System for Definition of Indicator Characteristics of Social Networks Participants Profiles

Oleg Bisikalo[0000-0002-7607-1943], Anton Kontsevoi

Department of Automation and Information-Measuring Technique of Vinnytsia National Technical University Vinnytsia, Ukraine

obisikalo@gmail.com, anton.96k@gmail.com

Abstract . The system that implements the process of automated determination of indicator characteristics of social network profiles, as well as determining the response of social network participants to certain events has been developed and tested on the real text examples and data. The developed system showed that proposed approach has substantially improved the accuracy and the amount of the characteristics of user profiles inside the Twitter social network by using the approaches that combine various natural language processing models and technologies.

Keywords : model, associative imaginative thinking, social network, participants profiles, indicators, users classification, android, mobile application.

1 Introduction

Social networks are very actual research field in computer sciences today because a vital information for most people on the planet is already in the digital form. Consider an approach to determining the indicator characteristics of profiles of participants in social networks (SN) based on the modeling of some generalized image. Typically, collecting data to solve such and similar problems is sought by 1) determining the set of significant (indicator) characteristics of related objects (profiles of SN participants), 2) obtaining values of such characteristics for one object, 3) accumulating statistics for values for of all objects under study. The next, fourth stage should ensure that you gain useful knowledge of statistics in one way or another - for example, statistical analysis, mashing learning, deep learning. The result is a breakdown of the primary set of objects into specific classes or categories [1]. The choice and effectiveness of a method depend, as a rule, on the composition and types of data being analyzed.

We take advantage of the fact that a large part of the indicator characteristics of the profiles of SN participants is verbally assigned, that is, belongs to a plurality of words of some natural language. The idea behind the proposed approach is to use linguistic methods of analyzing the plurality of words that characterize a particular profile of a SN participant in order to obtain a concise natural-language description of that participant, understandable to humans. We formalize the problem by using the model of associative imaginative thinking of the person, first proposed in [2], in particular, simulating the processes of solving a problem by a person.

The subject of formalization will be certain mnemonic processes inherent in man. In [3], in order to formally solve this class of problems, we propose to apply algebraic models of the artificial mechanism of the operative memory of an artificial linguistic system. The actual task in the development of the model [3] is to determine and detail the image-solution of the problem situation by using a group of formal operations to the ensemble of images and modeling the orienting human reflex. For the first time, an algebraic solution to this problem was obtained in [4], which considered the processes of generalizing verbal information as a whole.

In terms of the model of figurative thinking, a basic synthesis operation provides, according to Vygotsky [5], the influence (infusion) of the meaning of a particular event or situation into one image. Such an operation occurs in human memory, in particular on the basis of verbal features of the image, if the event (situation) became known through textual description. For example, a newspaper editor reads the news of an event and should give the message its own name - concise but striking and appealing to the reader.

2 Formal Model

We show the fundamental possibility of constructing a resolution image within the model of the mechanism of memory [4]. We will assume that previously N verbal contender images have already been selected into the Check – Set memory stack. Then it is necessary to definitively determine the components of the vector of emotions Vector – Set (otherwise - such verbal images that correspond to positive signs of the situation) and to select one image with N with the highest weight. The formal formulation of the problem of constructing an image of solving a problem situation will be considered

$$\begin{aligned}&Check-Set, Vector-Set \to Focus-Bi \,|\\ &Focus-Weight = \underset{i\in N}{Max}(Weight_i),\end{aligned} \tag{1}$$

where *Focus–Bi* is binary image code in focus of attention; *Focus–Weight* is the weight of the associative relationships of the image in the focus of attention with the images-components of the emotion vector *Vector–Set*; *Weight*$_i$ is the weight of associative relations of the i-th image of the image ensemble (IA) with the images *Vector–Set*.

Formally, we will look for the synthesis or construction of a solution image as an *InsZX* statement similarly to work [6]. The main sources of information for the model are short texts of the SN users, and according to [4] we use the following notation: *Bi*–

Vector$_i$ is binary code of the i-th image of the emotion vector *Vector–Set*;*Bi–OM* is binary code of IA RAM.

