Latent Semantics Approach for Network Log Analysis: Modeling and its application

Kazuki Otomo
University of Tokyo
Tokyo, Japan
otomo@hongo.wide.ad.jp

Satoru Kobayashi
National Institute of Informatics
Tokyo, Japan
sat@nii.ac.jp

Kensuke Fukuda
National Institute of Informatics
Tokyo, Japan
kensuke@nii.ac.jp

Hiroshi Esaki
University of Tokyo
Tokyo, Japan
hiroshi@wide.ad.jp

***Abstract*—Network log analysis helps network operators to troubleshoot their network. Many mathematical analysis methods rely on a set of time series corresponding to log type (log template) per device, as their input. However, they do not take full advantage of the meaning of logs despite log messages containing semantic information written in a free format. In this paper, we emphasize the use of this rich semantic information for network log analysis. More specifically, we propose an unsupervised latent semantics-based network log analysis. The key idea of the work is to build a latent semantics model of unobservable network functionalities (e.g., routing protocols, hardware) from generated logs, instead of inferring what is happening in the network from a network operator's knowledge with log messages. This approach enables us to numerically cluster/compare the meaning of logs because each log message obtains a numerical distributed representation (a topic distribution). We discuss the validity of our approach with a set of logs collected at a nation-wide academic network for a year. We first show that our approach outperforms a popular data-driven approach (i.e., word2vec), which does not require any assumptions on the data, by evaluating the quality of the distributed representation of logs. Furthermore, through two network analysis scenarios, we demonstrate several benefits of our approach: intuitive interpretability of analysis results and bridging the gap between multi-vendors log messages.**

I. Introduction

Network log analysis is a useful method to understand a network system's behavior. One of the characteristics of a network log is that, unlike other data, it is text data. By reading logs, network operators can understand what happened on their network, when it happened, and which devices are involved. In operational networks, syslog [1] is a widely used protocol for collecting network logs. With these logs, operators can investigate detailed statuses and events for each device in their network system. A difficult issue of log analysis is that the amount and diversity of logs is too much to manually inspect because network systems are growing rapidly. This diversity also comes from multiple vendors and software deployed in the network. Thus, automatic log analysis methods are highly required to handle large scale and a wide variety of logs.

Many log analysis methods for finding anomalies [2]–[5] and their root causes [6]–[9] have been proposed to achieve automatic log analysis. There is one analytical problem with network log analysis; we need to convert raw log messages to numerical representation for statistical / mathematical analyses. Many log analysis methods solve this problem by taking following common approach: classifying logs by their message type (i.e., log template) and then treating each classified group as time-series data by using timestamps written in the logs Then, the log analysis turns to a domain of multi-dimension time-series analyses; i.e., we can apply sophisticated statistical methods to the log analysis.

978-3-903176-32-4 © 2021 IFIP

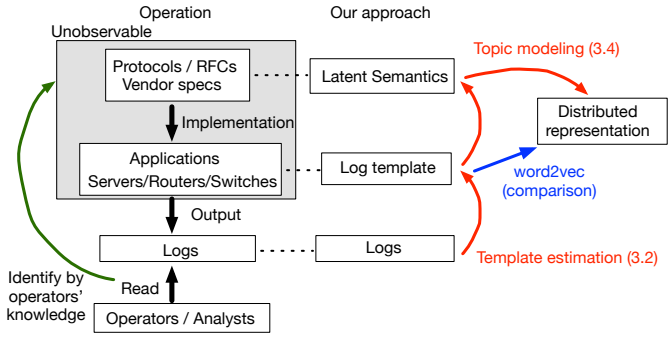

Figure 1. Log semantics approach overview

While log messages are meaningful text data, the current approach does not consider their content information (i.e., semantics), which is well studied in natural language processing (NLP). Thus, network operators/analysts are required to estimate or infer what the meanings of logs and their root causes are (Figure 1). This limitation prevents us to analyze logs using such rich information in the following two examples. (1) Input log time series are independently treated in matrix operations. Thus, several semantic relationships of logs (e.g., BGP related messages) are not captured within analysis. (2) Post analyses such as interpretation and root-cause analysis are not easy by automatic ways. For example, automatically finding related information or the root cause is not easy because the current approach can not semantically connect external text data (e.g., trouble tickets, documents, etc.) to log analysis results. Moreover, operators need to determine what the anomaly detected in the time series means without sufficient readability of the analysis result.

To overcome this problem, we propose an unsupervised latent semantics based approach for network log analysis. The key idea of the approach is to consider a model in which all the network log messages are generated by unobservable

and abstracted functionalities corresponding to network protocols and the details of software/hardware implementations of network devices (Figure 1). We build the model with a topic-based modeling in NLP and clustering techniques (section III) from generated logs. In this type of modeling, each log message has a statistical topic distribution consisting of network functionalities (e.g., BGP, OSPF, link, hardware), so two log messages characterized by more similar topic distributions share closer background context. Thus, we can numerically compare and/or aggregate log messages by their meanings of the logs. Through this validation, we can confirm that our topic modeling outperforms a commonly-used data-driven approach (i.e., word2vec [10]), which does not require any assumption on input data, in clustering log templates based on semantic information (section IV). Furthermore, we demonstrate the power of semantic information in two log analysis applications (section V): assigning suitable trouble tickets to log time series (subsection V-A) and bridging domain knowledge between two different vendor logs (subsection V-B).

Our contributions are as follow:

- We propose a latent topic modeling to introduce the semantics to log messages in network log analysis.
- Our validations show that our approach performs better than a data driven approach (e.g., word2vec).
- We demonstrate the effectiveness of our approach through two typical log analysis scenarios.

