UDC 004.89 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait/1995-5499/2022/2/145-156
Received 17.02.2022
Accepted 29.06.2022

INTELLIGENT ANALYSIS SYSTEM FOR MONITORING THE SPREAD AND INFLUENCE OF POLITICAL INTERNET MEMES

© 2022 M. G. Kozlova✉[1], V. A. Lukianenko[1], P. A. Gurzhiy[1], O. O. Makarov[2]

[1]V. I. Vernadsky Crimea Federal University, Department of Computer Science
4, Vernadskogo Avenue, 295007 Simferopol, Republic of Crimea
[2]Limited Liability Company «Adalisk»
10, Tarusskaya Street, 117588 Moscow, Russian Federation

Annotation. The paper explores the options for developing artificial intelligence technologies and tools aimed at enhancing the functionality of the system designed for monitoring and analyzing political Internet memes in the Russian-speaking segment of the Internet. Achieving this goal on the basis of deep machine learning methods and neural network technologies involves dealing with a number of correlated systemic interdisciplinary tasks (problems), which we discuss in the article. We introduce the architecture and functionality of the Memometrix software package, which can provide a partial solution to the problem of automatic image acquisition, tagging, filtering, grouping and exporting. The rationale for distinguishing different levels of macro- and micromodels, models of information interaction between intelligent agents, and control models, has been provided. Simulation modeling is helpful in adjusting the sets of algorithms to be then equipped by a search intelligent agent designed for monitoring and collecting Internet memes, and their analysis, as part of a multi-agent system for controlling the flow of Internet memes. Creating an intelligent system for monitoring Internet memes, its tuning and verification to real processes require the use of tagged databases. Building Internet meme databases involves accumulating and classifying Internet meme images, identifying their duplicates. It is required to use filtering, grouping and exporting the data relevant to Internet memes. Searching for memes involves making search queries and an option to perform a reverse image search, in that allowing to use the already existing databases for classifying meme images. We provide a detailed description of the package's relevant modules, as well as an example of an interface for filtering Internet memes based on using the templates agreed upon with sociologists. The software product was developed with the use of modular architecture and comprises several applications: a Web application (user interface), a server application (provides the operation of the package), and an application for automatically collecting Internet memes. The modularity of the package allows for making adjustments in the application (analysis of collected results, visualization of collected information, prediction of the further spread of Internet memes).

Keywords: Internet meme, social network, intelligent agent, software package.

INTRODUCTION

Web communities consume the flow of information, in that forming the "artificial intelligence" of their participants. Information spread in social networks includes its particular form coined Internet memes (IM). They are visual, imaginative, easy to comprehend and viral by nature. There is a developed process of spreading the IM flow that contributes to the formation of both positive and negative stereotypes. The socio-political effects of IMs are moderate and manageable. The information field self-organization process follows a destructive direction based on the principle of least resistance. Managing such a process requires great effort. Hence the relevance and demand for a comprehensive interdisciplinary research aimed at studying the flow of IMs. The emerging area of interdisciplinary research being

✉ Kozlova Margarita Gennadievna
e-mail: art-inf@mail.ru

coined "memetics" renders impact on developing algorithms for solving NP-hard discrete optimization problems in the form of evolutionary algorithms associated with the viral dynamics of information spread in the Web. And vice versa, methods for studying complex high-dimensional networks and relevant optimization problems contribute to the analysis of processes occurring on the Internet.

The goal of the present work is to explore the options for developing artificial intelligence technologies and tools aimed at enhancing the functionality of the system designed for monitoring and analyzing political Internet memes in the Russian-speaking segment of the Internet. Achieving this goal on the basis of deep machine learning methods and neural network technologies involves dealing with many correlated systemic interdisciplinary tasks (problems). Let us consider some of them.

Destructive (or positive) effects of political memes are associated with the constant flow of memes, their spread and viral nature. The source of the flow comprises a certain event (IM's eventful nature) and the response of the Internet community to the event. Appealing events are displayed online in the form of symbolic images. The symbolism of memes and their meaning can be instantly understandable when they comply with the established stereotypes. People inventing new memes have often creative personality, which leads to their rapid spread in various social networks. However, mass memes do not require any specific effort to promote their circulation (involvement in the flow of memes). Let us point out a few more problems relevant to the analysis of Internet memes. Among them are issues of information security within social networks.

To help develop controlling tools effective in preventing the spread of destructive Internet memes, it is necessary to know the state of the IM flow. Involvement of participants in the information flow is associated with their brain activity, concentration, attention. Processing the ever-increasing flow of information leads to losing track of time and weakens its perception. Self-awareness fades away, parts of the brain become inactive. It is important to know when this information doesn't arouse interest any more. IM's appeal and popularity work in a cyclical manner. The IM life cycle lasts 2–3 years and is underpinned by psychological perception according to the scheme shown in Figure 1 [1].

