Summarizing Industrial Log Data with Latent Dirichlet Allocation

Shunmuga Prabhu Siddharthan, Marcel Dix, Barbara Sprick and Benjamin Klöpper

Abstract Industrial systems and equipment produce large log files recording their activities and possible problems. This data is often used for troubleshooting and root cause analysis, but using the raw log data is poorly suited for direct human analysis. Existing approaches based on data mining and machine learning focus on troubleshooting and root cause analysis. However, if a good summary of industrial log files was available, the files could be used to monitor equipment and industrial processes and act more proactively on problems. This contribution shows how a topic modeling approach based on Latent Dirichlet Allocation (LDA) helps to understand, organize and summarize industrial log

Shunmuga Prabhu Siddharthan
ABB Automation GmbH
Hänchener Str. 14, 03050 Cottbus, Germany
🖂 shunmuga.siddharthan@de.abb.com

Marcel Dix · Benjamin Klöpper
ABB Corporate Research Center
Wallstadter Str. 59, 68199 Ladenburg, Germany
🖂 marcel.dix@de.abb.com
🖂 benjamin.kloepper@de.abb.com

Barbara Sprick
SRH Hochschule Heidelberg
Ludwig-Guttmann-Str. 6, 69123 Heidelberg, Germany
🖂 barbara.sprick@srh.de

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 6, No. 1, 2020

DOI: 10.5445/KSP/1000098011/14
ISSN 2363-9881

files. The approach was tested on a real-world industrial dataset and evaluated quantitatively by direct annotation.

1 Introduction

Industrial automation devices and systems such as robots, drives, machine tools or distributed control systems produce potentially large log files. These log files contain information on the actions the devices carried out and on possible issues and problems encountered during the execution of the actions. This information makes the log files an important source for troubleshooting, root cause analysis or performance optimizations. Today, human experts usually start their analysis from the raw log files. However, raw log files are an ill fit for direct human analysis for various reasons: There are usually far too many events logged; there is a large portion of very common events carrying no relevant information (like program start/stop); events are ordered by time and not causality; log entries are heavily customized to accommodate specific applications.

Data mining and machine learning are promising approaches to help human experts in analyzing log files or to use the contained information in different ways, e.g. to monitor equipment and production processes to detect anomalies, to try to predict certain events, or to support root cause analysis. A problem less investigated in the state of the art is the summarization of (potentially large) log files, to gain meaningful insights into the logs.

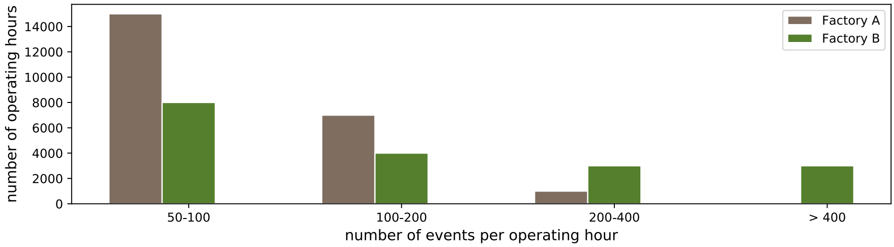

Figure 1: Different event frequency in one-hour-bins from two factories.

Figure 1 is a real example, taken from two European factories, showing that industrial devices can produce a lot of events per hour. In this case, the devices are industrial robots in those factories. For privacy-reasons, we changed the names of the factories to A and B. Our log files analyzed for factory A contained a total of 200,000 hours of robot operation across more than 150 robots in the factory. For factory B our logs contained a total of 300,000 operation hours across more than 250 robots. Both log files were collected over a period of 12 months. For most operating hours there are less than 100 events recorded per hour (70 % for factory A and even 90 % for factory B). What is shown in Figure 1 is that there are still many operating hours, where a very large number of events are being logged per hour. For example, in case of factory A, there were over 14,000 robot operating hours having 50–100 events per hour, and in case of factory B, more than 8,000 robot operating hours where 50–100 robot events were recorded per hour. The very different pattern numbers in the two factories (robots in factory B tend to produce more events than robots in factory A) shows that the content of robot event logs depends on the configuration and programming of the robots. Given hundreds of robots in a single factory and the large number of events per robot, this is a challenging cognitive task. Reducing the data points to be observed by a human while monitoring the robots or performing a root cause analysis without significant loss of information would be of great benefit.

