Analyzing Comments in Ticket Resolution to Capture Underlying Process Interactions

Monika Gupta[1], Prerna Agarwal[1], Tarun Tater[1],
Sampath Dechu[1], Alexander Serebrenik[2]

[1]IBM Research, India
[2]Eindhoven University of Technology, The Netherlands
[mongup20, preragar, ttater24, sampath.dechu]@in.ibm.com, a.serebrenik@tue.nl

Abstract. Activities in the ticket resolution process have comments and emails associated with them. Process mining uses structured logs and does not analyze the unstructured data such as comments for process discovery. However, comments can provide additional information for discovering models of process reality and identifying improvement opportunities efficiently. To address the problem, we propose to extract topical phrases (keyphrases) from the unstructured data using an unsupervised graph-based approach. These keyphrases are then integrated into the event log to derive enriched event logs. A process model is discovered using the enriched event logs wherein keyphrases are represented as activities, thereby capturing the flow relationship with other activities and the frequency of occurrence. This provides insights that can not be obtained solely from the structured data.
To evaluate the approach, we conduct a case study on the ticket data of a large global IT company. Our approach extracts keyphrases with an average accuracy of around 80%. Henceforth, discovered process model succinctly captures underlying process interactions which allows to understand in detail the process realities and identify opportunities for improvement. In this case, for example, manager identified that having a bot to capture specific information can reduce the delays incurred while waiting for the information.

Keywords: Process Mining · Ticket Resolution · Unstructured Data

1 Introduction

A lot of structured and unstructured data is generated during the execution of business processes which gets stored in the information systems [3]. The data captures the runtime process behavior which can be analyzed to discover process reality, and support process improvement. Previous studies show that such an analysis can be based on process mining [2]. Process mining consists of mining event logs generated from business process execution supported by information systems. Every entry in the event log is an event referring to a case, activity, time stamp, and optional attributes such as actor (resource), associated cost, and duration.

Ticket resolution process also has corresponding logs captured in the ticketing system. Also some of the activities in the ticket resolution process have comments associated with them. Existing process mining techniques leverage structured event logs for discovering process model and identifying process inefficiencies [2]. However, comments can provide additional information for effective process improvement decisions. In this study, we aim at discovering the detailed process model for ticket resolution process, using the information present in the comments. The discovered model can then be used to identify the inefficiencies.

To model the detailed process, we extract the topical phrases (keyphrases) from the comments generated during the process execution, using an unsupervised graph-based approach [8]. These keyphrases are then integrated into the event log to derive enriched event logs. A process model is discovered using the enriched event logs wherein keyphrases are represented as activities, thereby capturing the flow relationship with other activities and the frequency of occurrence. This provides insights that could not be obtained solely from the structured data (i.e., activities), and these insights could be used to perform the ticket resolution process more efficiently.

To evaluate the approach, we conduct a case study on the ticket data of a large global IT company. We first extract the keyphrases from the comments associated with the ticket activities with an average accuracy of around 80%. This enables us to succinctly capture the additional information about the activities influencing the ticket resolution process and often causing delays, such as extra information required, priority, and severity. The model allows the managers to understand in detail the process realities and identify opportunities for improvement. In this case, for example, the manager identifies that having a bot to capture the information or adding a mandatory field in the initial ticket template, so as to reduce the delays incurred while waiting for information, can reduce the time (he subsequently had his team implement the bot).

2 Usefulness of Information in Comments

A lot of rich information is present in the comments generated during the process execution, which needs to be integrated into the discovered process model for in-depth process understanding. The in-depth unstructured data-driven (e.g., comments) insights help effectively identify the inefficiencies and make informed process improvement decisions.

