UDC 519.816

A chatbot as an environment for carrying out the group decision making process

Olga Chukhno[*], Nadezhda Chukhno[*], Konstantin Samouylov[*†], Sergey Shorgin[†]

[*] *Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation*

[†] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (FRC CSC RAS), 44-2 Vavilov St, Moscow, 119333, Russian Federation*

Email: olgachukhno95@gmail.com, nvchukhno@gmail.com, samuylov_ke@rudn.university, ssorgin@ipiran.ru

Since the early 1960s, people have been interested in creating a robot that can communicate with humans (for example, ELIZA, ALICE and SmarterChild). Therefore, it's not surprising that chatbots are an incredibly sought-after topic now. This trend is due to the growing popularity of messaging platforms, as well as the development of artificial intelligence and machine learning. Firstly, the way of communicating is radically changed. We moved from phone calls to text messages, and then to messaging applications. Secondly, the current state of the field of artificial intelligence, natural language processing (NLP) and voice recognition allow to robots to understand user's requests better and respond to them accordingly. In fact, chatbots are becoming new interfaces in messaging applications, replacing many other mobile apps. Some experts argue virtual assistant of this type will be able to replace 99% of software in the near future. Thus, instead of developing a new application, it becomes possible to create a service that works in an already installed platform on the user's device. The goal of the work is to present a solution in which users will be able to participate in the GDM process using natural language at any time and any place. Moreover, the chatbot will make easier to provide experts preferences using an apprehensible interface. The paper proposes a virtual interactive system developed for implementing the method of ranking alternatives in the group decision making process. The designed digital assistant works on the base of a cross-platform messenger Telegram.

Key words and phrases: group decision making process, chatbot, conversational systems, machine learning, decision support system, mobile application.

CEUR-WS.org/Vol-2407/paper-02-164.pdf

Copyright © 2019 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.
In: K. E. Samouylov, L. A. Sevastianov, D. S. Kulyabov (eds.): Selected Papers of the IX Conference "Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems", Moscow, Russia, 19-Apr-2019, published at http://ceur-ws.org

1. Introduction

The group decision making process (GDM) in fact is choosing the best options from an acceptable set of alternatives and is presented in almost every task of daily life. The use of new technologies expands decision making capabilities and allows these processes to be conducted in situations that were previously impossible [1]. For instance, now experts from different countries of the world may be involved in one process of decision making. However, even with the development of mobile technologies, there is still an important need for new interaction tools between the experts and providing information to the system in cases where they can not meet in one place at the same time [2–5].

The implementation of mobile technologies in the GDM processes is based on the assumption that it will lead to an optimization of the group decision making process, namely, experts will be more focused on the problem and will be able to spend less time providing their assessments to the system [6]. These tasks differ from optimization problems traditionally solved the authors' research team, for example, optimizing the functioning of wireless telecommunication systems depending on interference [7, 8], or developing hysteresis network overload control mechanisms [9, 10].

The recent boom in artificial intelligence has increased people's attention to conversational entity, commonly referred to as chatbots [11–13]. Chatbot is a virtual assistant that is able to interact with the user using natural language. As a rule, it can ask questions to the user and answer them, suggest a topic to be discussed, etc. In the modern world, chatbots have been implemented for education, information search, customer service, communication, site navigation, analytic, design, developer tools, games, health, personal, productivity, HR, marketing, news, shopping, social, sports, travel, there are even bots that provide entertainment services [14–16]. However, we assume that in the field of group decision making, our work will be the first attempt to create a conversational program that has the ability to communicate with experts 24/7 hours (Fig. 1).

Figure 1. Chatbot's domains

The GDM method that can work with a large amount of information have been presented in [6]. The method of alternatives ranking was developed to carry out the process in social networks, when the number of experts involved in the process was large.

However, the procedure for providing assessments by experts was not automated and this step of the process took a lot of time over the experts.