This contribution discusses the application of topic modelling with Latent Dirichlet Allocation (LDA) with the objective to support human experts in using log files in monitoring large fleets of industrial assets. The aim of monitoring assets is to identify those assets that require special attention, e.g. by maintenance or tuning. Specifically, we introduce a process for topic modeling on industrial log files.

2 Related Work

2.1 Data Mining and Machine Learning on Industrial Log Data

Manual analysis of industrial log files is a cumbersome task and not surprisingly there exists prior work in applying data mining or machine learning to support this analysis. Mörchen (2007) introduces several temporal data models of which two are relevant when modeling industrial event logs for data mining or machine learning: The symbolic time sequence and the item-set sequence. A symbolic time sequence has nominal values that are observed at certain points in time and several values can occur at the same time. To model the robot log as a symbolic time sequence, the entries need to be transformed into nominal values with the help of some preprocessing strategy. An item-set sequence is a time sequence where each point in time is assigned to an item-set. Each item-set is a subset of the entire set of nominal values. The symbolic time sequence can be considered as a special case of an item-set sequence with item-set size of one.

A lot of prior work exists on mining log files based on statistical properties. For instance, Zöller et al. (2017) analyze the correlation between events and especially the distribution of the time lag between events with the objective to predict failure events based on prior events. Laxman et al. (2009) use frequent episode discovery algorithms to find sequences that contain common failure types and thus help in the root-cause analysis.

Frequent pattern mining algorithms are often used to analyze industrial log data. Folmer and Vogel-Heuser (2012) use frequent pattern mining to find frequently co-occuring alarms to reduce the number of alarms in alarm floods. An alarm flood is a situation in process plants where a high number of alarms are presented to the operator at once and result in an information overload. Similarly, Hadžiosmanović et al. (2012) and Czora et al. (2017) apply the frequent pattern mining algorithm FP-growth to perform data mining on industrial log files. Fullen et al. (2017) introduce a different approach based on item-set sequence modeling and compare several distance measures including the Jaccard distance, the Levenshtein distance and the TF-IDF, which are commonly used in text mining problems, to measure the similarity of alarm floods. For all three measures, the authors use the number of alarm types in the floods as basis for the distance calculation. Applications of machine learning to industrial log files is less common. Abele et al. (2013) use Bayesian networks refined by machine learning to support the diagnostic activities of operators in industrial plants. Atzmueller et al. (2017) use engineering data, in particular piping and instrumentation diagrams, to learn a compact graphical model with transition probabilities to detect anomalies in process plants.

None of the prior approaches addresses the topic of effective summarizing, which would be required to support the task of monitoring fleets of industrial assets.

2.2 Summarization on Log-like Data

Chandola and Kumar (2007) worked on summarization of network data which is structurally similar to industrial log data. They introduce three different techniques to summarize network transaction data: Cluster-based summarization and two iterative improvement algorithms (top-down-summarization and bottom-up-summarization), using a score based on summary size and information loss to improve an existing summary. The approaches are based on comparing network transactions based on categorical attributes only (numerical attributes are transformed by equal-binning). The application to robotics event data is difficult, because compared to network data the individual events (transactions) usually share very few common attributes. Robot events (and industrial events in general) have only few attributes that can be compared across different events and the concept of information loss introduced by Chandola and Kumar (2007) cannot be applied to robot and many other types of industrial event data. Wang et al. (2010) proposed a Hidden Markov model (HMM) to summarize events with short description length and high interpretability. However, the implementation of HMM demands data finger prints, internal states of the system and domain knowledge. Usually, this information is not readily available for data collected in industrial processes and thus this approach would require considerable human labeling effort.

