Topic Model Visualization with IPython

Sergey Karpovich[1], Alexander Smirnov[2,3], Nikolay Teslya[2,3], Andrei Grigorev[3]
[1]Mos.ru, Moscow, Russia
[2]SPIIRAS, St.Petersburg, Russia
[3]ITMO University, St.Petersburg, Russia
cims@yandex.ru, { smir, teslya}@iias.spb.su, a.grigoriev@crowd-systems.ru

***Abstract*—The paper introduces an approach to topic model visualization that is characterized by wide possibilities of choosing a method of visualization, user-friendly model representation, and simplicity of implementation for applications. The existing approaches to topic models visualization have been analyzed, and a system, which allows choosing data source for topic models, changing modeling parameters and visualizing the result of topic modeling with IPython has been developed. The example of topic model visualization has been built using the SCTM-en corpus of original news text.**

I. Introduction

Topic modeling algorithms are one of the promising directions in distributional analysis of natural language processing [1]. Distributional semantics is a language research method based on the study of the environment of individual units in text, which does not use the information on the full lexical or grammatical meaning of these units. The most widely known distributional semantic models are vector space model, latent semantic analysis, topic model, and predictive model. Distributional semantics is an area of linguistics that deals with the calculation of the degree of semantic proximity between linguistic units based on their distributional characteristics in large arrays of linguistic data. Every word has its context vector. The collection of vectors forms a vector space.

Topic modeling is a way to build topic model of a collection of text documents. A topic model (TM) [2], [3] allows to group text documents, to determine which topics include each document and what kind of words (terms) forms each topic. TMs, which are also called probabilistic topic models, softly cluster words and documents by topics (clusters) that means that a word or a document can refer to several topics with different probabilities. As an example, synonyms will be assigned to one topic with high probability because they are often uses in similar contexts. On the other hand, homonyms will be assigned to different topics because of different contexts. TMs are generally based on the “bag of words” or “bag of documents” hypothesis, which assumes that the order of occurrence of words, as well as the order of documents in corpus does not matter.

When creating a TM, a specialist should evaluate the quality of the constructed model to get an idea of the distribution of words and themes, and to identify bugs. Systems for building TM and visualization of TM results that allow to control the parameters, to rebuild TM, and to display the distribution of words and topics in a user-friendly way are urgent for topic model specialists. The simple user-friendly interface for reviewing the results of topic modeling allows specialists to quickly get an idea about the quality of the constructed TM, to analyze text collection, and to make decision.

The article is aimed to develop a system for analyzing the text data using a probabilistic topic model features. The system provides more capabilities for managing the construction of the TM process, the choice of suitable libraries and subsystems. It combines an open library for the creation and visualization of TM that allows the operator to focus on the analysis of the modelling results.

The paper is structured as follows. Section II provides analysis of the existing approaches to the problem of visualization for topic modelling results and methods of text visual analysis. Section III describes the mathematical tool used for topic model creation. Section IV describes the proposed approach to the topic model visualization. Section V provides an example of visualization techniques for topic model that is based on corpus SCTM-en. Section VI gives a conclusion for proposed approach to topic model visualization and indicates the direction for further research in the development of systems for topic model simulation and visualization.

II. Review of the existing approaches to visualize the results of topic modeling

The following requirements for the developed system have been formed based on the existing approaches review and explored business requirements with analyzes of text data.

- TM parameters control,
- data source choice,
- system's modularity; connection, removal and replacement of necessary software libraries,
- representation of significant for the theme words,
- mapping the proximity of topics,
- interactive user friendly visualization,
- display of changes in the popularity of topics over time, temporal TM visualization.

Paper [4] offers an open source software method of visualization. The main idea lies in visualization of TM that summarizes and organizes a collection of documents. The