The purpose of the paper is to develop a prototype of a mobile virtual assistant that automatically processes the answers of experts. Smartbot will provide a new approach to working with a dynamic group of experts (during the process the number of participants may change), making decisions about alternatives at any place and at any time. In fact, at each stage of the group decision making process, users:

- will be informed about a set of alternatives that need to be compared in pairs;
- will be able to select independently a scale for evaluating alternatives, guided by the degree of confidence regarding the level of knowledge about the problem/topic under consideration;
- will be able to participate in the process anytime and anywhere by connecting to the Internet.

Accordingly, developing the tasks posed in [6], the authors of this article create an environment for more extensive numerical experiments of the method, which will improve indicators such as the degree of participation of experts in the GDM process and guaranteed to increase the reliability of the results.

The rest of the paper is organized as follows. Section 2 provides an overview that is necessary for understanding the group decision making process, as well as explaining to the reader what chatbots are, why they are needed, and why these interactive agents are so popular nowadays. The third section describes the system architecture. The chatbot implementation for the GDM process is presented in the fourth section. The fifth part of the paper reveals the main advantages of introducing an artificial conversational entity into the area of group decision making. In conclusion, results are summarized and the tasks for further research are set.

2. Preliminaries

2.1. Ranking Method of Alternatives for the Group Decision Making Process

The group decision making process can be defined as follows [2,4]: the set of experts $E = \{e_1, ..., e_K\}$ should present their preferences regarding to the set of alternatives $X = \{x_1, ..., x_M\}$, where $|E| = K < \infty$ and $|X| = M < \infty$.

The goal of the group decision making process is to order different alternatives from a set of alternatives from the best to the worst with the help of the association of some preference degrees, taking into account the opinions expressed by group of decision-makers.

The problem becomes critical when the number of alternatives and experts evaluating them are large. In this case, it is inappropriate for each expert to carry out a comparison of all alternatives pairwise. The models described in [4,6] provide that the expert gives estimates for some subset of alternatives X_k from the set X, where X_k is the subset of alternatives chosen by the expert e_k, $X_k \subseteq X$, $e_k \in E$, $k = 1, ...K$.

So, each expert e_k provides the symbolic estimates for all pairs (x_i, x_j), according to their own scale (LTS), such that $i \neq j$ and $x_i, x_j \in X_k$. Expert's assessment shows how much the alternative x_i is better than the alternative x_j.

Denote the preference value of the expert for the alternative x_i with respect to the alternative x_j by the variable $p_{ij}(k)$.

The obtained values $p_{ij}(k)$ form the preference matrix $\boldsymbol{P}_k = (p_{ij}(k))_{i,j=1,...,M}$. For non-estimated alternatives, the symbol "0" is entered in the matrix in the appropriate position.

In addition, each expert may have their own scale of assessments of the level t (LTS): $S(t) = \{s_1(t), ..., s_{n(t)}(t)\}$, where $s_i(t)$ is the symbolic assessment of the scale t, $i \in \{1, n(t)\}, n(t)$ is the number of assessments, that is, $|S(t)| = n(t)$. Also, it should be noted that all assessments are arranged in ascending preferences: $s_1(t) \prec s_2(t) \prec ... \prec s_{n(t)}(t)$.

Linguistic expression set consists of symbolic assessments, according to which experts compare alternatives:

$S(1) = s_1(1)$: very bad, $s_2(1)$: bad, $s_3(1)$: normally, $s_4(1)$: good, $s_5(1)$: very good, $n(1) = 5$;

$S(2) = s_1(2)$: awful, $s_2(2)$: very bad, $s_3(2)$: bad, $s_4(2)$: normally, $s_5(2)$: good, $s_6(2)$: very good, $s_7(2)$: excellent, $n(2) = 7$;

$S(3) = s_1(3)$: awful, $s_2(3)$: very bad, $s_3(3)$: bad, $s_4(3)$: below the average, $s_5(3)$: normally, $s_6(3)$: above the average, $s_7(3)$: good, $s_8(3)$: very good, $s_9(3)$: excellent, $n(3) = 9$.

That structure have solved the problem of misunderstanding between a person and a computer. To convert the symbolic assessment to the numerical value one of the well-known methods can be used [17]. We omit the method description since it is not the subject of our paper.

