Using topic modeling for communities clusterization in the VKontakte social network

Sergey Gorshkov, Eugene Ilyushin, Anastasia Chernysheva, Viacheslav Goiko, Dmitry Namiot

***Abstract*—Topic modeling is one of the most widely used methods in text analysis. It can be used to select topics as well as to find the topics distributed in each document from the corpus. In this article, we present a method for clustering communities in the social network VKontakte (the most popular Russian social network) using topic modeling. As a communities sample a set of groups for which several students of Tomsk State University are subscribed was selected. There were about 7,000 of them in this set. The article describes the method by which the text corpus was formed, as well as mathematical modeling using two popular classical methods LDA and ARTM. A detailed description of these models, quality assessment criteria, and the main practical techniques used by the authors in training the models are given. The aggregated results of clustering communities by topic are also presented. There are also described a method for expert evaluation of community topics based on visualization of the words that make up the lexical core of the topic.**

***Keywords*—topic modeling, LDA, ARTM, social network analysis**

I. Introduction

Currently, social networks are one of the largest sources of data. Social network analysis [1] is used to solve a wide range of problems in science and business. In the modern Russian and global IT sector, popular social networks are part of large ecosystems that provide a diversity of services and products. Of course, to increase the user's interest in the social network, to sell him a product or service, the services use a diversity of recommendation systems [2]. They are based on the user's interests, the user's friend graphs, and other information about actions on the Internet. Based on the interests of "similar" users in a specific metric, you can improve recommendations for each person, increase the average time spent in the service, having some completeness of information, taking into account the privacy policy. For tasks that are solved from outside the ecosystem that includes the social network, you need to use open data that can be uploaded via the API or using web scraping. Such tasks are diversified. For example, for business, it can be conducting a targeted mailing of a specific product or service to a particular user group, offer to join a community, searching for potential employees, and so on. For researchers, it is essential to identify different groups to research in the area of sociology, attract applicants to universities, identify extremism and other violations of the law in social networks, and so on. However, how to distinguish all groups? How do we find out what users ' interests are? The user is subscribed to those communities that are interesting to him for a range of reasons. Based on this simple hypothesis, if you get the topics distribution of the user's communities, you can find out the distribution of their interests. Then you can improve recommendations by offering similar content to users with similarly distributed topics. It is no secret that every user leaves their digital footprint on the Internet. Based on the digital footprint in a particular social network, using the page estimation and the user's interests, it is possible to predict their preferences, various psychological and social characteristics, which can later be used for different science and business tasks. However, all thoughts about the user's interests, the allocation of groups, rest on some communities classification that persons subscribe to. In this article, we solve the problem of classification of communities by topics that are presented in the community content. Of course, that content can be attributed to different topics, so we look for the distribution of topics within each community. The task of such a classification can be solved by experts who identify the main topics and mark up a limited number of communities. Such a problem can be solved with the help of crowdsourcing projects in Russia, it is for example Yandex.Toloka. However, this task has problems with scalability and significantly depends on the human factor. We offer a solution based on machine learning methods with minimal involvement of experts in data markup and evaluation of the result.

Manuscript received April 19th, 2021

Sergey Gorshkov – Lomonosov Moscow State University; National Research University Higher School of Economics, (email: serggorsar@yandex.ru)

Eugene Ilyushin – Lomonosov Moscow State University, The faculty of Computational Mathematics and Cybernetics, (email: eugene.ilyushin@gmail.com)

Anastasia Chernysheva – Lomonosov Moscow State University; Skolkovo Institute of Science and Technology, (email: nastya.ch.9797@mail.ru)

Viacheslav Goiko – Tomsk State University; Sirius University of Science and Technology, (email: goiko@ftf.tsu.ru)

Dmitry Namiot – Lomonosov Moscow State University; RUT (MIIT), (email: dnamiot@gmail.com)

The data of the students were obtained with the financial support of the Russian Foundation for Basic Research within the framework of the scientific project No. 19-31-51024 "The influence of the structural and content characteristics of the Internet activity of senior pupils and students on their educational achievements."