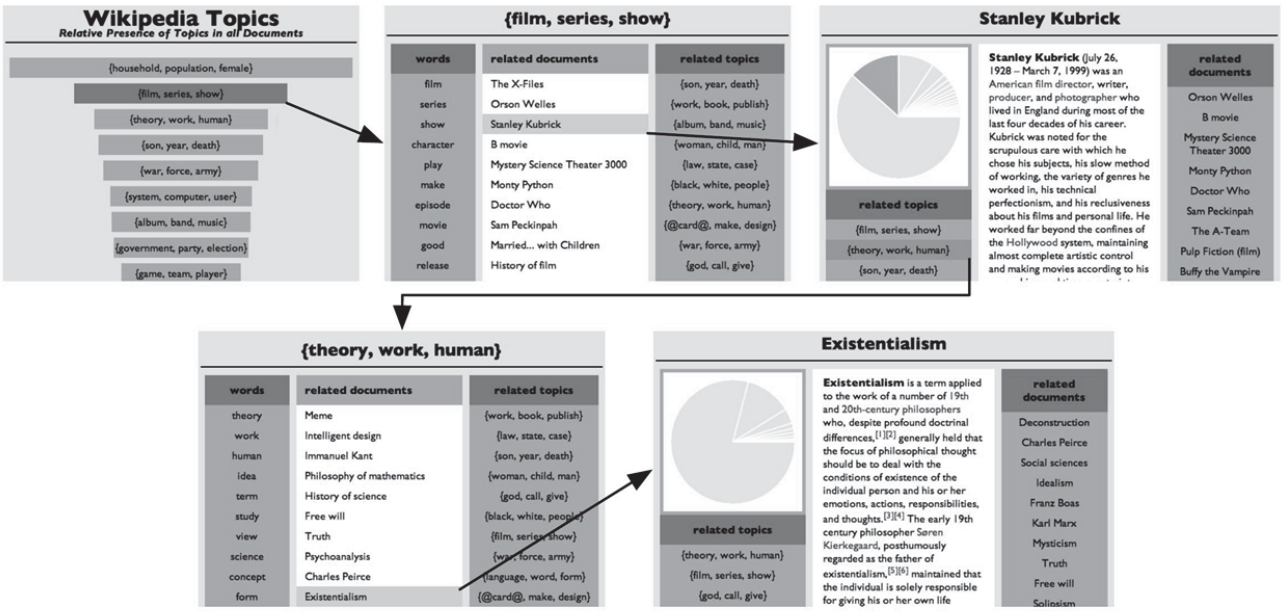

Fig. 1. The example of Blei's topic models visualization interfaces [4]

method aims to summarize the corpus for user, to reveal the relationships between the content and summaries (topics), and to reveal the relationships across content. The proposed navigator has two types of pages: topic pages and document pages. There are hyperlinks between the summary and documents used for binding pages and user navigation. Every topic unites several documents, and every document reefers to several topics. Fig. 1 illustrates an example of such visualization. The method of visualization has the following disadvantages: it is not possible to return to TM from the interface to change the parameters and to rebuild the model; it is necessary to involve professionals with specific programming skills to install and configure the system; summaries can only be presented as HTML pages; there are no user-friendly elements to represent relationships between topics and words like computer graphics and diagrams.

Paper [5] presents tools for visual analysis and evaluating topic model quality. The approach has two main interfaces: a view with term-topic relation matrix and a document view. Fig. 2 illustrates an example of the user interface for term-topic relationship. TM is presented as a matrix, where rows correspond to terms and columns to topics. The system has an advanced interactive graphic interface created with JavaScript library d3.js. The system has the following disadvantages: the program source code is not available for free use; programmers should be involved for development; topic change in the timeline is not displayed; it is difficult to assess topic proximity of the summarized clusters.

Paper [6] presents LDAExplore system for TM analysis and visualization. The system's main goal is to provide an interactive visualization environment for TM research. The interface allows filtering documents and words of the model. The main disadvantages of the system are the lack of temporal TM presentation and the complexity of its software implementation. Fig. 3 illustrates the LDAExplore system's interface.

In [7] the temporal topic model visualization technique is discussed. This Method a topic-based, interactive visual analysis tool, TIARA. Given a topic derived from a set of text documents, automatically splits it into a set of subtopics spanning over multiple linear, non-overlapping temporal intervals. In [8], [9] visualization for analysis of large text corpora is realized. One of them VarifocalReader in-depth visual analysis of text based on tf-idf Natural Language Processing. Another the hierarchical topics evolve presents interactive visual text analysis approach RoseRiver which allows users to progressively explore and analyze the complex evolution patterns of hierarchical topics. In paper [10] a visualization method for checking topic model is proposed, the humanistic interpretation of topics – rather than formal topic model evaluation. There are several systems and tools for

