Adaptive Food Suggestion Engine by Fuzzy Logic

Tousif Osman, Maisha Mahjabeen,Shahreen Shahjahan Psyche, Afsana Imam Urmi,
J.M.Shafi Ferdous, Rashedur M Rahman
Department of Electrical and Computer Engineering, North South University
Plot-15, Block-B, Bashundhara, Dhaka, Bangladesh
tousif.osman@northsouth.edu,maisha.mahjabeen@northsouth.edu,
shahreen.psyche@northsouth.edu, afsana.imam@northsouth.edu, shafi.ferdous@northsouth.edu, rashedur.rahman@northsouth.edu

Abstract—The research proposes an approach for searching and sorting food list according to user preference. The system maintains a fuzzy database to store restaurants and their menus with some specific parameters for each item. System allows the user to specify food tastes that are of their interest, location name where they prefer to eat, approximated budget for each person and few other defined parameters as searching keyword. While searching, the system will fetch the entries from the database according to user defined parameters, convert each item's metadata to fuzzy parameters and pass the list to a fuzzy controller. Then the controller assigns a score as an output to each item. Finally the list is sorted in descending order. System allows the users to provide the feedback about the food which makes the system to be adaptive. System also gets scaled for each individual user as food taste varies significantly for individual users; hence the system provides more precise result and makes it a complete food searching engine.

Keywords—Fuzzy Logic; Adaptive learning; Fuzzy Database; Soft Computing; Searching; Food Searching;

I. INTRODUCTION

Since the inception of human history the production of various kinds of data is increasing exponentially. In this era of 21st century human produces enormous data on a single second. As a result we require smart algorithms to find our desired information. At present there are many searching algorithms but in most cases they are text base string parsing and matching algorithms. However data can be divided into various subgroups and by processing them separately it is possible to provide more optimal and concise result using their unique metadata. There are many such systems that handle this subdivision of data in a more robust fashion by using their different parameters. In the field of food searching there is a scarcity of smart, optimal algorithm that solves this searching problem efficiently. As the parameters that signifies food suggestions are abstract, this sector of algorithm development remains poor. As a result most food searching problems are handled with traditional string matching algorithms. In most cases, systems allow the user to search food by the food name. However, generally users intend to search food by the taste they want to eat or the environment they prefer and get the listed food names as output. For instance two most domination food searching platforms, Yelp and Foursquare, does not consider what users felt about the food and environment while searching. So when users search by food attribute (i.e. taste: spicy) optimal result is not generated. Yelp lists a food that has spicy on the menu name and Foursquare considers a menu metadata as well while sea rching. In this research we convert this compound problem of food selection to a fuzzy domain and provide an optimal food selection algorithm using fuzzy logic. The major component of food, taste is a complex attribute that cannot be measured on a linear scale.

II. RELATED WORKS

The authors in [1] describe the essentiality and efficiency of using recommendation system using tags and latent factorials for sorting out the best solution in food searching. In this research when data is being searched, system evaluates tags and rating given by users to find out the best choice. In next paper [2], collaborative filtering and content based recommendation techniques have been used to build a hybrid fuzzy system for restaurants. The authors calculated similarity between restaurants using cosine distance which is also known as cosine similarity. Appetito el al.[3] propose a recommendation system which prioritizes the sentimental analysis to recommend a menu to the user. For the fuzzy system, they have used the extension of the Apriori algorithm. In the next related paper [4] a fuzzy recommendation system is proposed based on customer online review. According to the paper, the system takes the user review as one of its input which is mostly in natural language. In [5], the authors calculate food nutrition and body mass index of user and provides food recommendation by using fuzzy logic.

