Identification and Prediction of an Internet Meme Flow Lifecycle*,**

Mariia S. Germanchuk [1[0000-0002-7317-3768]], Margarita G. Kozlova [1[0000-0002-7467-8389]] and Vladimir A. Lukianenko [1[0000-0001-5271-031X]]

[1] V.I. Vernadsky Crimean Federal University, Simferopol, Russia
m.german4uk@yandex.ru
art-inf@mail.ru
art-inf@yandex.ru

Abstract. Memetics in social networks is currently quite a popular field of scientific research. The paper tackles the problems of spreading memes, mathematical modeling of spread processes, and developing tools for socio-political research. It is shown that the life cycles of an Internet meme flow and a separate meme have their specifics and ecology. The task of identifying the actual stage of a life cycle (LC) is much more difficult than that when applied to the economic LC of an enterprise. If taken in general, the problem is incorrect and dependent on the availability of data related to a selected meme flow in a network. Identification of a meme LC when monitoring a network is associated with a system for making queries, a technology for making an automatic database. Network monitoring to identify a meme LC is associated with a query system, a technology of automatic database formation, to then be able to make predictions based on the method of making conclusions by analogy with the use of the neural network approaches. The paper presents the results of the initial stage of work on the project designed to study the spread dynamics of Internet memes.

Keywords: Internet Memes; Analysis, Modeling, Identification, Prediction, and Management of the Flow of Internet Memes; Technologies of Socio-Political Use of Memes.

1 Introduction

Online social network communities act as consumers of information flows that build the "artificial intelligence" of their members. Among different types of information circulating within social networks, of particular interest are Internet Memes (IM). They

** The work is supported by the Russian Foundation for Basic Research, project № 20-011-31460\20 «Political memes in virtual communications: development of a software product for monitoring their spread and influence (analysis of the Russian-speaking segment of the Internet)»

are presented in a visual, easy to understand image-based form and have a viral spreading pattern. The already developed process of IM flow propagation contributes to the formation of both positive and negative stereotypes. The social and political effects of IM are mild and manageable. The information field is self-organized according to the principle of least resistance in a destructive direction, and a great effort is required to manage such a process. That said, comprehensive interdisciplinary studies aimed at examining the IM flow seem quite relevant and are in high demand. An emerging area of interdisciplinary research called "memetics" makes an impact on algorithms for solving the NP-hard problems of discrete optimization in the form of evolutionary algorithms related to the viral nature of information propagation on the Internet. And vice versa, the methods of studying high-dimensional complex networks together with associated optimization problems are implicated in the analysis of processes occurring in online networks.

The goal of the present work is to conduct the life cycle (LC) analysis of an IM flow or separate query selected memes. Various approaches to the meme definition, ways to classify them, research problematic, as well as developing tools for extracting, processing, and analyzing IMs have preliminarily been studied in [1, 2]. The problems of recognition of meme images, visualization of their spread on the Internet, certain aspects of intellectualizing the ways of the meme data stream processing have been considered in [3, 4], while relevant sociological and political approaches have been proposed in [5]. The articles [6, 7] present general ideas, approaches, and principles underlying the development of artificial intelligence systems. In the publication [8] the authors provide the analysis of memes in the context of linguistics. The materials of articles [9, 10] have previously been used by the authors in building a system designed to recognize labels on meme images.

While it is possible to examine the life cycle of a single meme, its circulation often gives rise to a flow of derivative memes. Depending on the introduced concept of meaning proximity (analogy, precedence), the calculated flow rates and intensity parameters will be different. Taking into account the probabilistic nature of the process, it is still important to be able to work with a single meme or a small number of memes. Typically, the life cycle is qualitatively displayed as a graph of a function that depends on time, with a characteristic increase, maximum value, the period of stabilization and degradation (the function value tends to zero). It is difficult to find out a stage related to the IM flow under examination. The needed parameters can be extracted from a close data set relevant to the analyzed memes as a result of regular monitoring of the process. At the same time, quantitative characteristics must be measured in different parts of a circulation network, which is complicated. As a result, to identify the life cycle of the IM flow, it is necessary to involve expert communities, mathematical modeling, as wells as the Big Data and Data Mining technologies. Based on the logic of dynamic systems, mathematical models of the spread of viral diseases, rumors, diffuse processes, etc., require adaptation to networks that change over time. In the simplest case scenario, it might be sufficient to obtain statistical data on the quantity, frequency, and so on for the flow of tested memes followed by regression and factor analyses. On the other hand, similar to high-dimension dynamical systems, one can expect the presence of channels

and jokers – low-dimension models that can qualitatively reflect the ongoing processes of IM propagation.