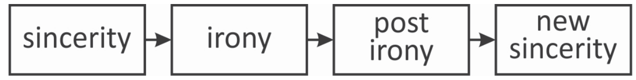

Fig. 1. Psychological perception of Internet memes

It is argued that placing the IM's cycle within a timeline may help predict the IM's status over specific segments of its life cycle. More accurate predictions require statistical, quantitative analysis and the use of time series models to describe IM propagation.

Social Web is the medium for IM propagation [1] and is defined by its agents' behaviors. Depending on modeling purposes, there can be distinguished several levels of macromodels, micromodels, models of information exchange between agents, control models [2–5].

Modeling can be performed based on Markov models, consensus models, finite automata, diffusion models, infectious disease spread models. The Reflexive Game Theory is used to describe interactions among agents leading to the spread of IMs. In [3], the authors considered DeGroot models, as well as statistical parameters and cases of a threshold model for behaviors of agents who make decisions based on opinions of other agents, which can be paralleled to behaviors of agents who spread IMs. Such terms as agents' trust and reputation are implemented. The models of influence and propagation (diffusion) of activities, information, memes implement the Linear Threshold Model (LTM) [6] and the Independent Cascade Model (ICM) [7]. The problem of maximizing the spread of influence is being elaborated. It is necessary to define the number of initial agentsdistributors (under condition of a limited budget) sufficient for providing the maximum spread of IMs in the network. The task of selecting an array of initial agents-detectors is NP-hard for the LTM and ICM models. In [7], the authors proposed a greedy heuristic $(1-e^{-1})$-optimal algorithm. In [4], it has been demonstrated that the probabilistic description of the model concurs with the results of simulation modeling and equals the results of calculations within the probabilistic model based upon the real distribution of degrees in the link graph for different social networks (the Russian segments of the Facebook, LiveJournal, Twitter networks). The mentioned approaches are only a preliminary step in accomplishing the task of creating an Internet search agent for intelligent monitoring and collecting IMs, and their analysis in a multi-agent IM flow control system.

Processing big data collected in the Internet social networks makes it possible to identify hidden changes occurring in the society. This information can be used by a decision support system (DSS) for preventing the spread of toxic, destructive information in the characteristic for memes comprehensible, symbolic and visual form. This requires predictive analytics based on big data, and exploratory, pilot analytics based on actual memetics. Building an intelligent system for monitoring memes, its adjustment and verification to real processes requires the availability of tagged databases. Involvement of experts competent in this field of analytics can help identify the destructive features, indicators of reciprocal influence (matrices), and elaborate integral characteristics. To effectively deal with the noted problems, the AI based Memometrix system is currently under development. This system does not have the capacity to automatically make decisions, but can interact with a DSS. Decisions are made by a decision maker (DM) — a specialist (group of experts) highly competent in the field. Another problem is modeling the DM, which best corresponds to the system of management in social networks (based on the information provided by Memometrix).

This paper is the follow-up to the series of works done by the authors [10–14].

1. PROCESSING THE INTERNET MEME DATA FLOW

Implementation of intellectual system for recognition of Internet meme data flow [14]. As Internet memes are images containing the pattern and text it is necessary to work out the recognition system (OCR) which detects the text in noisy image and then extracts it. To extract the text from the Internet meme OCR must: 1) find the text; 2) pre-process the image that contains the text; 3) recognize the text.

EAST algorithm (An Efficientand AccurateScene Text Detector) using the fully connected convoluted neural network which makes decisions based on the level of words and lines was selected to solve the tasks of text search on the image. This algorithm differs by the high precision and speed. The key algorithm component is represented by the model of neural network which is taught to directly forecast the existence of text instances and their geometry in the source images. This model is a fully connected convoluted neural network adopted to detect the text that produces the predictions of words and text lines for every pixel. This excludes such subphases as candidate offering, composition of text area and splitting of words. The subsequent processing stages include only the threshold value and threshold for the predicted geometrical patterns.

At the stage of pre-processing of images most of which is represented by text it is necessary to select only text and get away with the noises. The problem is that the text color is unknown and that text may embrace several tints of one color. The pre-processing includes the clustering of images, building of mask that separates the text from the rest background and determination of background color.

The convoluted neural network is used as the algorithm of symbols classification. The used convoluted neural network (CNN or ConvNet) belongs to the class of deep neural networks which are typically applied for the analysis of visual patterns. CNN networks are the regularized versions of multi-layered perseptrons. The multi-layered perseptrons usually belong to the fully connected networks, i.e. when every neuron of one layer is connected with all neurons in the next layer. The 'full connection' of these networks makes them prone to the data overrun. The typical ways of regularization include the adding of some form of scale measuring to the loss function.

