Some Features of Design of Intelligent Systems for Processing the Internet Memes Flow[1]

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

[1] V. I. Vernadsky Crimean Federal University, Simferopol, Russia

german4uk@yandex.ru
m.kozlova.art.inf@gmail.com
LukianenkoVA1@gmail.com

Abstract. The intellectualization of data processing for Internet memes flow includes problems of search and identification of network memes, text and meme image recognition, analysis of relevant information, detection of the graph structure of meme flow distribution network, clustering, and visualization. The paper focuses on viral pictures consisting of an image and a text. It is a challenging problem even with such a simplified representation of memes. Memes consisting of images and text are widely spread in social networks. The modeling process involves both an immediate content of memes and information contained in the environment and comments. The intellectualization of processing such specific information is accompanied by expert assessments. Tool design for processing and analysis of the Internet meme flow is based on a group of methods to solve the following problems: extracting and analyzing text from an image; text classification; implementation of different network metrics to adjust the classification, collecting different metrics to create a predictive system. The research paper introduces a novel approach to the development of methodology and software designed for automatic and intelligent detection of political Internet meme's influence on Russian-speaking Internet users. The social network Vkontakte was used as a meme dissemination medium.

Keywords: Internet meme, social network, image recognition.

1 Introduction

Social network research is getting increasingly popular nowadays in the world [1]. Social networks can be treated as a source of data about the ways of living and interests of real people. Various companies and research centers demonstrate an increased interest in Internet data. Experts use big data arrays from social networks to model economic, political, and social processes at distinct levels and to develop specific mechanisms, tools, and technologies to effectively influence them [1]. Of particular interest are Internet memes as Internet communication units that combine both verbal and non-verbal pieces of information in the form of text and images. Internet memes are intended for users to provoke intrigue and surprise and make them wish to distribute information over the network.

According to the Merriam-Webster dictionary, the meme is defined as "an idea, behavior, style, or usage that spreads from person to person within a culture". Internet memes are represented in the form of viral images visualized as a picture typically accompanied by a piece of text.

When studying the processes of meme dissemination and visualization, it is necessary to take into account particular semantic environments, the text itself, word combinations, words, and concepts as a whole, all of them being the elements of the world's language picture [2]. In linguistics, the representation of semantic fields is used for this purpose [3, 4]. Tags corresponding to certain memes are visualized as a tag cloud. Changes in tag clouds for selected communities characterize the distribution of Internet memes. To provide a more detailed analysis, it is necessary to implement different approaches. For example, the Thesaurus tool in Sketch Engine allows detecting words similar to selected ones. To calculate word similarity, it examines matching sets for word pairs based on their syntactic relationship. The similarity of word distribution is statistically calculated based on the lignite association measure following certain lexical and syntactic patterns. On the other hand, large arrays of data require rather simple algorithms to process them in real-time and a hierarchy of algorithms intended for more subtle analysis.

Geo-linking of memes to a certain distribution agent can be achieved in geosocial (location-based) networks. Relevant tools use GPS data in a mobile device and require access to its geolocation information. The network proximity parameter is used to find friends in a social network, make recommendations, and collect behavioral patterns for specific users. An overview of approaches to network proximity and the application that uses network proximity to distribute social content is provided in [5]. This type of application can be implemented for the analysis of meme distribution processes.

The current work aimed to develop an intelligent system designed to extract text fragments from images presented as Internet memes, to classify them, to use various network metrics to make corrections to its distribution forecast model, and to define particular classes where specific Internet memes might belong.

Internet memes analyzed within the present work can be classified as viral images consisting of some text combined with a picture. The social network Vkontakte was selected as an Internet medium for meme dissemination.

2 Main Tasks

Let there be a certain environment (in particular, a social network) for the Internet memes dissemination. The objects of this environment are Internet memes characterized by properties inherent to the distribution environment. The goal is to develop and implement specific tools intended to classify objects belonging to a given Internet meme dissemination environment and predict its distribution.