II. About topic modeling

A. Description of the topic modeling problem and its applications

Each community can be assigned its unique identifier in the social network (some string that allows you to access the community via the URL). We also match each community its content. The type of information can be different, for instance, text, images, audio, video, links, geolocation, list of subscribers, etc. As part of our research, we consider only its text component as content. So, each community is mapped

to a corresponding text document. The topic distribution of this text document is considered as the topic's distribution of the community content. Thus, we get a problem in which we have text documents collection and it is needed to select a set of topics and find its distribution. One of the most popular approaches to solve such problems is topic modeling [3]. A topic in topic modeling is a set of words or phrases that occur together in documents. Further, as a *word* word, we will understand one word or phrase. Each topic is represented by a dictionary and a probability distribution of words from this dictionary, i.e., the probability $p(\omega|t)$ to meet the word ω in the topic t. Denote the set of all topics T, the set of all words (dictionary) W. Classical models assume that word order doesn't important, and represents the document as a "Bag of words". The observed word frequencies in the documents are define the distribution $p(\omega|d)$ of the word ω in the document d. One document can be related to several topics. For its reflection, the concept of a document topic is introduced – the distribution of $p(t|d)$topics t in document d. This is our target variable which we need to find to solve the original problem. The distribution of words in document $p(\omega|d)$ can be represented by the distribution of words in topic $p(\omega|t)$ and topics in documents $p(t|d)$ as follows [4], [5]:

$$p(\omega|d) = \sum_{t\in T} p(\omega|t)p(t|d) \qquad (1)$$

The desired distribution $p(t|d)$ is usually estimated using various mathematical methods, which we will discuss later. In this paper, we consider Latent Dirichlet Allocation (LDA) and Additive Regularization of Topic Models (ARTM). Topic modeling is used to solve a wide range of tasks, such as automatic annotation of documents, creation of rubricators for text collections, classification of different text-type information – scientific papers, news, comments, etc. The development of the IoT technology, as well as the creation of smart home voice-controlled elements by IT ecosystems, led to the use of thematic modeling to detect groups of requests. Then, these groups can be used as a training sample to the problem of user commands classification for a smart home.

B. About LDA model

This method was proposed in [6] in 2003. LDA differs from earlier methods that the same words can occur in several topics, which reflects the flexibility of the language, and also avoids problems with homonyms and homographs. The probabilities $p(\omega|t)$ and $p(t|d)$ from (1) can be considered as elements of the matrix of word distributions in topics Φ, $\Phi = (\phi_{\omega t})$ and the matrix of topic distributions in documents Θ, $\Theta = \theta_{td}$ respectively. Then the problem can be considered as a matrix decomposition problem. The columns of these matrices can be generally considered as normalized, that is, $\sum\limits_{\omega\in W} \phi_{\omega t} = 1$ and $\sum\limits_{t\in T} \theta_{td} = 1$, and task decomposition can be led to maximizing the logarithm of likelihood

$$\sum_{d\in D}\sum_{\omega\in d} n_{d\omega} ln \sum_{t\in T} \phi_{\omega t}\theta_{td} \to \max_{\Phi,\Theta}, \qquad (2)$$

where D is the number of documents in the collection. Here $n_{d\omega}$ is the real number of words ω in the document d. Let's denote by n_d the number of elements in the document d. Then, using $n_{d\omega}$ and n_d can be obtained $p(\omega|t) = \frac{n_{d\omega}}{n_d}$ – the value for the word frequencies observed in a sample. The solution to the matrix decomposition problem will be non-unique, so we have to impose some additional constraints on $R(\Phi, \Theta)$, called regularizers. The regularizer function must be continuously differentiable. Then a new component will be added to the log-likelihood maximization problem, and the optimization problem will take the form

$$\sum_{d\in D}\sum_{\omega\in d} n_{d\omega} \ln \sum_{t\in T} \phi_{\omega t}\theta_{td} + R(\Phi, \Theta) \to \max_{\Phi,\Theta},$$

The first model, called PLSA (Probabilistic Latent Semantic Analysis), proposed in [7], assumed $R(\Phi, \Theta) = 0$. The LDA model assumes that

$$R(\Phi, \Theta) = \sum_{t\in T}\sum_{\omega\in W} (\beta_\omega – 1) \ln \phi_{\omega t} + \sum_{d\in D}\sum_{t\in T} (\alpha_t – 1) \ln \theta_{td}.$$

Here β_ω and α_t are positive tunable hyperparameters. The latent Dirichlet allocation model assumes that the document vectors θ_d are generated by the same probability distribution on normalized vectors, and the distribution is taken from the family of Dirichlet distributions [8] with the parameter α. Similarly, the topic vectors ϕ_t are generated by the Dirichlet distribution with the parameter $beta$.