However, CNN applies another approach to regularization. Using the advantages of hierarchical data framework the more complex frameworks are taken from the smaller and simpler ones. Thus, CNN is placed at the lower level according to the connection and complexity scale. ConvNet network may successfully capture the spatial and time dependencies of image by means of the relevant filters. The architecture of convoluted neural network ensures the better coordination with the image data base due to the less number of used parameters and opportunity to re-use the scales. In other words, the network may be taught to better understand the complex images. The architecture of convoluted neural network for letters classification is represented in Figure 2.

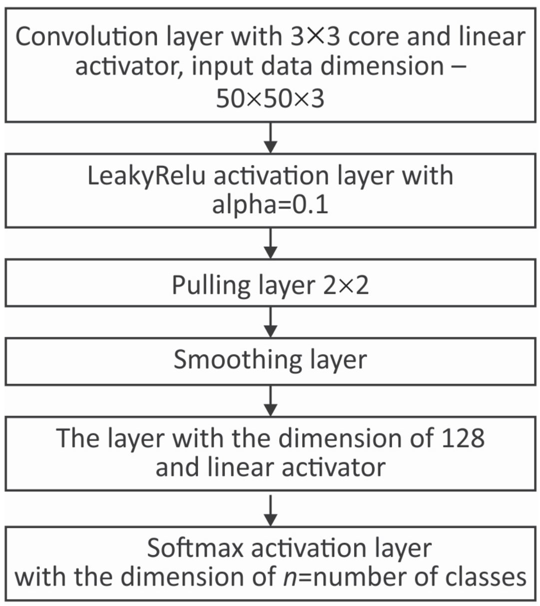

Fig. 2. Main steps of text determination algorithm

The input data are represented by the image of RGB color model that is divided by three color planes (red, green and blue). ConvNet role is to convert the image into form that is processed easier without losing the marker that play a pivotal role for obtaining the good forecast. The training sample including 59567 objects divided into 37 classes (33 upper case letters of the Russian alphabet and 4 lower case letters **«а»**, **«б»**, **«е»**, **«ё»**) was used as the data. 1814 fonts were used to create the sampling. The sampling is divided into the training one with 47653 objects and test sampling with 11914 objects. The neural network was implemented on Python with the use of Keras library. The model of convoluted neural network for letters classification was taught on 30 epochs demonstrating the quality of 98.8 % (for training data).

The centers of mass for dedicated keylines are found at the stage of lines recognition and combination of recognized letters into words. Thereafter, the main lines drawn up the text are built. If the upper and lower points are above and over the text line this keyline is also placed on this line. If the distance between keyline is larger than average distance multiplied by specified coefficient this means that there is a white space between the keylines. This allows not only define the regular sequence of symbols processing but also releases from the noises which could be left after image processing. Furthermore, the keylines are united into the words according to the intervals between the letters. The pseudocode of program for text recognition is written as:

```
Algorithm
Input:
   Sampling (X^m = {x_1,...,x_m})
   k – number of clusters;
   threshold – threshold of intersymbol
   distance;
Output:
   output_text – image text
1. Image clustering
2. Mask building
3. Letter contours extraction
lines = []
for cnt in contours:
   if moment of cnt not in lines then
      add cnt in line
for line in lines:
   mean_dist=mean_distance_between_con-
tours
for <i:=0 to len(line-1)> do:
   if    dist(cnt[i],cnt[i+1])>mean_dist+-
threshold then
      separate cnt[i] and cnt[i+1] by space
result = “”
for <line in lines> do:
   for <i:=0 to len(line)> do:
      result += Classification cnt[i]
```

The designed OCR properly highlights the text and classifies the short words. However, in the course of design when classifying the large text volumes the problems were arisen. Therefore, the final version of program for text extraction from the image includes two stages. At the first stage EAST algorithm extracts the text blocks. In case of numerous blocks which form the large group Tesseract is used to recognize them. If there are few text blocks the described approach to text extraction is applied (Figure 3).

Fig. 3. Work of algorithm recognizing the English text recognition

The pseudocode of the final algorithm for extraction and recognition of Internet meme text:

```
img; <S – is an input image>
n = minimal number of words;
textblocks = EAST(img);
<extract text from image> join all text
blocks which are crossed;
if len(textblocks) > n
   then txt = Tesseract(textblocks);
   else txt = TEXTEXTRACTOR(textblocks);
show txt;
```

Let's analyze the stage of classification of text extracted in the course of recognition of Internet meme text. The social network "VKontakte" is selected as the environment of Internet meme distribution. The classification objects are represented by the records obtained at the stage of text extraction. The records contain the text, meme image and comments.

The objects of selected distribution environment will be classified as "political" and "non-political" ones. The object class is determined by the class of record text, Internet meme text and comments text. Thus, it is necessary to build the classifier which can determine if the text belongs to the political class or not. As the social network "VKontakte" is mainly oriented toward the Russian-speaking population so initially the work was carried out with the memes containing the Russian text. Therefore, the sample consisting of 63 political and 44 non-political memes was provided as an example to solve this task. The text of comments taken from the social network "VKontakte" that features both political and non-political content was used as the additional objects. Generally, 208 sentences were used for analysis. The sample consisting of 16 political and 10 non-political memes was provided for validation. The data were normalized by the stemming algorithm and converted into the vector type before the classifier training. Considering the small size of training data the support vectors method was selected as the classification algorithm.