We add to the algebraic *BAS* system [3] formal operations and predicates on the boulevards that meet the problem:

1. Search operation of *Seek i-1* unit *(s)* in *Bi-OM* binary code and *End-Bi* predicate, which is true when not all bits of code have been viewed

$$Bi-OM \xrightarrow{Seek"1"(i)} k\,, \tag{2}$$

$$i \le n \to End-Bi\,, \tag{3}$$

where *i* is the number of the image in the RAM memory; *k* is the unit number of the binary code corresponding to the *i*-th image of the IA; *n* is the number of units in binary code (current RAM).

2. Operation *New–OM* transfer of images-components of emotion vector *Vector–Set* in IA RAM:

$$\bigcup_{i=1}^{n} Bi-Vector_i \xrightarrow{New–OM} Bi-OM\, \cdot \tag{4}$$

3. Operation *New–Vector* insertion into the vector of emotions *Vector – Set* values from the *Choice – Set* stack:

$$Choice-Set \xrightarrow{New–Vector} Vector-Set\,. \tag{5}$$

4. Operation *New–Choice* uploading to the *Choice – Set* stack of *RAM* memory:

$$Bi-OM \xrightarrow{New–Choice} Choice-Set\,. \tag{6}$$

In graph diagrams for algebraic constructs, we include the following notations of structural programming operators [7]:

Do – loop by parameter or condition;

* – composition.

Consider the solution of problem (1) on the basis of such constituent operators.

1. Operator *Pyramid–Images* of binary code decomposition *Bi–OM* to the codes of the image-components. As a result of the action of the operator presented in Figure 1, the well-known *Bi – OM* code defines the codes of the images that are part of the IA.

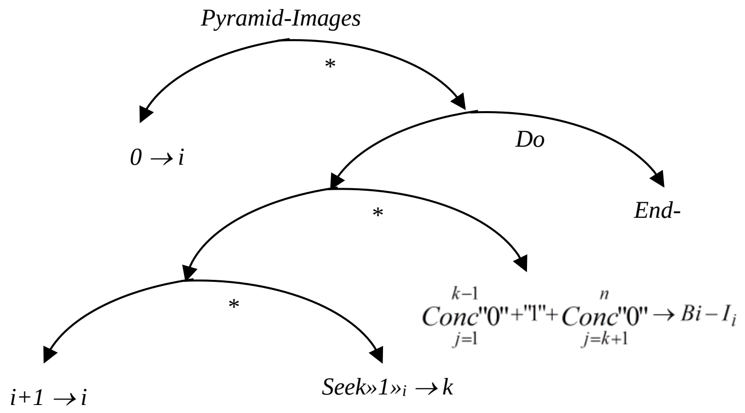

Pyramid-Images
*
0 → i
Do
End-
*
$\overset{k-1}{\underset{j=1}{Conc}}"0"+"1"+\overset{n}{\underset{j=k+1}{Conc}}"0" \to Bi - I_i$
*
i+1 → i
Seek»1»$_i$ → k

Fig. 1. A graph of the operator *Pyramid-Images*.

For this purpose, a cycle along the length of the *Bi – OM* code is organized, if there is a unit, then another element with a unit at this position and zeros on all others is added to the list of images (the cycle is limited by the number of units *n* in the *Bi – OM* binary code):

$$\begin{gathered} Pyramid - \mathrm{Im}\,ages ::= 0 \to i * \{[End - Bi] \quad (i+1 \to i * \\ Seek"1"_i \to k * \overset{k-1}{\underset{j=1}{Conc}}"0" + "1" + \overset{n}{\underset{j=k+1}{Conc}}"0" \to Bi - I_i)\}, \end{gathered} \tag{7}$$

where $Bi–I_i$ is the binary code of the *i*-th image in RAM; *Conc“0”* is concatenation of "0" characters into binary code digits.