Figure 1 shows the snapshot of a real example of a discovered process model for the ticket resolution process of a large global IT company. As part of the ticket resolution process, an analyst (person responsible for servicing the ticket) can ask the user to provide additional information by writing a comment, which gets captured in the information system as an event, *Need Info - Client*. An analyst can ask for different information, such as error messages and operating system, which gets recorded in the comments. A process model is discovered using only a structured event log where activity, *Need Info - Client* is not further decomposed (refer to Fig. 1, left panel). We extracted the keyphrases from all

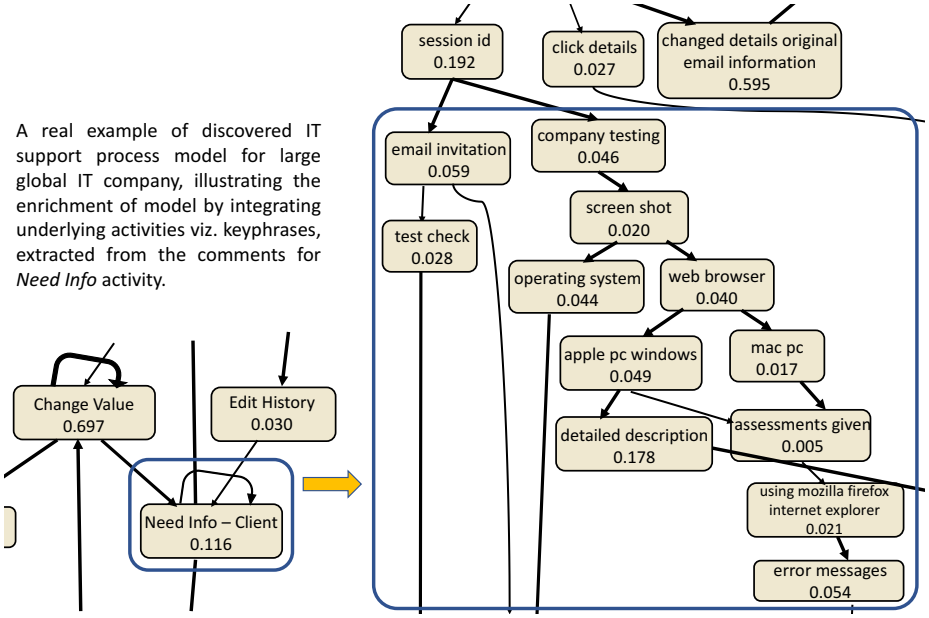

Fig. 1. A real example that compares process model from structured logs against the model capturing underlying activities as per the keyphrases extracted from comments.Here, node corresponds to the activity and edge represents the flow relationship

the comments corresponding to the activity *Need Info - Client*, using an unsupervised graph-based keyphrase extraction approach. The extracted keyphrases represented information typically asked by analysts, using which an enriched event log was derived where the activity, *Need Info - Client*, was mapped to relevant keyphrases on the basis of the comment. The process model discovered using the enriched event log (refer to Fig. 1 - right panel) presented the underlying interactions of the process with activities such as email invitation, screenshot, web browser, operating system and error message, each corresponding to an information asked by the analysts (highlighted in Figure 1, right panel). This allowed discovering the in-depth reality, which cannot be observed from Figure 1, left panel. Thus, the following informed improvement decisions could be made to mitigate the delays incurred while waiting for information from the user:

- As *detailed description* (relative frequency is 0:178) is asked more often, the IT company should deploy a system such that a user can be preempted at the time of ticket submission to provide the same upfront [6]. Advantage of having a preemptive model is that a user is preempted selectively based on the ticket requirement as learnt from the historical data [6].
- A user is typically asked to provide an error *screenshot* after asking for *company testing*; therefore, analysts can be preempted to ask both the screenshot and the company testing at the same time.

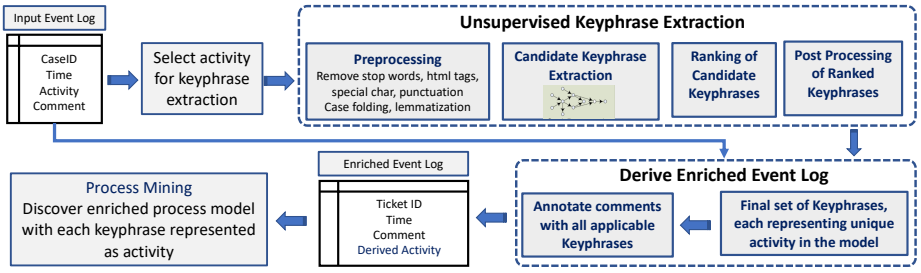

Fig. 2. Proposed approach to discover the underlying process using comments.