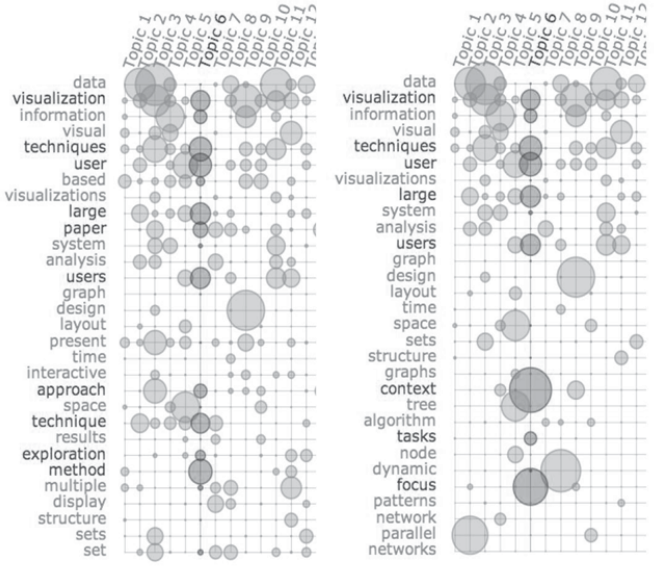

Fig. 2. The example of d3.js topic model visualization interface

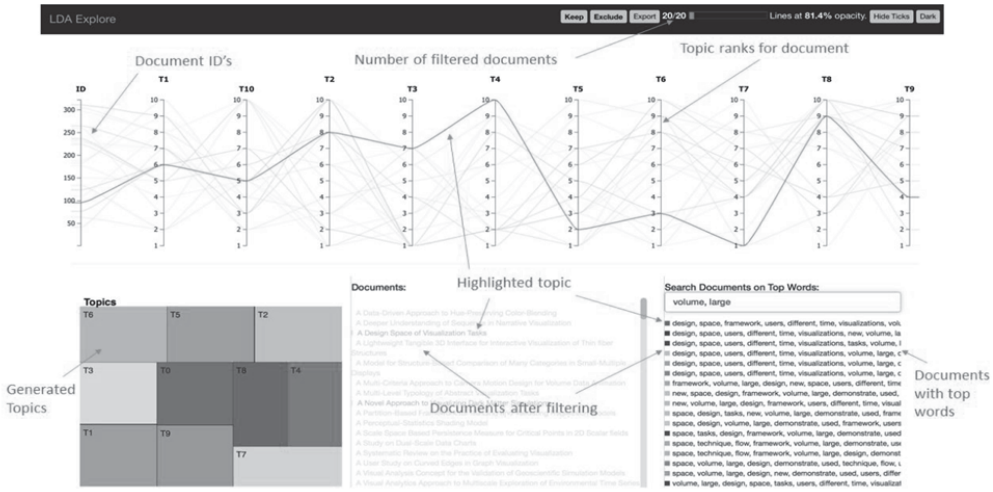

Fig. 3. The example of LDAExplore topic model visualization interface

visual analysis, but each of them has some disadvantages. That is why the following tasks are relevant: the visualization of TM summaries, and the development of a system, which would allow creating and rebuilding topic models, and then represent the intermediate and final results in a user-friendly way.

Table I presents comparison of existing approaches for topic model visualization based on requirements defined before.

TABLE I. APPROACHES TO TOPIC MODELS VISUALIZATION

Requirements	**Visualizing TM [4]**	**Termit [5]**	**LDAExplore [6]**	**TIARA [7]**	**VarifocalReader [8]**	**RoseRiver [9]**	**TM Checking [10]**
TM parameters control	-	-	-	+	-	-	-
Data source choice	-	-	-	+	-	-	-
System's modularity	-	-	-	-	-	-	-
Representation of significant for the theme words	+	+	+	+	+	+	+
Mapping the proximity of topics	-	+	+	-	-	+	+
Interactive visualization	+	+	+	+	-	+	+
Temporal TM visualization	-	-	-	+	-	+	-

III. THE APPROACH TO GRAPHICAL VISUALIZING OF TOPIC MODEL

TM induces relationship between topics and documents in text corpus. One of the most common TM is Latent Dirichlet allocation (LDA) [3], a generative probabilistic semantic indexing model. The other TM are usually LDA extensions. The result of topic modeling is multinomial probability distributions over terms generated by soft clustering of words based on document co-occurrence.