II. Semantics approach in network log analysis

In this section, we first describe general issues in existing log analysis. Then, we introduce a semantics approach in network log analysis. Finally we explain how our approach solves the general issues and expand the log analysis.

A. Background and problem statement

Automatic log analysis requires numerical representations of logs in order to apply mathematical / statistical methods to the log messages because logs contain unstructured data (i.e., set of texts). Existing approaches often classify log messages by using a log template [2], [3], [6], [8], [11]. The log template is a pattern of log message representation and there are many log template estimation or extraction methods [12]–[14]. Existing works just use log templates as a log classification criterion, and they build log time series for each group of logs in the same template and conduct multivariate time-series analyses. However, these existing approaches do not take full advantage of the meanings of logs, and the log messages are just separated by the format of logs.

Thus, the lack of semantics causes two fundamental problems in the current log analysis. First, it does not consider any semantic relationships among input time series (we refer to it as an "input issue"). This issue causes impractical analysis results because of the inability to extract sufficient information from logs. For example, without semantic information, we can not use relationships among logs and thus, we need to process each log time series independently. Second, analysis results are hard to interpret for operators because the results only focus on numerical behavior even though log messages indicate contextual information (we refer to it as an "output issue"). This issue forces the operators to go back in the raw log messages in order to understand what happened on their network system. In practice, categorizing or summarizing analysis results by their meanings (e.g., labeling issues as routing problem, hardware broken, etc.) is helpful information for troubleshooting or root-cause identification. However, the existing log analysis does not fully support this due to a lack of meanings in logs. These two issues create a gap between mathematical log analysis and practical operations.

These issues have not been well emphasized in log analysis for the application layer (e.g., HDFS(Hadoop Distributed File System) [2], OpenStack [8]). In application layer logs, the occurrence conditions and meanings of the logs are obvious for each log template because all logs were designed using the same software. Thus, the meanings of logs are clear for each log template, and the classification by log templates is sufficient to understand the meaning of logs. On the other hand, in network logs, each log is defined by different software and protocols because network system is multi-layered and distributed. Therefore, meaningful classification of log templates is very helpful for analysis.

Figure 2 shows examples of log templates in the application layer (OpenStack, shown in [2]) and in network. As shown in the figure, logs in OpenStack have a common prefix ("`instance: *`"). In addition, each log template represents the life cycle or a manipulation of the instance. In contrast, network logs in the figure have different topics such as NTP, SSH and Routing. Moreover, the first three logs have different formats despite having the common context of NTP. Since there are many different types of logs in a network, operators rely on abstracted classification (e.g., NTP, SSH, Routing in the figure). It is also difficult to know such abstracted classification of logs when only relying on the format of logs. One naive idea is using keywords (e.g., `NTP`, `BGP`, `SSH`') which specifically explain background applications or protocols. However, definitions of keywords are different among services and devices in target networks. Thus, such keywords must be automatically extracted.

B. Semantics approach

We explain the semantics approach in network log analysis for these issues. By adopting the semantics approach, the meaning of logs are statistically introduced in log analysis.

To overcome the two general issues (input and output), one might consider to directly use textual information of logs (e.g., word matching, edit distance of logs, TF-IDF). However, in particular in network logs, these approaches are not suitable because there are wide variety of literal differences among logs in spite of implying the same semantic content. For example, a log ntpdate: NTPDATE_TIME_CHANGED: step time offset and xntpd: precision = 10 usec do not share any words while both of them mention the same protocol "NTP". Instead, the key idea of our approach is bridging text data and numerical data by using the background context of

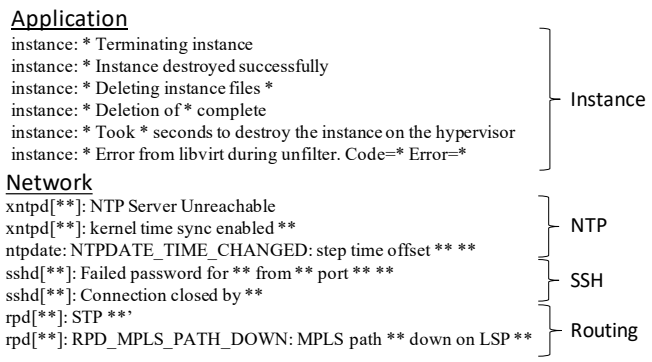

Figure 2. Examples of log template in application and network

logs. We consider the background context of logs as protocols, RFCs, or vendor requirements of the devices. We attempt to capture these unobservable background contexts and represent semantics of log data by using these background context. The root cause of these issues is the inability to convert between numerical data (analytical) and text data (readable). Thus, the solution is to appropriately convert raw log data (readable) to numerical data (analytical) without a lack of semantic information of logs and assign readable labels to the analysis results on the basis of numerical semantic information.

In this study, we refer to the background context of logs as *latent semantics*. We define latent semantics as latent contextual information that all the log messages have. As shown in Figure 1, we assume log messages are outputs of unobservable network functionalities. In network system, these functionalities are implemented along with the network common specification such as protocols, RFCs, etc. Thus, we model on the basis of that log messages are generated by latent context, which is the source origin of logs. For example, because logs related to BGP tend to include the words BGP, AS, notification, and message, operators are able to know that a log has the background context of BGP from such key words despite including many other words in a log.

A similar assumption is often used in NLP for document clustering. Topic modeling [15], a popular method in NLP, estimates latent topics from a group of documents. Each document is represented by a topic distribution and it statistically shows how much the document is related to each topic.