{ As information about the *operating system* and *web browser* is asked in the comments, a bot can be designed to automatically detect this information at the time of ticket submission.

This example highlights the potential of our approach for effective process improvement, by deriving keyphrases from comments corresponding to activities and representing them as part of the discovered process model.

3 Proposed Approach

To achieve the objective of integrating knowledge captured in unstructured data, namely comments into the discovered process model, we presented an approach consisting of multiple steps, as shown in Figure 2. First, a process analyst can select the activity for which the comments should be analyzed for in-depth process understanding. Performing this selection is important because a granular view of every activity can make the discovered process model look like spaghetti. Also, it needs to be decided on the basis of analysis to be performed, such that activities not captured in the structured logs are inferred from the comments. Thereafter, the comments corresponding to selected activities are preprocessed and used for candidate keyphrase extraction in an unsupervised manner. Extracted candidates are ranked and processed to select most relevant keyphrases which in turn are used to annotate the comments, thus, deriving an enriched event log. Finally, using the enriched event log, the process model capturing the flow relationship and frequency is discovered.

3.1 Unsupervised Keyphrase Extraction

Keyphrase extraction aims at automatic selection of important and topical phrases from the body of documents [16]. Broadly, keyphrases are extracted using two approaches: supervised and unsupervised. In the supervised approach, a model is trained to classify a candidate keyphrase, requiring human-labeled keyphrases as training data. It is impractical to label training data (in this case, process

Algorithm 1: Unsupervised Keyphrase Extraction

Input: Initial Event Log EL (CaseID, timestamp, Activity, Comments)
Output: Keyphrase List K for Selected Activity A
Variables: *Candidate keyphrase list* : $M \leftarrow []$; $lookup_{table} \leftarrow []$; $score \leftarrow []$, *Global Graph* : $G \leftarrow []$
$comments \leftarrow SelectActivity(EL; A)$
$sentences \leftarrow Preprocess(comments)$
$G \leftarrow GlobalGraph(sentences)$
for *sentence* in *sentences* **do**
    $G_c \leftarrow Graph(sentence)$
    $G_u \leftarrow G - edges(G_c)$
    $phrases \leftarrow G_u \cap G_c$
    $M \leftarrow M \cup phrases$
    $lookup_{table}[sentence] \leftarrow phrases$
for m in M **do**
    $pf = PhraseFrequency(phrase)$
    $cf = CommentFrequency(phrase)$
    $score[m] \leftarrow pf \times -log(1 - cf)$
$score_n \leftarrow Sort(score; n)$
$K \leftarrow PostProcess(score_n)$
return K

execution comments), given the effort required for manual labeling. Thus, we focused on the unsupervised approach for our purpose. Unsupervised approaches can be grouped as follows [9]: graph-based ranking, topic-based clustering, simultaneous learning, and language modeling. Graph-based ranking methods are state-of-the-art methods [12], based on the idea of building a graph from the input document. Nodes in the graph are ranked based on their importance to select the most relevant keyphrases. Therefore, we used CorePhrase [8], a graph-based algorithm for topic discovery, that is, keyphrase extraction from multidocument sets based on frequently and significantly shared phrases between documents. The algorithm is domain independent and thus suitable for our purpose with some adaptations. The algorithm first identifies a list of candidate keyphrases from the set of documents and then selects top n-ranked keyphrases for the output by using a ranking criterion. The ranked keyphrases are then postprocessed to be adapted according to the domain.

Preprocessing of Comments: The comments from event logs are preprocessed (*Preprocess* in Algorithm 1) including stemming, case folding, removal of HTML tags, stop words, and special characters. Further, we created a set of unique sentences across all the comments to improve the scalability of the approach. The sentences in the comment were demarcated by a period. A unique set was created such that if a sentence was present in multiple comments then it was considered only once. This significantly reduced the number of sentences to be processed in further steps because the sentences were repeated across various comments (emails). This preprocessing did not affect the final set of extracted keyphrases

because the keyphrase for a comment was a set of keyphrases extracted for its constituting sentences.