Let D be the set of text documents, and W be the list of terms. Every document $d \in D$ is a sequence of n_d terms $(w_1, w_2, \dots, w_{n_d})$ from the list W.

Considering the hypothesis of the conditional independence $p(w|d,c) = p(w|c)$ according to the formula of total probability let get a probabilistic model for the generation of the document d:

$$p(w|d) = \sum_{c \in C} p(w|d,c)p(d|c),$$
$$p(w|d) = \sum_{c \in C} p(w|c)p(c|d),$$
$$p(w|d) = \sum_{c \in C} \varphi_c \theta_d$$

The documents vectors $\theta_d = (p(c|d): c \in C)$ are generated by the same probabilistic distribution of normalized $|C|$ - dimensional vectors. It is convenient to take from a Dirichlet distributions $Dir(\theta, \alpha), \alpha \in R^{|C|}$. The topic vectors $\varphi_c = (p(w|c): w \in W)$ are generated by the same probabilistic distribution of normalized vectors of dimension $|W|$, this distribution is also convenient to take from a Dirichlet distributions $Dir(\theta, \beta), \beta \in R^{|W|}$. For computing φ_c– the topic vectors and θ_d– the documents vectors EM-algorithm is used.

According to probabilistic topic modeling, first proposed in the paper [2], a probabilistic model of the appearance for the pair "document-word" can be written in three equivalent ways:

$$p(d,w) = \sum_{c \in C} p(c)p(w|c)p(d,c),$$
$$p(d,w) = \sum_{c \in C} p(d)p(w|c)p(c,d),$$
$$p(d,w) = \sum_{c \in C} p(w)p(c|w)p(d|c),$$

where:

- C is the variety of topics;

- $p(c)$ is the prior distribution of topics in the collection.
- $p(d)$ is the prior distribution on a variety of documents, empirical evaluation $(d) = {}^{n_d}/_{n}$, where $n = \sum_d n_d$ is the total length of all documents, and n_d is the document length in words;
- $p(w)$ – is the prior distribution on a variety of words, empirical evaluation $(w) = {}^{n_w}/_{n}$, where n_w is the number of occurrences of the word w in all documents.

IV. Implementation

The following tools were selected for the implementation of the system:

- Python programming language,
- IPython interactive shell,
- Python Anaconda distribution kit,
- libraries: genism 0.13, pyldavis 2.0, matloblib 1.5, wordcloud 1.2.

Python is a high-level programming language used for general-purpose programming. *IPython*, a command shell for interactive computing, distributed with *Anaconda*, the Python distribution kit, allows to create an interactive interface for implementing and managing Python scripts.

Gensim library is used for preparing and creation of a topic model. The following libraries are involved for visualization of the TM results: *pyldavis, matloblib, wordcloud*.

Altogether, the selected tools and libraries provide the necessary flexibility and functionality to implement software package. The developed system provides to a specialist flexible configuration of parameters for the algorithms, and has convenient means of visualization and export of intermediate and final results of TM. The most important and often used setting for TM-LDA is the number of clusters to which a collection of documents has to be divided. This number is set using the num_topics parameter in *gensym* library. The number of clusters selected empirically for each collection of documents. Clusters should be separated as much as possible, it is better when the meaningful words are encountered in one or more topics, but not in too many topics. Fig. 4 illustrates the components of *IPython* TM visualization system. The data source for system implementation is text corpus SCTM-en. TM-LDA module creates topic models with gensim library. Calculating module carries out the estimation data for TM visualization in the time row. Visualization module is responsible for representing the results of topic modeling.

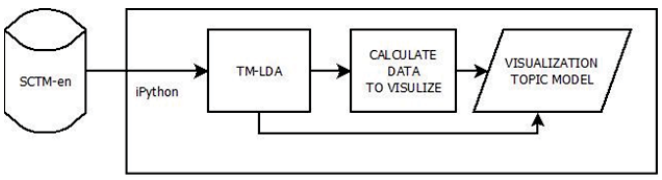

Fig. 4. The components of TM visualization system

V. Evaluation

For research, the text corpora SCTM-en has been created using the method described in [11]. The international news website "Russian Wikinews" (Wikinews) has been chosen as a data source, because texts are distributed through a free license of Creative Commons Attribution 2.5 Generic. Wikinews as well as other Wiki-based resources are websites of the second generation of Internet that are characterized by the fact that many ordinary users are involved to develop content of these sites, and those users help to expand them and update information. Large volume, continuous expansion, neutrality of opinions, and availability are viewed as the advantages of all wiki-resources, including Wikinews. According to the site statistics, English Wikinews contains more than 21 000 news. 20 100 of them contain information about the date of the described events. The body of SCTM-en takes news describing the events from 2005 to 2017. Wikinews texts in corpus are cleared of wiki markup and punctuation marks. Only words with more than three characters long are used for TM construction.