By applying the same idea to log analysis, we extract topics of logs (i.e., protocols or services) and each log can be represented by a topic distribution. Topic distributions enable us to analyze semantic relationships among logs. Thus, the input issue is solved by our semantics approach. In addition, the semantics approach enables us to connect numerical analysis results and external interpretable data (e.g., trouble tickets, documents etc.) through topic distributions; the output issue is also solved by our semantics approach.

As a further benefit, our semantics approach bridges different network environments at the layer of latent semantics. Currently, transferring examined knowledge in a specific network cannot be applied to different networks due to the large literal differences of logs. We can consider that the latent semantics is a high level representation of log data. The analysis results, or learnt models based on the latent semantics can be applied into other network environments (e.g., a network consisting of difference vendor devices) because the latent semantics represents common knowledge in network (see subsection V-B for more details).

C. Related Work

Many works have been devoted to statistical / mathematical log analyses [2], [6], [8], [11], [16]–[21].

A basic log analysis applies statistical methods to a group of logs classified by its log templates. Xu et al. [11] proposed a straightforward log anomaly method; log template estimation, manual feature creation, PCA based anomaly detection, and template-based workflow visualization. They focus on application layer logs, and many works use a similar flow of log analysis. As mentioned in subsection II-A, this approach does not consider the meaning of logs due to it being a simple log template-based analysis. Thus, this approach basically does not solve the input issue. A major difference in application logs compared with network layer logs is that the meaning of log templates is clearer and more distinguishable, and the analysis results in log templates are sufficient to understand for operators to understand. This means the output issue is not critical in application log analysis.

However, in network logs, showing analysis results by log templates is not practical because each log appears with completely different format. Kimura et al. [16] focused on anomaly prediction in network logs. They also took the same analysis framework; log template estimation, feature creation, and machine learning based proactive feature detection. However, due to a lack of semantics information in logs, they could not solve the two general issues in subsection II-A. In particular, the "output issue" seriously affects its operation because they notify detected anomalies to operators of anomalies detected by fault alarms without contextual information.

To solve these issues, a number of existing studies already focus on the meanings of logs. Meng et al. [19] proposed $template2vec$ based on a data-driven approach that does not require any assumptions on input data. They used a vectorized representation of a log template to detect anomalies. They confirmed that their template2vec approach achieved a higher precision compared with other traditional methods. These works showed that using semantic information improved the performance of log analysis in the application layer (i.e., solving the input issue in application layer). However, as shown by the blue line in Figure 1, the NLP methods (e.g., word2vec) they used do not model the generative process of logs, but obtain semantic information from the data trend (we refer to this approach as a "data-driven approach"). In addition, as shown in Table I, these studies focus on HDFS or web service logs, instead of network logs.

In this work, we model the latent semantics behind a generative process of network logs. Thus, we expect that it is more intuitive to obtain a distributed representation

Table I
RELATED WORKS

	Target system	Use semantics	Input issue	Output issue
Xu et al. [11]	application	no	no	no
Kimura et al. [16]	network	no	no	no
Aussel et al. [17], Guofu et al. [18]	application	model based	yes	no
Meng et al. [19], Zhang et al. [20]	application	data driven	yes	template base visualization
Our approach	**network**	model based	**yes**	connecting external data

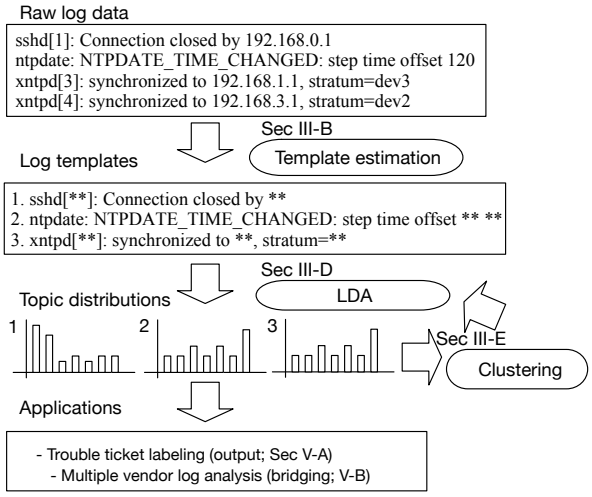

Figure 3. Methodology overview

with network log-specific assumptions (i.e., topic modeling) than to obtain a distributed representation from a data-driven approach (word2vec). In terms of the output issue, existing works attempted to visualize the analysis results by using log data itself. Additionally, in our work, our semantics approach focuses on solving the output issue by connecting the external data with logs through semantics.

To evaluate the ability of the latent semantics modeling, we compare the topic modeling based distributed representation and word2vec based one in the evaluation section.

III. SEMANTICS EXTRACTION FROM NETWORK LOGS

A. Methodology overview

In this section, we describe an overview of our semantics extraction method from network logs. Our purpose is to obtain a distributed representation of raw network log messages on the basis of their latent semantics.

Our method consists of the following steps (also shown in figure Figure 3): template estimation, pre-processing, topic modeling, and parameter tuning with log template clustering. The template estimation contributes to obtain fine-grained topic estimation. In topic modeling, we estimate topics from log templates by latent Dirichlet allocation (LDA). Then, we conduct log templates clustering and feedback clustering results in order to tune LDA's parameters.

After the modeling, each log template has a topic distribution (a vector value) and we can use a latent semantics-based representation of logs for analysis.