C. About ARTM model

In the likelihood maximization functional (2), we can add not one regularizer, but several, giving weight to each term. This approach is called additive regularization, and, as the name suggests, is the basis of ARTM [9], [10]. By specifying different regularizers, we can impose the desired restrictions on the matrix decomposition parameters. For example, if you want to add some regularization so that the divergence Kullback-Leibler [11] (it is essentially a distance between probability distributions) of the relevant rows of Φ and columns of Θ was minimal relative to the specified distributions β_ω and α_t for each word and each topic respectively, we get a smoothing regularization

$$R(\Phi, \Theta) = \beta_0 \sum_{t\in T}\sum_{\omega\in W} \beta_\omega \ln \phi_{\omega t} + \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_t \ln \theta_{td}.$$

ARTM with such a regularizer is the same as LDA, so ARTM can be considered as an extension of LDA. If we want the distance between the topics to be large, we need to zero out the values in the vectors, that is, to maximize the distance between the trainable and the fixed distribution. For this purpose, we can use sparsity regularization:

$$R(\Phi, \Theta) = –\beta_0 \sum_{t\in T}\sum_{\omega\in W} \beta_\omega \ln \phi_{\omega t} – \alpha_0 \sum_{d\in D}\sum_{t\in T} \alpha_t \ln \theta_{td},$$

choosing distributions β and α as uniform. By adding different regularizers, you can ensure that each topic has a set of terms that distinguish it from others, select topics (after applying regularization, the probabilities for the most insignificant topics will be small), get rid of linearly dependent topics, and so on [5].

D. Evaluating quality methods of topic models

There are two fundamentally different approaches to evaluate the quality of topic models. The first allows you to evaluate the quality based on the decomposition into matrices Φ and Θ. At the same time, it can't be said anything about how good the topics turned out to be for subsequent analysis. The most famous measure, called perplexity, is related to the likelihood that was used in the formulation of the maximization problem and has a cross-entropy nature.

The perplexity for the document collection D is calculated as follows:

$$P(D) = \exp(-\frac{1}{n} \sum_{d \in D} \sum_{\omega \in d} n_{d\omega} \ln p(\omega|d)).$$

This measure can be interpreted as a word's difference in the text. The less evenly words distributed in the topics, the lower the value of perplexity. Perplexity can only be compared within a single model, i.e. with a fixed number of topics and fixed regularizers. This metric is very convenient for selecting the required number of iterations of training a topic model. Perplexity can be considered as a loss function, and when its value changes insignificantly at each subsequent iteration, the learning process can be stopped. The second approach is based on expert estimation of the resulting topics. This is a more important indicator from a business point of view because it is clear and allows you to understand how the received topics are interpreted and suitable for further analysis. To evaluate the resulting topics, it is used the top-N words that have the greatest contribution to the topic, that is, have the greatest probability of meeting in this topic. There are several techniques here. For example, you can ask experts to give a title to a topic based on the top N words in it. Or you can add an extra word to the top-N words and ask the experts to find it; there are a lot of options where this extra word can be taken from. It is these estimates we use as final for our topic model. In [12], [13], it was shown that such an evaluation measure as coherence, calculated without the participation of experts, well correlates with expert estimates. For coherence calculations, we use the top-k words by the weight value in the matrix Φ, denote as ω_i the word with the i-th largest value $\phi_{\omega t}$. The topic's coherence shows words similarity, and is calculated as the average pointwise mutual information for top-k words of each topic by passing through the text fixed-size window:

$$PMI_t = \frac{2}{k(k–1)} \sum_{i=1}^{k–1} \sum_{j=i}^{k} PMI(\omega_i, \omega_j),$$

where k is the window size.