The standard TM output is a list of the most likely words for each topic with a numerical estimation of the probability, an example is shown in Fig. 5. This representation is not user-friendly. With *wordcloud* library the received topic is visualized in a more illustrative way (Fig. 6). Each word in the topic is compared by the probability; font size in the visualization corresponds to the word's significance in the topic: the higher is the probability, the greater is the font size of the displayed word. Fonts of words have different colors to provide better visibility. The method is suitable to evaluate of the intermediate and final results of the topic modeling.

Fig. 7 illustrates interactive user-friendly interface created with the library pyLDAvis. The topics are represented with the Venn diagram on the left. The representation allows to estimate how close the topics are: the close ones are situated closer, those in which the common words are less frequent are far apart from each other. A set of the most characteristic for the chosen theme words is on the right side. Such representation makes it possible not only to see the significance of the word to the chosen topic, but also reflects how often the word is found in the collection. The interface is interactive. A specialist can choose the topic that interests him, it is enough to click on the topic display circle, or use the filter. The chosen word or theme are highlighted. If you click the interesting word the topic display is updated. Topic in which the selected word significantly increase in size in proportion to its importance.

Representation allows the switch from topic to analysis of words. In practice, it is often necessary to deal with the analysis of the stream of text documents, such as the news stream. In this paper, a text stream is a sequence of text documents with a specific creation time of every document.

The topics representation on Fig. 7 also helps to understand the correctness of the chosen number of clusters in the constructed topic model. If topics on the Venn diagram are located in one area and intersect repeatedly, the constructed model does not accurately correspond the subject area of the analyzed text documents. It is necessary to change the number

of clusters and rebuild the model, so that the dimensions of topics were the same size, and that topics minimal overlap.

Fig. 6. The example of wordcloud TM visualization interface

The text streams processing refers to the complex task of clustering of incoming documents and analysis of their topic characteristics. Topic models, which analyze text streams, are called temporal topic models. To visualize the dynamics of topic change over time it is necessary to carry out a preliminary calculation of TM data (algorithm 1).

Algorithm 1 Preparing for temporal visualization

1: **For** $d \in D$:

$p(c), t(d)$, where $t(d)$ – the date of the described event

2: **For** c_d: – for all the topics in the document

$$p(c,t) = \sum_{d \in D} p(c|d)$$

3. the matrix topic-date is calculated, its value is the sum of the topic probabilities for the date.

4. the value of the sum of the probabilities at each date is normalized, the sum of all probabilities per day is equal to one.

There are several methods for the visualization of topic modeling results on the time scale. Fig. 8 illustrates the presentation with a simple one-line diagram realized with *matplotlib* library. Every line that reflects a topic is painted in a different color. The display allows to see rises of popularity of certain topics in the news stream. The diagram reflects that some popular topics had appeared in 2007 and 2010.