Due to the open nature of viral memes, there is no basic list that describes (or can describe) each meme, and there is no authorized Internet body to index new memes according to some conventions. Hence, a trained model cannot be effectively trained based on every class that existed before or may appear in the future. Thus, adaptive models and algorithms are needed for effective analysis and prediction.

The authors selected the social network Vkontakte as a dissemination medium. Records that contain text and an application in the form of an Internet meme will be referred to as objects of this environment. For example, we will classify each selected Internet meme as having political or non-political content.

The expected input to the system is a link to a meme in Vkontakte or a direct image with text. The ultimate goal of the project lies in creating a system that can correctly label Internet memes and predict their distribution.

Given the specifics of Internet memes, this problem cannot be solved by conventional algorithms. To solve the problem, it is necessary to design special methods and algorithms for detecting, recognizing, and classifying text, using network metrics, as well as clustering algorithms.

To design special tools for such a system, the following intermediate tasks must be solved:

1. extracting text from a meme image;
2. classification of a text piece associated with an image;
3. use of different network metrics to adjust the classification method for Internet memes and to create for them a distribution prediction system;
4. analysis of data obtained using the system;
5. clustering and visualization of the Internet meme database.

3 Implementation of Intelligent Data Processing System for Internet Meme Flow

3.1 Extracting Text From the Image

Since Internet memes are generally images containing a picture and a text, it is necessary to design a recognition system (OCR) capable of detecting a text fragment on a noisy image and extracting it. To extract text from Internet memes, OCR is supposed to: 1) find text fragment; 2) pre-process image containing text; 3) recognize text.

Initially, for text recognition, we used the freely distributed Tesseract text recognition software developed by Hewlett Packard. Tesseract allows recognizing text based on the LSTM neural network [6]. However, to achieve better recognition results, one should improve the image quality before it to Tesseract. Image processing includes several steps. First, paths are to be analyzed. Detection of path nesting and finding child paths make it possible to detect and recognize both black text on a white background and vice versa. At this stage, paths are assembled into strings, and strings into text. Text strings are divided into words depending on line spacing. The second step involves a two-step text recognition process. First, an attempt is made to recognize each word in turn. Each word recognized by the classifier with a high confidence level is passed to the adaptive classifier as training data. After that, the adaptive classifier is capable of recognizing the remaining text more accurately.

However, in the course of the development process, it became obvious that Tesseract algorithms were not always sufficient to effectively select text fragments superimposed on top of an image and achieve satisfactory results when classifying small phrases or phrases characteristic of Internet memes. Hence, it became necessary to develop a more specialized OCR.

To solve the first problem of finding text within the image, it was selected the EAST algorithm (An Efficient and Accurate Scene Text Detector) [7]. It uses a fully connected convolutional neural network that makes decisions based on word and string level. This algorithm is characterized by high accuracy and short operation time. The algorithm's key component is a neural network model that is trained to directly predict the existence of text instances and their geometry in source images. The model is a fully connected convolutional neural network adapted for text detection that outputs predictions of words or text strings for each pixel. This approach eliminates interim steps, such as the candidate's offer, the formation of the text area, and splitting words. Subsequent processing steps include only the threshold value and the threshold for predicted geometric shapes.

At the stage of preprocessing images, containing mostly text, one needs to select exclusively text and get rid of the noise. Problematic is that the text color is unknown while it may contain several shades of the same color. Preprocessing includes image clustering, creating a mask separating text from the background, and determining background color.