2. Operator *OM–Change–Vector* (figure 2) simulates the launch of the mechanism of constructing the image of the untying, because *New–Choice, New–OM* and *New–Vector* operations are performed sequentially:

$$\begin{gathered} New - Change - Vector ::= New - Choice * \\ New - OM * New - Vector. \end{gathered} \tag{8}$$

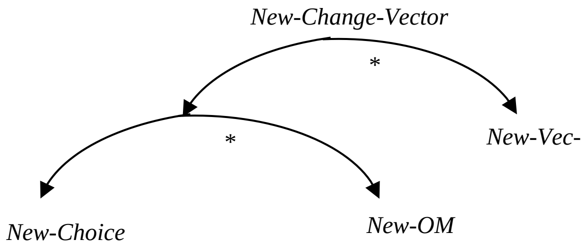


Fig. 2. A Graph of the operator *OM-Change-Vector*.

3. The *InsZX* unblocking constructor (Figure 3) runs the corresponding *OM – Change – Vector* mechanism and submits its results to the *Orient – Reflect* Oriented Reflex model of p.4.5.3.:

$$InsertZX ::= OM - Change - Vector * Orient - \mathrm{Re}\, flect \cdot \qquad (9)$$

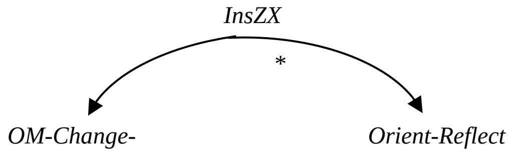


Fig. 3. A Graph of the operator *InsZX*.

The algorithm for constructing the image of solving a problem situation is implemented within the model of the orienting reflex [4] with the help of algebraic operators (2) - (9), as well as programmatically implemented and tested on the basis of Kotlin + Apache OpenNLP technology.

Let us illustrate the possibilities of the proposed approach by solving problem (1) for experimental data of a cross-sectional test example "Semantic WEB Standards" from [4]. Five linguistic images (LI) were selected to the *Check – Set* memory stack - "Ontology (111)", "XML (88)", "Tool (93)", "Resource (73)", "RDF (87)", and the constituents of the *Vector – Set* emotions vector have been set up by LI (56), Information (34) and Network (17).

Formally, in accordance with the statement of problem (1), we define *Check–Set=011111000000*, *Vector–Set={000000010000, 000000000010, 000000000001}*. As a result of the action of the *OM-Change-Vector* operator, the *Vector – Set* linguistic images are transferred to the IA and replaced by the LI from the *Check – Set*. The *Orient – Reflect* oriented reflex operator consistently determines the weights for all linguistic images (*Bi – OM* = 000000010110), which ultimately results in the largest of them being *Focus – Weight* = 8 for the LI language (56). However, even with the limited natural-language material of the cross-cutting example, the "network (17)" is gaining weight 4, which makes it a competitor to the "language (56)" LI in generalizing meaningful concepts of the Semantic WEB Standards topic.

Thus, based on a formal look at the phenomena of the ensemble of images, emotions vector, focus of attention and orientation reflex, the possibility of constructing an image of solving a problem situation by means of an algebraic system is shown. Unlike works [8, 9] the main source of information for the model is a textual description of the problem situation, which determine its features. The proposed approach to operator construction is based on already known models [3, 4] and provides an invariant representation of the content of a brief description of a situation, which is proved on the basis of a through test example [10].

3 Application Development

New application based on the model for determine and detail the image-solution of the problem situation is proposing. The program fulfills the tasks of analyzing the reactions of users of the social network Twitter to current news, as well as determining indicator characteristics of the profiles of participants in this social network.

This development is planned to be used to collect statistical information about the response of users to an event that occurs in real time, as well as the analysis of indicator characteristics of network user profiles. The goal is to analyze statistical information and display it in the form of histograms and tables. Data collection, analysis and processing takes place on the device itself, which increases the security of the data the user is working with [11].

