**LDAAnalyzer: A Tool for Exploring Topic Models**

Chunyao Zou, Daqing Hou  
Department of Electrical and Computer Engineering  
Clarkson University  
Potsdam, New York 13699  
zouc, dhou@clarkson.edu

**Abstract**—Online technical forums are valuable sources for mining useful software engineering information. LDA (Latent Dirichlet Allocation) is an unsupervised machine learning method which can be used for extracting underlying topics out of such large forums. However, the main output of LDA forum learning are usually huge matrices that contain millions of numbers, which is impossible for researchers to directly scrutinize the numerical distribution and semantically evaluate the relationship between the extracted topics and large collection of unorganized documents. In this paper, we present LDAAnalyzer, an LDA visualization tool that makes the hidden topic-document structures rise to the surface. From the functionality point of view, LDAAnalyzer consists of (1) LDA modeling, (2) LDA output analysis and (3) new corpus training. With the help of LDAAnalyzer, our semantic topic-modeling evaluation based on large technical forums becomes feasible.

**Keywords**—LDA; topic modeling; visualization; forum

**I. INTRODUCTION**

Nowadays online technical forum discussions contain considerable useful information that can be used to facilitate software development activities. Many researchers in this area have used Latent Dirichlet Allocation (LDA) as their analysis tool because of its impressive modeling power [7, 8, 9, 10]. However, little research has been focused on answering the question as to how well topics generated by LDA reflect the actual discussions in technical forums from a semantic point of view. Previous research has been paying attention to constructing new topic models, using purely mathematical approaches to evaluate modeling output. Validation of topic quality has mainly relied on checking topic words and their probabilities, rather than the consistency between topic words and meaning of the underlying documents, thus bypassing an important requirement that LDA should be semantic oriented. Mathematically good results do not necessarily imply good modeling from semantic point of view.

LDA is a high level statistical tool. The raw output of the model is usually large matrices that contain millions of numbers that are impossible for researchers to scrutinize directly. To overcome this barrier, we’ve developed an automated tool named LDAAnalyzer to translate and visualize the numerical distributions into human-friendly tables and graphs. Using the tool, researchers can then conveniently explore the otherwise hidden relationships between topics and documents, topics and words, and examine unstructured corpus in an organized manner. LDAAnalyzer consists of three main components (1) LDA output analysis, (2) new corpus training and (3) forum user behavior analysis. With the help of LDAAnalyzer, our semantic topic-modeling evaluation based on large technical forums has become feasible.

**II. RELATED WORK**

Chaney and Blei have created a similar tool for visualizing LDA output [1]. The most novel features of their tool include: (1) user friendly GUI for visualizing Theta table and Phi table that are the two main output of LDA; (2) convenient navigation among documents, topics and words; (3) links among similar documents and similar topics. Since their tool is designed for both researchers and non-technical users, it hides those numerical details and similarity algorithms. Unlike ours, it does not provide any interface for configuring and customizing the model visualization.

Our tool LDAAnalyzer is designed for software engineering researchers. Not only does it contain all their tool’s functionalities, but also enrich the numerical data visualization and enhance the user interaction design. In addition, to support our semantic research, we have added the following novel functions to LDAAnalyzer as well: (1) the visualization of training new documents based on previous topic models, and (2) forum user behavior mining.

Within Software Engineering, more often LDA has been used to model source code [4, 11-13]. For example, CodeTopics [4] shows the similarity between source codes being developed and related high-level artifacts (HLAs). CodeTopics is designed to helps developers to improve the understanding of source code identifiers. CodeTopics has the UI to present topic distribution and source code similarity. It is a good application of topic modeling in the area of development assistance and program comprehension.

Topic Browser developed by Gardner et al. [2] provides various functions and views for visualizing modeling output, however, only a limited number of documents and words related to a topic can be displayed. TopicNets [3], implemented by Newman et al., visualizes the topics and their relationships, but provides little support for analyzing corresponding documents and topic words.

There are other tools on visualization of large document corpus, such as FacetAtlas [5] and Exemplar-based Visualization [6]. They are both for understanding the entire corpus from a top-level viewpoint. They do not have the functionalities to zoom into individual documents and depict the relationship between topics and documents.
III. LDAA NALYZER IN ACTION

Figure 1 depicts the main user interface of LDAA NALYZER, where its functionalities are divided into the three tabs on top: “LDA Result Analysis”, “New Document Training” and “Forum Mining”. This section describes these functionalities, its architecture and implementation, and our experience of using it to conduct semantic evaluation of generated topics.

A. Loading LDA modeling result into LDAA NALYZER

The main purpose for the “LDA Result Analysis” component is to visualize the relationships between documents and topics, and between topics and words. It also allows for the navigation from topics to documents and from topics to words, enabling us to further explore the hidden semantic ties that LDA discovers. The starting point of this component is the “LDA Result Analysis” tab in Figure 1.

