Leveraging Unstructured Data to Analyze Implicit Process Context

Renuka Sindhgatta[1(✉)], Aditya Ghose[2], and Hoa Khanh Dam[2]

[1] IBM Research, Bangalore, India
renuka.sr@in.ibm.com
[2] University of Wollongong, Wollongong, Australia
{aditya.ghose,hoa}@uow.edu.au

Abstract. Adapting a business process to different context requires identifying various situations and evolving the process to support such situations. Previous work focused on modeling, observing and collecting contextual information. Furthermore, impact of context on process or resource performance has been studied. However, much of the work considers explicit contextual information that is defined by domain experts. There are several implicit contextual dimensions, that are difficult to model as all situations cannot be anticipated a priori. Context mining involves analysis of process logs to identify context and correlate with process performance indicators or outcomes. In this work, we leverage unstructured data available in user comments or mails to discover implicit context of the process. We automatically analyze textual data and group process instances by applying information extraction and text clustering techniques. Groups of process instances are correlated to their process outcomes to filter irrelevant information. We apply the approach on real-world process logs to identify contextual information.

Keywords: Process context · Natural language processing
Cluster analysis · Process execution logs

1 Introduction

Analyzing and (machine) learning impact of the business process context (or the environmental factors), on its execution helps adapting and improving the process [9]. There exists many interpretations of the notion of context in various disciplines including mobile applications and eCommerce personalization. In one of the early works by Dourish [5], two views of context are presented. First, a *representational view*, where context is defined as information that is stable, can be defined for an activity and is separable from the activity. Hence, context is described using a set of attributes or dimensions. An example of a representational process context is the *hour of the day* when the process executes. It is independent of the activity and yet has an impact on the execution of activity (peak workload). Second, an *interactional view*, where context is dependent on

M. Weske et al. (Eds.): BPM Forum 2018, LNBIP 329, pp. 143–158, 2018.
https://doi.org/10.1007/978-3-319-98651-7_9

the activity, and can be dynamically produced by the activity. An example of interactional view is the non-availability of a customer to confirm the purchase of an insurance claim. While the situation is not a part of the process, it is dynamically created as the activity requires confirmation by the customer.

Business process context modeling considers the representational view, which we term as explicit context: information that is identified by domain experts and can be defined a priori. Saidani et al. [23] define a meta-model of context for a business process. The meta-model comprises of context entity, context attributes and context relationships. A domain expert can define a context model based on the meta-model and the contextual information can be observed from the process execution logs. For example, in the insurance claim process, a domain expert would indicate that the location of customer as contextual information, as the process path and outcome could vary for customers in different locations. These attributes are characterized as *explicit contextual dimensions*. Existing approaches extract contextual dimensions from structured information in process logs, and use supervised learning methods to predict process or resource performance [10,11,25,26].

There are situations that arise as a part of performing a task or an activity (interactional view), and may not be known a priori. These *implicit contextual dimensions* need to be discovered from various sources of information. For example, in an IT application maintenance process, when performing the task of resolving IT problem, the worker or resource may find that, certain legacy applications require more time to resolve as multiple interlinked applications need to be restarted, while a new application using web services takes less time as it requires restart of just that specific web service. This information is implicit and once identified, the process redesign could assign different resolution times based on the new contextual dimension of type of application - legacy application or service based application. The source of identifying the underlying implicit context can be unstructured information available as textual comments that are recorded during the process execution.

In this work, we study the problem of exploiting unstructured textual data to discover implicit context. In the proposed framework, textual data is extracted from execution logs of process instances. Commonly occurring situations are identified by applying text clustering methods. A few relevant clusters are semi-automatically selected by applying filtering rules and choosing clusters with significantly different process performance. The clusters of textual information, can be considered as input to identifying contextual information. This approach helps domain experts discover possible contextual dimensions. To the best of our knowledge, discovery of process context from unstructured or textual data available with process execution histories has not been considered so far. To summarize, the following are the main contributions of our work:

- Introduce the research problem of mining context from textual information available during the process execution.
- Propose a semi-automated approach of identifying context using textual information available in process execution logs.