$$PMI(a, b) = \ln \frac{P(a\, near\, b)}{P(a)P(b)},$$

where $P(a)$ and $P(b)$ are the probabilities of appearing the words a and b in the corpus, and $P(a\, near\, b)$ is the probability of meeting words a and b side by side in a window of width w, by default $w = 10$. The more likely two words are not coincidentally sided by side, the greater the coherence. However, as a result of the model operation, several identical or very similar topics can be identified, but coherence cannot detect this. Many topic models, including non-classical ones, are subject to this since there is nothing to prevent the models from learning two identical or very close distributions. To solve this problem, it is either need to reduce the number of topics or add some regularizers. In our problem, we will use coherence as assistance during selecting the hyperparameters of the model, including the number of topics.

III. Related works

It was not possible to find articles that solve exactly the problem we set. However, many papers use topic modeling to solve various types of social network analysis problems. In [14], the study of extremist groups is carried out using the intellectual analysis of texts. The problem is to find group leaders on the forum, as well as to identify subgroups. Data from forums is used as a text corpus. 36 relevant topics were identified using the LDA. In [15], the authors apply topic modeling to build a category of topics in social networks. Email messages are used as data, and the results are taken to improve the product support service. In [16], the authors solve the problem of users clustering based on their "vocabulary" and then use obtained information for recommendations. That article also uses a modified version of the LDA, where customers are represented through topics, and this correspondence changes over time, resulting in a dynamic recommendation model that allows you to take into account the changing interests of users over time. Topic models are also used, for example, to analyze the mood that arises as a result of conversations in social networks [17]. The LDA-based model allows you to find topics in a conversation and connect selected topics with emotions. A subset of Twitter posts with a single hashtag was taken as a sample. Dmitry Sergeev did a very fascinating job in [18] for topic modeling of comments on VKontakte and Youtube, linking groups of users by the words they use in the comments and genres of Russian music. Based on this, clusters of music that people with similar comments topics listen to were identified. ARTM is a newer and evolving approach, so it's not surprising that researchers use LDA in most of their articles. A non-standard application of the LDA idea is used, for example, in [19] to analyze the graph of relationships in social networks. It deals with the problem of clustering users and their thematic interests for the social network Twitter.

IV. Proposed Method

A. Data collection

The social network VKontakte was chosen for our study since it has an open and simple API [20] (hereinafter referred to as the VK API). As a sample, we took communities that were subscribed to by at least 5 students of Tomsk State University. The selection of communities was carried out as follows: the profiles of 4,513 students were taken, after which the open profiles were selected from them. Further, users who hid the list of their communities were excluded. Then, using the VK API, a list of all the communities they subscribed to was uploaded for each user. The result consisted of 6967 communities. There are quite a lot of people in the sample, therefore all possible frequency topics of the communities probably met. Further, since all users study in the same region, they are subscribed to similar communities with a regional link (pizza delivery in Tomsk, a nail salon in Tomsk, driving school in Tomsk, etc.). This is great for model training because it looks for differences in terminology, rather than differences between toponyms, store names, and so on. If we need to calculate a community topic of another city, the trained model will easily do this, since the significant words will be directly defining for the topic. At the same time, such words as «Tomsk» and popular brands in the region will become the background, as they will be found in most of the communities. Anyone can use the VK API to upload various information about the community. We are interested in some text and service information, namely the name of the community, its description, the last 50 entries on

the wall (or all of them, if there are less than 50). For each record, information is stored about whether it is advertising or not. We discard advertising posts, because in most cases they do not correspond to the topic of the community, but are selected based on the user's preferences, their request history, location, or other considerations.