B. Template estimation

Log template estimation is a well-known problem in log mining [12], [13]. A log template is a word occurrence pattern of a log message, and represents the structure of log messages, variables, and meaningful words (descriptions). Figure 3 shows an example; variables are replaced with wildcards (shown as "⋆"). To use a log template as an input of topic modeling is effective due to the following: we can omit variable words from topic estimation, and grouping log messages based on their templates mitigates the biased appearance of log messages. In this work, we adopt a supervised learning approach proposed by Kobayashi et al. [7]. This algorithm is based on a conditional random field (CRF) [22], which is well-studied in NLP, and generates log templates composed of description words and variable words from raw log messages.

C. Preprocessing to log templates

Before applying topic modeling, we additionally preprocess estimated log templates in order to normalize the log template representation. Normalized words contribute to obtain appropriate latent semantics. In this work, we adopt both typical preprocessing and converting domain specific words to normal words by using a manually generated dictionary (e.g., make 'addr' to 'address'). By using the above preprocessing, "alarmd[**]: connection succeeded after ** retries" is converted to a list of words "alarmd connection succeed after ** retry".

D. Topic modeling to log analysis

We use preprocessed log templates as the input of topic modeling. In this work, we selected a well-known Latent Dirichlet Allocation (LDA) [15] for the topic modeling method. LDA is a generative probabilistic model of a corpus (a set of documents), and assumes that each word in a document is subjected to a word distribution conditioned by a topic. It also learns the probability of a topic's occurrence in a corpus and the probability of a word's occurrence for each topic. Using these probabilities, LDA can assign a plausible topic to each word and obtains a topic distribution for each document.

Next, an inference process is conducted so that the probability of a document occurrence is maximized. There are several implementations of the document probability maximization. In this work, we adopt collapsed Gibbs sampling [23] for ease of implementation. LDA estimates a topic distribution and word

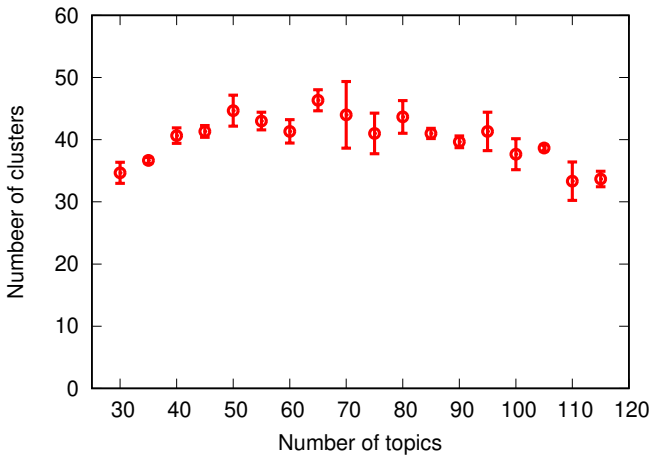

Figure 4. Dependency of number of topics on clustering result

distributions for each topic in input documents. To apply LDA to log templates, we treat one template as one document and a group of templates as a group of documents. The input of LDA is bag-of-words (BoW) representation of log templates. Note that variable words (i.e., wildcards) in log templates are ignored. The output of LDA is a topic distribution for each log template. Topic distribution has a fixed dimensions specified by a pre-defined parameter of LDA. After LDA estimation, each log template is represented by a vector of the number of topic dimensions. We refer to the output vector as a *topic distribution*. We consider that have a similar topic distribution share the same latent semantics estimated by LDA. Now, each log template is numerically comparable in terms of semantics. LDA also outputs topic representative words on the basis of estimated word distribution for each topic. Topic representative words are useful for adjusting parameters of LDA.

E. Topic distributions clustering and parameter tuning of LDA

We show a log template clustering method on topic distributions for parameter tuning of LDA results. We observe the peak of topic distributions is unstable due to short size of input documents (i.e., log templates) while the shape of topic distributions has a stable trend for log templates. Hence, we use a similarity of topic distributions, not a peak of the distribution for clustering criteria of log templates.

Next, we discuss the parameter tuning to determine the number of topics. We empirically observe that the number of DBSCAN clusters formed by topic distributions has an upper limit regardless of the number of topics (Figure 4). In this study, we take the number of that produces the largest number of clusters in order to capture the most fine-grained unit of semantics from log templates. In general, the observation of clustering results is useful to decide the number of topics or controlling the granularity of semantics to be extracted.

IV. Evaluation

In this section, we evaluate how the abstraction level of estimated topics are similar to our expectations, which are the latent semantics of network logs such as protocols, RFCs, etc. We apply the topic modeling to log messages gathered from a real network system. We evaluate the results of the topic estimation by comparing them with manual labeling. We also compare clustering results with topic modeling and word2vec.

Table II
SINET4 DATASET

logs	devices	templates	term
34.7M	130	1789	456 days

A. Dataset

We use log messages collected from SINET4 [24]. SINET4 is a nation wide academic network in Japan that connects more than 800 universities and research institutes. The network consists of eight core routers and more than 100 layer 2 switches. A summary of the dataset is shown in Table II. We manually annotate protocol level labels to estimated log templates as the ground truth.

B. Log template clustering based on topic distribution

We evaluate obtained topic distributions by comparing them with manual labeling on log templates.

We extract 1,789 templates from the dataset with the template estimation method. Then, we apply topic modeling to preprocessed log templates as shown in section III. We set the number of topics to 60 throughout our experiment on the basis of the parameter tuning criteria described in subsection III-E. Then, we adopt DBSCAN [25] with the Manhattan distance to obtain clusters of log templates as mentioned in subsection III-E. DBSCAN has two parameters; the minimum number of cluster candidates and the maximum distance (ε) between two samples in the same cluster. We empirically fix the minimum number of clusters to five and adjust ε to maximize the number of clusters. DBSCAN outputs outliers that are not included in any clusters under specified parameters. We re-assign outliers to the most similar cluster by comparing topic distributions between an outlier and the mean topic distributions of clusters.