The paper is organized as follows. Section 2 presents a real-life motivating example, followed by a background of concepts used in our work (Sect. 3). The overall approach is outlined in Sect. 4, and a detailed empirical evaluation is presented in Sect. 5. Related work is presented in Sect. 6, followed by conclusions and future work in Sect. 7.

2 Motivating Example

Table 1, contains textual information logged by workers or resources involved in the process of maintaining IT applications. A problem is reported by a customer. The resource or worker allocated to the task, evaluates the problem, identifies and executes relevant resolution, confirms with the customer if the problem has been resolved. At every step in the process of analyzing and resolving the problem, the details are recorded in an incident management system (process aware information system). Examples in Table 1 are representative of typical challenges with textual logs of business processes: (i) varying informativeness from being very brief to very detailed, (ii) containing ill formed sentences with grammatical errors, typographical errors and abbreviations. The entry numbered 2, has detailed information of the steps taken to resolve the issue. The entry 4, has very limited information and hence is of little value. The characteristics of the textual information available in the maintenance of 4 IT applications is shown in Table 2. Textual data is small in terms of the number of words in a process instance log.

Table 1. Unstructured textual information captured during IT maintenance process

No.	**Communication log of the problem tickets recorded by knowledge workers**
1	emailed user. ***waiting for user to get back to me*** emailed user. looking for response User confirmed that the issue is not replicated. Hence closing the incident
2	Left a voicemail for customer at the number provided in this ticket Requested he call option (one) for further assistance **Validated userid in the portal, made in Synch** **Manually made in Synch with that of GUI** Call made both on office phone and cell ***Voice sent on cell and office phone is not reachable*** ***2nd call made to the customer***. No response.. ***3rd call made to the customer*** No response. Call closed due to no prior response from the customer
3	**incorrect logon locks**. unlocked the ID and reset the password pinged user via IM ***John confirmed to close the incident***
4	Received confirmation from user, closing the incident

Table 2. Characteristics of textual data in process logs of real-life IT application maintenance process

Application	Number of process instances	Number of sentences	Average number of words per sentence	Average number of words per process log
Application security	684	2235	10.25	44.35
Portal	210	1569	14.11	118.02
HR system	490	1482	11.87	41.38
Reporting	832	1267	9.71	20.02

However, these logs reflect some common situations that arise when performing an activity. For example, 'Unavailability of the customer' could be a situation or a task context, and could impact the time taken to perform the task. The log contains both, (i) information relevant to the specific process or task, and (ii) information that represents context. Hence, the textual data can refer to multiple topics. In the following section, we describe the background of concepts that can be applied to mine relevant information from the logs, specifically related to identifying multiple topics from textual documents.

3 Background

This section presents well known natural language processing techniques that can be used together to mine contextual information from process logs.

3.1 Notations

The textual information logged during the execution of a process instance can be considered as a text document. Let each document $d_i \in D$ represent textual information logged for respective process instance $p_i \in P$. Each document could comprise information on activities being performed, the actions taken when performing the activity and the situation or conditions during the execution of the activities. Hence, document d_i comprises of one or more topics of the topic set $T = \{t_1, t_2 \dots t_T\}$ with some topics representing the context of the process instance. The problem can be represented as a multi-label categorization of textual logs.

We further assume that each document d_i is represented by smaller constituents that relate to one or more topics. The smaller constituents or chunks of text are called *segments*, which in turn contain one or more sentences. A segment is small enough to contain information relevant to a single topic. We believe that, in general, this assumption holds for communication logs containing short descriptions. Hence let S_i be the set of segments of document d_i, then $S = \bigcup_{i=1}^{|D|} S_i$, is a set of all segments. The goal is to find the topics T over S, and further find the topics for each document $T_i \subseteq T$ based on topics of the segments S_i of the document d_i, and hence the process instance p_i.

3.2 Segmenting Document

The goal of breaking down the document into segments, is to identify smaller constituents that represent distinct information related to tasks or their context. There are multiple ways of segmenting text. The suitability of the method is based on the characteristics of the textual information in the process logs.