Documents probabilities sum are not normalized for topics model visualization. It allows to estimate the periods in which the number of the events described in the news from various topics is increasing and decreasing. Because of the vast amount of events and news occurring in the world every day, topic model presented in the Fig.8 shows the volatility of these events. A detailed study of the topic model allows to set rhythms of increasing and decreasing of the number of events.

```
%%time
lda = gensim.models.ldamodel.LdaModel(corpus, id2word=dictionary, num_topics=30, alpha='auto', eval_every=5)
```

```
Wall time: 2min 6s
```

```
for l in lda.show_topics(30, 5):
    print (l)
```

```
(0, '0.012*"steel" + 0.008*"slam" + 0.007*"bridge" + 0.007*"ship" + 0.006*"muslimmajority"')
(1, '0.026*"russian" + 0.021*"russia" + 0.019*"putin" + 0.015*"medal" + 0.014*"paralympics"')
(2, '0.020*"minister" + 0.011*"that" + 0.011*"said" + 0.009*"leader" + 0.009*"government"')
(3, '0.034*"election" + 0.028*"party" + 0.020*"vote" + 0.011*"votes" + 0.011*"electors"')
(4, '0.015*"that" + 0.012*"from" + 0.011*"space" + 0.009*"with" + 0.006*"launch"')
(5, '0.017*"rugby" + 0.007*"translated" + 0.007*"with" + 0.007*"from" + 0.006*"munich"')
(6, '0.032*"that" + 0.012*"from" + 0.011*"with" + 0.008*"said" + 0.007*"states"')
(7, '0.022*"that" + 0.013*"will" + 0.011*"party" + 0.011*"with" + 0.010*"said"')
(8, '0.039*"states" + 0.025*"united" + 0.025*"obama" + 0.025*"president" + 0.018*"presidential"')
(9, '0.022*"image" + 0.013*"eggs" + 0.011*"polish" + 0.009*"school" + 0.007*"that"')
(10, '0.026*"fisher" + 0.014*"flynn" + 0.009*"with" + 0.009*"that" + 0.009*"will"')
(11, '0.018*"football" + 0.014*"club" + 0.013*"league" + 0.011*"soccer" + 0.010*"bayern"')
(12, '0.035*"korea" + 0.031*"match" + 0.028*"cricket" + 0.018*"jong" + 0.016*"coast"')
(13, '0.027*"prison" + 0.013*"murder" + 0.011*"angeles" + 0.008*"inmates" + 0.008*"america"')
(14, '0.042*"dies" + 0.014*"categoryobituaries" + 0.009*"with" + 0.008*"michael" + 0.008*"church"')
(15, '0.012*"states" + 0.007*"santiago" + 0.006*"chile" + 0.006*"illinois" + 0.006*"america"')
(16, '0.018*"with" + 0.009*"music" + 0.009*"from" + 0.008*"categoryculture" + 0.008*"that"')
(17, '0.019*"with" + 0.014*"team" + 0.013*"first" + 0.011*"world" + 0.010*"after"')
(18, '0.021*"that" + 0.016*"they" + 0.015*"with" + 0.012*"have" + 0.012*"what"')
(19, '0.014*"wikinewsie" + 0.013*"from" + 0.012*"categoryiain" + 0.012*"macdonald" + 0.012*"were"')
(20, '0.021*"that" + 0.011*"with" + 0.011*"from" + 0.008*"trumps" + 0.007*"this"')
```

Fig. 5. The example of TM visualization interfaces

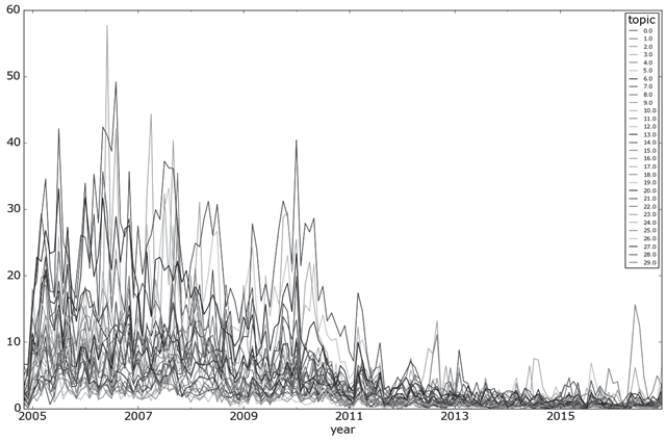

Fig. 8. The presentation of topic modeling results over 2005-2017

The second method to present the TM results over time is an area chart that is realized with *matplotlib* library. The result is shown on Fig. 9. Each topic has a different color in the chart. The area graph displays all topics. Their value on the ordinate scale is the sum of the topic probabilities at the specific date. The diagram reflects that some popular topics had appeared in 2005 and 2008. There was a noticeable reduction in the number of news since 2010. This representation makes it possible to track changes in the popularity of each topic over time and see the overall growth and reduction in the number of documents. As the previous method, this view allows to evaluate periods of increasing and decreasing of the number of events for all topics.