3 Topic Modelling with Latent Dirichlet Allocation

Topic modeling is a machine learning technique which helps to analyze large collections of unclassified text. Topic models (also referred to as probabilistic topic models) use algorithms which help to identify thematic information from large archives of documents (Blei, 2012). The topics delivered by these algorithms emerge from the analysis of the unlabeled documents of original texts. This enables organizing, understanding and summarizing large collections of documents of textual information that would be impossible by a human to annotate.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data, which Blei et al. (2003) successfully used to generate topic models. Following the notation of Blei et al. (2003), LDA assumes a fixed number K of *topics* in a document collection with D *documents*. *Words* are the basic unit of the discrete data and indexed by $\{1, ..., V\}$ and documents are sequences of N words denoted by $d = (w_1, w_2, ..., w_N)$ and a corpus is a collection of M documents $D = \{d_1, d_2, ..., d_M\}$. The basic idea of LDA is to represent documents as random mixtures over latent topics, where each topic β is characterized by a distribution over the words. LDA assumes that a document collection is drawn as follows (Blei, 2012):

1. K Topics are drawn from a symmetric Dirichlet distribution $\overrightarrow{\beta}_k \sim Dir_v(\eta)$.
2. For each document d topic proportions are drawn from a symmetric Dirichlet distribution $\theta_d \sim Dir_k(\overrightarrow{\alpha}_d), d \in \{1, ..., D\}$.
3. For each word w in each document d
   - a topic assignment is drawn from the topic proportions
   $$(z_{d,n}|\theta_d) \sim Multinomial(\theta_d), \tag{1}$$
   - a word is drown from the corresponding topic
   $$(w_{d,n}|z_{d,n}, \beta_{1:K}) \sim Multinomial(\beta_{z_{d,n}}) . \tag{2}$$

Here $\overrightarrow{\alpha}_d$ is the prior information about the topic mixtures for a document and $\overrightarrow{\beta}_k$ is the prior information about the word distribution in topic k. A full description on estimating the prior and the parameters of the Multinomial distribution can be found in Heinrich (2008) and Wallach et al. (2009). Software packages like Gensim (Řehůřek and Sojka, 2010) are able to chose these parameters automatically. It has to be noted that the only observable variable in the model is the distribution of words of documents while topics, topic proportions, and topic assignment are latent. LDA analyzes a corpus performing a Baysesian inference to compute the following posterior distribution:

$$p(\beta_{1:K}, \theta_{1:D}, Z_{1:D,1:N}|w_{1:D,1:N}) . \tag{3}$$

Collapsed Gibbs sampling as described by Porteous et al. (2008) is an efficient inference procedure with a time complexity in each iteration of O(nK). Sontag and Roy (2011) showed that exact inference – the process of finding the mixture over the topic $\beta_{1:K}$ for a given document with a word sequence $d = (w_1, w_2, ..., w_N)$ is polynomial for small number of topics and NP-hard for a large number of topics.

4 Topic Modelling Applied to Industrial Log Files

Table 1 shows an anonymized nine-second excerpt from an event log of an industrial robot. Each row corresponds to a specific event that was considered worth recording in the log. Some important characteristics of the data can be observed. First, each row has a timestamp, but the recording of events is not equidistant. The timestamps provide an order over the events. However, the order is not unambiguous, since several events share the same timestamp. Timestamps only capture a point in time and the events carry no notion of duration. Second, the tabular structure is composed by a combination of a fixed time-format and categorical columns (e.g. code, category) on the one hand and free text on the other. Regarding the free text, the text is highly standardized. For instance, the two lines with code 71439 share exactly the same title and very similar descriptions.

Table 1: Example except from a typical robot event log.