In [6], the researchers suggested a book recommendation methodology which integrates fuzzy concepts with link mining. They took the book list of one of their course of top 50 universities. Then they tabulated the books and assigned a score to each book using fuzzy quantifier. In [7] the authors used multi-criteria collaborative filtering (MC-CF) recommendation system that uses Neuro fuzzy techniques along with dimensional reduction. The research of this paper was done to reduce two drawbacks,known as sparsity, and scalability. There was another research based on personalized recommendation system [8]. They had done this research mostly to provide suggestion to ecommerce portals related product suggestion. This research work has used Fuzzy Near Compactness (FNC) concept to provide solution.

Considering the drawbacks and advantages of mobile phones, a content based recommendation system has been built by the researchers to improve user experience [9]. Here information

retrieved from sensors and frequently used personal information has been used to build a context aware system. Through mapping mobile user context and properties of contents, they have modeled a modified reasoning search engine. Finally, the last related research work to our research is on efficient advertisement recommendation system for TV program [10]. Though the goal of the paper is completely different to ours but their research methodology includes some concepts that helped building our system. For providing TV program recommendation they have used fuzzy logic techniques and have taken into account various parameters like program content, users interests, providers' preferences, program timing, etc. Based on those parameters they have provided keywords to each program and generated TV-program recommendation.

III. THEORY

The primary goal of this research is to build a fuzzy system that will take different input parameters as search query and as an output the system will assign scores or precedence level to individual menu as a fuzzy output. More score means more preferred menu for a specific search query. Explanations of the system components are given below.

A. Input Variables:

Our primary objective is to make a flexible food searching engine. We have considered 5 attributes as input variables, these are: taste, environment, budget, location and user review. For each input variable we have defined 3 membership levels.

1) Taste:

Taste is the primary component that distinguishes different foods. Our system majorly focuses on searching food by different tastes. User can give any kind of taste as input. For example: single taste like sweet, sour etc. or even multiple tastes like sweet-sour, hot-spicy etc. We also do not want to limit the user to any specific number of taste attributes. So in the fuzzy controller we need N number of variables but it would not be a good design choice. To solve this issue we have defineonly one variable for taste in the fuzzy system. This single variable is a generalized variable taste, which has 3 fuzzy levels with their membership functions. The levels are: Low, Medium & High.This variable can represent any taste of user's interest at a time. The system relates the user input with the database and send test level of single taste to the fuzzy system as an input. In the case of multiple tastes for example for sweet-sour the calculation will be done twice. So the server will call the fuzzy controller twice for a single search and hence we will have two separate scores for two different tastes for the same set of menus. System will combine multiple scores to get a single score.

Let us assume a user is searching for food names that are "sweet and spicy". Initially system will derive all the food names from the database that has entries named sweet and spicy in the menu and taste table. After that system will assign fuzzy score of spicy and sweet to each menu item. As fuzzy controller cannot take two taste inputs at the same time, system will pass each menu'sscore of spiciness with other parameters and the score of

sweet and other parameters separately to the fuzzy controller. Then the system will combine resulting two scores for each menu to get the final score for individual menu item.

menu1_score_spicy
  = fuzzycontroller (menu1, spicy, …other_parameters)

menu1_score_sweet
  = fuzzycontroller (menu1, sweet, …other_parameters)

menu1_score= menu1_score_spicy + menu1_score_sweet

menu2_score_spicy
  = fuzzycontroller (menu2, spicy, …other_parameters)

menu2_score_sweet
  = fuzzycontroller (menu2, sweet, …other_parameters)

menu2_score= menu2_score_spicy + menu2_score_sweet
.
.
menuN_score_spicy
  = fuzzycontroller (menuN, spicy, …other_parameters)

menuN_score_sweet
  = fuzzycontroller (menuN, sweet, …other_parameters)

menuN_score= menuN_score_spicy + menuN_score_sweet

2) Environment:

This input variable is just like the taste variable. The difference is the input that will be environment type like formal, casual etc. and we do not allow the user to specify multiple environments at the same time. Except this slight difference the handling process and calculation are almost as same as taste and will have a separate fuzzy variable called environment.