1. *Phrase extraction* using parts-of-speech (POS) patterns has been used to extract text segments [6,21]. These are similar to regular expression patterns based on parts of speech. While, pattern based extraction has a high precision in extracting information, it has low recall as it filters phrases that do not match the POS pattern. For example, the phrases 're-provisioning completed', 'has been re-provisioned' and 're-provisioned and sent confirmation', have the same information, and yet have different POS tag patterns: 'VBG VBN', 'VBZ VBN VBN', 'VBN CC VBN NN' respectively (VBN is verb, CC is conjunction, and NN is noun, based on the listing of POS tags by Penn Treebank Project [18]). This method of segmentation is suitable when information logged by process participants is based on standardized templates.
2. *Parse Tree* is a rooted tree that represents the syntactic structure of a sentence based on a grammar. There are two ways of constructing parse trees: (1) constituency relation that is based on phrase structure grammar, (2) dependency relation that is based on relations among words. Constituency parser can be used to break down the sentence to extract smaller noun or verb phrases. Noun and verb phrases can be used as segments of the document. Parse trees are suitable when there is very sparse data reported by the process participants. In such scenarios the information extracted, is limited to key actions recorded during process execution. For example, from the communication log on the first row in Table 1, verb phrases such as 'emailed user', 'waiting for user', 'looking for response' can be extracted by using constituency parser.
3. *Extractive summarization* is an automatic text summarization method that, produces a summary of the text while retaining key information in a document [2]. There are two well known methods to summarization (i) abstractive summarization, and (ii) extractive summarization. Extractive summarization identifies important sections of the text and generates them verbatim. Distinct sentences of the document summary can be used as segments. Summarizing text is suitable when there verbose comments logged by process participants.

3.3 Clustering Methods

The extracted text segments can be categorized and grouped using different clustering methods. We briefly discuss common clustering methods and their suitability to grouping textual data available in process logs:

1. *Topic Modeling* Clustering approaches such as latent semantic analysis [20], probabilistic latent semantic analysis (pLSA) [14] and latent Dirichlet allocation (LDA) [3] have been used to identify representative set of words or topics. These approaches identify topics by exploiting the co-occurrence of

words within documents and are well suited for multi-topic text labeling. However, they are not suitable for short documents containing limited number of words and sentences. Hence, while these methods are widely used in multi-class text categorization, they are unsuitable for textual data available in process logs.
2. *Partition based clustering* such as k-Means, k-Mediods, are the most widely used class of clustering algorithms [13]. These algorithms form clusters of data points, by iteratively minimizing a clustering criterion and relocating data points between clusters until a (locally) optimal partition is attained. An important requirement of partition based methods is the number of partitions or K as input.
3. *Affinity Propagation* is one of the recent state-of-the-art clustering methods that has better clustering performance than partition based approaches such as k-Means [7]. Affinity propagation identifies a set of 'exemplars' and forms clusters around these exemplars. An exemplar is a data point that represents itself and some other data points. The input to the algorithm is pair-wise similarities of data points. Given the similarity matrix, affinity propagation starts by considering all data points as exemplars and runs through multiple iterations to maximize the similarity between the exemplar and their member data points.

3.4 Text Similarity

Next, we focus on the key aspect of any clustering algorithm; the choice of (dis)similarity function or distance metric between data points (text segment pairs). A text segment, is represented as a vector and distance functions such as Euclidean distance or similarity functions such as cosine similarity are used.

1. *Bag-of-Words (BOW)*: Each text segment is represented as vector of word counts of dimensionality $|W|$, where W is the entire vocabulary of words.
2. *TF-IDF*: The bag-of-words representation divided by each word's document frequency (number of text segment it occurs). The representation ensures that commonly occurring words are given lower weight.
3. *Neural Bag-of-Words (NBOW)*: Each text segment is represented as a mean of the embeddings of words contained in the text segment. The embeddings of words are obtained using the word2vec tool [19]. As the word vectors retain the semantic relationships, the distances between embedded word vectors can be assumed to have semantic meaning.
4. *Word mover distance (WMD)*: WMD is suitable for short text documents (or text segments). It uses word2vec embeddings [16]. The word travel cost (or euclidean distance), between individual word pairs is used to compute document distance metric. The distance between the two documents is the minimum (weighted) cumulative cost required to move all words from d_i to d_j. When there are documents with different numbers of words, the distance function moves words to multiple similar words.