To implement the symbol classification algorithm, the convolutional neural network was used. Such a network (CNN or ConvNet) belongs to the class of deep neural networks and is most often used for the analysis of visual images. CNN networks are regularized versions of multilayer perceptrons. Multilayer perceptrons usually belong to fully connected networks, i.e. each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" makes these networks predisposed to data overload. Typical regularization methods include adding some form of weight measurement to the loss function. However, CNN takes a different approach to regularization: they take advantage of hierarchical templates applied to data and collect more complex templates using smaller and simpler templates. Thus, on the scale of connectivity and complexity, CNN can be placed at a lower level. The ConvNet network can successfully capture spatial and temporal dependencies in an image using appropriate filters. The convolutional neural network architecture provides better alignment with the image data set by reducing the number of parameters involved and allowing re-use of weights. In other words, the network can be taught to better understand complex images [8]. The architecture of a convolutional neural network for classifying letters is shown in Fig. 1.

The input data is an image in the RGB color model, separated by three color planes – red, green, and blue. The ConvNet is used to transform images into a form that can be more easily processed without a loss of properties critical for making a good prediction. The data used as a training sample included 59567 objects divided into 37 classes: 33

letters of the Russian alphabet of upper case and 4 letters of the lower case – "а", "б", "е", "ё". 1814 fonts were used to create the sample. The sample was divided into a training sample containing 47,653 objects and a test sample containing 11,914 objects. The neural network was implemented using Python and the Keras library. The convolutional neural network model for letter classification was trained on 30 epochs and showed 98.8% quality on training data.

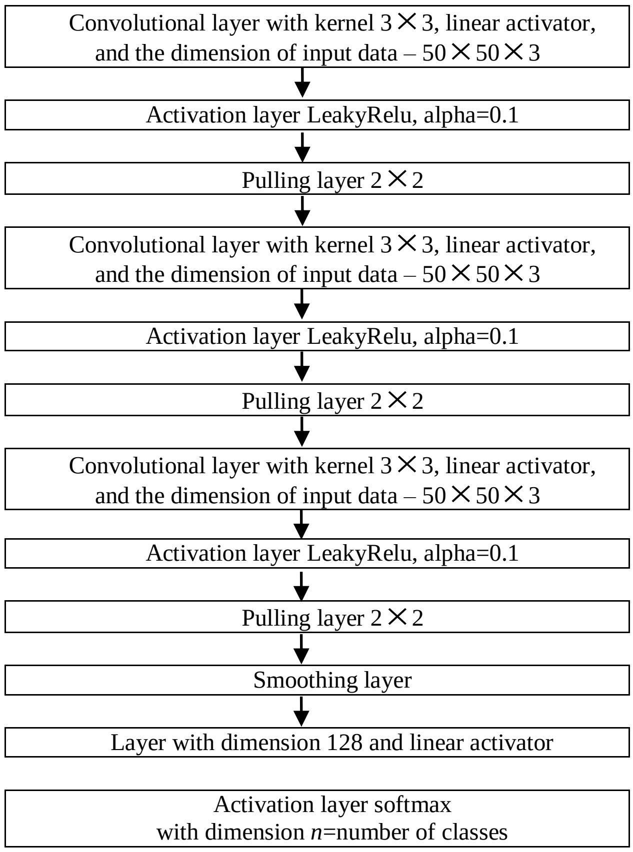

Fig. 1. Algorithm basic steps for text detection.

At the stage of line recognition and combining recognized letters into words for selected paths, centers of mass were found, and then basic strings making up the text were constructed. If the top and bottom points lie above and below the text line, respectively, then this path also lies on it. If the distance between paths is greater than the average distance multiplied by a certain coefficient, it means that there is a gap between paths. This not only allows to determine the correct sequence of character processing but also eliminates the noise that could remain after image processing. Then, based on the intervals between letters, paths are combined into words.

As a result, the pseudocode of the text recognition program looks like this Fig. 2:

```
Algorithm
Input:
Sample X^m={x_1,…,x_m}
k – number of clusters;
treshold – line spacing threashold;
Output: output_text – image text
```

```
1. Image clustering
2. Masque building
3. Letter paths selection contours

lines = []
for cnt in contours:
if moment of cnt not in lines then add cnt on line
for line in lines:
mean_dist = mean distance between paths
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] with space

result = “”
for <line in lines> do:
for <i:=0 to len(line)> do:
result += Classification cnt[i]
```

Fig. 2. The pseudocode of the text recognition program.