3) Budget:

User's budget will be taken as an integer input. This input variable has 3 levels in the fuzzy system: Less Preferred, Preferred&More Preferred.Database has a budget for single person for each menu.When user specifies a budget system will find menu names in the range of $4.5 more or less of the user defined budget. System finds the absolute difference of users' budget and different foods' price. If a user's budget is 'x' and one of the food's price is 'a'. Then the difference 'D' is given below:

$$S = |x - a|$$

Hence the system creates a list of menus with theprice in between x – 4.5 and x + 4.5. Therefore the difference will be in the range of $9. If the user difference between budgets is 0 ~ 3 it will be in the level of More Preferred, if the difference is between 3 ~ 6 then it will be in the preferred level and if the difference is between 6 ~ 9 then the menu will be in Less Preferred Level. The lower the differenceis, the higher the preference.

4) User Review:

System also allows the user to search according to users reviews. Users can assign 5 levels of satisfaction. Review is in

the scale of 0 to 5.While making the query user can provide a score from 0 to 5 and system will find all the menus that are 1.5 more or less of the user defined score. We have again defined 3 levels of fuzzy membership asLess Preferred, Preferred &More Preferred for the user review variable. User review will be handled similarly as Budget variable.

5) Location

Users can mention their preferred location. According to their preferred location we can calculate the distance of the restaurants and preferred location. User will give the name of the area of their choice. Using Google API system gets the latitude and longitude of the preferred location. Restaurants will have their own location in the database as latitude longitude form. Using this information we can calculate the distance.

Suppose we have a restaurant named 'A' which has latitude '$lat' and longitude '$lon'. User gives a preferred location 'B' which has latitude 'lat' and longitude 'lon'. So the distance between these two places is:

$$AB = \sqrt{((lat - \$lat)^2 + (lon - \$lon)^2)}$$

System considers only the restaurants within that range of 3KM.Hence we have defined the location variable by 3 membership functions which are Close, Near and Far. When the difference is between $0 \sim 1$ km then it will be on the close level, if the difference is between $1 \sim 2$ km then it will be on near level and if the difference is between $2 \sim 3$ km thenit will be on the far level. The closer the distance, the higher the preference is.

B. Input Membership Function:

 In the fuzzy system for input we have used the 'Gaussian Curve' because it is relatively more precise than the triangular, trapezoidal and other curves. For all of the five variables we have used curves like fig 1 but the range and labeling is different.
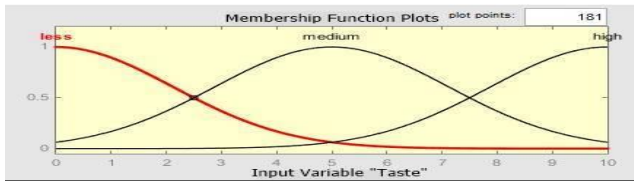

Fig. 1.   Input membership function

C. Output Variable:

In the fuzzy system we have defined one output variable 'Preference' which we infer from the five variables using the fuzzy if-else logic. The preference is defined by 5 functions or levels. Higher preference level means -that is more similar to the users' requirements. The division of the preference level is with respect to the logic below:

We have defined all of the input variables in three levels. We have considered each variable as a ternary number.Each can have maximum values up to 3. Let us assume that variables taste, environment, budget, locationand user review are x_1, x_2, x_3, x_4 &

x_5.If all the variables are equals to highest value which is 3 we get, $\qquad \sum x_i = 15$

If all the variables are equals to the lowest value which is equals to 0, we get $\qquad \sum x_i = 0$

So if we declare 5 levels of preference, the range of each level is, $\qquad \sum x_i \div 5 = 15 \div 5 = 3$

Therefore we have defined 5 output membership function for 5 levels. The preference levels are given below with their range in table 1.Let us consider an example where a menu the following properties.

Taste = high = 3, Environment = Medium = 2

Budget = medium = 2,   Location = Low = 1

User Review = High = 3

So, Total = 3 + 2 + 2 + 1 + 3 = 11.