The word clouds from which it can be observed the class of "political memes" embraces the images with the names of countries, regions as well as names of political leaders are built for both classes. The class of "non-political memes" didn't feature the clearly distinguished group of words because the sentences of this class were of various themes. The classifier accuracy for test sampling was 90.25 %. The classifier accuracy will be improved with the increase of training data. 33 political and 24 non-political English memes were sampled in the course of preparation of this article. 112 sentences were analyzed in total. The sampling consisting of 11 political and 7 non-political memes was provided for validation. The English memes were taken not from the social network "VKontakte" but selected with the help of Google search engine. The word clouds of both classes with English memes are shown in Figure 4. It interesting that here the classifier accuracy for test sampling was 94.05 %.

Fig. 4. Cloud of English words specifying the objects of non-political class (left) and of political class (right)

The word clouds of both classes with Russian memes are shown in Figure 5.

Let's consider the process of data collection and analysis by means of system processing

Fig. 5. Cloud of Russian words specifying the objects of non-political class (left) and of non-political class (right)

and analyzing the Russian Internet memes. Five groups of the social network "VKontakte" were selected for analysis. It was able to extract 43 political memes by means of the designed toolkit. The lists of participants who assessed the selected record were provided after obtaining the list of political memes.

The research aimed at studying the characteristics of the flow of Internet memes is conducted by a group of mathematicians and sociologists from the V. I. Vernadsky Crimean Federal University. For that purpose, it is necessary to collect actual samples of images relevant to Internet memes, and then track their spread dynamics. Manual tracking and monitoring of Internet memes is a time-consuming procedure. Hence, it is vital to have access to an easy-to-use and functional software to help conduct research in an efficient and handily manner.

The project's goal is to develop a modular software package capable of automatically collecting Internet meme images, store them in a structured form, automatically tag and visualize them. This package will serve as a tool for sociologists and political scientists to help them conduct a comprehensive analysis and monitoring the search for Internet memes in social networks. Memometrix is the prototype of such an intelligent system [11–13]. The previously designed module of the system in the form of a software package (TagRun) provides some functions applicable to the work done by sociology experts. The AI technologies make it possible to provide novel functions. However, certain problems arise.

The first problem for experts to tackle is the large amount of information to be classified. A regular search for popular Internet memes via the search engines typically generates more than 100 meme images for each query. To make them further operable, each of them has to be marked with at least several tags. Given it takes 5 minutes to classify a single IM, it takes more than 8 hours to process the whole array of results generated by a search query.

The second problem is about identifying the IM duplicates. When accumulating data, visually similar images of a certain Internet meme can be obtained from different sources. In order to combine such objects into one, manual control by an expert is necessary.

The third problem has to do with the limitations in the package's functionality, namely in filtering, grouping, and exporting the collected information related to Internet memes. Working with TagRun made it clear that filtering by tags has to be complemented by additional criteria, such as search dates, source, number of appearances, initial query, etc. Besides, an option to export the collected data for further analysis is needed.

Automatic image tagging

To generalize, the procedure of image tagging is based on recognizing objects belonging to a certain image. To date, there can be distinguished a few basic research directions: image retrieval by sample (for example, the Image Retrieval in Medical Applications project [9]), object detection in static images (Google Picasa, iPhoto, Microsoft Live Photo use this technology) and object classification in static images (implemented in Google Googles, TextonBoost and Linkoln). The first and second approaches produce sufficiently effective solutions widely implemented in automatic systems. Regarding the third problem, which can be paralleled to the task we have to do with, the proposed solutions for it are so far quite weak. When selecting an approach to the task of meme classifying, their origin specifics is to be taken into account. Unlike photographs, which may be unique, all meme images are derived from the Internet. It follows that there exists at least one copy of it, and, more importantly, it has already been indexed by search engines, such as Google or Yandex.

Search engines operate upon extensive databases created via indexing web pages. However, in addition to the web page addresses, associations are also made with images, videos, and other media items located on the web page. Hence, it is possible to perform a reverse image search to identify not only its likely source, but also a feasible search query for it to be found. For example, the Google's reverse image search service provides a likely query for which the image was found, while Yandex provides a list of tags that belong to the image. To sum up, opting for a reverse image search when using search engines makes it possible to recruit the already existing databases to help classify the collected images.