The identification and prediction of the IM flow life cycle are primarily centered on studying the IM effects on the activist youth audience and effective management needed to eliminate possible destructive influences. For example, the life cycle of the IM "cats" would let us study the audience most sensitive to such an influence, and a corresponding cluster of related communities. Of note is that only indirectly measured data would be available for further analysis. The acts of creating and propagating one's own IM flow must comply with actual legal prohibitions and regulations. Of most interest is to find a prospective test IM, which appears to be quite doable given the contingent nature of meme emergence. That said, studying the IM life cycles is of great importance and implies the creation of relevant tools for accumulating data, analyzing the processes of IM propagation, and making a corresponding software product to help process memes in automatic mode.

2 Internet Meme Life Cycle

Internet meme can be defined as a piece of information or phrase, witty or ironic, often meaningless, spontaneously gaining popularity when spread on the Internet environment (by e-mail, in instant messengers, on forums, blogs, etc.). The term "meme" came into use in the middle of the first decade of the 21st century, it is used in mass media, daily lexicon, and Internet communication. In the Internet era, almost all emerging memes automatically become IMs. Words and their combinations, as wells as images, can be considered memes. That is any auditory or visual segments of the Internet, statements, pictures, or sounds that provoked interest and got spread on the Internet.

New IMs keep constantly appearing to then disappear after a while. The main characteristic for IMs is their relevance: memes appear in users' news feeds, they are sent in private messages, mentioned in informal conversations. The life cycle for an IM comprises the moment of its appearance, its spread dynamics, loss of relevance, and eventually the end of use. It is difficult to trace back a precise moment of IM's appearance, as it is possible for even a few years old contents to become "viral". Such quantitative characteristics, often used in social networks or forums, as the number of "likes", "reposts", rating, etc. can be seen as indicators of the IM's relevance or dynamics. However, besides the already mentioned characteristics, the relevance of IMs can be affected by their circulation among users of social networks, instant messengers, blogs, etc. Such quantitative indicators can hardly be properly estimated. Collecting quantitative indicators from social networks is also quite complicated. Let's say we intend to examine a certain IM, and for this, we will track its relevance by monitoring quantitative characteristics in several social networks (VK, Twitter, Facebook). In this case, we are facing several difficulties at the same time.

First, the recognition of IMs and they're distinguishing from the rest of the content. It is not a trivial task to classify some images as containing a certain meme. Moreover, the recognition of IMs in audio and video materials would demand very much resources and might be close to impossible.

Second, monitoring IMs in all groups, channels or topics is an extremely complex processing task. The reason for this is a large number of the above-mentioned means of information spread in social networks. There are hundreds of thousands of groups on social networks. It is impossible to process the whole content circulating within all these channels of IM propagation. Studying IM life cycles would be significantly simplified when performed within the framework of a system for total accumulation, processing, and prediction of all Internet content.

3 Mathematical Model of the IM Viral Spread Among Social Network Users

Let's look at a simple mathematical information model of the IM spread dynamics among network users. Such a model makes it possible to determine the necessary input data and to give a qualitative and rough quantitative estimate for the predicted number of new consumers (and distributors) of the selected IMs, given the number of the already active IM consumers.

Let N be a potential number of IM consumers $(N >> 1)$. Let us denote the number of users affected by the IM at time t as $x(t)$, and denote the number of those who have not yet received and responded to the IMs spread by the community $x(t)$ as $y(t)$, i.e. $x(t) + y(t) = N$. The number of possible participants in the interval Δt is proportional to the number of contacts between already active consumers $x(t)$ and potential ones $y(t)$, i.e. the increment $\Delta x = \alpha x(t) y(t) dt$, where α is the coefficient of proportionality. Taking the limit of the increment $\Delta x = x(t + \Delta t) - x(t)$ at $\Delta t \to 0$, we obtain the logistic equation

$$\begin{aligned} \dot{x} &= \alpha x(N - x), \\ x(t_0) &= x_0 \end{aligned} \tag{1}$$