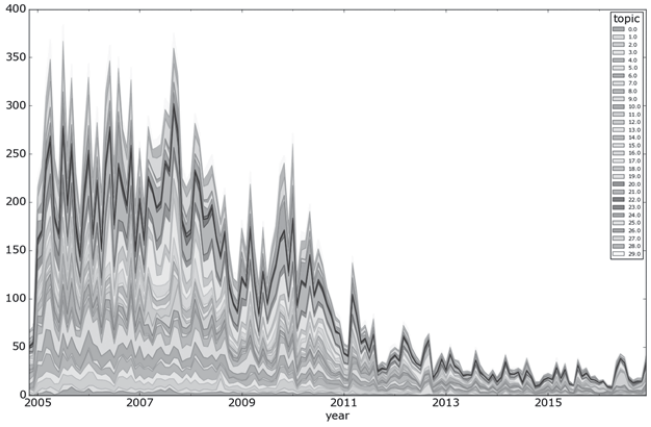

Fig. 9. The unnormalized presentation of topic modeling over 2005-2017

Fig. 10 illustrates the third method of TM visualization over time. The area chart is based on the normalized data. Each topic has a different color like in the previous chart. The presentation allows to track the growth of popularity or the formation of a new topic and the decline of popularity for another one. Noticeable are the changes of some topics in 2014-2016. It is difficult to track which topic has changed notably on the time scale 2005-2017. The system has everything to make a cross-section of the data over time taking the range from 2014 to the end of 2016, and to visualize it. The result is shown in Fig. 11. The representation is similar to the TIARA method used in [7] and [9], but much easier to implement. The presentation features combine an intuitive graph of themes changing during the time interval, standard library with rich display settings, and simplicity of implementation.

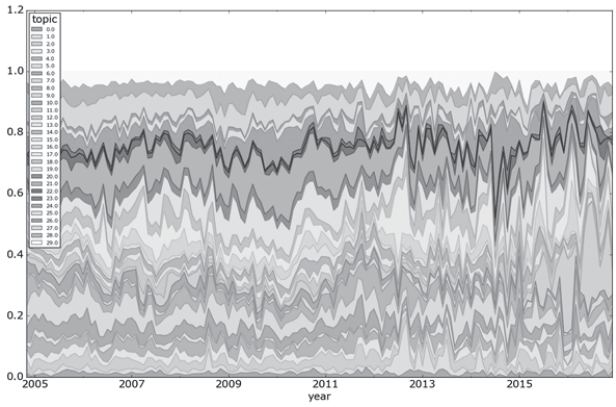

Fig. 10. The normalized presentation of topic modeling over 2005-2017

Representation allows to estimate the quality of the constructed topic model. If the diagram contains the set of topics with similar behavior during the same time interval, but at each interval they are insignificant in comparison to other topics, then it should be tried to change the number of topics in topic model constructing and then redraw the diagram.

The fourth method is implemented by means of histograms. The result is shown on Fig. 12. This representation is convenient for tracking changes in the popularity of topics on a specific date. The chart shows clearly that the topic 1 (colored green) rose sharply in June 2016, and then its popularity began to decline.

To improve the presentation an optimum number of topics should be selected for the topic model construction. That allows correctly correspond to all topics changes during the time, not too small changes on a time interval and not too large to be noticed as insignificant topic. Such selection allows to make the right conclusion about the characteristics of the analyzed text corpus.

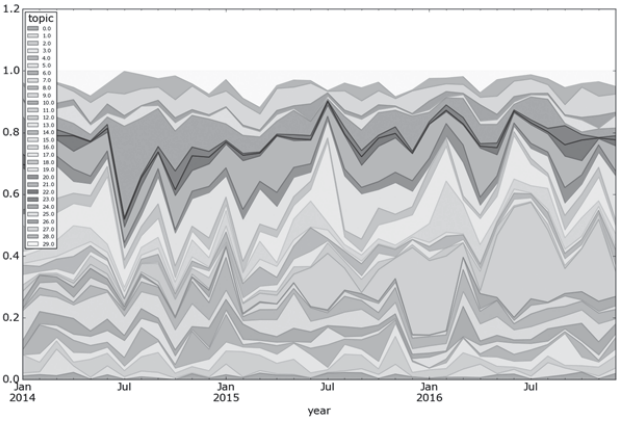

Fig. 11. The normalized presentation of topic modeling over time for the period 2014-2016