Like the vast majority of programs for Android, the program consists of a graphical interface and other application resources (graphical, text and sound resources), as well as the business logic of the application.

For the program to work, you need to configure the access settings for the Twitter social network API. These parameters are key: value pairs:

- consumer key;
- consumer secret;
- access token;
- access secret.

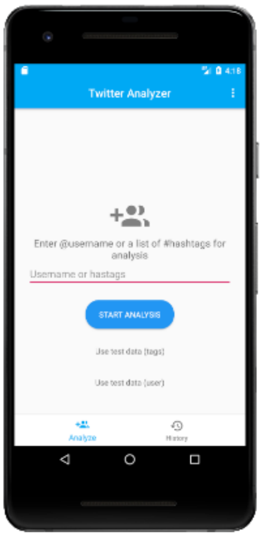
These values are used when sending requests to the servers of the social network and necessary for the process of collecting information. To get these parameters, you must have a user account with developer privileges on Twitter. These parameters are unique for each user and allow identifying his program when accessing data on the server Data collection, analysis and processing takes place on the device itself, which increases the security of the data the user is working with [12].

User interaction with the program starts from the main screen. In the center of the screen, there is an input field where the user can enter a user nickname whose profile he wants to analyze or a hashtag to which some discussion on the network is attached. Nicknames should be entered with the use of the symbol "@" in front of the nickname itself without spaces between them. In case it is necessary to analyze the reaction to the news, the user enters the corresponding hashtag starting with the input symbol "#".

After that, the user can click on the “Start analysis” button, which will start the process of analyzing user reactions to events on the network or analyzing the profile of the network member depending on the input.

In addition, on the screen there are two additional buttons “Use test data (tags)” and “Use test data (user)”. Each of them starts the process of analyzing relevant content using pre-stored data from the Twitter network. This is necessary if the user wants to see the result of the work of the program, but does not have a Twitter developer account and therefore cannot access its API. In this case, the program will not access the Twitter API, but will use pre-prepared response files from Twitter servers.

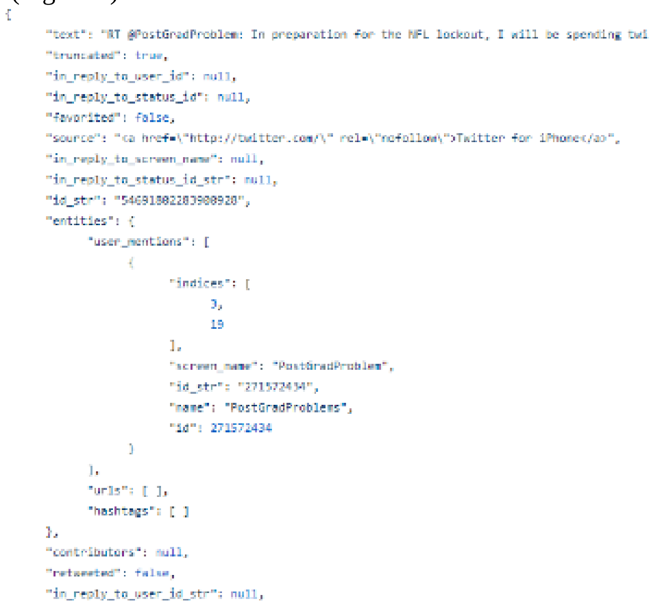
The main screen of the application is shown in Figure 4.



Enter @username or a list of #hashtags for analysis

Fig. 4. The main screen of the application.