Time	**Code**	**Category**	**Severity**	**Title**	**Description**
03:25:12	71439	IO	Warning	Motors On rejected	Motors On, via System IO, was rejected
03:25:16	71439	IO	Warning	Diagnosing Profinet-I/O Unit	I/O Unit *X* diagnostic data on Slot *Y*
03:25:21	71444	IO	Info	Alarm in Profinet-I/O Unit	I/O Unit *X* has send an alarm on Slot *Y*
03:25:21	10010	Op	Info	MOTOR-OFF State	The system is in the Motors OFF state. ...
03:25:21	71439	IO	Warning	Diagnosing Profinet-I/O Unit	I/O Unit *X* diagnostic data on Slot *Y*
03:25:22	10011	Op	Info	Motors ON state	The system is in the Motors ON state.
03:25:23	80002	User	Warning	User Defined Error	User Defined Description

However, especially the description contains variables embedded in the text: For example, in the case of the IO category, events, the identity of the exact I/O unit, and the slot within the unit that triggered the event. This specific information can be very important when resolving an issue on the robot system. The last row in the table shows a user-defined event. These events have the event codes 80002 and 80004 (all 80002 event titles and description have been altered in order to anonymize the analyzed event log). With user-defined events, the robot programmer can define own events for a warning and for an error with his own user-defined titles and descriptions. As a consequence, very different types of events can be captured by events with these two event codes. Examples of such events are problems in material supply towards the robot system or problems with tools attached to the robot system. This mix of containing both structured information as well as largely unstructured information (e.g. variables embedded in free text) is very common for industrial log data.

In general, industrial log data is an ordered sequence of events with attributs $e = (ts, ca_1, ...ca_n, tx_1, ..., tx_m)$ where ts is the timestamp, $ca_1, ..., ca_n$ are categorical attributes of the event and, $tx_1, ..., tx_m$ are textual attributes of the event. Summarizing such industrial log files is not very well supported today. From the perspective of industrial practice, methods to summarize robot event logs must meet certain requirements:

1. The method should be able to concisely summarize large sets of industrial event log entries.
2. The method should not require labeled data, because upfront labeling of raw log data requires expensive expert time.
3. The method should not require prior domain knowledge, because industrial robots are used across various manufacturing domains and serve very different purposes.
4. The method should be able to capture the usual problem situations present in the data.
5. The method should follow a clear methodology to support applications by domain experts instead of machine learning experts.

4.1 Applying LDA on Industrial Logfiles

LDA does not require any prior labeling of the data or formal domain model, it generates topic assignments only based on the observable words. LDA should be also able to capture normal and common error situations present in the data, because these should be reflected in the probability distribution of the events. This section describes how LDA can be applied to logfiles given a fixed number of topics K and a vocabulary V.

In order to apply LDA to industrial logfiles, it is necessary to define the corpus (of documents) and the vocabulary in the context of industrial logfiles. Following the temporal data models of Mörchen (2007), an item-set sequence can constitute the corpus and each item-set corresponds to one document. Item-sets can be generated by combining all events within a start time t_s and an end time t_e, where t_s and t_e can be defined based on calendar time (e.g. days, hours, shift start and end) or based on production information (e.g. start and end time of manufacturing steps or manufacturing batches).

The vocabulary V defines the possible item in the item-sets or documents. Considering the data in Table 1, two general strategies exists to define the vocabulary:

1. Text-based vocabulary: The vocabulary is created from the words information in the event log and is the union of the words of any subset $v \subset \{cx_1, ..., cx_n\} \cup \{tx_1, .., tx_m\}$ of the categorical and textual columns. The items (or words in the document) will be by the union of all the words of all events between the start time t_s and the end time t_e.
2. Event-based vocabulary: The vocabulary is created from the types of event present in the event log. Two events e_1 and e_2 are of the same type if for a subset $t \subset cx_1, ..., cx_n \cup tx_1, .., tx_m, e_1(a) = e_2(a) \quad \forall a \in t$. The items of the item-set (or words in the document) will be the event types of the events between the start time t_s and the end time t_e.

Both approaches have their benefits and drawbacks. Using the textual information like in the first approach might lead to easily interpretable topics containing words in natural language. On the other hand, the fact that categorical information often carries a lot more information than textual information is easily lost. In the robot log for instance the code captures a very specific type of event that the robot manufacturer wanted to capture. At the same time, the textual information in the title is largely redundant because there exists a 1:1 mapping between code and title (except for user defined events) and even descriptions are very similar with respect to the present words. At the same time, the description contains also very specific data like the I/O Unit or in some cases even program pointer addresses. Excluding this very fine grained, very divers textual information, for instance with the help of stemming and stop word lists, is very difficult. Predefined stop word lists available for natural-language-processing tasks will be of limited use.