Candidate Keyphrase Extraction: To extract candidate keyphrases, the algorithm compares every pair of sentences to identify the common phrases. If there are n sentences in the corpus, comparing every pair is inherently $O(n^2)$. However, as highlighted in the CorePhrase [8] algorithm, using a data structure called the *Document Index Graph* (DIG), the comparison can be done in approximately linear time [7]. For our purpose, the DIG stored a cumulative graph representing the entire set of unique sentences (e.g., *GlobalGraph* function in Algorithm 1). When the keyphrase for a sentence has to be extracted, its subgraph is matched (by performing graph intersection) with the cumulative graph (viz. Global Graph) except for the sentence (Line 9 and 10 in Algorithm 1). It gives a list of matching phrases between the sentence and the rest of the sentences. This process generates matching phrases between every pair of sentences in near-linear time with varying length phrases. A master list M is maintained that contains unique matched phrases for all sentences that will be used as a list of candidate keyphrases. A *lookuptable* is also maintained that contains all sentences and the corresponding matching phrases (Line 12 in Algorithm 1) irrespective of whether the phrase gets selected after ranking or not (in the follow up steps of algorithm). Therefore, if a comment remains unannotated as phrases for none of its constituting sentences are in the selected set then the *lookuptable* is referred for annotation (discussed in Section 3.2).

Ranking of Candidate Keyphrases: Quantitative phrase metrics are used to calculate the *score* representing the quality of the extracted candidate keyphrase. The *score* is computed as $pf \quad \log(1 \quad cf)$, where cf is the comment frequency and pf is the average phrase frequency. Inspired by term frequency-inverse document frequency (TF-IDF) [8], the *score* rewards the phrases that appear in more documents (high cf) rather than penalizing them. For a phrase p, the comment frequency $cf(p)$ is the number of comments in which p appears, normalized by the total number of comments: $\frac{|\text{comments containing } p|}{|\text{all comments}|}$.

The average phrase frequency pf is the average number of times p appears in one comment, normalized by the length of the comment in words: $\arg\,\text{avg}[\frac{|\text{occurrences of } p|}{|\text{words in comment}|}]$

Postprocessing of Ranked Keyphrases: Selected top-ranked keyphrases contain some nonrelevant phrases, that is, phrases that fall out of the domain but still are common in most of the comments. Examples of such phrases are *thank you for contact,* and *contact helpdesk*. Such keyphrases are removed by creating a common domain dictionary that contains unwanted words to be removed from the keyphrases (function *PostProcess* in Algorithm 1). This domain dictionary thus postprocesses the keyphrases to obtain the final set of keyphrases. The following is an example of how postprocessing is applied to the keyphrases:

Extracted Phrase: Please tell unemployment benefits

Postprocessed Phrase: Unemployment benefits

Here words *please* and *tell* belong to an unwanted dictionary, as they are not relevant in keyphrases and thus removed. Also, if a keyphrase is a proper sub-

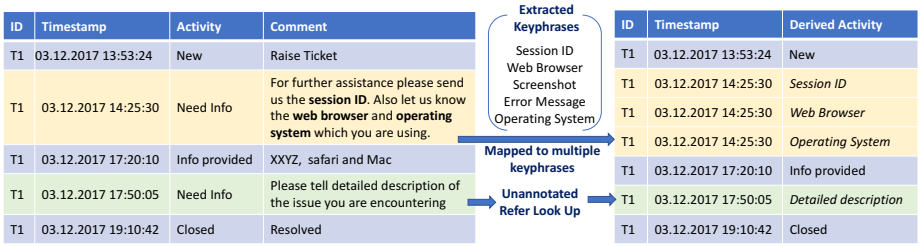

Fig. 3. Example to illustrate comment annotation for deriving an enriched event log.

string of any other selected keyphrase, then it is removed to resolve the spurious multilabel assignment.