B. Data preparation

We convert the resulting set of information for each community into a text document. At the same time, it should be taken into account that in some communities, there is no text information outside of advertising posts. All information can be represented by images or links to other resources without a text description. To classify such communities, we also suggest using its name and description, which often reflect the topic of the community's posts. If the community posts contain very little text information, but mostly images, then there is a high risk of encountering hidden advertising in the community (such posts are not marked as advertising and therefore we impossible remove them from consideration after uploading them from the Internet). Such information distorts the thematic profile of the community. When composing a text document, we write all the text information from the 50 last uploaded records that are not marked as advertising once, 5 times the community's name, and 5 times its description. This gives more weight to the last two fields of the description and reduces the impact on the subject of the document of random entries from other topics. At the same time, if there is a lot of text content in the community, then the name and description added 5 times will not play a significant role. So, we got a text document for each community. Next, we process the text. We perform it in Python, as well as the further practical part of the study. In each document, we replace all punctuation marks with spaces (except for underscores, octothorpes, and hyphens that separate two parts of the word), after which we replace several consecutive spaces with one. Remove Unicode characters that do not have a text representation. We also reduce all words to lowercase, after which we use the pymorphy2 library to perform lemmatization, that is, to bring word forms to normal form. Remove the stop words from the text (we take them from the nltk library), that is, frequently encountered words of the language that do not carry a semantic load. To perform this processing in parallel for all the documents in the collection, we use the dask library. After that, we check that the received documents were not empty. Three documents turned out to have an empty text description, these communities were closed and had no name. So, we have a collection of text documents, which we classify using topic modeling. Next, we use the following practical technique – we train an LDA model with a sufficiently large number of topics, set other parameters as default, and look at the top-K words in the topics. We expect that some «trash» topics should stand out. And so it happened, it turned out that the corpus contains quite a lot of separate Russian, Greek, and other letters, as well as short letter combinations, which are either cut off parts of words, according to which words cannot be restored, or nonsense. We immediately remove all these letters and letter combinations from the corpus.

C. Application of LDA model

Let's use the implementation of the LDA model in the gensim library. We use the LdaMulticore class, because it allows us to parallelize training, and corpora in our sample are quite large. In gensim's LDA implementation, the alpha and eta hyperparameters specify the parameters of the a priori Dirichlet distribution. We iterate over the various values of these parameters. These can be either vector with different values («asymmetric»), or vectors consisting of the same numbers («symmetric»), and you can explicitly specify the vector. Also, for both parameters, you can specify the value «auto», at which the asymmetric distributions will be trained on the source data. In the last case, we can't use multicore LDA with auto-tuning alpha, so we just use LdaModel. Let's fix the number of topics equal to 50, and run a search through the grid of different combinations of the $alpha$ and eta, parameters to select the three best models. At this stage, we evaluate the quality of the constructed model based on coherence. The three best combinations for parameter pairs $(alpha, eta)$ were $(symmetric, None)$, $(auto, None)$ and $(asymmetric, None)$. As we can see, the defined a priori probability eta of the words does not lead to winning as a model. Let's determine the optimal number of topics for each model and go through the number of topics in the range from 20 to 70, evaluating the coherence. The paper [21] describes how to choose the optimal number of topics, but for our task, it is easier to focus on the interpretability of topics due to the nature of the data. As described above, this is not enough, and therefore, for reasonable values of coherence, we manually look for collinearity and appearance of trash topics at the top-K topic words. Defined the optimal number of topics equal to 40, for each of the three leading models we proceed to select the optimal number of iterations, estimating the percentage of improvement in perplexity at each step. Note that after 10 iterations, the perplexity value changes in all three models by less than 1% per subsequent step, so after 10 passes through the collection, we interrupt training to save the algorithm time. Finally, let's look at the topics and give them an expert estimation, as well as look at the distribution of document topics in the collection. Here, for each document, we consider only the topic that has the largest share in the distribution of topics in the document. As a result, we select the LDA model with the parameters $(auto, None)$. With this combination, coherence takes the best value, the topics are well interpreted and there are no duplicates in it. We visualize the top 50 words in each topic and use experts to choose its name. Here and further, we select the major topic for each document – it is the topic that has the highest probability in the distribution of all topics for this document. Here are the names obtained for the most frequently encountered topics (topics that were major in at least 200 documents) – «beautiful photos», «about love», common words, trash topic, «music», «photographers», «literature», «education», «travel», «design and fine arts», «politics», «shops and purchases». Under the «trash» topic we imply a set of words parts and whole words that do not have any meaning.