Finally, we obtain 43 clusters from the SINET4 dataset. We expect that one cluster must be a group of templates sharing the same semantics. We evaluate the validity of the topic-based clustering by the adjusted rand index [26], [27] and compare it with that of the manual annotated labels.

1) Comparison method: word2vec: Here, we explain a data-driven approach (i.e., word2vec) to compare with our topic-based approach. In the view of document clustering or numerical document representation, word2vec is a popular method. Recent works on log analysis also adopt word2vec as a distributed representation of log templates [19], [20]. Word2vec estimates distributed representation of words from a group of documents. In network logs, we can apply word2vec to logs by regarding a log template as a document. Every log template is numerically represented by the mean distribution of word vectors in a log template. This vector representation

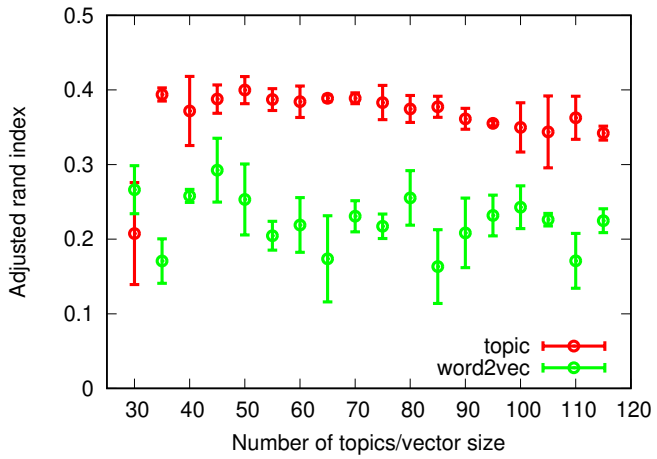

Figure 5. ARI for topic distribution and word2vec based clusterings

is also measurable (e.g., BGP templates are closer to OSPF than NTP). We apply word2vec to the dataset and then apply DBSCAN clustering. Finally, we evaluate the adjusted rand index to be the same as that in topic modeling evaluation.

2) Results: First, we show that our topic-based method outperforms word2vec in terms of the goodness of clusters (i.e., adjusted rand index). The adjusted rand index is a popular metrics for clustering. The rand index is the score to determine each pair in a dataset with the same category is placed in the same cluster or not. The adjusted rand index solves the problem with the rand index where it shows a non-zero score even for random classification. Higher values of the adjusted rand index score in the range [0, 1] indicates the generated labels are more similar to manual labeling. As shown in Figure 5, our proposed method achieves a higher score than word2vec (0.36 and 0.22 on average, respectively).

Overall, our semantics approach outperforms the data-driven approach. A plausible reason for the performance difference is due to the nature of log data: a shorter length of documents (template) and a smaller number of documents, compared to usual data-driven NLP tasks. While this difficulty is also true for LDA, the latent semantics is fits well to network system logs because of its explicit model.

C. Examples of estimated topics

By investigating clustering results, we found several large clusters: the largest cluster is related to network interfaces (169 log templates), the second largest one is MPLS (85 templates), and the third one is BGP (51 templates).

Table III shows examples of the estimated topics. Clusters A, B, and C correctly capture routing protocol specific words (e.g., 'lsp' and 'mpls' for MPLS, 'bgp' and 'session' for BGP, and 'state' and 'neighbor' for OSPF). Cluster D captures router's hardware topic (e.g., fpc, initialize, and pfe). Note that these topics are estimated by the unsupervised method (i.e., without a priori knowledge). These results indicate that the latent semantics estimated from log messages correspond to the similar abstraction level of protocols, as we expected.

Table III
EXAMPLES OF LOG TEMPLATE CLUSTERS

ID	Representative words
A	lsp, mpls, jflow, flag, peer, backup, secondary, family, primary, connect
B	rpd, bgp, session, use, task, bandwidth, jtask, valid, reinitialize, show
C	state, neighbor, ospf, realm, mpcs, v2, idx, program, vpls, smem
D	fpc, initialise, pfe, cmt, eachip, asic, share, logger, fetch, initrd

V. APPLICATIONS

Here, we introduce two log analysis applications using the topic model to show the effectiveness of our approach.

A. Network event labeling by trouble tickets

In the first application, we apply the topic distribution of log templates to external data (i.e., trouble tickets) and use them as a knowledge base inspired by LogCluster [28]. By using topic distribution, the numerical representation of external data is also available if those data can be related to log messages. We demonstrate that labeling logs with external data such as trouble tickets gives more readable and useful analysis results to network operators.

1) Baseline: LogCluster and its simplified implementation: There are a number of works to label results of applications analysis [28], [29]. LogCluster first learns patterns of a log sequence related to trouble tickets on a target system. Then, it attempts to report past sequences of logs similar to those of the target trouble ticket. Note that log sequences are chunks of log messages split by any rules (e.g., time bin or same event). It is useful to identify problems and refer to past recovery processes in a similar incident. To calculate similarity among sequences of logs, LogCluster partly uses an IDF-based vector representation of sequence of logs. Here, we apply the same approach in network logs as a baseline method. In network logs, we assume a sequence of log templates as a document and each template as words. Then, IDF is calculated by $IDF(t) = \log(\frac{N}{n_t})$, where t is an index of log templates, n_t is the number of sequences including log template, N is the total number of log sequences. Now, we construct a log sequence vector with the number of log templates: $v(i) = IDF(t_i)$, where v is a vector of the log sequence, i is an index of the vector, and t_i is a log template with index i.