It lies in level 4 which ranges from 9 to 12. So, the rule 4 will be triggered or the 4th membership curve will be used for generating the score.

TABLE I.   RULES SEPARATION TABLE

Rule #	Preference Levels	Range
1	Level 1	0 – 3
2	Level 2	3 – 6
3	Level 3	6 – 9
4	Level 4	9 – 12
5	Level 5	12 -15

D. Output Curve:

For output curve, each preference level is represented with triangular shaped membership functions to keep the computation simple.Each curve overlaps on the midpoint of the adjustment curve. Hence we have the output in on the range of 0 to 9 that is,output score lays on the range of 0 to 9.
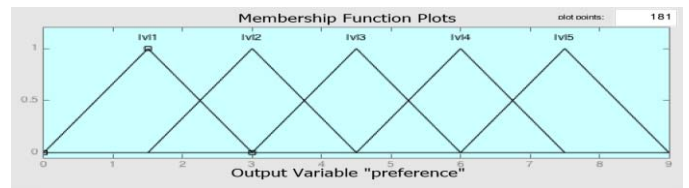

Fig. 2.   Output function

Figure 2 shows the output curves with the label of the preference levels.

E. Fuzzy Conditioned Rules:

Each variables with 3 membership functions and 3 levels. However some variable may not be present on the search query. For instance a user may not be interested in the budget and he/she may not provide budget in the search query. If a variable is not present we use 'none' or 0 as the input. Therefore,each variablein the system can have at most 4 values including the option 'none'. We have associated each combination of input variables with the output using the logic described in "Output Variable" section (III.C).Example of a rule is given below:

If taste = medium
      and environment = low
      and budget = more preferred
      and location = closer
      anduser_review = high
then output level 4

As there are 5 input variables all the variables cannot be visually represented in a single plot. So, as a demonstration of accuracy Fig.3 maps two variables as a graphical representation. It is the surface view of the input variable taste and location. From the graph we can see that more the taste and location is closer to the desired taste and location, the higher the score will be.
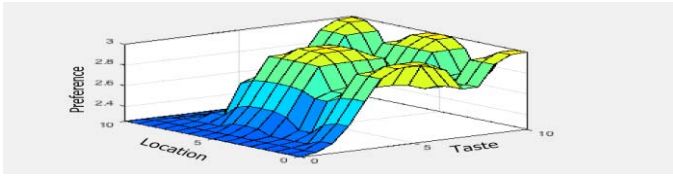

Fig. 3.   Graphical representation or surface view of taste and location

F. Adaptative System Based on User Weight:

In our system Taste, User review and Environment scores are given by the user. In the Input Variable section (III.A), we have taken the average of scores to get the scores for the input parameters. We can further enhance this scoring by neural networks adaptation techniques. We have assigned each user a weightW_i. This score will be calculated by the frequency f_i where f_i stands for number of reviews that users have given on the system. To get individual user score for a given parameter (i.e. taste) we will multiply user given score by the user's weight. To get the final score, we will take the mean of all the scores for that given parameter. We will compute the adaptive score with the following equations-

$$W_i = \frac{2\arctan(fi)}{\pi}; \; fi > 0 \quad \text{Total Sore} = \frac{\sum_{i=1}^{n} XiWi}{n}$$

Here X_i stands for individual user, W_i stands for weight of that user and n stands for total number of user.User frequency is an integer. To generate the weight we need to convert it to a range of $0 \sim 1$. Hence we have formulated the above function that generates our desired output. Fig.5 shows the diagram of the weight function.