D. Application of the ARTM model

We use the BigARTM library for working with ARTM models. BigARTM can operate with several data modalities. According to our research, it can be the text, hashtags, links, images, locations, and so on. During training, different weights can be assigned to different modalities to adjust the strength of the modalities' influence on the model. In this article, we use only the text modality. In the same way as in

LDA, we need to configure an important hyperparameter – the number of topics. We add metrics of the Φ and Θ matrices sparsity to evaluate the advantages of its decomposition. Let's try to train a model with the number of topics equal to 40 (as we got in LDA) without regularizers. After the model convergence (that is, the perplexity has started to change insignificantly at each step), we look at the sparsity values of the matrices. The percentage of zero values in the matrices Φ and Θ turned out to be approximately 0.63 and 0.01, respectively, which resulted in a rather poor decomposition and mixed topics. To solve this problem, we configure additive regularizers. To remove the common words that abound in social networks, we use a sparse regularizer of the topics words distributions matrix Φ. We choose the value of the sparse coefficient to be negative, thus the more a word occurs in the entire collection, the less it occurs at each topic. Similarly, we add a regularizer for the matrix Θ, to escape topics collinearity. By grid search, we iterate over the number of topics and regularization coefficients, where for each pair for Φ and Θ we evaluate the quality by coherence and by the percentage of zeros in the decomposition matrices. With the help of experts, we also evaluate the topics' interpretability for the best combinations. As a result, we get the optimal number of topics equal to 40, and a set of regularization coefficients such that the percentage of zeros in the matrices Φ and Θ is approximately 0.9. The required number of iterations is estimated by the convergence of the perplexity and it is equal to 10. Here are the names suggested by experts for the most frequently encountered topics (topics that were major in at least 200 documents): «girls and beauty», «family and relationships», «private blogs», «music», «about friendship», trash topic, «movies», «literature», «concerts and events», «about love».

V. Results and Discussion

Let's compare the results obtained after applying the LDA and ARTM models to the assembled case. The LDA model for each document always produces a distribution of topics in which at least one topic has a non-zero contribution. Respectively, using LDA it is possible to classify all documents. The ARTM model, in turn, can give out all zeros for individual documents in the topic's distribution, which turned out to be 13%. Failures in classification related can be divided into the following groups:

1) Documents with a very small number of words that were not included in the top-k words in any of the selected topics (most often these words were common).
2) The topic is low-frequency and therefore did not get into the list of popular topics during topic modeling.

Note that the graph of the number of documents in each topic is smoother for ARTM than for LDA. For LDA, there are topics with a very large number of documents (maximum 1206, the second-largest number is 661), also there are several with a very small number (about 10, minimum 6 documents). As a result of the ARTM work, the topic distribution is more linear (maximum 541 documents, minimum 30). In the distribution of the number of documents by topic, many documents are related to topics of medium frequency. To determine the name of each topic, we used the convenient tool wordcloud, which allows us to visualize the most significant words in the topic. Then the expert chose the name for each topic based on provided visualizations. The names obtained by keywords for the most frequent topics which were identified by two considered models were given in sections 4.3 and 4.4. for both cases, we should notice, that the selected popular topics are consistent with the experience of using and analyzing communities in social networks. Trash topics were found by using both models. ARTM identified two of them, it had a smaller size than trash topics identified by LDA. As a result of the work of LDA, the third most popular topic was the topic consisting of common words, while in ARTM, thanks to the regularizers, this problem did not arise. There is a natural desire don't classify documents as belonging to a trash topic and a topic with common words but to use the next most popular topic, which does not belong to the two classes described above, as a major one. In Figures 1 and 2, we present the major topics distribution of the document collection after all described corrections. At the same time, we renumber the topics in the order of descending the number of documents in which they are major and specify these numbers on the abscissa axis. The corresponding number of such documents is the value on the ordinate axis.