Identifying image duplicates

Many techniques can be used when searching for duplicates or similar images, all of them having specific advantages and disadvantages. Choosing a certain approach well depends on the task at hand. When developing a solution for the search of duplicates in the already existing image databases, the main criteria are speed of operation and ease of implementation. We opted for the family of algorithms based on perceptual hashing to be implemented in the system being under development. In contrast to the traditional cryptographic hash functions (e.g., sha256 or md5), introducing a slight change in the input data will not lead to a significant change in the result produced by the hash function, but rather will lead to a similar result (stability). This feature allows the use of perceptual hashes to detect the degree of difference in the input data. For the perceptual hash to be used more effectively, it is necessary to perform a preliminary preparation of the image to bring all the images to the same format. This is necessary for further hash calculation. Among the advantages of this solution, of note is an option of one-time calculation of the image hash, and then comparing the new hash with the already existing ones to help search for similar objects. If the accuracy of a single perceptual hash is insufficient, it is possible to increase the amount of input data or add a second perceptual hash via using a different algorithm belonging to the same family.

2. MEMOMETRIX SOFTWARE PACKAGE MODULES

The software package should provide the following functions:

1) automatic tagging of images when they are added;

2) search for images which are duplicates or similar among already collected ones when adding a new image;

3) advanced system for grouping and filtering images;

4) data export options.

The package consists of several linked applications. Communication between applications is realized via an application program interface based on REST technology (RESTful API). To transfer information, the package components exchange text messages in JSON format. The general scheme of the package components and their interaction is shown in Figure 6.

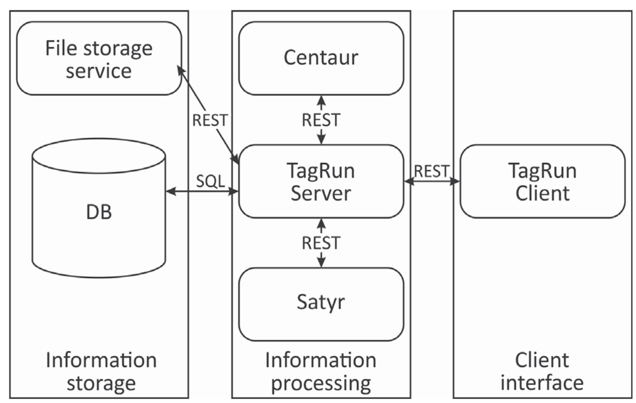

Fig. 6. Package components

The Satyr service counts image hashes in the duplicate image detection component. The Centaur service collects and automatically tags images. Both services interact directly with the TagRun Server. The Server API provides sufficient data for Centaur to operate. Satyr provides its own APIs for the server to be able to retrieve image hashes from the file storage service. Considerable changes in the service structure are allowed, but the user interface and information storage module are not affected. Changes are introduced only to the binding component, which essentially saves the costs of adding, changing or deleting the package components. In that, the modular structure has an important advantage for software development over the monolithic one. The already developed modules are applicable to creating intelligent search agents (IA) designed to monitor, collect and analyze IMs. Such IAs can work as part of a multi-agent system (MAS), exchanging the obtained data.

2.1. Development of the package modules

Satyr Component. Satyr is an application written in Python using the Flask framework and the OpenCV library. It is used to calculate the perceptual hash for an image. Based on hashes, the similarity between images is measured. Satyr consists of two parts: the first ensures uploading the file and sending the result, the second processes the image and calculates the hash.

The logic of the first part of the application is as follows. Create a new Flask application. Describe its only API method — `upload_file()`, which is available in the root of the application address. Then launch the application. The `upload_file()` method waits for a POST request to the application with a file to process. If the file is an image, it is saved in the temporary directory. Then the hash of the image is directly calculated. After the hash has been calculated, the image file is deleted and the client receives the hash value in JSON format.

The second part recruits an algorithm for calculating the image's hash function. First, the data on the image is read from the file. Then, the image is compressed to a size of 8×8 pixels. Next, the color image is converted into grayscale. Further on, the grayscale image is converted into black and white one. At the end, the resulting matrix is transformed into a vector of 0 and 1, according to the color of the matrix items.

Centaur Component. The application for collecting popular Internet meme images and their automatic tagging is implemented in C#. We have created the library via implementing the logic built on the acquisition process and uploading to the main server, as well as a console application used to start the acquisition process. Let us describe the behavior of the object responsible for retrieving information about images for a particular query (hereinafter, the grabber). Since the Google search service is not the only source for acquiring images (e.g., one can also use the Yandex search), it is reasonable to generalize the rules of its usage. To that end, we used the interface that describes the grabber methods. The interface has 3 public methods. The **SearchText** method makes a query to the search engine and returns the processed information on the first 100 retrieved images as well as their total number. The **SearchImage** method does a reverse image search and returns the processed information on the first 100 resulting images and their total number. The **GetTags** method uses the search engine to determine the content of the image and present this information as tags. The **ITagRunApi** interface is used to provide interaction with the main server. This interface methods allow to retrieve from the server the information needed for the grabber to operate and upload the results to the server. On having the interfaces for the grabber and for communicating with the server, there was built the grabber meant for the Google and Yandex search engines. For this purpose, there was created the class that implements the IGrabber interface.