Algorithm 1

```
procedure ROBOT_PREPROCESSING ($logfile$)
    $e$ = read($logfile$)
    $e[event\_type] = e[code] + e[title]$                     ▷ Create Event Type
    $e$.drop($code, category, severity, title, description$)   ▷ drop columns not required
    $[(t_s, t_e)] = (e[ts]$.truncate($day$), $e[ts+1]$.truncate($day$))   ▷ item-sets border from calendar days
    return $e, [(t_s, t_e)]$
end procedure

procedure LDA_LOGDATA ($e, k, stop, [t_s, t_e]$)
    $V = e[event\_type]$.unique $- stop$     ▷ Create vocabulary and remove stop events
    $C = []$                                  ▷ Initialize empty corpus
    for all $t_s, t_e \in [t_s, t_e]$ do
        $d = e[ts > t_s$ and $ts \leq t_e]$   ▷ Select events in timeframe
        $d = d$.groupby($event\_type$).count     ▷ Transform into “bag of events”
        $C$.add($d$)
    end for
    return LDA($C, V, k$)   ▷ Execute LDA with generated Corpus, Vocabulary and for k topics
end procedure
```

Using event types as vocabulary lacks an easy interpretability, but captures the domain knowledge encoded in categorical texts much better. In addition, event codes often have a very clear and concise meaning to domain experts. For the data set in this study we used an event type vocabulary. Algorithm 1 shows in pseudo-code the robot log data specific preprocessing steps as well as the generic steps required to perform LDA on industrial log data.

4.2 Topic Modelling Workflow for Domain Experts

The LDA procedure described in the previous subsection assumes a predefined list of start and end times to derive documents from the event log, a list of stop events, and a fixed number of k topics. Selecting these required input values is very difficult for a domain expert. We suggest the following iterative approach to define these hyper-parameters:

1. Pick initial slicing of the event logs.
2. If the resulting corpus ...
    a) ... contains less then 250 items-sets, then decrease slice size,
    b) ... contains less then 100 item-sets with 75 or more items, then increase the slice size, or
    c) ... contains less then 250 items-sets and less then 100 item-sets have 75 or more items, then the data is probably not suitable for LDA.
3. Train LDA models for $k = 5, \ldots, 100$ with steps of 5 between each k.
4. Calculate and check topic coherence score.
5. If the domain expert is ...
    a) ... satisfied with the topics for one k, then use the selected topic model, or
    b) ... not satisfied with the topics, then...
        i) ... review topics for noisy and redundant events and add them to the stop event list and repeat training from step 3, or
        ii) ... increase or decrease the size of slices considering the constraints from step 1, if no more noisy or redundant events are present in the topics.

The recommendations or constraints regarding the number of item-sets and the size of item-sets is derived from the investigation of limiting factors of topic modelling with LDA by Tang et al. (2014). Their experiments on both synthetic and real-world data show that LDA does not produce good results on a corpus smaller then 250 elements and a document length considerably smaller then 100 words. In our workflow, we generate topic models with k between 5 and 100 topics. For each of the generated topic models the coherence score is calculated.

Syed and Spruit (2017) introduce the topic coherence score as a proxy for human topic ranking to examine LDA models. The topic coherence score can be defined as a distributional hypothesis which conveys that words with similar meaning tend to co-occur within a similar context. The coherence measures estimate the probabilities $\mathcal{P}$ of word co-occurences based on segmenting documents (word sets) into smaller pieces and calculate a confirmation measure $\mathcal{P}$ and aggregate the confirmation measures of the top-n words in each topic. The specific coherence measures C_V used in the study shows according to Röder et al. (2015) the best correlation with human topic rankings. C_V generates virtual documents by a sliding window of size 110 words, uses normalized pointwise mutal information (NMPI) as coherence measure and aggregates the NMPI of the top-n words of the topics by calculating the average. Details on how to calculate C_V can be found in Syed and Spruit (2017). The domain expert will review only the topic models with a relatively high topic coherence score. In the study based on the robotic data, this workflow has been very effective to generate topic models summarizing the robot log data per calendar day.