3.2 Annotating Comments with Keyphrases to Derive Enriched Event Log

The initial event log, *EL*, contains activities and corresponding comments. Once the keyphrases are extracted, each comment in the dataset is analyzed to determine whether one of the keyphrases matches with it. To make the matching consistent, we performed the same preprocessing as mentioned earlier. If a comment contained a keyphrase, it was annotated with the corresponding keyphrase. As we only extracted the top *n* most relevant keyphrases, some comments might be annotated by one or more keyphrases, while other comments might not be annotated at all.

To tackle the latter cases, we referred to the *lookup* table and retrieved the keyphrases for that comment. These keyphrases were added as labels to the comment. Therefore, maintaining the *lookup* table helped in assigning labels to otherwise unannotated comments.

Figure 3 depicts a real example of an event log where the first comment for the activity, *Need Info*, is mapped to three keyphrases. However, the second comment is not annotated with any of the selected top-ranked keyphrases and, therefore, is mapped to *detailed description* after referring to the lookup table. The enriched event log is generated with a new attribute, *derived activity*, replacing *activity* and *comment*, and representing the extracted keyphrases. To capture the ordering relationship of activities in the enriched event log, keyphrase(s) derived for a particular comment are assigned the timestamp of original comment.

3.3 Evaluating Keyphrase Extraction and Annotation

There are two aspects for evaluation, that is, the identification of informative keyphrases and correct annotation of comments with keyphrases. We evaluated the quality of extracted keyphrases manually, that is, checked whether they conveyed the required information. Further, we needed to evaluate the annotation

of comments. As discussed in Section 3.2, a comment can be mapped to multiple keyphrases. Therefore, we used multilabel evaluation metrics that could be *example-based* or *label-based* [15]. We chose the *example-based evaluation metrics* that could capture the average difference between the predicted labels and the actual labels for each test example, and then averaged over all examples in the test set. Thus, unlike *label-based evaluation metrics*, these metrics took into account the correlations among different classes [17], which is of interest here. To evaluate the quality of the classification (here, annotation of comments) into classes (here, keyphrases), we used the following set of metrics, thus capturing the partial correctness [15]:

Let T be a multilabel dataset consisting of n multilabel examples $(x_i; Y_i); 1 \; i \; n; (x_i \; 2 \; X; Y_i \; 2 \; Y = f0; 1g^k)$, with a labelset $L; jLj = k$. Let h be a multilabel classifier (here, annotator in Section 3.2) and $Z_i = h(x_i) = f0; 1g^k$ be the set of label memberships predicted by h for the data point (i.e., comment) x_i.

$$Accuracy; A = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (1) \qquad Recall; R = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \quad (3)$$

$$Precision; P = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \quad (2) \qquad F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \quad (4)$$

$$HammingLoss; HL = \frac{1}{kn}\sum_{i=1}^{n}\sum_{l=1}^{k}[I(l \in Z_i \wedge l \notin Y_i) + I(l \notin Z_i \wedge l \in Y_i)]; \quad (5)$$

where I is the indicator function which is equal to 1 if $Z_i = Y_i$, else 0. Since HL is a loss function, it should be minimum for better performance.

4 Case Study: IT Support Ticket Resolution Process

To illustrate the value of integrating knowledge from unstructured data into the discovered process model, we performed a case study on the IT support process data of a large global IT company. The dataset represented interactions between the users and the support team (analysts), and thus, comments were present with relevant activities. While the IT support process was continuously monitored by the process analyst, the unstructured data, for example, comments, were not taken into account.

Data extracted from the organization's ticket system includes the required information about a ticket starting from the time of ticket submission until it is closed. Downloaded data consists of 2620 tickets with 15; 819 events in total. We observed from the dataset that two activities (out of 19), *Change Value* and *Need Info*, existed where analysts wrote comments. The number of events with the activities *Change Value* and *Need Info* was 4036 and 280, respectively.

In *Change Value*, changes in the ticket attributes were captured by a descriptive comment as shown below with an anonymized example (for confidentiality):
Changed Category from "" to "Y". Changed Sub-Category from "" to "test reset". Changed Severity from "" to "Sev 4". Changed Summary from "" to "reset the test". Changed Support Contract from None to Contract1.