The automatic search via search engines usually consists of two parts that include creating and executing a query to retrieve the content of a web page, and processing the web page to retrieve the information of interest. For the Google search, a separate query is used to retrieve the total number of search results for the query. The image search procedure does not differ much from the text search in terms of the programming code. The only difference is a one more jump between the search pages and the result. The page content is then filtered to acquire information on images. For determining the image content, let us use the Yandex search service as it proved to be more suitable for tagging. Similar to processing queries via Google, we make a query, get the web page, then get the desired information from the web page.

Extracting data on images from web pages turns out to be the main difficulty when gathering information via search engines. For that purpose, it is necessary to take into consideration the main methods of creating web pages widely implemented in the modern Internet environment. The first method involves generating pages on the server side and then sending them to the client (to the web browser). This is an older method of generating pages, which involves a larger amount of data to transfer, implemented, e.g., in php. The second, more advanced method implies creating a minimal html page on the server and then uploading the data from the client via additional queries. This method saves traffic and speeds up the page load time. The Google search uses a modified version of the first method. Upon request, the client receives a partly filled html document, then supplemented with more data by means of javascript.

On accomplishing the analysis of the query results it has been established that the basic information on images (links to the image and its source) is located at the end of the document in the form of the javascript object which is then used to visualize the elements of the page. To sum up, to obtain information on the image it is necessary to perform a series of operations as it is demonstrated in Figure 7.

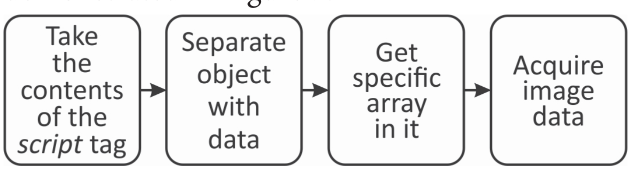

Fig. 7. Process of data acquisition

The result of collecting the data of interest is loaded into TagRun. When images are uploaded, the information on the date of acquisition and collected tags is added. The process of uploading images is shown in Figure 8.

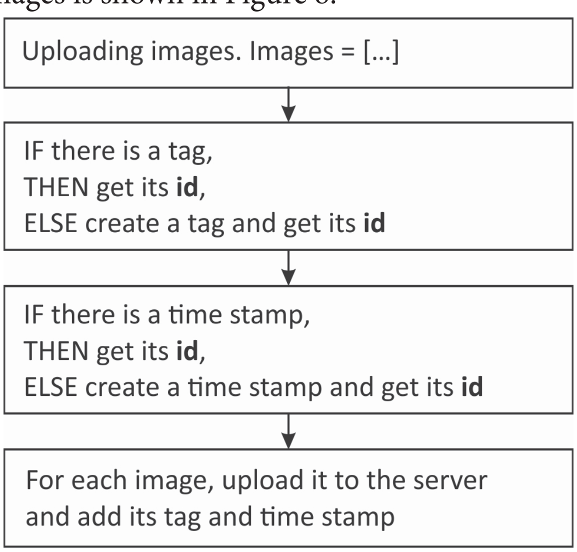

Fig. 8. Process of uploading an image

2.2. Filtering and exporting data

In addition to collecting data for further analysis, it is necessary to have specific tools for its filtering. This is the case not only because the analysis requires grouping IMs according to general criteria. Effective filtering is needed in order to get rid of images that are not IMs, but were added to the database accidentally when collecting information in automatic mode. A better selection of material for analysis can also be achieved by manual filtering of information, but it requires involvement of expert users. As the package has already found its use by sociologists for as long as a few months, they managed to formulate the basic criteria for filtering the previously collected Internet memes. For the convenience of the experts' work, the following parameters have been selected as the main ones:

1) the date of image acquisition to provide an option to track the dynamics in the number of meme appearances over a certain time period;

2) the number of images in the collection;

3) domain name of the source from which the image was collected;

4) the content of the query by means of which the image was retrieved, in order to be able to distinguish memes collected by different search queries.

The implementation of these filtering functions is shown in Figure 9.

Extended filtering options have made it possible to implement the function of saving filtering templates for later use. Experts in sociology proposed the use of a convenient template for filtering images collected from social networks. Implementing these templates provided good filtering results: the collected images turned out to be 90 % Internet memes without any incidental contaminants. Besides filtering, exporting data for further analysis appeared essential for our colleagues as well. A corresponding function was added to the package implemented in the form of exporting data into a universal Excel spreadsheet format.