Accordingly, the matrix $\boldsymbol{P}_k$ is modified into a matrix $\boldsymbol{V}_k=(v_{ij}(k))_{i,j=1,\dots,M}$, which consists of numerical values, such that $(s_i(t), 0) \to i$, $v_{ij}(k) = 1, \dots, n(t)$.

Then, in order to form a matrix of average preference values based on information obtained from all experts, it is necessary to select a basic linguistic term set (BLTS) and convert the value from other scales to basic.

Any available LTS can be selected as a BLTS, and by default, the largest scale is selected, assuming that the scales are ordered by increasing their power, i.e. $S(1) \prec S(2) \prec \dots \prec S(T)$.

To convert estimates from LTS of the level t to BLTS of the level t', we use the method described in detail in [6]:

if $S(t) \prec S(t')$ and $\frac{\beta(t')}{\beta(t)} = \frac{n(t')}{n(t)}$, then $F : B(t) \to B(t')$ mapping is one-to-one and is determined by the formula (1).

$$F(s_{[\beta(t)]}(t), \beta(t) - [\beta(t)]) = \Delta(\frac{\Delta^{-1}(s_{[\beta(t)]}(t), \beta(t) - [\beta(t)])}{n(t)}, \tag{1}$$

where $\beta(t)$ denotes the shift relative to the nearest integer value in accordance with the symbolic estimate.

In view of the foregoing the GDM process includes 2 steps.

1) Aggregation of information received from all experts. Individual preferences of experts are aggregated into the matrix of collective preferences (2). Some problems may require making a decision in specified restrictions - depending on certain conditions and situations. To take into account all sorts of restrictions, numerous methods for aggregating estimates have been developed, for example, aggregating operators. One of these operators is an ordered weighted average (OWA), as well as its various interpretations. In our paper we use the simple (i.e. not weighted) average. $\overline{\boldsymbol{P}}=(\overline{p}_{ij})_{i,j=1,\dots,M}$, the elements of which are calculated as follows

$$\overline{p}_{ij} = \frac{\sum\limits_{k=1}^{K} p_{ij}(k)}{\sum\limits_{k=1}^{K} n_{ij}(k)}, \tag{2}$$

where

$$n_{ij}(k) = \begin{cases} 1, & \text{if} x_i, x_j \in X_k, \\ 0, & \text{elsewhere}. \end{cases}$$

2) Exploitation of the information received in step 1. This step forms the final ranking of alternatives taking into account experts' preferences.

We can perform the ranking of alternatives using the mean value between the operator GDD (Quantifier Guided Dominance Degree) and GNDD ((Quantifier Guided Non Dominance Degree).

The calculation of the GDD (3) and GNDD (4) is carried out according to following formulas:

$$GDD_i = \sum_{j=1}^{M} \overline{p}_{ij}, i = 1, ..., M; \tag{3}$$

$$GNDD_i = \sum_{j=1}^{M} max\{\overline{p}_{ji} - \overline{p}_{ij}, 1\}, i = 1, ..., M. \tag{4}$$

So, we got the Ranking Value. It should be noted, the higher the value, the better the alternative is considered.

This information is necessary to understand the purpose of developing a chatbot. We do not give all the formulas of the method of ranking alternatives in the group decision making process [6], since this work is focused on obtaining expert assessments, specifically we are developing an environment in which experts better express themselves in presenting preferences.

2.2. Chatbots

A chatbot is an artificial intelligence software that can imitate a conversation with a user in natural language using messaging tools, websites or mobile applications [11, 12]. Simply, this is an account that is managed not by people, but by software. The chatbot works with a less confusing web and mobile application that is easy to install because there is no need for installation additional packages. Bots are completely different from human accounts because they do not have online status.

The benefits of such services include efficiency: artificial conversational entity can combine the steps of complex processes to optimize and automate routine and repetitive tasks through several simple voice or text requests, reducing execution time and increasing the efficiency of the task.

Virtual interactive agents can also be deployed in platforms that potential experts have already used such as Facebook or Twitter, so it is possible to contact with users in a familiar environment, which increases convenience and makes expert participation in the process more comfortable and convenient.

Finally, chatbots can simplify and speed up the process of grading or expressing opinions from a mobile device, browser or any convenient platform. Services maintain context and control dialogue by dynamically customizing responses based on conversation.