An analyst asks information from the user (here, customer) by writing a comment, which is captured as activity *Need Info* in the database. For example,

Table 1. Experimental Results where **K**: Total extracted keyphrases in final set, **L**: average number of labels for each comment, **P**: Precision, **R**: Recall, **A**: Accuracy, **F1**: F1 measure, and **HL**: Hamming Loss.

Data	K	L	P	R	A	F1	HL
Change Value	33	3.63	84.74 %	81.27%	80.26%	82.12%	8.15%
Need Info	16	4.28	84.71%	89.97%	80.21%	86.57%	6.21%

Dear ABC, Thank you for contacting the Support Center. In order to assist you more e ectively we ask that you please provide the following information: Are you using a Macintosh Computer (Apple) or a PC (Windows)?: What web browser are you using (Internet Explorer, Mozilla Firefox, Safari, Google Chrome)?: Website you were directed to access: Session ID/login info: Detailed description of the issue you are encountering: Screen shot of error message: Thank you in advance!

To enrich the event logs, we performed keyphrase extraction for these two activities. This allowed us to precisely capture what values were changed and what information was typically asked by the analysts, in coherence with the complete process flow. IT support data were not made publicly available for confidentiality reasons; however, examples and results were included for an explanation.

4.1 Unsupervised Keyphrase Extraction and Enriched Event Log Derivation

Comments for the selected activities, namely, *Change Value* and *Need Info*, were preprocessed. All the preprocessing steps as discussed in Section 3.1 were performed and the final set of preprocessed unique sentences was used for candidate keyphrase extraction. As per Algorithm 1, a set of candidate keyphrases is extracted for both the activity sets. Extracted keyphrases were ranked using the scoring function. We selected top 50 keyphrases from the ranked list for *Change Value* and *Need Info* respectively which were postprocessed as per the data properties. These postprocessed set of final keyphrases were used for annotating the respective comments as discussed in Section 3.2, thus generating enriched event logs. Some comments for both the activities were left unannotated for which we referred to the lookup table, and hence assigned keyphrase. Effectively, the total number of unique keyphrases (K) in the resulting enriched event log was 33 and 16 for *Change Value* and *Need Info*, respectively (refer to Table 1). The average number of keyphrases, $L \simeq 4$ for *Change Value* and *Need Info* indicated that multiple important topics were present in a comment, that is, multiple ticket attributes were changed and multiple information was asked in the same comment.

4.2 Visualizing and Analyzing an Enriched Process Model

Enriched event logs are used for process discovery using ProM. We presented and compared *Need Info* for the original and enriched process model of IT support process in Figure 1. Here, we presented the process model snapshot for IT support process, specifically highlighting the *Change Value* derived activities (refer to Fig. 4).

As shown in Figure 4, the individual activity *Change Value* was replaced with more specific activities such as *changed category*, *changed company name*, and specific instances of *changed summary*, each corresponding to extracted keyphrases.