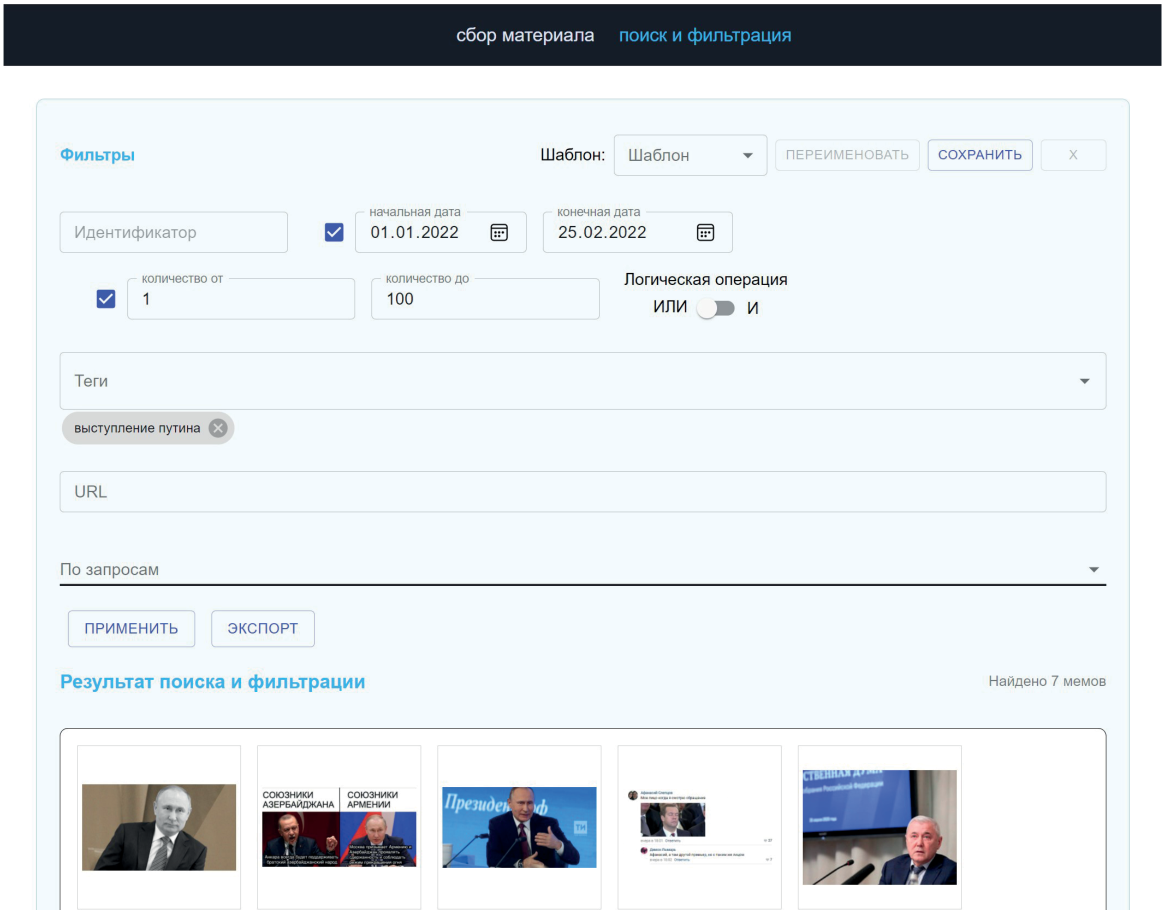

Fig. 9. Interface of the section for filtering Internet memes

CONCLUSION

There is developed the Memometrix software package that provides the functionality for an automatic acquisition of images, their tagging, filtering, grouping and exporting. We used the modular architecture of the software product that consists of a web application (provides the user interface), a server application (ensures the operation of the package), an application for automatic acquisition of Internet memes. In the future, the modular design of the package will allow to apply further modification to the application (analysis of results, visualization of collected data, predicting the further spread of Internet memes).

It is to note that the development of the Memometrix package is the result of an interdisciplinary research that involved mathematical analysts, programmers, sociologists and political scientists. The formalized IM properties, the logic of queries, the logical scheme of the software product, testing and feedback from users allow to further improve the software. The data collected with the help of the package allows to estimate how strong is the response produced in the Web by a certain social or political event; which changes occurring in real world correlate with changes in the flow of IMs, how much IMs reflect the real world events or turn out to be a result of manipulation and "injecting" destructive information. The resulting IM database is tagged and allows for statistical research. There are statistics for specific IMs and those related to a certain information query (quantity of IMs received for a query, time series, dynamics of change). Based on statistical data relevant to the identified IM's life cycle and its spread dynamics, integral indices for IM's dynamics can be calculated. The testing IM data can be found on the Memometrix software package website https://memometrix.ru.

ACKNOWLEDGMENT

The research was supported by the Russian Foundation for Basic Research (RFBR), project No 21-011-31733 "Development of a software package for automatic monitoring of the influence of political memes on the Russian-speaking segment of the Internet (stage — building the database for training a neural network)". The authors would like to express gratitude to Tigran Gabrielian for his valuable comments on the research tasks and assistance in software development.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Farmer J.* The life cycle of internet memes. Available at: https:// medium.com/@Farmer__Joe_/the-life-cycle-of-internet-memes-7d6722121e48.