The developed OCR effectively highlights text and classifies small words. However, during the development process, there were problems when classifying large volumes of text. Therefore, the final version of the program for extracting text from images includes two stages. At the first stage, the EAST algorithm selects text blocks. If their amount is big and they form a large group, then Tesseract is used to recognize them. If the number of text blocks is not large, the formerly described approach for text extraction is used.

Pseudocode for the final text extraction and recognition algorithm "Internet meme":

```
img; ◂<S – input image >
n = minimal number of words;

textblocks = EAST(img); ◂<extracting text from image>
combining all intersecting text blocks;
```

```
if len(textblocks) > n then
txt = Tesseract(textblocks);
else
txt = TEXTEXTRACTOR(textblocks);
show txt;
```

Fig. 3. Algorithm basic steps for text detection.

3.2 Text Classification

The social network Vkontakte was chosen as a medium for Internet memes dissemination. The objects of classification are records containing text, an application in the form of an Internet meme, and comments. Such objects can be classified as political or non-political. An object is classified according to the class of the record text, the Internet meme text, and comments text. So, it is necessary to build a classifier detecting whether the text is political or not.

To solve this problem, we used a sample consisting of 63 political memes and 44 non-political ones. Some additional objects were taken from the social network VKontakte and were represented by comments having both political and non-political content. In total, 168 phrases were used for training. A validation sample contained 11 political and 7 non-political memes. The data preprocessing stage included data normalization with the stemming algorithm and converting data into a vector representation.

The support vector machine [9] was chosen as a classification algorithm taking into account the small size of the training data.

Word clouds were built for both classes (Fig. 4, 5). Using these clouds it’s possible to note that the "political memes" class includes images containing the names of countries, regions, and political leaders. The "non-political memes" class does not have a clearly defined group of words, because sentences related to this class have different topics.

The classifier demonstrated an accuracy of 81.25 % on the test sample. With the growth of the training data, the accuracy of the classifier will be improved.

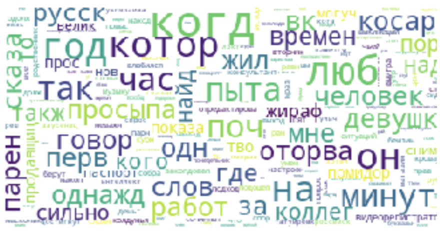

Fig. 4. A word cloud of objects from the non-political class

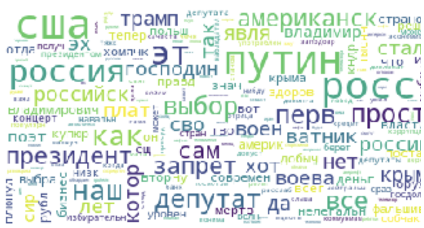

Fig. 5. A word cloud of objects from the political class

4 Data Collection and Analysis Using the Internet Meme Processing and Analysis System

Five groups from the social network Vkontakte were selected for analysis. We managed to extract 43 political memes using the developed tools. A list of extracted political memes allowed to obtain lists of participants who "rated" the selected entry.

A total of 66,184 participants were collected. A separate list was created for cities where the selected participants live. This list was used to construct a pie chart displaying the most politically active regions of Russia. The most active regions are Moscow, St. Petersburg, Yekaterinburg, Novosibirsk, Krasnodar. Cities with a frequency less than 0.4% were assigned to a separate category "other".

To identify groups of cities with the same political activity in the obtained data we used the K-means algorithm [9]. The data was divided into 5 clusters. The first cluster included cities with the least interest in political memes: Korolev, Pskov, Nizhnevartovsk, Blagoveshchensk, Engels, Taganrog. Moscow was extracted into the second cluster. The following cities were grouped into the third cluster: Yekaterinburg, Novosibirsk, Krasnodar, Rostov-on-don, Nizhny Novgorod, Chelyabinsk, Perm, and Samara. The fourth cluster was represented by the city of Saint Petersburg. The fifth cluster included the cities of Tomsk, Minsk, Saratov, Tyumen, Kaliningrad, Vladivostok, Yaroslavl, and Irkutsk.