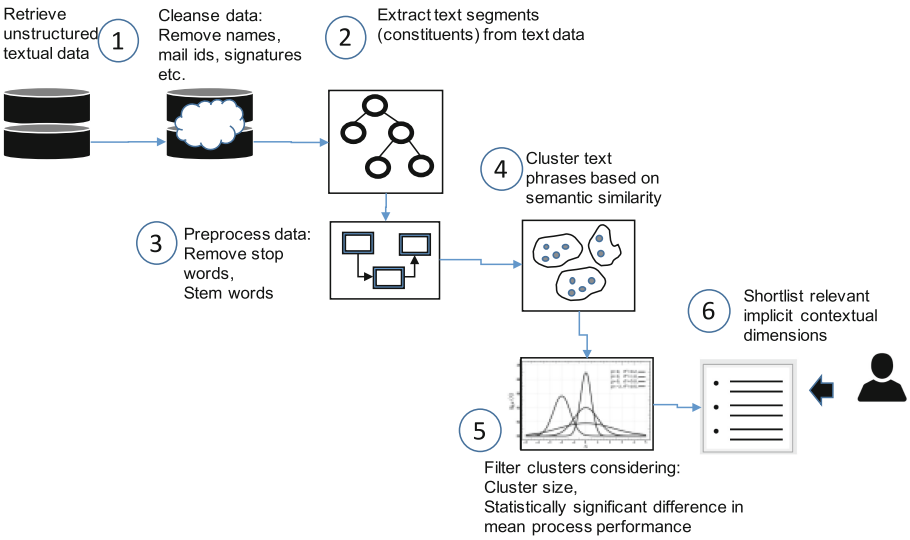

Fig. 1. Overall approach to identify implicit contextual dimensions

4 Overall Approach

Our approach to infer or identify implicit context is organized into multiple steps, as shown in Fig. 1. The approach comes down to answering three key questions: (i) What are the common situations and actions taken by the performers of a process during its execution? (ii) How many process instances are related to these situations? - is this a common or a rare situation? and (iii) Are these representative of process context and do they impact the performance outcome of the process? The steps of the approach are discussed in detail:

4.1 Text Retrieval and Cleansing

A tuple $\langle pid, ppi, text\_data \rangle$ containing the process instance identifier (pid), the process performance indicator (ppi) [4], and the unstructured textual information is extracted from execution logs. The use of each of these attributes, will be described in the following steps. The $text\_data$ for each process instance is referred to as a document. The document is processed to remove the names of people, IP addresses, HTTP addresses, and other textual data such as email signatures, phone numbers, that would not represent common actions or situations. The cleansing uses named entity recognizer[1], to detect person names, organization names. IP addresses, phone numbers, email addresses are cleaned from the text using regular expression parsers.

[1] https://nlp.stanford.edu/software/CRF-NER.html.

4.2 Text Segmentation

In this step the document is broken down into text segments by extracting summaries, or by extracting phrases using constituency parsing. A suitable method is chosen based on the characteristics of textual log (sparsity, verbosity, or variety), as described in Sect. 3.2. Hence we have $\langle pid, text\_segment \rangle$.

4.3 Text Preprocessing

Each text segment goes through standard preprocessing steps (i) lemmatization, where the base form of the words in the text segment are derived (e.g. - allocate, allocation, allocating are replaced by their lemma 'allocate'). (ii) Stop word removal, where very frequent words that are likely to appear in all the documents and contain little information, are removed.

4.4 Clustering

The text segments are clustered using one of the similarity measures described in Sect. 3.4. This step results in grouping process instances having similar text segments. The process instance associated to each text segment and its performance indicator is used to form a tuple $\langle pid, cluster\_id, text\_segment, ppi \rangle$.

4.5 Filtering Clusters

The goal of this step is to identify clusters of text segments, that are important and useful to a domain expert and help discern contextual dimensions. Two filters can be applied:

Size Filter: The number of process instances associated with a cluster is a good indicator of its importance. Intuitively, if the size is very large, then the information content is a part of normal execution of the task. For example, if the number of process instances associated to the phrase 'confirming and closing loan application' is very large, it is indicative of a normal procedure. Similarly, a cluster containing very few process instances may not be useful as it may indicate an exception and has to be handled as a part of the process exception or process error management. An upper and lower bound on number of process instances is set to filter clusters.