When entering a username or hashtag, the system first calls the Twitter API. The corresponding module of the program sends the locks to the server using the previously specified access parameters, as well as the user's request. The server returns data in JSON format (Figure 5).



```
{
        "text": "RT @PostGradProblem: In preparation for the NFL lockout, I will be spending twi
        "truncated": true,
        "in_reply_to_user_id": null,
        "in_reply_to_status_id": null,
        "favorited": false,
        "source": "<a href=\"http://twitter.com/\" rel=\"nofollow\">Twitter for iPhone</a>",
        "in_reply_to_screen_name": null,
        "in_reply_to_status_id_str": null,
        "id_str": "54691802283900928",
        "entities": {
                "user_mentions": [
                        {
                                "indices": [
                                        3,
                                        19
                                ],
                                "screen_name": "PostGradProblem",
                                "id_str": "271572434",
                                "name": "PostGradProblems",
                                "id": 271572434
                        }
                ],
                "urls": [ ],
                "hashtags": [ ]
        },
        "contributors": null,
        "retweeted": false,
        "in_reply_to_user_id_str": null,
```

Fig. 5. The example JSON response from the Twitter API.

At the stage of parsing each reaction of the user, the text of the reaction itself is extracted and divided into tokens, after which these tokens are added to the list of tokens. The list of tokens is necessary for further text processing, since many other text processing methods are needed in preliminary tokenization.

Next, the program determines the names of people, locations and dates that are in the text of tweets using the Named Entities Recognition API library Apache OpenNLP.

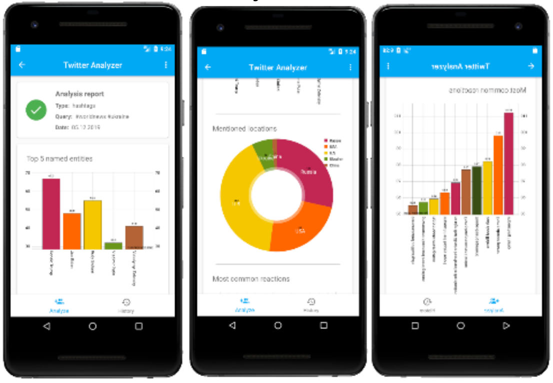
Then, using the Chunker API, the program selects keywords and phrases (sets of words combined in meaning and grammatically) in tweet texts. For this function to work, the program needs a list of tokens, as well as the result of the POS Tagger API, whose task is to determine the parts of speech for each word in the incoming text. This will allow the Chunker API to identify words and phrases related in meaning in the text, which in the end result will help determine the reaction of a particular user to an event on the network, as well as to the general reaction of users.

In the case of analyzing the profile of a Twitter participant, the Language Detector API is additionally applied, which, based on the model embedded in it, recognizes the language of the text and presents a list of languages that it was possible to recognize together with the probability coefficient that this language is the language of the text.

To study and analyze the reactions of users to events, the responses of users on the Twitter network regarding events in the network united by the tag “#ukraine” were selected as input data for analysis, which means that if a given tag or word is present on a tweet, it will fall into search results.

To conduct the study, 1,500 user responses in English were collected, since the recognition model works with English speech. This figure is small for real analysis, since it covers only a small percentage of network users' reactions, but it is dictated by the limitations of the Twitter API. This limitation can be circumvented by sending requests at fixed intervals, after which the request counter is reset to zero and new requests can be sent. However, this approach will not work in the context of a mobile application, since even if it will collect data on demand in the background for a certain period of time, there is a risk that the system will stop the background process. For the test needs, this number of tweets is enough, because the goal is not to collect the most accurate statistics, but to demonstrate an approach to solving the problem.

Figure 6 shows the results of an analysis of user reactions to events on the network.



Type: hashtags
Query: #worldnews #ukraine
Date: 05.12.2019

Fig. 6. User response to events on the network.

The user can interact with the graphs, zooming in and scrolling through them to see all the data.

In the course of processing the results of the analysis of user reactions, the program identified the five most common names in the text of tweets, they turned out to be "Donald Trump", "Joe Biden", "Rudy Giuliani", "Vladimir Putin", "Volodymyr Zelensky" (Figure 7). With small amounts of data (1-5 thousand tweets), there is no need to display more results, since often the names of other people are rarely found in tweets, more often they are in the @nickname format.