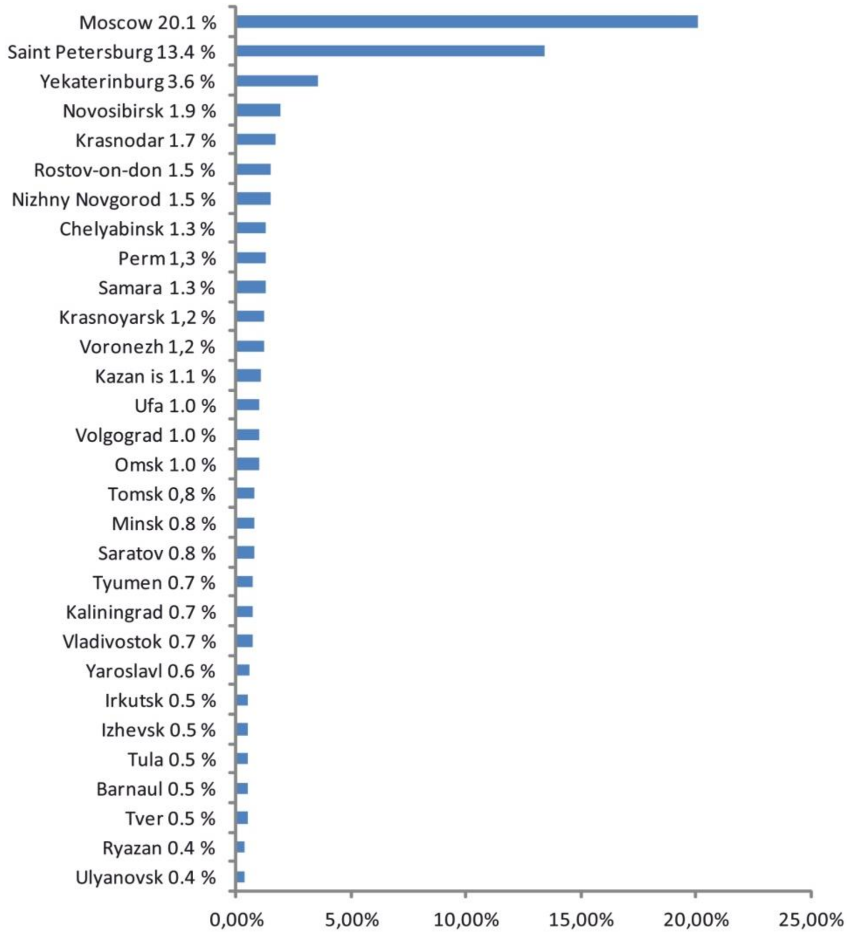

Fig. 6. Frequency diagram of Russian cities' political activity.

Conclusions

Algorithms and software to select text from images, classify text, and cluster obtained data are elaborated. Distinct algorithms of text and image classification are subjected to comparative analysis. The algorithms to extract text from an image and to render images are implemented:

- finding text using the algorithm EAST;
- grouping images containing text information;
- preprocessing images containing text information;
- detecting text lines on images;
- extraction of letter paths;
- classification of letters;

– combining letters to get a text.

A system to process and analyze the Internet memes flow is designed.

The drawbacks of the obtained model refer to the accuracy of meme classification. So, it demonstrated only 85% accuracy in determining whether a record belongs to political memes or not. It is mainly caused by an insufficient set of training data, lack of image classification, errors when extracting text from images. To improve the quality of classification, it is necessary to increase the amount of training data, improve the image extraction algorithm, and implement an image recognition algorithm. Some of the obtained results are represented in [10, 11]. The further work supposes the implementation of algorithms that can predict meme popularity using network metrics.