2) Our approach: We use topic distributions as a vector representation of sequences of logs instead of IDF. Each log message has a topic distribution given by its log template. We use a mean distribution of a sequence of logs as a sequence vector. Then, we evaluate the IDF-based vector approach (i.e., data driven) and topic distribution-based vector approach (i.e,. model based). Note that LogCluster focuses on application systems (e.g., HDFS). Thus, they define a sequence of logs as log messages that have the same task ID that appears in every log message. In a network system, the task ID is not available in logs. Here, we define a sequence of logs as a group of log messages related to a trouble ticket.

3) Experiment setup: SINET4 has over 200 trouble tickets in a period. We manually merge related logs for each trouble ticket. Each trouble ticket has related logs and each logs has a topic distribution. Thus, the topic distribution for trouble tickets is calculated by using the mean distribution of each related logs. We refer to the mean topic distribution in a trouble ticket as "trouble ticket topic distribution".

4) Trouble ticket assignment: Next, we implement a simple trouble ticket assignment method and compare the performance of both approaches. This method assigns suitable trouble tickets into relevant time steps in a log time series. Thus, operators can understand what happens at the time suggested by reviewing a previously reported readable trouble ticket.

We assign trouble tickets to a given time series as follows. First, we randomly select a few number of trouble tickets as a reference data. We chose corresponding log messages for each ticket in advance. Then, we obtain the topic distribution of the trouble ticket by calculating the mean topic distribution of the log templates belonging to the trouble ticket. Second, we split the testing log time series into one-hour intervals and convert them into the mean topic distribution. We refer to a mean distribution of one-hour interval log time series as a "time series topic distribution". Note that a time series topic distribution has the same dimensions as a trouble ticket topic distribution. Third, we calculate a cosine similarity between all the time series and reference trouble tickets by the topic distributions. Once the similarity is above a pre-defined threshold, the reference ticket is assigned at the time step. We expect that when an event similar to a trouble ticket occurs at the time, a similar type of trouble ticket is assigned.

We split SINET4 log data into a training term (2012/1/1 - 2012/9/30) and testing term (2012/10/1 - 2012/3/31). We evaluate the ticket assignment accuracy for different sizes of the reference data. The accuracy of the ticket assignment is determined by whether the same category of ticket is assigned or not at the time. For example, when our method assigns a BGP-related trouble ticket at time X, it is correct if a BGP-related ticket is actually reported at the time. The category of a trouble ticket is also manually annotated in advance.

Figure 6 shows the results of the trouble ticket assignment. The X axis shows the number of referenced tickets and the Y axis shows recalls. We calculate the recall by: $\frac{\#\text{Correct assignment date}}{\#\text{Date with trouble tickets}}$. As shown in the figure, the topic-based method shows a higher recall than the IDF-based method for the different number of the trouble tickets. From this result, we confirm that the topic-based approach (model based) outperforms the IDF-based one (data driven) especially for smaller sizes of reference data. This means that the model based approach can capture more information than the data driven approach from the same data.

5) Case studies: Finally, we show an example of semantics working effectively for the trouble ticket assignment.

Figure 7 shows the results of the IDF-based and topic vector-based methods on a day when a trouble ticket about BGP was reported. The IDF-based method did not assign any tickets, whereas the topic-based method correctly assigned

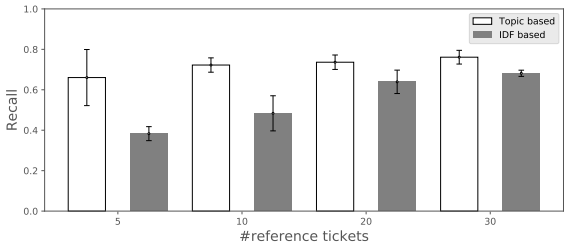

Figure 6. Recalls of trouble ticket assignment: topic distribution based and IDF based

Log templates in a reference ticket

A-1: rpd[**]: bgp_hold_timeout:**: NOTIFICATION sent to ** (** AS **): code ** (Hold Timer Expired Error), Reason: holdtime expired for ** (** AS **), socket buffer sndcc: ** rcvcc: ** TCP state: **, snd_una: ** snd_nxt: ** snd_wnd: ** rcv_nxt: ** rcv_adv: **, hold timer

A-2: ** rpd[**]: RPD_BGP_NEIGHBOR_STATE_CHANGED: BGP peer ** (** AS **) changed state from ** to ** (event **)

Log templates in testing data

B-1: rpd[**]: RPD_BGP_NEIGHBOR_STATE_CHANGED: BGP peer ** (** AS **) changed state from ** to ** (event **)

B-2: rpd[**]: bgp_process_caps: mismatch NLRI with ** (** AS **): peer: <** **>(**) us: <**>(**)

B-3: rpd[**]: bgp_read_v4_message:**: NOTIFICATION received from ** (** AS **): code ** (Cease) subcode ** (Maximum Number of Prefixes Reached) AFI: ** SAFI: ** prefix limit **

Figure 7. Example of trouble ticket assignment; Reference and testing data share one log template (blue)

BGP tickets. At this time, three types of logs related to BGP occurred. However, only one ticket ("A-2" in the reference ticket and "B-1" in the testing data) related to the BGP is included in the training data. Since the IDF-based method only considers a type of the log template, the similarity between the log time series and trouble ticket depends on whether the appeared log template is identical or not. Thus, in Figure 7, only one type of log templates contributes to the similarity because the testing time series includes only one type of log template, which also appeared in the training data. On the other hand, the topic-based method is able to infer semantics of log templates even if the exactly same log template does not appear in the training data. In this case, the topic-based method successfully captures the same semantics (i.e., BGP) both in the reference ticket and testing log data.