2. *Gubanov D. A., Novikov D. A. and Chkhartishvili A. G.* (2010) Social networks: models of information influence, management and confrontation. *Moscow : Fizmatlit*. 244 p. (In Russian)

3. *Breer V. V., Novikov D. A. and Rogatkin A. D.* (2016) Micro- and macromodels of social networks. I. Theory fundamentals. *Automation and Remote Control*. 77:2. P. 313–320.

4. *Batov A. V., Breer V. V., Novikov D. A. and Rogatkin A. D.* (2016) Macro- and micromodels of social networks. Part 2. Identification and imitational experiments. *Automation and Remote Control*. 77:2. P. 321–331.

5. *Breer V. V., Novikov D. A. and Rogatkin A. D.* (2016) Stochastic models of mob control. *Autom. Remote Control*. 77:5. P. 895–913.

6. *Granovetter M.* (1978) Threshold Models of Collective Behavior. *The American Journal of Sociology*. Vol. 83, No 6. P. 1420–1443. doi:10.1007/978-3-658-21742-6_54

7. *Kempe D., Kleinberg J. and Tardos E.* (2003) Maximizing the Spread of Influence through a Social Network. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. P. 137–146. doi:10.1145/956750.956769.

8. *Olamide T. K.* (2015) Classification of Internet Memes. The University of Texas at Austin. Available at: https://repositories.lib.utexas.edu/bitstream/handle/2152/35301/KOLAWOLE-MASTERSRE PORT-2015.pdf?sequence=1&isAllowed=y

9. *Siong L., Zaki W. and Hussain A.* (2015) Image Retrieval System for Medical Applications. *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. P. 73–77. doi: 10.1109/ISCAIE.2015.7298331.

10. *Gabrielyan O. A. [et al.]* (2021) Intellectualization Of The Sociometric Data Processing Of Internet Memes Within Virtual Communication Structure. *International Scientific Forum «National Interest, National Identity and National Security». The European Proceedings of Social & Behavioural Sciences*. Vol. 102. P. 274–278. doi: 10.15405/epsbs.2021.02.02.35.

11. *Germenchuk M. S., Kozlova M. G. and Lukyanenko V. A.* (2020) Software tools and technologies for analyzing the flow of Internet memes. *Taurida Journal of Computer Science and Mathematics.* 3 (48). P. 37–58. (In Russian)

12. *Germenchuk M. S., Kozlova M. G. and Lukyanenko V. A.* (2021) Some features of design of intelligent systems for processing the internet memes flow. *CEUR Workshop Proceedings*. Vol. 2834. P. 148–158.

13. *Germenchuk M. S., Kozlova M. G. and Lukyanenko V. A.* (2021) Identification and Prediction of an Internet Meme Flow Lifecycle. *CEUR Workshop Proceedings*. Vol. 2914. P. 112–123.

14. *Kozlova M., Lukianenko V. and Germanchuk M.* (2021) Development of the Toolkit to Process the Internet Memes Meant for the Modeling, Analysis, Monitoring and Management of Social Processes. P. 189–219, Chapter 6, doi.org/10.1002/9781119751991.ch6 *in book "Recognition and Perception of Images, Fundamentals and Applications"* Edited by Iftikhar B. Abbasov. Wiley-Scrivener. 464 p.

15. *Osterroth A.* (2018) Semiotics of Internet Memes. doi: 10.13140/RG.2.2.12320.89605

16. *Zhang P., Zakharov V.* (2020) Computerized Visualization of the Russian Language Picture of the World. *International Journal of Open Information Technologies*. Vol. 8, No. 1. P. 58–62. doi: 10.24412/FfNhqoIQLC0.

17. Tian Z., Huang W. and Tong H. (2016) Detecting Text in Natural Image with Connectionist Text Proposal Network. LNCS. Vol. 9912. P. 56–72. doi: 10.1007/978-3-319- 46484-8_4.

Kozlova Margarita Gennadievna — PhD in Physics and Mathematics, Associate Professor of Computer Science Department, V. I. Vernadsky Crimea Federal University
E-mail: art-inf@mail.ru
ORCID iD: https://orcid.org/0000-0002-7467-8389

Lukianenko Vladimir Andreevich — PhD in Physics and Mathematics, Associate Professor of the Department of Differential Equations and Geometry, V. I. Vernadsky Crimea Federal University
E-mail: art-inf@yandex.ru
ORCID iD: https://orcid.org/0000-0001-5271-031X

Gurzhiy Pavel Andreevich — master's student of the second year of study of the Department of Computer Science, V. I. Vernadsky Crimea Federal University
E-mail: pavelgur5@gmail.com
ORCID iD: https://orcid.org/0000-0001-7567-8594

Makarov Oleg Olegovich — JS Programmer (Middle), Limited Liability Company «Adalisk», Moscow; postgraduate.
E-mail: fantom2.00@mail.ru
ORCID iD: https://orcid.org/0000-0002-0502-631X