Process Performance Filter: This filter helps identify clusters that have an impact on the performance indicators of the process. The performance indicators of a process can be the completion time, the quality outcome of the process, or any other process indicator as detailed in [4]. To verify if the performance indicators of the process instances of a cluster are significantly different from other process instances, we consider two sample groups - (i) cluster group, and (ii) other group. Performance indicators of all process instances in a cluster are taken as one sample (cluster group). Performance indicators of a randomly chosen set of process instances from other clusters are considered as the second

independent sample (other group). The Mann-Whitney U test is used to compare statistically, the difference in the performance indicators of the two groups. The test is run with multiple random samples of *other group* to reduce false positives or Type 1 error. The Mann-Whitney U test is one of the powerful non-parametric tests that makes no assumption on the distribution of data and is relevant for groups with small sample sizes (as clusters could be containing 10 process instances).

4.6 Context Identification

The final step of the approach is a manual verification by domain experts on the filtered set of clusters. The description in the text segments of filtered clusters are used by the domain experts to identify contextual situations that impact the performance of the process.

5 Experimental Evaluation

We first evaluate and compare the segment based clustering using different clustering methods, and similarity measures, on a benchmark set of multi-topic documents, as there is no benchmark textual data of business process available to evaluate the approach. Next, the overall approach detailed in Sect. 4, is used on a real-life business process textual log to identify the clusters that indicate contextual information.

5.1 Evaluating Clustering of Text Segments

The Reuters-21578 text categorization collection is a text categorization benchmark [29]. The *Mode Apte* evaluation, is used in which unlabeled documents are removed. There are 10787 documents that belong to 90 categories. The collection has a training set containing 7768 documents and a test set containing 3019 documents. Two main constraints are set up on the data: (1) each document should be assigned to at least 3 topics or categories, (2) each category or topic must have at least 1% of the documents. The training set is used to set the parameters for affinity propagation and choosing K for k-Means, and group text segments into the same number of clusters as the categories in the collection (68 categories in our case).

The quality of segment based clustering is evaluated on the test data containing over 900 segments on 95 multi-labeled documents, using the commonly used criterion of *precision, recall* and *F1 measure* [27]. Two approaches are used to compute the measures for multiple categories. The Precision, Recall, F1-measure is computed for each category. Finally, the overall measure is obtained by averaging category specific Precision, Recall and F1 measure. This is known as macro-averaging ($Prec_M, Rec_M, F1_M$). The other approach is based on computing a confusion matrix of all the categories by summing the documents that fall in each of the four conditioned sets, namely true positives, true negatives,

false positives, and false negatives. The Precision, Recall and F1 measure is computed with the overall confusion matrix. This second measure is known as micro-averaging ($Prec_\mu, Rec_\mu, F1_\mu$).

The results are presented in Table 3. Text segments for each document are created by using extractive summaries. As K-Means algorithm is based on euclidean distance between two pairs, word mover distance is not evaluated. The results indicate that using affinity propagation based clustering, provides better F1 scores as compared to K-Means. Euclidean distance of NBOW and WMD measures result in higher macro-average and micro-average F1.

Table 3. Comparative evaluation of multi-class categorization for various distance measures and clustering methods

		Macro-average			Micro-average		
Clustering	Similarity	$Prec_M$	Rec_M	$F1_M$	$Prec_\mu$	μ	$F1_\mu$
K-Means	BOW	0.772	0.442	0.491	0.385	0.490	0.431
	TF-IDF	0.583	0.586	0.534	0.552	0.447	0.495
	NBOW	0.665	0.538	0.530	0.55	0.467	0.503
Affinity propagation	BOW	0.705	0.450	0.448	0.341	0.535	0.417
	TD-IDF	0.648	0.548	0.568	0.614	0.483	0.541
	NBOW	0.637	0.626	**0.580**	0.570	0.516	**0.542**
	WMD	0.652	0.593	**0.584**	0.631	0.470	**0.540**

5.2 Context Mining from Text Logs

The overall approach of identifying contextual information is evaluated on an IT maintenance process of 3 different applications of a large media and entertainment organization. The textual data recorded varies significantly for different application domains such as security, human resources, finance and web portal. The process consists of four main tasks: (1) customer creates an application problem ticket, (2) the worker acknowledges the receipt the ticket, (3) the worker analyzes the issue and resolves the problem, (4) on resolving the problem, the worker confirms with the user, and (5) the worker closes the ticket. At each step, the workers log their findings or progress. In some cases, emails sent or received by the customer and the worker is logged in the system. We analyze the communication or task logs associated with each process instance.