Therefore, the topic-based analysis can extract more information from a smaller dataset thanks to the semantics.

B. Multiple vendor log analysis

In this section, we show that our semantics approach can bridge a gap between multiple vendor's logs.

1) Domain bridging by semantics: The latent semantics information enables us to compare the similarity on the basis of the semantic content, regardless of the literal differences. Here, we consider that log messages are generated from two different vendors (Juniper and VyOS [30]). For instance, a log template of VyOS "ospf_nexthop_calculation(): could not determine nexthop for link" has the same semantics as a log template of JUNOS "**:rpd[**]: RPD_OSPF_NBRDOWN: OSPF neighbor ** (realm ** ** area **) state changed from ** to **" in terms of the OSPF protocol from our domain knowledge. As we can see, the literal differences are clearly non-negligible

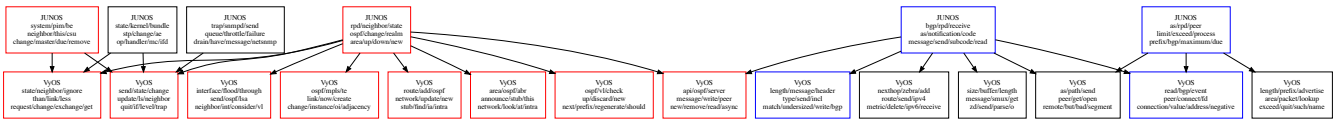

Figure 8. Graph structure of topics between JUNOS and VyOS: rectangles show topics with representative words, and edges connect bridged topics. Red for "OSPF" and blue for "BGP" (manually labeled)

Table IV
TOPIC ESTIMATION RESULTS IN JUNOS AND VYOS

OS	templates	topics	#linked topics
JUNOS	1,789	60	14
VyOS	2,109	66	20

to considering that they both refer to the same protocol (OSPF) by an unsupervised manner. However, network systems often consist of multiple vendor devices in practice. Thus, handling multiple vendor's logs (i.e., largely different format of logs) is an important and practical issue for network log analysis.

Here, we demonstrate that our semantics approach is useful in analyzing multiple vendor's logs. We apply LDA to several log datasets gathered from different OSes and software. Topics from LDA are expressed by representative words. We consider that both topics show the same latent semantics if the representative words of topics are matched more than a certain number of words. Since topic representative words are normalized in preprocessing, and the words related to topic (i.e., latent semantics of logs) have exactly the same words. Thus, the topics with the same latent semantics are linked to each other by word matching.

2) Experiment setup: We conduct an experiment using logs obtained from JUNOS and open source router software VyOS [30]. We extract log templates from VyOS's source code and obtained 2,109 templates. We apply LDA to both JUNOS and VyOS templates and estimate topics, respectively. Then, we link topics that commonly appeared in JUNOS and VyOS. Note that we also preprocess VyOS's log templates as shown in subsection III-C. We first connect topics between JUNOS and VyOS if two topics share two ore more words. Next, we drop nodes that have less than three edges. Finally, we link topics that have two or more shared words with the remaining nodes. Table IV show the summary of the data.

3) Results: Figure 8 shows an example of the graph structure of routing-related logs. The top rectangles in the graph correspond to JUNOS, and the bottoms are VyOS. Red and blue colors indicate the routing protocols (OSPF and BGP, respectively). We confirm that related topics are well connected in the two different systems. Through this experiment, we do not manually provide neither key words "BGP" and "OSPF" nor related words "AS" and "neighbor". All the connections are obtained by an unsupervised manner. This result indicates that key words of logs are same and correctly captured by LDA regardless of data source domain.

Focusing on OSPF and BGP topics, one topic in JUNOS connects to multiple VyOS topics. The templates in VyOS are covered all possibilities of log templates since log templates are generated from the source code analysis. On the other hand, the estimated topics in JUNOS are biased by practical network event occurrences because the log templates in JUNOS are estimated from raw log messages. Thus, the topics of VyOS are more detailed and covered than those of JUNOS.

One future direction based on this experiment is to apply mined knowledge to other network environments. In general, analysis results in a particular network cannot be applied to other environments due to large differences of network systems. If an upgrade in the same network system occurs, we need to re-learn the models because the format of the log templates might also be updated. However, our results show that the latent semantics approach captures abstracted representations of logs. We confirm that the representations can be shared in different environments even by a simple word matching. Thus, this approach is promising for transferring knowledge in multi-domain data. We confirm that latent semantics bridges the gap between multiple domain logs. It is a large benefit to handle data from different sources without being affected by its literal differences.

VI. CONCLUSION

In this work, we emphasized the use of semantics information in network log analysis. We first addressed the general issues of existing log analysis; not using semantic information (input issue) and interpretability of analysis results (output issue). In order to solve these two issues, we proposed a semantics extraction approach and describe its practical applications. We modeled unobservable network functionalities (e.g., protocols, RFCs) that generate human-readable network logs by latent semantics. With a log dataset collected at a nation-wide academic network, we confirmed that the topic modeling approach is superior than a data driven approach such as word2vec in the network log analysis. Furthermore, we demonstrated the effectiveness of the proposed approach through two log analysis scenarios: (1) improving readability of trouble tickets (for the output issue), and (2) bridging multiple vendor logs with the latent semantics. As a further perspective, we will improve our topic modeling to use more sophisticated methods [31], [32], in addition to integrating other external data (e.g., RFC).