Fig. 4.   Neural networks logic for adaptive system

In Fig.4, we have demonstrated our neural networks logic that we have used to make the system adaptive.
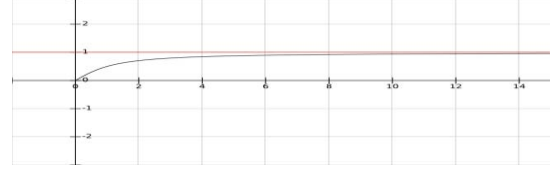

Fig. 5.   Plot of weight function

G. User Specific Adaption:

Taste has no particular measurement. For example, a food can be spicy for a person while the same food can be less spicy for other person. Therefore taste is very much user specific. To resolve this, a simple solution can be to assign a user preference scale for each attribute based on the user previous feedbacks. This preference scale will be updated every time a user provides a review.

For instance let us assume a user is saying a food little spicy out of 3 options little spicy, spicy and very Spicy. System will list feedbacks where he/she has provided spicy as taste feedback. System considers at most 5 recent feedbacks from the list. System will compare user's feedback with average feedback of the each item in the list. System will store the difference in feedback as the preference scale of that attribute. Let us assume, in the database, there are foods with medium spicy scoring as previous feedback like below:

Fried Chicken, KFCx1= 5.6

Beef & Bacon,Take Out,x2 = 4.8

Chicken Cheese Burger,Preetom,x3 = 6

Each of the time the user said "it was too spicy", the score is 10. Then the system will find the difference and will take the average difference.

$$\sum(10 - xi)/3 = ((10\text{-}5.6)\text{+}(10\text{-}4.8)\text{+}(10\text{-}6))/3$$
$$= 4.53$$

Now this score will be saved in the user account. Next time whenever the user will search for "something spicy", taste score 5, it will be scaled. In this case its (5-4.53) = 0.47 which means system should search for less spicy food.

IV.   METHODOLOGIES

Our goal is to develop a system that will allow the users to assign different taste parameters with different values. Besides, it will allow few other defined parameters and magnitude. System will allow the users to search fuzzy database and provide a list of

preferred menu-restaurant pairs.System also maintains a user profile.

The system consists of 3 major components: server, database & fuzzy system. System performs two major tasks, search for preferred menus and takes user review. While searching server receives data from the user interface(UI), server fetches relevant data from database, matches and calculates parameters and passes it to the fuzzy system. Fuzzy system uses the attributes level to calculate a preference score. Combining the preferences score with the database, system generates a sorted menu list as output. Fig.6 shows the three components of the system.

Fig. 6.   Block Diagram of the system

A. Server:

This part of the system is responsible for data fetching from the database, preprocessing the data for accruing fuzzy scores for each parameter and communications between the user and the fuzzy system. We have used Apache web server as the server software and PHP as the processing language. One of the primary task of the server is to serve user with web interface. There are two separate different forms, one for searching menus and one for taking feedback about specific items. The server keeps a profile of each user.

For searching, user can give different taste input using their natural language for example something spicy. For multiple tastes, they can provide the input as comma separated taste parameters like "sweet, sour". For location parameter, user provides location address just using strings. Using Google Map Application Program Interface (API), the system will derive the latitude and longitude values for that location. For the budget user has to give estimation of their budget for each parson.User can provide their preferred environment but this input takes only one environment. The other form is intended for User Review. Users will be asked to give the review under a specific restaurant's menu. User will be asked several questions like 'was the food too spicy?' or 'was the environment as expected?' System will store these information in the database. One of the major task of the server is to preprocess the data before passing it to the fuzzy controller. For preprocessing server uses the methods described in the theory section. There may be multiple review available for each menu in the database. Converting those multiple entries to a single score is the primary task of preprocessing.

B. Database:

Initially every menu in the database will have scores for each attributes.This score is provided by system administrators or voluntaries (i.e. restaurants).