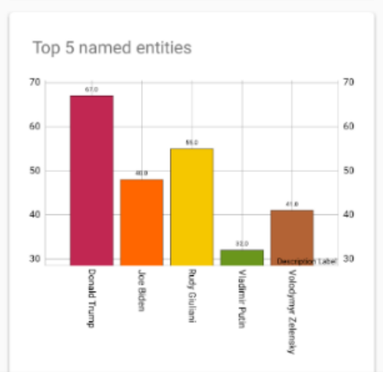


Fig. 7. The diagram of the most frequently encountered names in tweet texts.

Figure 8 shows a pie chart with the mentioned locations.

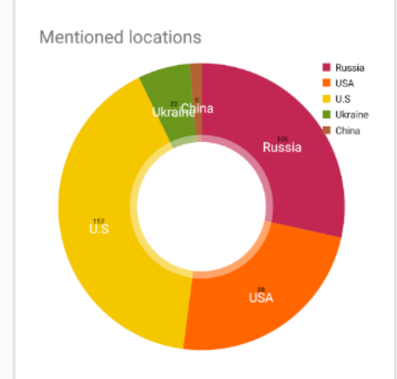


Fig. 8. The most frequently mentioned locations.

Named Entity Recognition API was able to highlight the names of locations in the text, in this case, the names of the most mentioned countries. It is worth noting that, despite the high accuracy of the results obtained, the model may not determine the names of little-known cities, which may be a problem when analyzing more “local” events in the network.

Figure 9 shows a diagram with the most common keywords and phrases in the texts of reactions to news on the network.

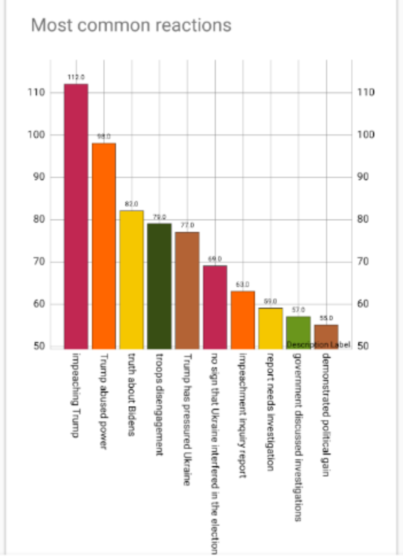


Fig. 9. The most common keywords and phrases.

It is worth noting that the approach using Tokenization, POS Tagging and Chunking API gives much greater accuracy and greater semantic load embedded in the final results compared to using Tokenization and Tuples (for generating pairs of keywords in the text). The data approach works well in English, since it has a strict word order in a sentence, therefore there is a high probability of finding matches among the received words and phrases.

To study and analyze the profile of a member of a social network, the profile of the current US president was chosen, as well as his tweets in the amount of 2500 pieces. This figure is small for real analysis, since it covers only a small number of recent user tweets and is dictated by the limitations of the Twitter API.

Figure 10 shows the results of the analysis of the profile of a member of the Twitter network.

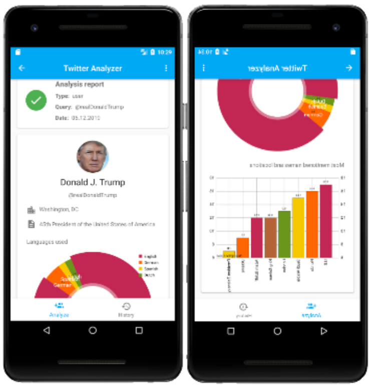


Fig. 10. Results of the analysis of the profile of a member of the social network Twitter.

The analysis results show that the system recognized four languages that were supposedly used when writing tweets by a user. The result with highest balls was the accuracy of the English language, which is true. The remaining languages were recognized, since the text could contain names, names or locations that are written similar to these languages, as well as due to errors in the recognition model. Figure 11 shows the pie diagram with the languages detected in the user tweets.