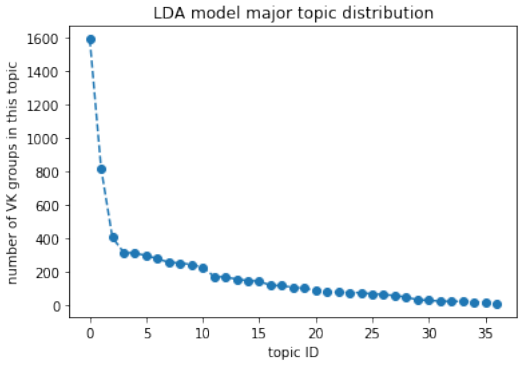

Figure 1. Distribution of topics obtained as a result of applying the LDA model in corpus

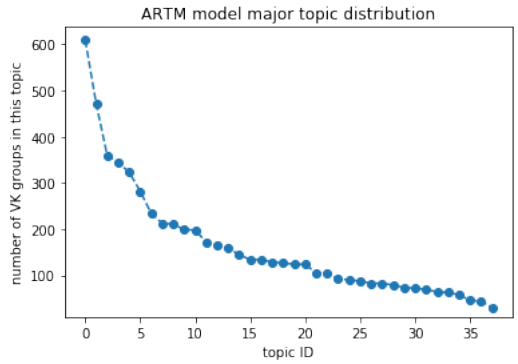

Figure 2. Distribution of topics obtained as a result of applying the ARTM model in corpus

There is a table with the names of the first 15 major topics in the order of the descending number of documents below.

Note that some topics are either found in the results of both algorithms or have close analogs in the results of the other model. In connection with the choice of the student focus group, the topic of "student events" was highlighted. In this regard, most likely, the topic of "computer games and esports" got to the top. The model's training time is also

	LDA	ARTM
1	Nice photos	Girls and beauty
2	About love	Family and relationships
3	Music	Literature
4	Traveling	Private blogs
5	Education	Music
6	Photographers	About friendship
7	Literature	Movies
8	Design and fine art	About love
9	Shops and purchases	Concerts and events
10	Politics	Shops and purchases
11	Goal setting and motivation	Drawing and creativity
12	Regional communities	Languages studying
13	Sport	Student events
14	Cooking recipes	Computer games and esports
15	Computer games and esports	Politics and news

Table I
Names of the top 10 most frequently encountered major topics

important. The LDA model was trained for 10 iterations in an average of 10 minutes 47 seconds, 10 iterations of ARTM — 3 minutes 44 seconds. All calculations were performed on the Google Colab platform.

VI. Conclusion and plans for further research

As a research result, topic modeling of the social network VKontakte communities was performed using the LDA and ARTM models. The necessary hyperparameters and regularizers were selected using various estimations of the model quality. The results of these two topic modeling approaches were compared, and the topics that were most often the most significant in the documents were considered. Thus, each group can be assigned to a topic number (or several topics with corresponding weights) and then use the topics distribution across all user subscriptions as attributes that reflect the user's interests. Obtained results are already applicable to many practical problems. Each of the two discussed approaches has its strengths and weaknesses. However, a rather long search of hyperparameters, which can be largely considered manual, including the need to specify the number of topics, can be safely attributed to the disadvantages of topic modeling. As plans for further research, we can identify the following areas:

1) Using n-grams in dictionary formation.
2) Using neural networks for topic modeling tasks.
3) Separately, it is necessary to consider the application of BERT [22] to solve the considered problem.
4) Using other modalities for the ARTM model, primarily hashtags.
5) Using multimodal clustering [23] for community detection

References

[1] N. Aydin, "Social network analysis: Literature review," vol. 9, no. 34, pp. 73–80. [Online]. Available: https://dergipark.org.tr/tr/pub/ajit-e/issue/54418/740686

[2] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," vol. 17, no. 4, pp. 305–338. [Online]. Available: http://link.springer.com/10.1007/s00799-015-0156-0

[3] A. Korshunov and A. Gomzin, "Tematicheskoe modelirovanie tekstov na estestvennom yazyke," vol. 23, pp. 215–244. [Online]. Available: https://www.elibrary.ru/item.asp?id=18361454