The solution to the Cauchy problem (1) is obtained in the form

$$x(t) = \frac{x_0 N e^{\alpha N (t - t_0)}}{N - x_0 (1 - e^{\alpha N (t - t_0)})} = \frac{x_0 N}{(N - x_0) e^{-\alpha N (t - t_0)} + x_0}.$$

Note, that $\lim\limits_{t \to \infty} x(t) = N$. Let's find the rate of change for the IM spread rate:

$$\ddot{x}(t) = \alpha \dot{x}(N - 2x) = \alpha^2 x(N - x)(N - 2x).$$

Only the third factor can take a zero value

$$N - 2x = 0 \Leftrightarrow (N - x_0) e^{-\alpha N (t - t_0)} - x_0 = 0.$$

As a result, $\ddot{x} > 0$ for $t \in \left(0,\ t_0 + \frac{1}{\alpha N} \ln \frac{N - x_0}{x_0}\right)$ and $\ddot{x} < 0$ for $t \in \left(\frac{1}{\alpha N} \ln \frac{N - x_0}{x_0},\ +\infty\right)$. Then, the growth rate $\dot{x}$ of interest to the IM increases up

to the moment $t^* = \frac{1}{\alpha N} \ln \frac{N - x_0}{x_0}$ and then decreases. The parameter t^* is needed for designing a control system for IM spread dynamics.

Concerning various IM flows, the community breaks up into clusters by their interests (including destructive and constructive ones). For each community that consists of N_i $(i = \overline{1,n})$ participants $\left(N = \sum_{i=1}^{n} N_i \right)$, one can formulate the Cauchy problem (1) for $x_i(t), t_{0i}, x_{0i}, \alpha_i$.

The spreading process will be characterized by characteristics integral for all communities. Note that the task of restoring initial parameters based on the present ones is incorrect. Destructive participants who spread IMs often appear to be network users with a video game addiction. While there have been developed programs designed to prevent, diagnose and treat people with a video game addiction, so far no programs have been created to identify users prone to get "infected" by destructive IMs. Video game addiction is supported by a constant flow of game updates, while the IM life cycle – by introducing new memes. The graph of the IM life cycle curve can be represented as a linear combination of logistic curves (solutions to the Cauchy problem (1)) when initial information is available.

To collect data on the meme's life cycle, information is needed regarding its relevance over a certain period. Since the search services prioritize relevant information, issuing a search service is a means to obtain needed information. Although the search results may appear somewhat outdated, they help provide a realistic picture of the meme spread on the Internet. Hence, the Google search results are used to retrieve meme samples.

Since the Google search tool is a website that has no open API to let extract the needed data, there appears a side task of extracting information from web pages. Information extraction can be performed either manually when working on small amounts of data or can be automatized in that allowing to extract large amounts of information from websites.

The process of extracting structured and useful information from a website is called parsing, and the tools for conducting such a procedure are called parsers. Websites are normally designed based on an assumption that provided information is meant for people to read. However, the format of data that is easy to understand by people is often not that well understood by program tools. Besides, different websites may vary much in their ways of structuring data. So, no universal tool for information extraction has so far been created.

Parsers are programs designed to collect and structure information extracted from websites. They are typically developed independently for each website based on its structure and technical properties. Ready-made solutions for extracting information from websites following some preliminary settings adjustment are also available, without writing a single line of code. Such solutions are often quite expensive and rarely possess a needed level of flexibility like those developed specifically for a certain website.

The process of information extraction can be simple: download the URL, read the information and pass it to the recipient; it can also be more difficult: log in to the system, generate a query based on the information located in the headers and values of JavaScript variables on the page, while their names may vary with each query and a JS code being in a minified or obfuscated form. The first case scenario looks quite easy to follow, while in the second one, to create a parser within a reasonable time window, it is effective to use headless-browsers (no GUI) that support scenarios such as Phantom JS for fetching data to optimize the amount of time needed to learn how the website interacts with the backend. Also, for these purposes, the Selenium WebDriver can be used with one of the typical browsers.

4 Structure of a Software Product for Processing Internet Memes

The main task of the current work was developing a cross-platform software package possessing the following functions:

1) adding images (memes) in an automated or manual mode;
2) image storage;
3) adding tags to images;
4) aggregation of images by tags;
5) adding timestamps to images.