Regardless of the created computer program and platform, human intervention is crucial in setting up, training and optimizing the system. In order to achieve the desired results, a combination of different forms of artificial intelligence, such as natural language processing and machine learning, may be the best choice.

Chatbot is currently described as one of the most advanced and promising ways of interaction between people and machines. From a technological perspective, such a program is a natural evolution of the system of answering a question using natural language processing (NLP) [13, 15].

3. Proposed chatbot system for the group decision making process

In this work, we offer a solution that can be launched inside the Telegram application. We arrived at decision to work in this cross-platform messenger due to the rich user interface of the platform bots.

The algorithm of the chatbot in Telegram is quite simple (Fig. 2).

Messages, commands and requests that are sent by users transmit to the software running on the developers' servers. Intermediary Telegram Server Webhook handles

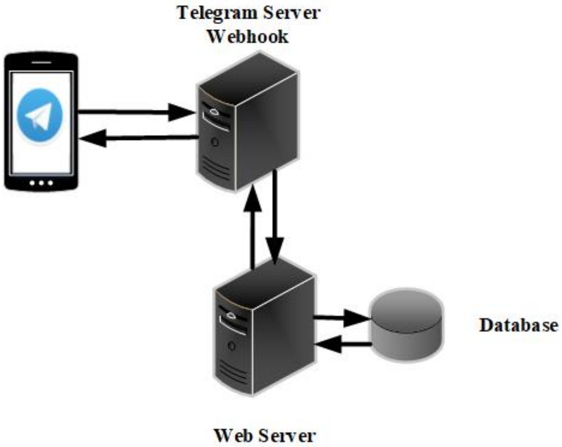

Figure 2. The architecture model of the chatbot

encryption and provides feedback between the utility and the user. The messenger uses Webhook technology for authentication and sending notifications about events to your application. From a programming point of view, it comes down to the work of ordinary callback functions for processing HTTP requests that will receive data about events, such as messages received by a chatbot.

Consider the work of the application from a conceptual point of view: when a user interacts with a chatbot in Telegram, the API (the Application Programming Interface) sends information about the activity to the code using an HTTP request, and then the code sends information about how the program should respond (see Fig. 3).

Thus, the Bot API is an intermediary between the Telegram bot and the application logic. It consists of two main parts: updates and methods. The developer receives updates reflecting user interaction with the bot. While calling methods are necessary for a bot to perform predefined actions, such as sending messages to users.

To create a chatbot, it is necessary:

a) Download the Telegram application to computer, phone or to another device.

Telegram is primarily a mobile software tool, but for development purposes, it is possible to install it on the computer where the code is written. Thus, it is a quickly way to check the correctness of the work.

b) Get an API key using @BotFather.

After sending the command /newbot, the "name" and the "username" for the bot should be selected. The name is what other users see in their contact list, and the username is how they find it.

After completing the steps above, we got an API key. The API key is how Telegram finds out that the code we wrote is associated with a specific bot. Each chatbot is assigned a unique API key.

If the name is taken, the bot will ask to enter a new one, if not, it will issue a token to access the API.

Next, should copy this token to the Telegram driver configuration file. After that start working with the driver. Another way is to customize bot settings in the bot @BotFather, including the icon that will be displayed to users.

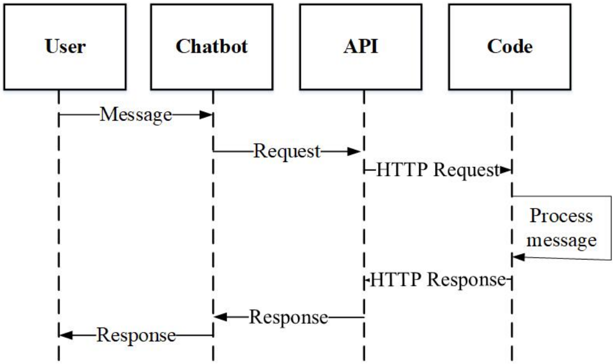

Figure 3. A sequence diagram describing the response of the chatbot to an incoming user message

c) Set up the application configuration (see listing below) and proceed to writing the code.