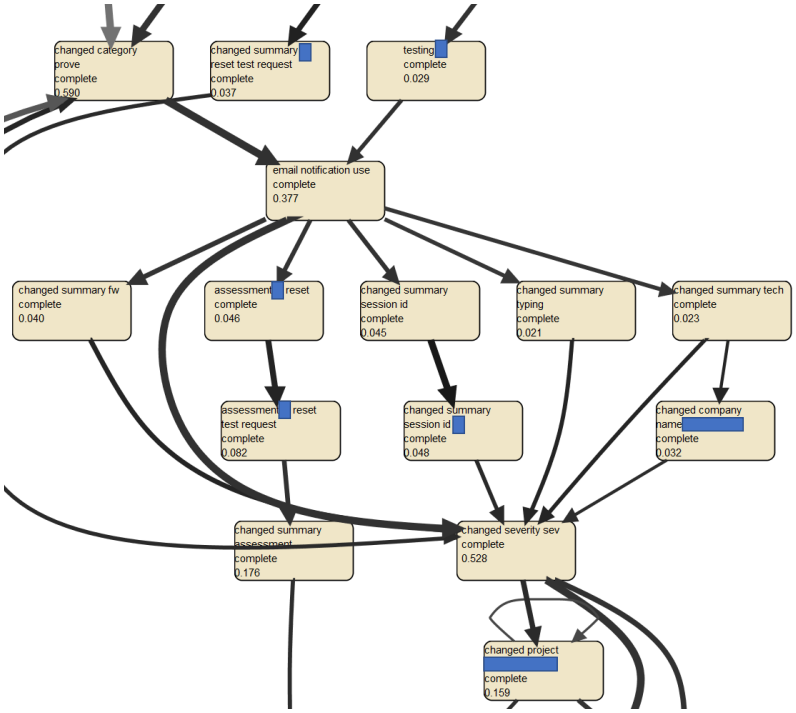

Fig. 4. A real example of discovered IT support process model for a large global IT company, illustrating enrichment of model by integrating keyphrases extracted from the comments for the *Change Value* activity. Some words are masked for confidentiality.

As the information captured in comments is integrated into the model, it is possible to derive insights as follows:

- The category was changed for a high percentage of tickets (as the relative frequency was 0.615), highlighting the need for a system to automatically assign a category based on the content of the initial ticket, thus optimizing the time spent for category assignment.
- The summary was changed for various tickets, and some of the most frequent instances were captured as keyphrases in our approach. Hence, we observed many states with a changed summary (suffixed with specific terms), although they all indicated some change in the summary.
- The company name was changed for a small percentage of tickets, which usually happened after the summary was changed in a specific manner. Therefore, the analysts could be preempted in those instances to change the company name in parallel with the summary, thus eliminating the delay.

We showed the process model discovered using structured logs and the enriched discovered process model side-by-side to the manager. He acknowledged that the enriched model helped in making effective process improvement decisions. One of the actionable insights he decided to take forward was to design a robotic process automation solution

for the automatic category assignment to a ticket. This could not have been possible without integrating knowledge from the comments into the discovered process model.

4.3 Evaluation of Keyphrase Extraction and Annotation

Establish the Ground Truth: To evaluate keyphrase extraction (as discussed in Section 3.3), we established a ground truth for comments corresponding to the selected activities. Thus, we needed to first manually identify a set of keyphrases for comments corresponding to selected activities and annotate comments with the same. First, we identified the ticket attributes typically changed (as part of *Change Value* activity) and information typically asked by the analysts (as part of *Need Info* activity) on the basis of managers' domain knowledge and manual inspection of the comments. Manual inspection was performed by two authors for a disjoint set of comments (random sample of around 25% comments for each) to identify lists of changed attributes and asked information, respectively. Lists by both of them were compared to create a consolidated list. Both the authors used different terms to represent the same information, which were made consistent. Both of them identified the same list with a few exceptions (i.e., rarely occurring content), which were resolved. The final list was verified with the manager. Each item in the list was considered as a keyphrase for the data set.

Now that the list of ground truth keyphrases was identified, to establish the ground truth, comments were annotated with keyphrases using a keyword-based dictionary [14]. A list of keywords corresponding to each keyphrase was prepared iteratively, for example, the keyword "summary" for the keyphrase "changed summary". If the comment contained keywords, it was annotated with the corresponding keyphrase. Thereafter, authors of the paper manually investigated the disjoint set of randomly selected comments to distill the wrongly annotated comments. This process was repeated two to three times until very few/no updates were made in the set of keywords.

As an example, ground truth keyphrases for *Change Value* and *Need Info* were {Changed Category, Changed Sub-Category, Changed Severity, Changed Summary, Changed Support Contract}, and {mac pc, web browser, website directed, session id, detailed description, screen shot}, respectively.

Analysis of Results: Automatically extracted keyphrases can be structurally different from the human-identified ones, although both represent the same topical information. To avoid spurious penalty on the metrics, we took this into account by manually creating a mapping between the two. Table 1 shows that the proposed approach performed with an accuracy of around 80% and had a low hamming loss. High F_1 measure ensured that the approach achieved a good balance between precision and recall. Hence, the proposed approach efficiently derived an enriched event log for the given data set.