Let's consider a task to study a certain meme's life cycle. To automatize the process of collecting the Google search results, it is necessary to take into account the basic principles underlying the use of Internet services, the Google search service being one of them. When creating Internet services, a developer or a company opts to create an application programming interface (API) to provide third-party developers an easy and convenient way to use their services. Unfortunately, the Google search service doesn't have a free public API, so we will receive the search results directly from HTML pages. This approach is used for the following part of the program (see fig 1.):

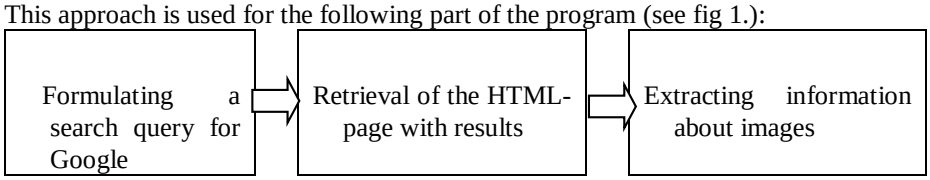

Fig. 1. Google search service doesn't have a free public API, so we will receive the search results directly from HTML pages.

Formulating a web query and retrieving a corresponding result employing modern software tools is not a difficult task. To obtain search results in the form of images, the following URL query is to be used:

```
https://www.google.com/search?q={Query}&tbm=isch,
```

on the place of {Query} one is expected to put a certain query such as "meme", for example.

More challenging is the task to directly extract information about an image based on available results. For this, the basic methods of creating web pages typical for the modern Internet are used. The first of them involves the procedure of generating pages on the server-side and then sending them to the client (web browser). This is an older method of generating pages that require more data to transfer and is implemented, e.g., in PHP. The second method is a more up-to-date one and implies a procedure of creating a minimal HTML-page on the server followed by uploading client data on it using additional queries. This method helps reduce network traffic and speed up page load times. The Google search service uses a modified version of the first method. When requested, a half-filled HTML document is sent to the client, with a portion of information complemented employing the javascript.

The analysis of the query results showed that the basic information about images (a link to the image and the page where it was retrieved from) is located by the end of the document in the form of a javascript-object, which is then used to display page elements.

Hence, to retrieve information on images it is necessary to accomplish the following operations (see fig. 2):

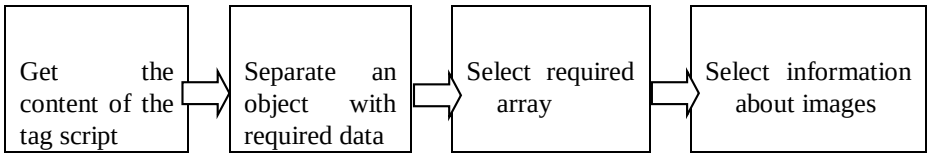

Fig. 2. Retrieve information on images it is necessary to accomplish the following operations.

As a result, we are receiving information about all images associated with the search query. Next, quantitative indicators on how long a selected meme (IM flow) has dwelled in the information space within certain communities are monitored.

Let's consider the case of implementing the technology for researching the life cycles of memes using the developed web application FrontEnd.

As part of joint research work with a group of sociologists working in the Crimean Federal University, to study the IM life cycle and build a system for opinion polls, a control sample of images from the Internet was selected. Preliminary preparation of collected data was done using the developed software tools to:

-IM grouping by an expert group according to selected criteria;
- IM sample according to certain criteria;
- generation, storage, and execution of expert queries in the form of expressions of mathematical logic for the IM sample;
- selection of the IM image areas with an option to set their attributes.

The FrontEnd application is written in TypeScript, which is an extension of the JavaScript language. The use of basic JavaScript in large applications increases the complexity of software development, as this programming language is dynamically typed

and as a result, unexpected errors may occur during code execution. TypeScript is different from JavaScript in its capacity to directly assign statical types, the possibility to use typical classes (as in traditional object-oriented languages), and an option of using plug-in modules. This was done to speed up development, facilitate readability, refactoring, and reuse of code, help find errors in development and compilation stages. Ultimately, TypeScript can be compiled to JavaScript and executed either in the browser or on the NodeJS platform [https://nodejs.org/en/].