Since there may exist multiple topic models produced as a result of running different configurations of LDA parameters, we first have to load a model from one specific LDA run via the dropdown box shown in Figure 1. Notice that in Figure 1 the result of an LDA run is named using the modeling parameters. After selecting a model and clicking “Load LDA Run” button, an overview of all the topics (with 20 keywords) is shown in the table in Figure 1.

Topic 0009 highlighted in Figure 1 is regarding discussions about uploading files to servlet, which we will use as a running example:

“0009 - file, content, stream, response, read, upload, filenam, length, getoutputstream, am, clos, wrt, write, attach, setcontenttyp, save, flush, ioexcept, disposit, sethead”

B. Visualizing Theta matrix (from topics to documents)

Recall that the theta table is a matrix where a row represents a document and a column represents a topic. Each value in the matrix is the probability that the current document (row) is about a topic (column). Since it is undesirable to display all the documents in the corpus for analyzing a topic, as shown in Figure 2, under tab “Theta Table,” we provide three approaches for filtering the documents: by “High Probability Docs,” “Percentile,” and “Mean Deviation.”

We select “High Probability Docs” in the following procedure. For each document (row), we mark the topic (column) that has the highest probability value. All the marked documents under a topic (column) then become its high probability documents. “Percentile,” as its name suggests, returns the documents that fall in the first percentile in the descending order of topic probabilities. Lastly, given a delta value from a user, “Mean Deviation” calculates a probability threshold using the following formula:

\[
\text{mean probability} + \delta \times \text{standard deviation}
\]

All documents with a probability higher than the threshold are displayed. The second column of the table displays the amount of documents loaded for the corresponding topic.

When we double click topic 0009 in Figure 2, it will show
all of the topic’s documents in descending order of probability values (Figure 3). Double clicking a document can trigger the loading of the document and the document content will be shown in the right side text area.

C. Visualizing Phi matrix (from topics to words)

To explore the probability distribution of keywords for a selected topic, click “Phi Table” in Figure 2. This will lead to Figure 4. Double clicking a topic number in the left pane triggers the loading of topic words in descending order of word probabilities. The right-hand side graphical view displays the words probabilities for the selected topic. Three kinds of charts (bar chart, column chart, pie chart) can be chosen to observe the words distribution.

D. Training new documents

Based on an existing topic models, give a new document, we can easily find its topic probabilities in the topic model. We prepared a thread in forum which was regarding file upload/download errors discussion. As shown in Figure 6, the new document was input into the left text area, after clicking the lower left button, the document’s topics were listed in probability descending order in the right table. There we see topic 0009 was ranked as the top topic. Double clicking the topic loads all the topic’s high-probability documents. In this way, we can also find other documents in corpus that are similar to the new document.
E. Exploring forum user behavior

This component collects and analyzes participants’ information from every thread. Figure 7 depicts all the users in the forum with their amounts of posts they ask and answer, respectively. Some interesting conclusions can be drawn from the data. For example, from the data we can find who are technical experts: The less than 1% of users who answer questions more than 100 times but seldom ask questions. For another example, we find that novices ask a few questions but seldom answer questions.

F. Implementation and Study

LDAAnalyzer was developed as a web application. As shown in Figure 8, in its frontend, we use apache Flex, a highly productive, open source application framework for building and maintaining expressive web applications. In the backend, we adopt the Spring and Hibernate integration solution, which offers system flexibility and extensibility.

In our semantic evaluation research, we used LDAAnalyzer on a corpus extracted from servlet forum in JavaRanch. We collected a total of 111,132 posts from 24,798 threads in the Servlet forum, from April 3, 2000, when the forum was founded, to July 25, 2013. With the assistance of this tool, we have made the following novel contributions: (1) categorizing all the topics found by LDA from semantic viewpoint; (2) proposing six kinds of semantic relationship between topics and documents; (3) finding three approaches to improve the quality of LDA output. More details can be found in our forthcoming paper.

IV. CONCLUSION

We have presented LDAAnalyzer, a tool that visualizes the output of topic modeling. It has three components (1) LDA output analysis, (2) new corpus training, and (3) forum user behavior analysis. It reveals the relationships between topic-document and topic-word, and those across documents. Its design has been motivated by supporting the whole process of our topic modeling semantic evaluation, but it should also be useful to other researchers who use LDA. The screencast and code for LDAAnalyzer can be found at: http://serl.clarkson.edu:8080/ldaanalyzer/res/download.html. Given that the LDA output is only one sample for the underlying distribution [8], a useful feature to add in future would be the comparative visualization of multiple samples.

REFERENCES