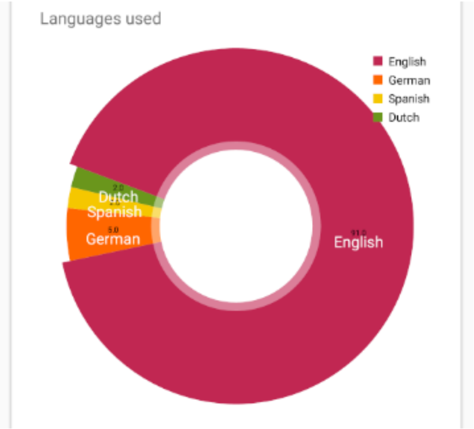


Fig. 11. The languages detected in the user tweets.

We also analyzed the most mentioned places and personalities on the same principle as when analyzing user reactions to events on the network. These results were put on one chart in order to determine the dependence between the name and location.

With this approach, between related pairs of names and locations, the difference in the frequency of references will be insignificant. However, in the case of this user and data set, this dependence was not detected.

Summing up, it can be argued that the created program successfully fulfills the tasks of analyzing the profiles of members of a social network, as well as determining the response of users to events in it.

In the course of the work, a new coordinated approach to solving the above problems was proposed, implemented using the Apache OpenNLP library. It made possible to increase the accuracy of the received results, which was confirmed by the testing results of the developed program.

4 Conclusion

The system for implement the process of automated determination of indicator characteristics of social network profiles, as well as determining the response of social network participants to certain events has been proposed in the article. Based on a formal look at the phenomena of the ensemble of images, emotions vector, focus of attention and orientation reflex, the possibility of constructing an image of solving a problem situation by means of an algebraic system is shown. The architecture and main modules of the proposed system were developed and tested on the real text examples.

To develop the software for solving the goal of research, the Kotlin programming language was chosen because of it and its ease of use when creating Android applications, as well as Apache OpenNLP library for parsing data about news and network users.

So, the developed system showed that proposed approach has substantially improved the accuracy and the amount of the characteristics of user profiles inside the Twitter social network by using the approaches that combine various natural language processing models and technologies.

According to the authors, the further development of the research subject is the study of different natural language processing models, which can be used as base of the proposed system for definition of indicator characteristics of social networks participants' profiles.

References

1. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning. Springer Science+Business Media, New York (2013).
2. Bisikalo, O., Cięszczyk, S., Yussupova, G.: Solving problems on base of concepts formalization of language image and figurative meaning of the natural-language constructs. In: Proc. SPIE, vol. 9816, Optical Fibers and Their Applications, pp. 419–432 (2015).
3. Bisikalo O., Ivanov, Y., Karevina, N.: Encoding of Natural Language Information on the Basis of the Power Set. In: 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 17-20. Lviv, Ukraine (2018).
4. Bisikalo, O., Ivanov, Y., Sholota, V.: Modeling the phenomenological concepts for figurative processing of natural-language constructions. In: CEUR Workshop Proceedings, vol. 2362, pp. 1–11 (2019).
5. Vygotskiĭ, L.: Thought and language. MIT press, Cambridge, MA (2012).
6. Lund, K., Burgess, C., Audet, C.: Dissociating semantic and associative word relationships using high-dimensional semantic space. In: Cognitive Science – COGSCI, vol. 18, pp. 603-608 (1996).
7. Eisenbud D.: Commutative Algebra with a View Toward Algebraic Geometry. Springer–Verlag, New York, NY (1995).
8. Zaki, M.: Scalable algorithms for association mining. In: IEEE Transactions on Knowledge and Data Engineering, vol. 12, issue 3, 38 p. (2000).
9. Cabrera, I.: A note on the envelopes of an associative pair. In: Comm. Algebra, vol. 32, pp. 4133–4140 (2004).
10. Mitkov, R.: The Oxford Handbook of Computational Linguistics. Oxford University Press (2005).
11. Lee, V.: Mobile Applications: Architecture, Design, and Development. Prentice Hall (2004).
12. Murphy, M.: The Busy Coder's Guide to Advanced Android Development. In: CommonsWare, LLC (2011).