To create the UI, the ReactJS library was chosen [https://ru.reactjs.org/], which allows the creation of graphical user interfaces for web applications. To work with the Document Object Model (DOM) in React, the Virtual DOM (virtual tree of web page elements) is used. Changes in the Virtual DOM are automatically applied to the DOM in a browser for it to match to the Virtual DOM. React allows developing interfaces using component-oriented programming. React is based upon a set of components which are self-contained, reusable blocks, each having its state and functions.

To control the state of the application, the Redux library was chosen, which is positioned as a predictable state container for JavaScript applications.

React.js together with Redux on the client-side allow creating an MVC application architecture. The MVC (model, view, controller) architecture implies that the model is the only source of truth and the state is stored in it. Views are derived models that need to be in sync when the model changes its state. Applications written using React + Redux are a non-deterministic state machine.

To route, the application (routing inside the application on the client-side), React.Router is used. The router determines which view to display to the user. Thanks to the router, it became possible to make a single-page application (SPA). This means that the web application uses a single HTML document and implements the interaction with the client using dynamic loading of styles and scripts. The advantage of SPA is that they are similar to native applications, except that they run within a web browser. Transitions between pages look more seamless and invisible for the user, which positively affects users' experience (UX) and allows to improve the page response time, as the application does not need to load an entire HTML file.

The interface is assembled using the web pack (JavaScript module assembly system). The webpack is used to assemble a web application in a single bundle based on JavaScript modules, CSS, and HTML files.

The Babel.js tool is also implemented. It is used to convert the code written in the latest standard JavaScript (ECMAScript 6 is used in the project) into the code compatible with the older standard JavaScript, which is often needed for it to be supported by older browsers.

The Bootstrap 4 library, developed by Twitter, is used to apply styles for page elements. Setting page styles is simplified, as there is no need to write a custom CSS code; the already prepared classes can be used for setting layouts.

The application works with entities:

Area – an area selected in an image – a class that contains information about the area. It stores the id of the image in the database, the id of the area in the database, its coordinates, and the size of the area, as well as a list of tags related to it.

Tag – a tag entity, an object with a tag id field in the database and its name.

ImageInfo – a basic entity that unites other atomic entities. It is a representation of an image in an application; it aggregates the id, width, and height of the image, the date it was uploaded to the server, an array of image areas, an array of tags related to the image itself, and an URL – its location in the server file system.

Query – a boolean expression for fetching images from a database, i.e., it represents a structure in JSON format, in the form of a tree, where each element has a type (boolean operator or operand), a text value, and an array of child elements (see fig. 3).

Components of the application:

TagSuggestSelect — tag suggestions from the database.

TList — a list of entities conveyed to it, displayed in a column.

QueryBlockSuggestSelect — suggestions for selecting elements of a visual representation of the query for image selection.

QueryBlock – a block of visual representation of an expert query.

Modal – a popup window overlapping the main content of the page.

ImageStrip – a block that displays pictures in the form of "tiles".

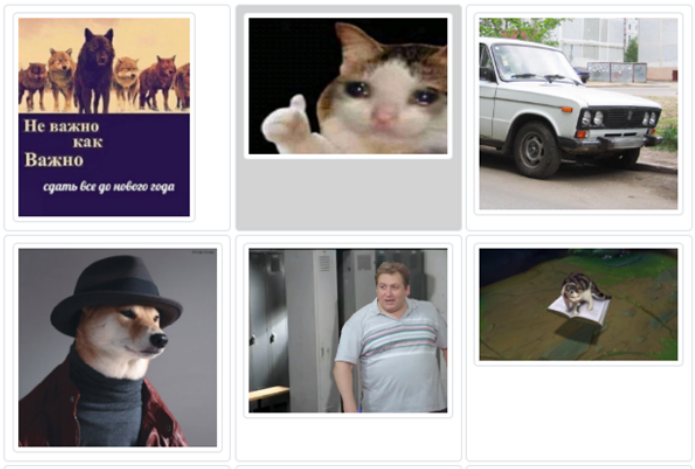

Fig. 3. Pictures arranged as "tiles" (ImageStrip).

ImageDetails – a block to display information about the image, and to do basic work on the image.

ColoredRect – a block for working with the image areas.

New areas are created using drag-n-drop. Blocks are based on the React-konva (HTML5 canvas) component, adapted to work with react and its data-flow.