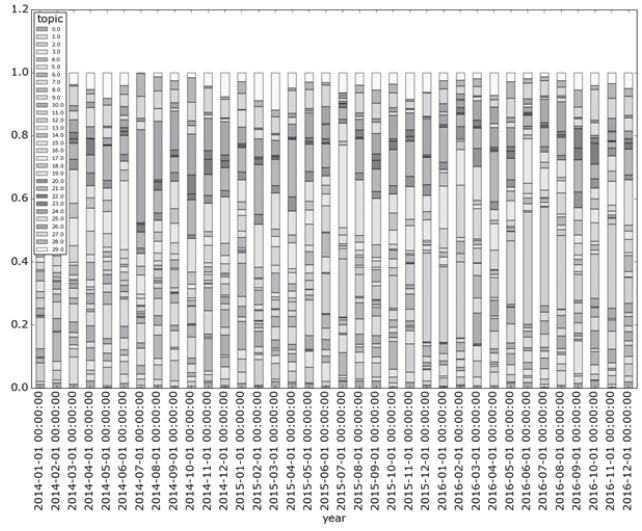

Fig. 12. The normalized presentation of topic modeling over time

VI. Conclusion

The paper presents the system for topic model visualization. It allows to select an appropriate way of visualization, and provides user-friendly presentation of topic modeling results. The most important features of the system are flexibility to configure and manage the creation of TM process, the choice of data source, and wide range of possible result visualizations due to the used libraries. A specialist that will work with the system can analyze the result, return to the TM settings, change them and rebuild the model to get new results. It is important to have a user-friendly diagram for analyzing the changes of topic popularity over time. Such diagram allows to get an idea about the features of the corpus for a specialist in the text mining. An average user can understand the meaning of the data presented on the diagram as well. The system source code is available on GitHub under free non-commercial GIT license [12]

The future work in topic model visualization is going to be concentrated on the improvement of the natural language processing quality.

Acknowledgment

This work has been supported by projects funded by grants 17-07-00327 and 17-07-00328 of the Russian Foundation for Basic Research. The work has been partially financially supported by state research № 0073-2014-0005 and by Government of Russian Federation, Grant 074-U01.

References

[1] K. Vorontsov, A. Potapenko, A. Plavin, Additive regularization of topic models for topic selection and sparse factorization, *International Symposium on Statistical Learning and Data Sciences. – Springer International Publishing,* 2015, pp. 193-202.

[2] T. Hoffman, Probabilistic Latent Semantic Indexing, *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval,* 1999, pp. 50-57.

[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet Allocation, *Journal of Machine Learning Research*, 2003, pp. 993-1022.

[4] A.J.B. Chaney, D.M. Blei, Visualization Topic Models, *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media — ICWSM,* 2012, pp. 419-422.

[5] J. Chuang, C.D. Manning, J. Heer, Termite: Visualization techniques for assessing textual topic models, *Proceedings of the International Working Conference on Advanced Visual Interfaces. – ACM*, 2012, pp. 74-77.

[6] A. Ganesan, LDAExplore: Visualizing Topic Models Generated Using Latent Dirichlet Allocation, *arXiv preprint* arXiv:1507.06593. 2015.

[7] S. Pan, M.X. Zhou, Y. Song, W. Qian, F. Wang, S. Liu, Optimizing temporal topic segmentation for intelligent text visualization, *Proceedings of the 2013 international conference on Intelligent user interfaces.* ACM, 2013, pp. 339-350.

[8] S. Koch, M. John, M. Worner, A. Muller, T. Ertl, VarifocalReader—In-Depth Visual Analysis of Large Text Documents, *IEEE transactions on visualization and computer graphics*. 2014. Vol. 20, issue 12, pp. 1723-1732.

[9] W. Cui, S. Lui, Z. Wu, H. Wei, How hierarchical topics evolve in large text corpora, *IEEE transactions on visualization and computer graphics.* 2014. Vol. 20, issue 12. pp. 2281-2290.

[10] J. Murdock, C. Allen, Visualization Techniques for Topic Model Checking, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015, pp. 4284-4285.

[11] S.N. Karpovich, The Russian language text corpus for testing algorithms of topic model. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2015. vol. 39. pp 123–142. (In Russ.).

[12] Source code. Web: https://github.com/cimsweb/Topic-Model-Visualization-With-IPython