[4] "Topic models in practice. specialization "machine learning data analysis" on coursera." [Online]. Available: https://www.coursera.org/lecture/unsupervised-learning/tiematichieskiie-modieli-na-praktikie-O5QDm

[5] V. Bulatov, "Metody otsenivaniya kachestva i mnogokriterial'noy optimizatsii tematicheskikh modeley v biblioteke TopicNet." [Online]. Available: https://mipt.ru/upload/medialibrary/c25/bulatov_dissertation_topicnet_signature.pdf

[6] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," vol. 3, no. 4, pp. 993–1022. [Online]. Available: https://dl.acm.org/doi/10.5555/944919.944937

[7] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*. ACM Press, pp. 50–57. [Online]. Available: http://portal.acm.org/citation.cfm?doid=312624.312649

[8] S. Kotz, N. Balakrishnan, and N. Johnson, "Chapter 49: Dirichlet and inverted dirichlet distributions," in *Continuous Multivariate Distributions. Volume 1: Models and Applications*. [Online]. Available: http://www.ru.ac.bd/wp-content/uploads/sites/25/2019/03/201_09_Kotz_Continuous-Multivariate-Distributions-Models-and-Applications.pdf

[9] K. V. Vorontsov, "Additive regularization for topic models of text collections," vol. 89, no. 3, pp. 301–304. [Online]. Available: http://link.springer.com/10.1134/S1064562414020185

[10] K. Vorontsov and A. Potapenko, "Additive regularization of topic models," vol. 101, no. 1, pp. 303–323. [Online]. Available: http://link.springer.com/10.1007/s10994-014-5476-6

[11] S. Kullback and R. A. Leibler, "On information and sufficiency," vol. 22, no. 1, pp. 79–86. [Online]. Available: http://projecteuclid.org/euclid.aoms/1177729694

[12] D. Newman, J. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," pp. 100–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=1857999.1858011

[13] D. Newman, S. Karimi, and L. Cavedon, "External evaluation of topic models," pp. 11–18.

[14] G. L'Huillier, S. A. Ríos, H. Alvarez, and F. Aguilera, "Topic-based social network analysis for virtual communities of interests in the dark web," in *ACM SIGKDD Workshop on Intelligence and Security Informatics - ISI-KDD '10*. ACM Press, pp. 1–9. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1938606.1938615

[15] Muon Nguyen, Thanh Ho, and Phuc Do, "Social networks analysis based on topic modeling," in *The 2013 RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*. IEEE, pp. 119–122. [Online]. Available: http://ieeexplore.ieee.org/document/6719878/

[16] S. S. Lee, T. Chung, and D. McLeod, "Dynamic item recommendation by topic modeling for social networks," in *2011 Eighth International Conference on Information Technology: New Generations*. IEEE, pp. 884–889. [Online]. Available: http://ieeexplore.ieee.org/document/5945352/

[17] D. Naskar, S. Mokaddem, M. Rebollo, and E. Onaindia, "Sentiment analysis in social networks through topic modeling," pp. 46–53. [Online]. Available: https://www.aclweb.org/anthology/L16-1008

[18] D. Sergeev, "Python for topic of VKontakte comments. PyDaCon meetup, 2019." [Online]. Available: https://youtu.be/MEBjnGaHsmw

[19] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*. ACM Press, p. 565. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2348283.2348360

[20] API of vk.com. [Online]. Available: https://vk.com/dev/openapi

[21] F. Krasnov, "Evaluation of optimal number of topics of topic model: An approach based on the quality of clusters," vol. 7, no. 2, pp. 8–15. [Online]. Available: http://injoit.org/index.php/j1/article/view/656/659

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, pp. 4171–4186. [Online]. Available: http://aclweb.org/anthology/N19-1423

[23] D. I. Ignatov, A. Semenov, D. Komissarova, and D. V. Gnatyshak, "Multimodal clustering for community detection," in *Formal Concept Analysis of Social Networks*, R. Missaoui, S. O. Kuznetsov, and S. Obiedkov, Eds. Springer International Publishing, pp. 59–96, series Title: Lecture Notes in Social Networks. [Online]. Available: http://link.springer.com/10.1007/978-3-319-64167-6_4