5 Example Results and Evaluation

The approach suggested in the previous section was tested on a real-word industrial data set. One year of data from more than 150 robots resulted in 2260 item-sets having between 8 and 280 events per item-set (24 events on average). 105 item-sets contained more than 75 items. Thus, the requirements of the workflow described in Section 4.2 where just meet. The size of the dictionary was 388 event types, after removing 12 event types as stop items in the iterative process described in Section 4.2. The topic models where generated with Gensim (Řehůřek and Sojka, 2010) version 3.5 using the option to choose the hyper-parameters α and β automatically. With help of the workflow from Section 4.2 a topic number of $k = 20$ was identified as most suitable. In 50 experiments, the training of the 20 models and calculating the corresponding topic coherence scores for the models took between 28 and 31 seconds in a sequential execution (the experiments where run on a Windows 7 64-bit machine with Intel i5 2.4GHz processor and 8GB of RAM). The training of the individual models for the different k took between 1.1 and 3.3 seconds (average 1.6 seconds) and the calculation of the topic coherence score always took below one millisecond. This performance was sufficient for an efficient usage of the workflow from Section 4.2 and could be easily improved by concurrent generation of the topic models.

Table 2: Topic weights summarizing the logs of two robots for one week.

Robot A

Day	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
1	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
2	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
3	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
4	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
5	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
6	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02
7	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02

Robot B

Day	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
1	.01	.01	.01	.01	.01	.90	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01
2	.00	.00	.00	.00	.00	.63	.00	.00	.00	.00	.15	.06	.00	.00	.00	.00	.00	.12	.00
3	.01	.01	.01	.01	.01	.66	.01	.01	.01	.01	.01	.23	.01	.01	.01	.01	.01	.01	.01
4	.01	.01	.01	.01	.01	.86	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01	.01
5	.10	.00	.00	.00	.00	.73	.00	.00	.00	.00	.00	.1	.00	.00	.00	.00	.00	.00	.00
6	.01	.01	.01	.01	.01	.51	.01	.01	.01	.01	.01	.26	.01	.01	.01	.01	.01	.01	.01
7	.02	.02	.02	.68	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02	.02

Table 2 shows the daily topic distribution from two different robots over a week (the values are rounded to two digits). Both tables show a very distinctive topic distributions over the entire week plotted. Table 3 shows the top events of the two most prominent topics. The top three events of topic 3 are very common and non-critical events like an opening of safety cage of the robot cell or the start and completion of a scheduled brake check. The topic basically denotes that the robot was in normal operation. For robot A, LDA assigned 68 % of the probability mass to this topic and spreads the remaining probability equally across all remaining topic. Topic 5 denotes a problem with a tool attached to the robot arm along with communication problems to a specific device. This hints that there is a communication issue between the robot controller and the attached device. This problem is persistent for robot B almost over the entire week. At the end of the week, the robot B seems to return to normal operation, possible due to some maintenance activities while the robot was switched off. This type of visualisation can quickly summarize the operational status of robots over longer periods of times and help for instance maintenance managers to plan and prioritize their efforts for the time when robots are not producing.

Table 3: Example topics with top 3 event types and probabilities.

Topic	Top 3 Event Types	Probability
Topic 3	20206 (General Stop Open)	0.335
	10271 (Cyclic Brake Check Started)	0.076
	10270 (Cyclic Brake Check Done)	0.076
Topic 5	80002 (Tool A Error)	0.185
	71276 (Communication established with I/O device)	0.101
	71058 (Lost communication with I/O device)	0.1

In order to assess whether the topic models generated using the proposed iterative approach described in Section 4 are suitable for summarizing industrial logfiles, we adapted a method suggested by Han Lau et al. (2014), namely direct annotation of topic assignments. For this method, three domain experts were presented 287 test documents together with the top-*N* words for the top-ranked topic for each document. The domain experts were asked to assess the *matching* of the top ranked topic for the presented document on a rating scale from 0 to 3:

0: If the topic is not at all present in the event log item-set

1: If the topic has very low relevance for item-set

2: If the topic is present in the item-set but not the most relevant one

3: If the topic is present in the item-set and it is the most relevant one

Table 4 shows the outcome of this assessment. Overall, for 161 documents, the assigned topic was rated the most relevant topic for the document, for 70 documents, the assigned topic was rated as present in the document but not the most relevant topic, for 51 documents, the assigned topic was rated as present with a low relevance and for 5 documents, the assigned topic was rated as not relevant. The arithmetic mean value of this evaluation is 2.34. In fact, about

80 % of the topic assignments were relevant or most relevant for the assessed documents, whereas for only 1.7 % of the documents the assigned topic was not present in the document.

Table 4: Direct annotation of topic assignment for test item-sets.

Evaluator	Rating 0	Rating 1	Rating 2	Rating 3	Test Documents
A	1	38	38	129	206
B	3	10	9	22	44
C	1	3	23	10	37
Total	5	51	70	161	287

6 Conclusion

In this work, we proposed a topic modeling technique for analyzing industrial log files with the help of the Latent Dirichlet Allocation model and a workflow suitable for domain experts to generate useful topic models. The subsequent domain expert's direct annotation evaluation showed that the Latent Dirichlet Allocation model's result delivered captured useful information about a real-world industrial log file in a compact way. The proposed iterative approach can certainly be useful in monitoring industrial systems and serves human analysts as a decision support tool. Log file analysis using the proposed topic modeling technique helps to identify the industrial system's behavioural trend. Run-time consumption was found to be well within acceptable limits which in return supports larger timescale analyses.

The approach in its current implementation has several limitations that indicate future research needs. We showed that robots can produce considerable amounts of data on a smaller timescale than days. In our example data sets, such periods were to infrequent to enable topic modelling. Data augmentation for industrial log files could be an important mechanism to overcome this limitation. Second, the approach is only able to capture information that is present in the training data. In the current application this is acceptable, because the possible robot event types are statically defined with programs of robots. In other cases or when using a different approach to create event types, new event types might appear over time. In this case, an online version of topic modelling without the need for expert evaluation would be useful. Additional future research directions are to leverage the present natural language in log files to augment the summarization for the human user, and a less heuristic workflow leveraging methods of hyper-parameter optimization in order to require less expert input.

References

Abele L, Anic M, Gutmann T, Folmer J, Kleinsteuber M, Vogel-Heuser B (2013) Combining Knowledge Modeling and Machine Learning for Alarm Root Cause Analysis. IFAC Proceedings Volumes 46(9):1843–1848. DOI: 10.3182/20130619-3-RU-3018.00057.

Atzmueller M, Arnu D, Schmidt A (2017) Anomaly Detection and Structural Analysis in Industrial Production Environments. In: Haber P, Lampoltshammer T, Mayr M (eds), Data Science–Analytics and Applications. Springer Vieweg, Wiesbaden (Germany), pp. 91–95. DOI: 10.1007/978-3-658-19287-7_13.

Blei DM (2012) Probabilistic Topic Models. Communications of the ACM 55(4):77–84. DOI: 10.1145/2133806.2133826.

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet Allocation. Journal of Machine Learning Research 3(Jan):993–1022, Lafferty J (ed). URL: http://www.jmlr.org/papers/v3/blei03a.html.

Chandola V, Kumar V (2007) Summarization – Compressing Data Into an Informative Representation. Knowledge and Information Systems 12(3):355–378. DOI: 10.1007/s10115-006-0039-1.

Czora S, Dix M, Fromm H, Klöpper B, Schmitz B (2017) Mining Industrial Logs for System Level Insights. In: Mitschang B, Nicklas D, Leymann SH F., Herschel T M., J., Härder T, Kopp O, Wieland M (eds), Datenbanksysteme für Business, Technologie und Web (BTW'17, Workshopband), Gesellschaft für Informatik e.V., Bonn (Germany), pp. 57–64. URL: https://dl.gi.de/handle/20.500.12116/941.