Views. The user interface is divided into three views, each of which has its own set of functions. A first view is a form for uploading images to the server. It allows uploading. The user clicks on the "Select File" button and selects an image located on his local storage. The file is written to the state of the component. After clicking on the "Upload Image" button, the action upload image is called and the image is sent to the server for storage. The second view is the main window for working with uploaded images. The

column on the far left is the TList block. In this column, one can remove tags and filter images by a specific tag. There is also a button to display all images, regardless of their tags.

The middle column is the ImageStrip component. After clicking on an image, the image is selected and the rightmost component ImageDetails is drawn with the data relevant to the selected image.

The third view is the query page and consists of three blocks: TList, ImageStrip, and QueryBlock. Operations with queries are conducted here, such as: creating, saving, and searching images for a given query.

The back-end component of the service is written in C#, MySQL is used as a database management system.

The key entity in the application is image-related information, which is stored in the Images table, which is linked to the Areas table by a one-to-many relationship and to the Tags table by a many-to-many relationship. The Areas table is also kinked by a many-to-many relationship to the Tags table (see fig. 4).

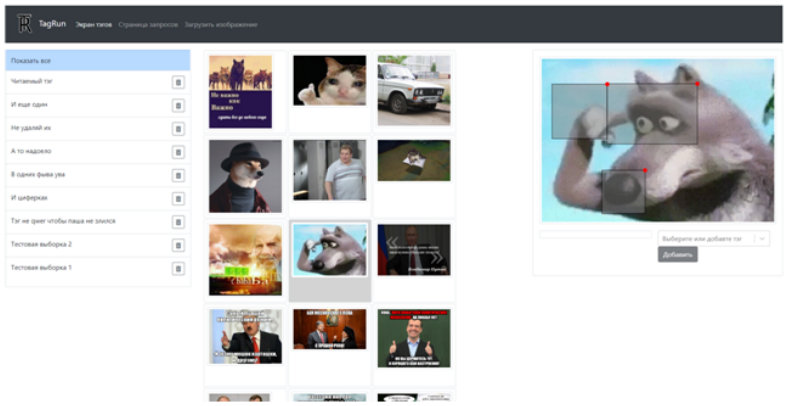

Fig. 4. User interface screen form.

Controllers. The API consists of three controllers, each responsible for working with its entity. Each controller is divided into endpoints, each performing its task depending on the HTTP query method and the URL the query was sent to. As a rule, the endpoints return data to the client in JSON format, but there are also the endpoints that return files located on the server.

5 Visualization of Internet Meme Life Cycle Models

The IM life cycle model can be visualized by plotting the dependence of the IM popularity on time. The number of times the IM has appeared in the search results for a certain period will be used as a measure of popularity. For example, if a time interval of one week is used and the IM has been found in 5 search results, then it receives a value of 5.

As part of this work, daily searches were performed over several months. Search results for these queries were stored in the database. The collected data were visualized with the help of the developed software in the form of graphs (see fig. 5):

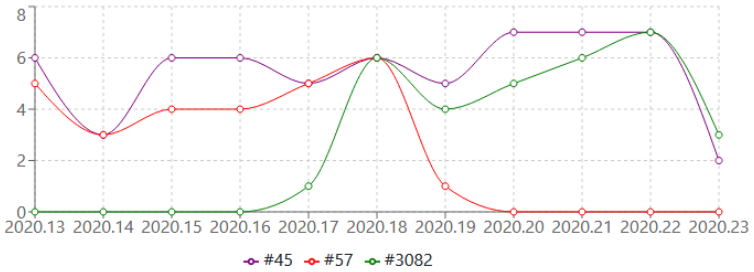

Fig. 5. Data visualized in weeks.

The graph above shows the popularity dynamics for three IMs. The X-axis represents time, measured in weeks for clarity. There is also an option of using months as time intervals, however, due to a limited period of collecting information, such a graph is not sufficiently informative (see fig. 6).

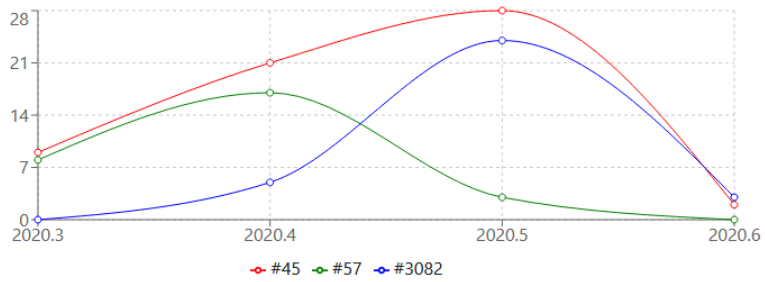

Fig. 4.