To evaluate the overall approach of mining contextual factors from textual data, the pipeline of steps detailed in Sect. 4 is executed. Table 4 presents the descriptions derived from the text segments in the filtered clusters. As shown, for the 'Security' application, of the 2493 text segments extracted from all the process instance documents, clustering using affinity propagation with WMD, results in 119 groups or categories. The mean completion times of the process instances in these groups is compared to mean completion time of a random

number of other process instances. A statistical significance in the mean completion time (the performance outcome), is used to filter few clusters. Further, a filtering of clusters is done based on the size of the cluster. For example, 'confirm and close incident' is a very common text segment that is identified and associated with several process instances. It occurs in 50% of the process instances. It may hence, be a process completion step and not a situation or context. The highlighted descriptions in the table are examples of context.

Based on the cluster labels in Table 4 (that are derived from common text in the clusters), for the security application, it is observed that any process instance associated with *reset password* has lower completion time (indicated with a + sign in the table), as the task is extremely specific. The clusters further highlight a key situation of not being able to contact the customers, leading to the process being set to 'pending' status and the completion time being much higher than other process instances. Identifying such a situation can help re-design the process to account for customer unavailability. Similarly in the maintenance of the portal application, waiting for *more information* from the user leads to higher completion time of such tasks. A template with all relevant information recorded by the customers when creating the problem ticket, could be a plausible solution. In the HR domain application, the number filtered clustered were limited and the clusters did not provide useful insights on context.

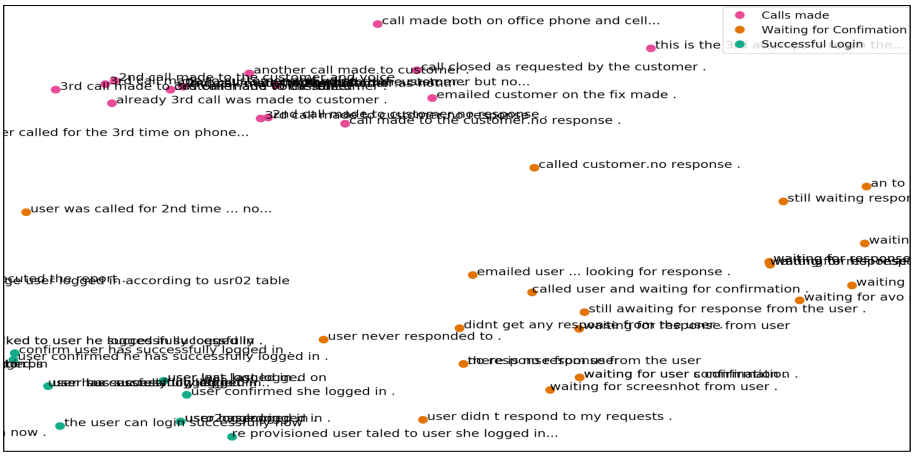

Fig. 2. Visualization of a subset of clusters

Figure 2 visually depicts a subset of clusters of the textual segments. The NBOW vectors of text segments is represented on a two dimensional space. The textual segments are the noun and verb phrases extracted using constituency parser.

Table 4. Filtered clusters of IT application maintenance process logs, (+) indicates clusters has lower completion times

App. Domain	# Text Segments	#Clusters	# Filtered	Cluster labels
Security	2493	119	13	1. **(2nd call, 3rd call) made to the customer**
				2. (researching, working, fixing) issue
				3. (asked, sent, mailed) to check again
				4. **waiting for (approval, confirmation)**
				5. could not (read, get, contact) user
				6. waiting for user
				7. **reset password (sent, mailed) user** (+)
				8. **changing status to pending**
				9. **tried calling the user**
				10. . . . .
Portal	2025	170	22	1. **sent to the user for (confirmation, information)**
				2. **waiting for user (confirmation, email)**
				3. **moved support issue to development**
				4. **getting more details on the issue**
				5. called and left a voice mail
				6. . . . .
HR system	2092	189	27	1. **(were, tied to, failed) data issues**
				2. closing the incident (+)
				3. **need to upgrade to breakfix**
				4. (write, call) back to me
				5. . . . .

