculate $Z_{Technical} = P(T_j/C_j)$

$$P\left(\frac{Technical}{S}\right) = \prod P\left(\frac{Ci}{Technical}\right) * P(Technical)$$

Where $j = 1, 2, \ldots, n$ categories.
$C_j$ = Categories that belong to Set S.
Thus

$$P\left(\frac{Technical}{S}\right) = P\left(\frac{Computer}{Technical}\right) * P\left(\frac{Hardware}{Technical}\right) * P\left(\frac{Musical\_Instrument}{Technical}\right) * P\left(\frac{Composition}{Technical}\right) * P(Technical)$$

Similarly, we can follow step 1,2,3 to calculate

$$Zmelomane = P\left(\frac{Melomane}{S}\right)$$

Where,

$$P\left(\frac{Melomane}{S}\right) = \left[\begin{array}{l} P\left(\frac{Computer}{Melomane}\right) * P\left(\frac{Hardware}{Melomane}\right) * \\ P\left(\frac{Musical\_Instrument}{Melomane}\right) * \\ P\left(\frac{Genre}{Melomane}\right) * P\left(\frac{Composition}{Melomane}\right) \end{array}\right] * P(Melomane)$$

Similarly, we calculate "*Z*" for every trait defined in the system.

### 4. Sort all the "*Z*" values

After calculating "Z" for every trait, sort these values in the descending order.

### 5. Find the average number of traits that are assigned to the existing users from the Knowledge Base

After sorting "Z" value for each trait, we assign traits to the new user based on the average number of traits assigned to existing users.

For example: If the average number of traits assigned to the user in the existing Knowledge Base is "*n*", then top "*n*" number of traits from step 3 are assigned to the new user.

Thus, if $Z_{Technical}$ comes under the top "*n*" values then "Technical" is assigned as a trait to the user.

## 3.4 LEARNING

Learning is achieved by inserting new user information in the Knowledge Base. This is done for the purpose of accuracy. User information is added in the Knowledge Base when any trait assigned to him has weight that exceeds the threshold of that particular trait (defined in Training Set). Now, whenever a new user arrives whose weight is lesser than the threshold, such user's information will not be added in the Knowledge Base. For such users, Naïve Bayesian classification will be applied to find his trait. We will consider the new updated knowledge base as the training set for classification, which will give more accurate results as the number of records in the Knowledge base increases. There may be a case wherein such user (who has a weight less than the threshold that has prevented his record being entered into the knowledge base) has few more search queries added to his account of search queries in due course of time. In that case, the whole procedure is followed again and if by this time his count crosses or equates to the threshold, his record will be entered into the Knowledge Base. Or else, if the weight is still less than the threshold, simply the trait is found and assigned to the user using Naïve Bayesian classification.

## 4. IMPLEMENTATION

The proposed algorithm can be implemented in any language. The pseudo codes for the components of the system are given below.

**DEFINITION:**

```
user {traits [ ]} //user can have many traits;
category {(based on keywords)} // ontological definition;
$threshold \rightarrow TH = \{TH_1, TH_2,...., TH_m\}$
for each trait j;
$trait = \{category[]\}$ // weight list
```

**Steps:**

**TRAINING PART**

```
do for each well-known user $U_k$
{
    note down the trait of user $U_k$;
    extract words from log;
    find category count;
    find probability for each category; /* this ratio is
                                           weight.*/
    set $R_j = \sum Weight$; /* sum of probability of categories
                        that belong to that trait. */
}
$TH_j$= Average of $R_j$ of trait j;
```

**FINAL TRAINING SET**

```
$TH = \{TH_1, TH_2,...., TH_m\}$ For each of the "m" traits;
```

**CATEGORISING EXISTING USER (BUILDING KNOWLEDGE BASE)**

```
for (each user)
{
    get log;
    extract keywords from log;
    find category count;
    find probability for each category; /* this ratio is
                                           weight.*/
    $Uw \rightarrow weight\ list$
    $Uw = \{ W_1, W_2, .... , W_n \}$
    $R_j = \sum W_i$ categories i which belong to trait j
    $R = \{R_1, R_2, ......, R_m\}$ /*set of values of R for
                                   every trait*/
    if( $R_j \geq TH_j$ )
        assign trait to the user based on $R_j$;
}
```