Created by the authors of the article bot’s name is GDMbot. The purpose of developing this virtual assistant is to facilitate the process of interaction of the expert with the system. The bot configuration code with the user name @GroupDecisionMakingBot is shown below.

```
<?php
$BOT_NAME = ’GroupDecisionMakingBot’;
define(’BOT_TOKEN’, ’774803280:AAHKYDfHlJe_ghDmTjAR6HHZ56FKk89QGC4’);
define(’API_URL’, ’https://api.telegram.org/bot’.BOT_TOKEN.’/’);
define(’WEBHOOK_URL’, ’https://detsadbot.ru/opros_bot/bot.php’);
$id_admins = array(3238601, 293277190);

$arr_alternatives = array(

’Vkontakte’,
’Twitter’,
’Odnoklassniki’,
’YouTube’,
’MoyMir’,
’BabyBlog’,
’GooglePlus’,
’WhatsApp’,
’Viber’,
’Social Network from A to Ya’,
’MySpace’,
’LinkedIn’,
’Professionale’,
’MySpace’,
’LiveJournal’,
’Spaces’,
’AlterEgo’,
’Shararam’,
’Moya Shkola’,
’Kindernet’,
```


```
’Telegram’,
’Instagram’,
’DeviantArt’,
’Ask.fm’,
’Snapchat’,
’Vine’,
’Tumblr’,
’Facebook’
);

$db_hostname = ’localhost’;
$db_username = ’***************’;
$db_password = ’***********’;
$db_basename = ’***************’;
$user_table = ’opros_users’;
$data_table = ’opros_users’;
?>
```

We want the expert to feel as comfortable as possible in the rating process. Therefore, when writing a chatbot program, the authors planned that in order to communicate with the bot, the user would not have to type the message text, since the bot interface would provide a set of user buttons. To do this, we implement a virtual menu.

After all the code is ready, ought to register the hosting and add the domain name to it, transfer the script to the database and start the chatbot. At this stage, it is needed to have a web host. It is most convenient to create a separate subdomain for the bot - for example, bot.example.com.

4. Experimental results

The authors of this very article have developed the @GroupDecisionMakingBot for the group decision making process. Computer program of chatbot will help experts to express their opinions and set ratings in any place and at any time, using a mobile device, tablet or personal computer.

Developing the concept described in [6], we propose to automate the entire process of group decision making. According to the method of this article, a software tool has

Figure 4. Chatbot interface: Chatbot response snapshots

already been developed that defines ranked lists of alternatives. However, the bulk of the time was spent communicating with experts and receiving assessments. We also noticed that the experts were tired in the process of providing preferences, which adversely affected the reliability of the data obtained. Therefore, we consider this work to be the completion of the full automation of the entire group decision making process.

To pass a survey and provide estimates to the system, each expert needs to find our chatbot in Telegram by its name, more specifically, call @GroupDecisionMakingBot.

After the greeting, the user will be asked to participate in the survey and compare several of the 27 social networks pairwise. Experts can manage their actions using the "Forward" / "Back" buttons, so the expert may not be afraid of random errors, since it is always possible to correct any answers.

The second step is the definition of the user rating scale: the system offers 3 options, discussed in detail in [4, 6].

The final step is a pairwise comparison of the selected alternatives: the bot automatically displays a couple of alternatives and user response options, according to the rating scale. Note that the expert does not need to remember which social networks he chose, because the interactive system will independently request all the necessary information.

Finally, the bot sends to the creator of the application the results obtained from the experts. Further alternatives can be ranked using a previously developed software tool [6].

5. Features of chatbot use in the group decision making process

The @GroupDecisionMakingBot has the following advantages:
- simple and interactive real-time chat system;
- an effective way for experts to interact with the system;
- providing preferences anywhere and anytime;
- ability to send survey results to the administrator;
- work in cross-platform devices;
- easy integration and upgrade of the application;

– unlimited members.

The advantages listed above make the chatbot unique from the point of view of its use in the group decision making process, since in this area a set of such characteristics is implemented for the first time.