5.3 Threats to Validity

Threats to *external validity* concerns the generalization of the results from our study. We have tried to limit this threat by evaluating it on textual data of 4 application domains, with over 300 users logging comments on over 2000 process instances. While insights can be drawn from our study, we do not claim that these results can be generalized in all business processes. However, the results serve as the basis of using textual data to discern relevant process context. Threats to *internal validity* arise when there are errors or biases. In our study, we have used standard implementations of distance functions and cluster analysis. The clustering and filtering approach required some configuration parameters such as the minimum and maximum size of the clusters. These should not impact the applicability of the approach. The choice of measurements is considered as a threat to *construct validity*. Appropriate measures such as precision and recall were not used on textual data in process logs due to non-availability of labeled data. However, we evaluated metrics on a multi-labeled benchmark data set to compare various methods of grouping textual information used in our study.

6 Related Work

The business process management community has experimented with use of unstructured textual data for various use cases:

Generating business process model from textual documents has been studied in some of the earlier work. Ghose et al. [12] propose a Rapid Business Process Discovery (R-BPD) framework and toolkit that employs text-to-model translation. Templates of commonly occurring textual cues or patterns are used to derive processes or task descriptions. Information extraction based approach is used to identify verb and noun phrases. In addition, recent work by Friedrich et al. presents an automatic approach of generating BPMN models from natural language text [8]. Sentence level analysis is done to extract performers and actions. This is followed by text level analysis where the relationships between sentences is used to determine links between actions and the control flow.

Teinemaa et al. exploit both unstructured text and structured attributes of cases for predictive business process monitoring [28]. The authors present a framework that extracts features from textual documents and evaluate different combinations of text mining and classification techniques to label executions as positive or negative.

There have been several efforts on using unstructured textual information available in problem tickets raised during IT application or service maintenance. There are approaches that use supervised learning to identify the right team or service agents for efficient ticket assignment [1,24]. Automatic recommendation of resolution for problem ticket based on similar nearest neighbors has been studied [30]. The underlying approach evaluates semantically similar past problem tickets and recommends appropriate resolution. Automatically analyzing natural language text in network trouble tickets has been studied by Potharaju et al. [21]. The authors present Netseive, a tool that infers problem symptoms, troubleshooting activities and resolution actions. Mani et al. [17] use clustering techniques and assign salient labels to group similar problem tickets. They use a combination of Lingo, a phrase based clustering method and N-gram extraction to identify phrases or cluster labels. However, they do not evaluate the clusters and their performance outcomes. In this work, we use an IT service management process for our study and evaluate different segment based clustering methods. Our approach further evaluates the clusters and analyzes the performance of process instances in these clusters.

Context-aware business process modeling has focused on design and specification of contextual attributes or dimensions [22,23]. There have been efforts on designing and evaluating impact of context on the process performance [10,11], and task allocation decisions [25,26].

Kiseleva et al. [15] introduced the notion of implicit and explicit context for predicting user behavior in eCommerce applications. The web user's age, gender and other known attributes are considered as explicit context, while information such as the purchase intent of the user is not known and is considered to be hidden context.

We propose a method of using textual information available in the process execution logs to uncover contextual dimensions.

7 Conclusion and Future Work

In this study, we proposed a novel approach of leveraging textual logs captured during a process execution for identifying useful and relevant situations or context. Using unstructured information extraction methods, we developed our approach of clustering process instances or tasks into unified groups, correlating them with process outcome and identifying a subset of situations that are correlated to the performance outcome. Our approach is quite general, and can be applied to different application domains. In future, we intend to explore filtering approaches beyond cluster size and performance outcomes. We also want to explore possibilities of automating identification of contextual situations by using labeled dataset and supervised learning techniques.