**CATEGORISING NEW USERS**

```
for (each new user)
{
    get log;
    extract keywords from log;
    find category count;
    find probability for each category; /*this ratio is
                                          weight.*/
    $Uw \rightarrow weight\ list$ ;
    $Uw = \{ W_1, W_2, .... , W_n \}$;
    $R_j = \sum W_i$ categories i which belong to trait j
    $R = \{R_1, R_2, ......, R_m\}$
    if ( $R_j \geq TH_j$ )
        assign trait to user based on $R_j$;
    else
    for (each category i)
    {
        find nearest neighbors for category i
            /*neighbor: existing user having same category in
                    search
            distance: difference between weight of neighbor and
                    this user. $|W_i' - W_i|$.
                    Nearest neighbors are the users having
                    minimum distance from this user.
            */
        $total\ users\ fall\ in\ nearest\ neighbor\ category \rightarrow Un$
        Find $Pi\left(\frac{Ci}{Tj}\right) = \frac{Un}{total\ number\ of\ users\ that\ belong\ to\ Tj}$
    }
    for (each trait)
    $Pi\left(\frac{Ci}{Tj}\right) = \frac{Un}{total\ number\ of\ users\ that\ belong\ to\ Tj}$
    find the average of number of traits "z" assigned to each
        of the existing users.
    Assign "z" traits to this new user.
}
```

## 5. BENEFITS / STRENGTH OF PROPOSED SYSTEM

The proposed system is domain-independent and solely based on history of users' search pattern. Our system tries to solve the problem of assigning viable traits to usersby comparing the traits with the threshold generated from search history of well known users; that provides a good base for finding the threshold. The system allows assigning multiple traits to the user that matches the real world scenario for distinguishing users from the others.

## 6. CONCLUSION

Our proposed system is successful in suggesting personality trait for the user depending on the search queries entered by her/him. These search queries are processed using Data Mining algorithm and Naïve Bayesian Classifier algorithm. Data mining algorithm uses threshold for comparing user’s data with the training set. Hence, threshold value is crucial and deciding factor for populating knowledge base which is required for further processing of the system.

Personality trait suggested for the user serves as rich input source for the recommendation systems. Recommendation systems are deployed by various types of advertising companies, e-commerce sites, e-mail services, social networking sites, etc, to gain profit from targeted advertisements and by providing required information which user has not asked explicitly but it may be useful to the user. Our system does not require explicit user efforts for getting recommendations from recommendation systems. It can also help in providing various deals, advertisements and even latest information update about particular events in which user may be interested. Our system, thus, benefits the users as well as the companies providing recommendation systems.

## 7. REFERENCES

[1] Jingyu SunP, Xueli YuP, Xianhua LiP and Zhili Wu Research on, “Trust-Aware Recommender Model Based on Profile Similarity,” 2008 International Symposium on Computational Intelligence and Design.

[2] Tadachika Ozonoand and Toramatsu Shintani, “Paper Classification for Recommendation on Research Support System Papits,” IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.5A, May 2006.

[3] Gokul Chittaranjan, Jan Blom, Daniel Gatica-Perez, “Who's Who with Big-Five: Analyzing and Classifying Personality Traits with Smartphones”.

[4] Jasmeen Kaur, Vishal Gupta, “Effective Approaches For Extraction of Keywords,” IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010.

[5] Ying Wang, Weiru Liu, and David Bell, “A Concept Hierarchy based Ontology Mapping Approach”.

[6]<http://www.swharden.com/blog/category/php/> “This website illustrates structure of the Web Search Query Log of Google Searches.”

[7]<http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf> “This website presents the Naïve Bayesian Algorithm – An algorithm for Classification in Data Mining.”