Fig. 7 shows the design of the database. To maintain a simple relationship in the database we have defined six tables in the database. These tables are, (i) Restaurants that has the necessary information for that particular restaurant, (ii) Menu holds menu items and its parameters paired with the primary key of restaurant,(iii) Taste holds several user feedback about the test of the food (iv)Review stores the user reviews against each menu, (v) Environment stores the environment suggestion of a restaurant, (vi) User for maintaining user profile. A restaurant can have multiple types of environment and food can have different type of taste at the same time with different scoring. That is why we have used a separate table for environment and taste.
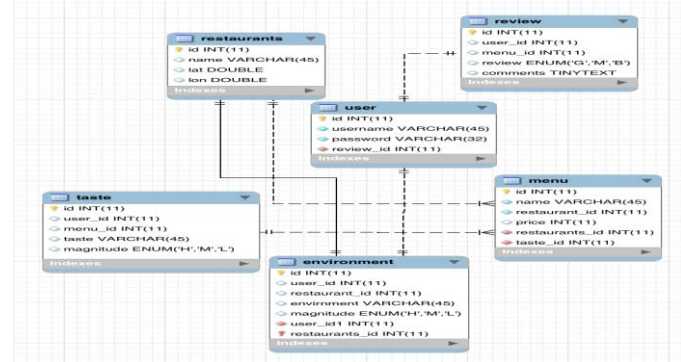

Fig. 7.   Database Tables Schema

We have used MySQL database, as it is easily integrated with the Apache server. Database does not deal with the fuzzy system rather the server acts as an intermediary and communicates with the database. Server fetches the data from the database and fuzzifies the data, hence the database acts as a fuzzy database.

C. The Fuzzy System:

The fuzzy system is the key factor of the system as it calculates the preference level of the restaurants with foods. Fuzzy System acts as a fuzzy controller in our system.The one and only job of the fuzzy system is to take in different score of input parameters and provide a score for that given set of inputs.We have used the set of instructions described in the Theory section (III) to construct the fuzzy controller. MATLAB fuzzy toolbox has been used to build the fuzzy system. Fig 8 shows the overview of the fuzzy system which takes taste, environment, budget, location and user review as fuzzy input variable and gives the output of preference level. We have used Mamdani inference rule to calculate the preference level.

Server fetches data and passes data to Fuzzy Controller. On receiving the input data, fuzzy system passes those data through the fuzzy conditions and calculates the preference level for each item. MATLAB fuzzy system has been wrapped as an external executable Command-Line (CLI) program. The program can take multiple set of input parameters from the CLI.

Fig. 8.   The Fuzzy System

When the system will get the preference levels of various restaurants menu's from the fuzzy system, it will order menus by scores from high to low.

V. RESULTS AND DISCUSSION

For testing the system we have inserted 20 restaurant and 10 menu items for each of the restaurant. We have done 200 survey on the university campus to get reviews on the inserted menus. We have run the system on simulated data to verify the results.

For example, user searched for a food using the keyword 'something spicy and crispy', '400/- budget', 'Gulshan 1 Area'. Firstly the system will find all the menus within the ranges as mentioned in the theory section and make a list. Now let us consider the system has a menu item 'fried chicken' under the menu of KFC, Banani. Suppose it is .8km away from the suggested area. It will give input .8km in location variable. While in the menu it is suggested that the spiciness is 7.5 for this item. So the input variable for taste will be 7.5. The user's budget is 500/- and fried chicken bucket which contains two pieces of chicken fry, costs 400/- . System will take the difference of menu price and the user preferred budget and send the difference to the fuzzy system which is 100/-. In this query user review and environment are none. Now using these input parameters we can calculate the preference level using the Mamdani Min Implication. The Mamdani Min Implication is:

$$\phi_{c[}\mu_a(x), \ \mu_b(y) \ _{]=} \mu_a(x) \wedge \mu_b(y)$$

Table II lists the sample input of this example and table III shows the corresponding outputs of this sample table. Using this example we get the preference level 3.32 for the above situation. This is also enlisted as the first sample output of table III.