Folmer J, Vogel-Heuser B (2012) Computing Dependent Industrial Alarms for Alarm Flood Reduction. In: International Multi-Conference on Systems, Signals & Devices, Institute of Electrical and Electronics Engineers (IEEE), New York (USA), pp. 1–6. DOI: 10.1109/SSD.2012.6198008.

Fullen M, Schüller P, Niggemann O (2017) Defining and Validating Similarity Measures for Industrial Alarm Flood Analysis. In: IEEE 15th International Conference on Industrial Informatics (INDIN'17), Institute of Electrical and Electronics Engineers (IEEE), New York (USA), pp. 781–786. DOI: 10.1109/INDIN.2017.8104872.

Hadžiosmanović D, Bolzoni D, Hartel PH (2012) A Log Mining Approach for Process Monitoring in SCADA. International Journal of Information Security 11(4):231–251. DOI: 10.1007/s10207-012-0163-8.

Han Lau J, Newman D, Baldwin T (2014) Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. Conference of the European Chapter of the Association for Computational Linguistics (EACL'2014), pp. 530–539, Wintner S, Goldwater S, Riezler S (eds), ACL Anthology, Michigan (USA). DOI: 10.3115/v1/E14-1056.

Heinrich G (2008) Parameter estimation for text analysis. University of Leipzig. Tech. Rep., Tech. Rep.

Laxman S, Shadid B, Sastry P, Unnikrishnan K (2009) Temporal Data Mining for Root-cause Analysis of Machine Faults in Automotive Assembly Lines. arXiv preprint, arXiv:0904.4608. URL: `https://arxiv.org/abs/0904.4608`.

Mörchen F (2007) Unsupervised Pattern Mining from Symbolic Temporal Data. ACM SIGKDD Explorations Newsletter 9(1):41–55. DOI: 10.1145/1294301.1294302.

Porteous I, Newman D, Ihler A, Asuncion A, Smyth P, Welling M (2008) Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), Association for Computing Machinery (ACM), New York (USA), pp. 569–577. DOI: 10.1145/1401890.1401960.

Řehůřek R, Sojka P (2010) Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, European Language Resources Association (ELRA), Paris (France), pp. 45–50. DOI: 10.13140/2.1.2393.1847.

Röder M, Both A, Hinneburg A (2015) Exploring the Space of Topic Coherence Measures. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM'15), Association for Computing Machinery (ACM), New York (USA), pp. 399–408. DOI: 10.1145/2684822.2685324.

Sontag D, Roy D (2011) Complexity of Inference in Latent Dirichlet Allocation. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ (eds), Advances in Neural Information Processing Systems (NIPS 2011), Curran Associates, Inc., Vol. 24, pp. 1008–1016. URL: `https://papers.nips.cc/paper/4232-complexity-of-inference-in-latent-dirichlet-allocation`.

Syed S, Spruit M (2017) Full-text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation. In: IEEE International Conference on Data Science and Advanced Analytics (DSAA'17), Institute of Electrical and Electronics Engineers (IEEE), New York (USA), pp. 165–174. DOI: 10.1109/DSAA.2017.61.

Tang J, Meng Z, Nguyen X, Mei Q, Zhang M (2014) Understanding the Limiting Factors of Topic Modeling Via Posterior Contraction Analysis. In: Xing EP, Jebara TS (eds), Proceedings of the 31st International Conference on Machine Learning (ICML'14), Association for Computing Machinery (ACM, New York, Vol. 32, pp. 190–198.

Wallach HM, Mimno DM, McCallum A (2009) Rethinking LDA: Why priors matter. In: Advances in neural information processing systems, pp. 1973–1981.

Wang P, Wang H, Liu M, Wang W (2010) An Algorithmic Approach to Event Summarization. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD'10), Association for Computing Machinery (ACM), New York (USA), pp. 183–194. DOI: 10.1145/1807167.1807189.

Zöller MA, Baum M, Huber MF (2017) Framework for Mining Event Correlations and Time Lags in Large Event Sequences. In: IEEE 15th International Conference on Industrial Informatics (INDIN'17), Institute of Electrical and Electronics Engineers (IEEE), New York (USA), pp. 805–810. DOI: 10.1109/INDIN.2017.8104876.