References

1. Agarwal, S., Sindhgatta, R., Sengupta, B.: SmartDispatch: enabling efficient ticket dispatch in an IT service environment. In: KDD, pp. 1393–1401 (2012)
2. Allahyari, M., et al.: Text summarization techniques: a brief survey. In: CoRR abs/1707.02268 (2017)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. del-Río-Ortega, A., Resinas Arias de Reyna, M., Durán Toro, A., Ruiz-Cortés, A.: Defining process performance indicators by using templates and patterns. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 223–228. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_18
5. Dourish, P.: What we talk about when we talk about context. Pers. Ubiquitous Comput. **8**(1), 19–30 (2004). ISSN 1617-4909
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545 (2011)
7. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007). https://doi.org/10.1126/science.1136800
8. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 482–496. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21640-4_36
9. Ghattas, J., Soffer, P., Peleg, M.: A formal model for process context learning. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 140–157. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_14
10. Ghattas, J., Soffer, P., Peleg, M.: Improving business process decision making based on past experience. Decis. Support Syst. **59**, 93–107 (2014)

11. Ghattas, J., Peleg, M., Soffer, P., Denekamp, Y.: Learning the context of a clinical process. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 545–556. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_53
12. Ghose, A., Koliadis, G., Chueng, A.: Process discovery from model and text artefacts. In: 2007 IEEE International Conference on Services Computing - Workshops (SCW 2007), 9–13 July 2007, Salt Lake City, Utah, USA, pp. 167–174 (2007)
13. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a K-Means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979). ISSN 00359254, 14679876
14. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR, SIGIR 1999, 15–19 August 1999, Berkeley, CA, USA, pp. 50–57 (1999)
15. Kiseleva, J.: Context mining and integration into predictive web analytics. In: 22nd International World Wide Web Conference, WWW 2013, Rio de Janeiro, Brazil, 13–17 May 2013, Companion Volume, pp. 383–388 (2013)
16. Kusner, M.J., et al.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, pp. 957–966 (2015)
17. Mani, S., et al.: Panning requirement nuggets in stream of software maintenance tickets. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, (FSE-22), Hong Kong, China, 16–22 November 2014, pp. 678–688 (2014)
18. Marcus, M., et al.: The Penn Treebank: annotating predicate argument structure. In: Proceedings of the Workshop on Human Language Technology, HLT 1994, pp. 114–119. Association for Computational Linguistics, Plainsboro (1994). ISBN 1-55860-357-3
19. Mikolov, T., et al.: Efficient estimation of word representations in vector space. In: CoRR abs/1301.3781 (2013)
20. Osiński, S., Stefanowski, J., Weiss, D.: Lingo: search results clustering algorithm based on singular value decomposition. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. AINSC, vol. 25, pp. 359–368. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39985-8_37
21. Potharaju, R., Jain, N., Nita-Rotaru, C.: Juggling the Jigsaw: towards automated problem inference from network trouble tickets. In: Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2013, Lombard, IL, USA, 2–5 April 2013, pp. 127–141 (2013)
22. Saidani, O., Nurcan, S.: Context-awareness for adequate business process modelling. In: Proceedings of the Third IEEE International Conference on Research Challenges in Information Science, RCIS 2009, Fès, Morocco, 22–24 April 2009, pp. 177–186 (2009)
23. Saidani, O., Rolland, C., Nurcan, S.: Towards a generic context model for BPM. In: 48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, 5–8 January 2015, pp. 4120–4129 (2015)
24. Shao, Q., et al.: Efficient ticket routing by resolution sequence mining. In: Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining, KDD 2008, Las Vegas, Nevada, USA, pp. 605–613 (2008). ISBN 978-1-60558-193-4

25. Sindhgatta, R., Ghose, A., Dam, H.K.: Context-aware analysis of past process executions to aid resource allocation decisions. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 575–589. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_35
26. Sindhgatta, R., Ghose, A., Dam, H.K.: Context-aware recommendation of task allocations in service systems. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 402–416. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_25
27. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. **45**(4), 427–437 (2009)
28. Teinemaa, I., Dumas, M., Maggi, F.M., Di Francescomarino, C.: Predictive business process monitoring with structured and unstructured data. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 401–417. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_23
29. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd Annual International ACM SIGIR, SIGIR 1999, Berkeley, California, USA, pp. 42–49 (1999)
30. Zhou, W., et al.: Resolution recommendation for event tickets in service management. IEEE Trans. Netw. Serv. Manag. **13**(4), 954–967 (2016). ISSN 1932–4537