TABLE II.   SAMPLE INPUT

Taste	Spicy, Crispy
Environment	None
Location	Gulshan 1 Area
Budget	500/-
User Review	None

System will have several menu names and score for a given input parameters. After preparing the list with processed parameters server will call the fuzzy controller and pass the list. As a response fuzzy controller will return score for each menu.

TABLE III.   RESULTING SCORE OF SAMPLE INPUT

Food	Price	Location	Taste	Preference
Fried Chicken,KFC	400	.8	medium	3.32
Burger,KFC	350	1	medium	4.46
Pizza,Pizza Hut	700	.2	High	4.04

In case of multiple taste search server will call the fuzzy controller multiple times for each test keywords and finally combing multiple results system constructs the final score for each menu and sorts the menus by score and provide the user with order list of menus according to preference. The searching

features are compared with two popular food search engines are given in table III in terms of searching features.

TABLE IV.   COMPARISON WITH EXISTING SYSTEM

Features	Yelp	Foursquare	Our System
Search by taste	Not Available	Not Available	Taste based search is available
Search by price	Available	Available	Available
Search by distance	Available	Available	Available
Search by environment	It searches using the keywords, pub, good for kids, has TV etc.	It uses the key words Coffee, Nightlife, Fun etc.	It uses the key words, formal , romantic, casual
Search by User Review	Uses ratings or review numbers to sort	Uses ratings to sort	Uses user review by parsing the keywords like good, bad, excellent etc.

VI. FUTURE WORK

Currently we have implemented our fuzzy recommender system as a web application. In future we want to build an android application for this fuzzy recommender system which will be more precise and easily accessible. We wouldalso like to modify the triangular output to get more optimized output.

REFERENCES

[1] M. Ge, M. Elahi, I. Fernaández-Tobías, F. Ricci and D. Massimo, "Using Tags and Latent Factors in a Food Recommender System", *Proceedings of the 5th International Conference on Digital Health 2015 - DH '15, pp.105-112,* 2015.

[2] R. Priyono and K. Surendro, "Nutritional Needs Recommendation based on Fuzzy Logic", *Procedia Technology*, vol. 11, pp. 1244-1251, 2013.

[3] Xochilt Ramírez-García and Mario García-Valdez, "Restaurant Recommendations Based on a Domain Model and Fuzzy Rules", *Recent Advances on Hybrid Intelligent Systems*, vol. 451, pp. 533-546, 2013.

[4] S. Sohail, J. Siddiqui and R. Ali, "OWA based Book Recommendation Technique", *Procedia Computer Science*, vol. 62, pp. 126-133, 2015.

[5] J. Han, H. Schmidtke, X. Xie and W. Woo, "Adaptive content recommendation for mobile users: Ordering recommendations using a hierarchical context model with granularity", *Pervasive and Mobile Computing*, vol. 13, pp. 85-98, 2014.

[6] N. Zhao, Q. Wang and J. Zhong, "Research on Fuzzy Intelligent Recommendation System Based on Consumer Online Reviews", *Knowledge Science, Engineering and Management*, vol. 8041, pp. 173-183, 2013.

[7] M. Trevisiol, L. Chiarandini and R. Baeza-Yates, "Buon appetito", *Proceedings of the 25th ACM conference on Hypertext and social media - HT '14, pp. 327-329,* 2014.

[8] M. Nilashi, O. Ibrahim, N. Ithnin and R. Zakaria, "A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques", *Soft Comput*, vol. 19, no. 11, pp. 3173-3207, 2014.

[9] B. Ojokoh, M. Omisore, O. Samuel and T. Ogunniyi, "A Fuzzy Logic Based Personalized Recommender System", *IRACST - International Journal of Com puter Science and Information Technology & Security (IJCSITS)*, vol. 2, no. 5, pp. 1007-1015, 2012.

[10] S. Velusamy, L. Gopal, S. Bhatnagar and S. Varadarajan, "An efficient ad recommendation system for TV programs", *Multimedia Systems*, vol. 14, no. 2, pp. 73-87, 2008.