Fig. 6. Data visualized in months.

The prediction procedure for the IM life cycle is performed based on the accumulated data presented in Fig. 3, 4. To make predictions more reliable, it is necessary to accumulate a sufficient amount of data in the IM database, to then identify characteristic classes of curves representing the IM life cycles. The prediction is calculated by using the already developed enterprise life cycle algorithm (based on economic statistics) with the help of the principal component analysis and inference by analogy.

6 Conclusions

The paper provides the results of the initial stage of work on the project designed to study the spread dynamics of Internet memes. The importance of developing specific

tools for collecting, processing, and studying the IM life cycle is emphasized. Here we have elaborated a general structure, visualization methods, and ways of implementing a software product in the life cycle analysis. For its further development, we intend to implement a neural network approach for the tasks of intellectualized processing of the flow of Internet memes to give an estimate of their impact on the audience of Internet communities.

The authors express their gratitude to T. Gabrielyan for his participation in formulating the work goals, and to P. Gurzhiy and M. Travintsev for their active involvement in software development.

References

1. Germanchuk, M.S., Kozlova, M.G., Lukianenko, V.A.: Problems of Modeling the Processes of Distribution of Internet Memes. Analysis, Modeling, Management, Development of Social-Economic Systems: a collection of scientific papers of the XII International School-Symposium AMUR-2018, September 14-27, 2018. Pp. 136-139. (2018) (In Russian)
2. Germanchuk, M.S., Kozlova, M.G., Lukianenko, V.A., Pivovar, A.E.: Development of Tools for Processing and Analyzing the Flow of Internet Memes: a collection of scientific papers of the All-Russian Scientific-Practical Conference MICME-2019 (Eds V. A. Lukianenko). Simferopol, 2019. Pp. 121-127. (2019) (In Russian)
3. Germanchuk, M.S., Kozlova, M.G., Lukianenko, V.A.: Features of the Development of an Intelligent System for Processing the Flow of Internet Memes: a collection of scientific papers of the Distant learning technologies, Proceedings of the IV All-Russian Scientific-Practical Conference (Eds V.N. Taran). Simferopol, 2019. Pp. 258-265. (2019) (In Russian)
4. Germanchuk, M.S., Kozlova, M.G., Lukianenko, V.A.: Intellectualization of Processing of the Stream of Internet Memes: a collection of scientific papers of the III International Scientific Conference "Autumn math readings in Adygea". Maykop, 2019. Pp. 139-143. (2019) (In Russian)
5. Gabrielyan, O.A., Lukianenko, V.A., Kozlova, M.G., Gasparyan, M.V., Gabrielyan, T.O.: Intellectualization of the Sociometric Data Processing of Memes within Virtual Communication Structure. NININS 2020: International Scientific Forum «National Interest, National Identity, and National Security» (in press). (2020)
6. Gabrielyan, O.A., Suleimenov, I.E., Vitulyova, Y.S.: Artificial Intelligence In The Context Of Noosphere Studies. SCTCMG 2019. DOI: 10.15405 / epsbs.2019.12.04.130 (2019)
7. Suleimenov, I., Panchenko, S., Gabrielyan, O., Pak, I.: Voting Procedures from the Perspective of Theory of Neural Networks. Open Engineering. 2016. V. 6. № 1. Pp. 318-321. DOI: 10.1515 / eng-2016-0048 (2016)
8. Osterroth, A.: Semiotics of Internet Memes. DOI: 10.13140/RG.2.2.12320.89605 (2018)
9. Zhang, P., Zakharov, V. P.: Computerized Visualization of the Russian Language Picture of the World. International Journal of Open Information Technologies. *ISSN: 2307-8162. Vol. 8, No.1*., pp. 58-62. DOI: 10.24412/FfNhqoIQLC0. (2020)
10. Tian, Z., Huang, W., Tong, H., He, P., Qiao, Y.: Detecting Text in Natural Image with Connectionist Text Proposal Network. LNCS, vol. 9912, pp. 56-72. DOI: 10.1007/978-3-319-46484-8_4. (2016)