References

1. Gubanov D.A., Novikov D.A., CHkhartishvili A.G. Social'nye seti: modeli informacionnogo vliyaniya, upravleniya i protivoborstva [Social networks: models of information influence, management and confrontation]. Pod red. chl.-korr. RAN D. A. Novikova [Ed. Corresponding Member RAS D. A. Novikov]. M.: Publishing house of physical and mathematical literature, 2010. (In Russ.).
2. Chzhan P., Zaharov V. P. Komp'yuternaya vizualizaciya russkoj yazykovoj kartiny mira [Computer visualization of the Russian language picture of the world]. International Journal of Open Information Technologies. 2020, vol. 8, № 1, pp. 58-62. DOI: 10.24412/FfNhqoIQLC0 (In Russ.).
3. Kutuzov A., Kuzmenko E. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science. 2017, vol 661. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-52920-2_15
4. Zaharov, V. P. Funkcional'nost' instrumentov korpusnoj lingvistiki. [Functionality of corpus linguistics tools]. V I. S. Nikolaev (Red.), Strukturnaya i prikladnaya lingvistika. Mezhvuzovskij sbornik: Vypusk 12. K 60-letiyu otdeleniya prikladnoj, komp'yuternoj i matematicheskoj lingvistiki SPbGU [In I.S. Nikolaev (Ed.), Structural and Applied Linguistics. Interuniversity collection: Issue 12. To the 60th anniversary of the Department of Applied, Computer and Mathematical Linguistics of St. Petersburg State University]. Publishing House of St. Petersburg University. 2019, pp. 81-95. (In Russ.).
5. Namiot, D. E., Makarychev, I. P.: On an alternative model of location marking on social networks. International Journal of Open Information Technologies. 2020, vol. 8, № 2, pp. 74–90. Available at: http://www.injoit.org/index.php/j1/article/view/887
6. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. Neural computation. ISSN: 0899-7667. 1997, vol. 9, № 8, pp. 1735-1780. DOI: 10.1162/neco.1997.9.8.1735
7. Tian Z., Huang W., He T., He P., Qiao Y. Detecting Text in Natural Image with Connectionist Text Proposal Network. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. Lecture Notes in Computer Science. 2016, vol. 9912. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-46484-8_4
8. Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. Neural Networks. 2015, vol. 61, pp. 85-117. DOI: 10.1016/j.neunet.2014.09.003
9. Voroncov, K. V. Matematicheskie metody obucheniya na precedentah. Kurs lekcij po mashinnomu obucheniyu [Mathematical teaching methods on precedents. Machine Learning Lecture Course]. Available at: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-

1.pdf (In Russ.).

10. Germanchuk, M. S., Kozlova, M. G., Luk'yanenko, V. A. Problematika modelirovaniya processov rasprostraneniya internet-memov [The problem of modeling the processes of the distribution of internet memes]. V sbornike: Analiz, modelirovanie, upravlenie, razvitie social'no-ekonomicheskih sistem. sbornik nauchnyh trudov XII Mezhdunarodnoj shkoly-simpoziuma AMUR-2018. Pod obshchej redakciej A. V. Sigala [In the collection: Analysis, modeling, management, development of socio-economic systems. collection of scientific papers of the XII International School-Symposium AMUR-2018. Edited by A.V.Sigal]. 2018, pp. 136-139. (In Russ.).
11. Germanchuk, M. S., Kozlova, M. G. Raspoznavanie, analiz i vizualizaciya internet-memov [Recognition, analysis and visualization of Internet memes]. Matematicheskie metody raspoznavaniya obrazov: Tezisy dokladov XIX Vserossijskoj konferencii s mezhdunarodnym uchastiem [Mathematical methods of pattern recognition: Abstracts of the XIX All-Russian conference with international participation]. Moscow: Russian Academy of Sciences. Pp. 351-355. (In Russ.).