6. Conclusions

The novelty of this paper consists in that a chatbot is used in the group decision making process. Moreover it becomes possible for an unlimited number of experts to participate, and also provides comfortable conditions for their work according to advanced technologies.

In the future, it is planned to develop a software tool that can recognize speech and the image of experts to provide an environment in which experts will communicate with a machine like with people.

Acknowledgments

The publication has been prepared with the support of the “RUDN University Program 5-100” and funded by RFBR according to the research projects № 18-07-00576 and № 18-00-01555 (18-00-01685).

References

1. Scott, J. Social network analysis. Sage, 2017.
2. J. Kacprzyk, Group decision making with a fuzzy linguistic majority, Fuzzy sets and systems 18, 105–118, 1986.
3. E. Herrera-Viedma, F. J. Cabrerizo, J. Kacprzyk, W. Pedrycz, A Review of Soft Consensus Models in Fuzzy Environment, Information Fusion, 17:4–13, 2014.
4. E. Herrera-Viedma,F. J. Cabrerizo, F. Chiclana, J. Wu, M. J. Cobo, K. Samuylov, Consensus in Group Decision Making and Social Networks, Studies in Informatics and Control 26:3, 259-268, 2017.
5. J. A. Morente-Molinera, I. J. P´erez, M. R.Ure˜na, E. Herrera-Viedma, On multi-granular fuzzy linguistic modeling in group decision making problems: a systematic review and future trends, KnowledgeBased Systems 74, 49–60, 2015.
6. N. Chukhno, O. Chukhno, A. Gaidamaka, K. Samuylov, E. Herrera-Viedma, A New Ranking Method of Alternatives for Group Decision Making in Social Networks, 10th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), 2018.
7. V. Begishev, R. Kovalchukov, A. Samuylov, A. Ometov, D. Moltchanov, Y. Gaidamaka, S. Andreev, An analytical approach to SINR estimation in adjacent rectangular cells, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9247, 446-458, 2015.
8. A. Samuylov, D. Moltchanov, Y. Gaidamaka, S. Andreev, Y. Koucheryavy, Random Triangle: A Baseline Model for Interference Analysis in Heterogeneous Networks, IEEE Transactions on Vehicular Technology, vol. 65, no. 8, art. no. 7275184, 6778-6782, 2016.
9. K.E. Samouylov, P.O. Abaev, Y.V. Gaidamaka, A.V. Pechinkin, R.V. Razumchik, Analytical modelling and simulation for performance evaluation of sip server with hysteretic overload control, Proceedings - 28th European Conference on Modelling and Simulation, ECMS 2014, 603-609, 2014.
10. Y. Gaidamaka, A. Pechinkin, R. Razumchik, K. Samouylov, E. Sopin, Analysis of an MG1R queue with batch arrivals and two hysteretic overload control policies, International Journal of Applied Mathematics and Computer Science, 24 (3), 519-534, 2014.

11. S. A. Abdul-Kader, J. Woods, Survey on Chatbot Design Techniques in Speech Conversation Systems, International Journal of Advanced Computer Science and Applications, 6 (7), 2015.
12. M. Dahiya, “A tool of conversation: Chatbot, International Journal of Computer Sciences and Engineering, vol. 5, no. 5, 158–161, 2017.
13. T. Kluwer, From chatbots to dialog systems, in Conversational agents and natural language interaction: Techniques and effective practices. IGI Global, pp. 1–22, 2011.
14. S. Sayed, R. Jain, B. Lokhandwala, F. Barodawala and M. Rajkotwala, Android based Chat-Bot, International Journal of Computer Applications, Vol. 137, No. 10, 29-32 , 2016.
15. F. A. Mikic Fonte, M. L. Nistal, J. C. Burguillo Rial, and M. C. Rodríguez, NLAST: A natural language assistant for students, IEEE Global Engineering Education Conference (EDUCON), 709-713, 2016.
16. M. N. Kumar, P. C. L. Chandar, A. V. Prasad and K. Sumangali, Android based educational Chatbot for visually impaired people, 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, 1-4, 2016.
17. V. Torra, Hesitant fuzzy sets, International Journal of Intelligent Systems, 25(6), 529-539, 2010.