VII. ACKNOWLEDGEMENT

This work is supported by JSPS KAKENHI Grant Number JP19K20262, and the MIC/SCOPE #191603009.

REFERENCES

[1] R. Gerhards, "The Syslog Protocol," Internet Requests for Comments, RFC 5424, 2009. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5424.txt

[2] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. ACM, 2017, pp. 1285–1298.

[3] M. Astekin, S. Özcan, and H. Sözer, "Incremental analysis of large-scale system logs for anomaly detection," in *Proceedings of the International Conference on Big Data (Big Data)*, 2019, pp. 2119–2127.

[4] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California fault lines: Understanding the causes and impact of network failures," in *Proceedings of the ACM SIGCOMM Conference*, ser. SIGCOMM'10, 2010, pp. 315–326.

[5] S. He, Q. Lin, J.-G. Lou, H. Zhang, M. R. Lyu, and D. Zhang, "Identifying impactful service system problems via log analysis," in *Proceedings of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2018, 2018, pp. 60–70.

[6] S. Lu, B. Rao, X. Wei, B. Tak, L. Wang, and L. Wang, "Log-based Abnormal Task Detection and Root Cause Analysis for Spark," in *Proceedings of the International Conference on Web Services*, ser. ICWS. IEEE, 2017, pp. 389–396.

[7] S. Kobayashi, K. Otomo, K. Fukuda, and H. Esaki, "Mining causality of network events in log data," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 53–67, 2018.

[8] B. C. Tak, S. Tao, L. Yang, C. Zhu, and Y. Ruan, "LOGAN: Problem diagnosis in the cloud using log-based reference models," in *Proceedings of the International Conference on Cloud Engineering*, ser. IC2E. IEEE, 2016, pp. 62–67.

[9] C. Scott, A. Wundsam, B. Raghavan, A. Panda, A. Or, J. Lai, E. Huang, Z. Liu, A. El-Hassany, S. Whitlock, H. Acharya, K. Zarifis, and S. Shenker, "Troubleshooting blackbox sdn control software with minimal causal sequences," in *Proceedings of the ACM Conference on SIGCOMM*, ser. SIGCOMM'14. Association for Computing Machinery, 2014, pp. 395–406.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in neural information processing systems*, ser. NIPS, 2013, pp. 3111–3119.

[11] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting Large-Scale System Problems by Mining Console Logs," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, ser. SOSP'09, 2009, pp. 117–132.

[12] M. Mizutani, "Incremental mining of system log format," in *Proceedings of the International Conference on Services Computing*, ser. SCC'13. IEEE, 2013, pp. 595–602.

[13] R. Vaarandi, "A data clustering algorithm for mining patterns from event logs," in *Proceedings of the 3rd IEEE Workshop on IP Operations Management*, ser. IPOM'03, 2003, pp. 119–126.

[14] P. He, J. Zhu, Z. Zheng, and M. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *Proceedings of the International Conference on Web Services*, ser. ICWS. IEEE, 06 2017, pp. 33–40.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The journal of machine learning research*, vol. 3, pp. 993–1022, Mar. 2003.

[16] T. Kimura, A. Watanabe, T. Toyono, and K. Ishibashi, "Proactive failure detection learning generation patterns of large-scale network logs," in *Proceedings of the 11th International Conference on Network and Service Management*, ser. CNSM, 2015, pp. 8–14.

[17] N. Aussel, Y. Petetin, and S. Chabridon, "Improving performances of log mining for anomaly prediction through NLP-based log parsing," in *Proceedings of the 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, Sep. 2018, pp. 237–243.

[18] G. Li, P. Zhu, and Z. Chen, "Accelerating system log processing by semi-supervised learning: A technical report," 2018.

[19] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, ser. IJCAI-19, 7 2019, pp. 4739–4745.

[20] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," in *Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2019, 2019, pp. 807–817.

[21] S. Zhang, Y. Liu, W. Meng, Z. Luo, J. Bu, S. Yang, P. Liang, D. Pei, J. Xu, Y. Zhang, Y. Chen, H. Dong, X. Qu, and L. Song, "Prefix: Switch failure prediction in datacenter networks," in *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, Apr. 2018, pp. 1–29.

[22] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML'01, 2001, pp. 282–289.

[23] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD'08, 2008, pp. 569–577.

[24] S. Urushidani, M. Aoki, K. Fukuda, S. Abe, M. Nakamura, M. Koibuchi, Y. Ji, and S. Yamada, "Highly available network design and resource management of SINET4," *Telecommunication Systems*, vol. 56, no. 1, pp. 33–47, 2014.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

[26] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[27] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, ser. ICANN'09. Springer, 2009, pp. 175–184.

[28] Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, and X. Chen, "Log clustering based problem identification for online service systems," in *Proceedings of the 38th International Conference on Software Engineering Companion (ICSE-C)*. IEEE/ACM, 2016, pp. 102–111.

[29] W. Zhou, L. Tang, C. Zeng, T. Li, L. Shwartz, and G. Ya. Grabarnik, "Resolution recommendation for event tickets in service management," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 954–967, 2016.

[30] "VyOS open source router and firewall platform," https://www.vyos.io/.

[31] Y. Hu, J. Boyd-Graber, and B. Satinoff, "Interactive topic modeling," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1. Association for Computational Linguistics, Jun. 2011, pp. 248–257.

[32] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'09. Association for Computational Linguistics, 2009, pp. 248–256.