5 Related Work

Unstructured text can be analysed to accrue benefits for process understanding and improvement at different levels however, sort with various challenges [1]. Some of the example use cases are discovering process models from natural language text [5][4] and searching textual as well as model-based process descriptions [11]. Automatic keyphrase extraction is used for a wide range of natural language processing and information retrieval tasks such as text clustering and summarization [13][18], text categorization [10], and interactive query refinement [16]. However, the application of keyphrase extraction to process model enrichment is not explored which is the focus of this work.

6 Conclusion

Process mining techniques are activity focused and do not consider comments generated during process execution. We presented a multistep approach to integrate hidden knowledge captured in unstructured text, namely comments, into the discovered process model. This was achieved by extracting keyphrases in an unsupervised manner and using them to annotate the comments thus deriving enriched event logs. We observed that the keyphrase extraction and annotation approach performed with an average accuracy of around 80% across different activities (*NeedInfo* and *ChangeValue*) for a data set. Further, we discovered the process model using a derived enriched event log and highlighted the value of enhanced process model in deriving actionable insights.

Our future plan is to extend the keyphrase extraction approach, such that the semantics is leveraged, and compare it with the proposed approach to analyze whether the insights derived from the discovered business processes are further enhanced.

References

1. Van der Aa, H., Carmona Vargas, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: COLING 2018: The 27th International Conference on Computational Linguistics: Proceedings of the Conference: August 20-26, 2018 Santa Fe, New Mexico, USA. pp. 2791–2801. Association for Computational Linguistics (2018)
2. van der Aalst, W.M.P.: Process mining - discovery, conformance and enhancement of business processes (2011)
3. van der Aalst, W.M.P., La Rosa, M., Santoro, F.M.: Business process management (2016)
4. Chen, Y., Ding, Z., Sun, H.: Pewp: Process extraction based on word position in documents. In: Ninth International Conference on Digital Information Management (ICDIM 2014). pp. 135–140. IEEE (2014)
5. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: International Conference on Advanced Information Systems Engineering. pp. 482–496. Springer (2011)
6. Gupta, M., Asadullah, A., Padmanabhuni, S., Serebrenik, A.: Reducing user input requests to improve it support ticket resolution process. Empirical Software Engineering pp. 1–40 (2017)
7. Hammouda, K.M., Kamel, M.S.: Efficient phrase-based document indexing for web document clustering. IEEE TKDE **16**(10), 1279–1296 (Oct 2004). https://doi.org/10.1109/TKDE.2004.58
8. Hammouda, K.M., Matute, D.N., Kamel, M.S.: Corephrase: Keyphrase extraction for document clustering. In: Machine Learning and Data Mining in Pattern Recognition. pp. 265–274 (2005)
9. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: A survey of the state of the art. In: ACL. vol. 1, pp. 1262–1273 (2014)
10. Hulth, A., Megyesi, B.B.: A study on automatically extracted keywords in text categorization. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. pp. 537–544. Association for Computational Linguistics (2006)
11. Leopold, H., van der Aa, H., Pittke, F., Raffel, M., Mendling, J., Reijers, H.A.: Searching textual and model-based process descriptions based on a unified data format. Software & Systems Modeling **18**(2), 1179–1194 (2019)
12. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: EMNLP. pp. 366–376 (2010)
13. Manning, C.D., Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT press (1999)
14. Pletea, D., Vasilescu, B., Serebrenik, A.: Security and emotion: sentiment analysis of security discussions on github. In: MSR. pp. 348–351 (2014)
15. Sorower, M.S.: A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis **18** (2010)
16. Turney, P.D.: Learning algorithms for keyphrase extraction. Information retrieval **2**(4), 303–336 (2000)
17. Zhang, M.L., Zhang, K.: Multi-label learning by exploiting label dependency. In: KDD. pp. 999–1008 (2010)
18. Zhang, Y., Zincir-Heywood, N., Milios, E.: World wide web site summarization. Web Intelligence and Agent Systems: An International Journal **2**(1), 39–53 (2004)