Event Log Construction from Customer Service Conversations Using Natural Language Inference

Christoph Kecht*[†][§], Andreas Egger[†][¶], Wolfgang Kratsch[‡][¶], Maximilian Röglinger[§][¶]
*Technical University of Munich, Germany [†]University of Augsburg, Germany
[‡]University of Applied Sciences Augsburg, Germany [§]University of Bayreuth, Germany
[¶]FIM Research Center, Project Group Business and Information Systems Engineering of the Fraunhofer FIT, Germany
Email: christoph.kecht@tum.de, andreas.egger@fim-rc.de, wolfgang.kratsch@fim-rc.de, maximilian.roeglinger@fim-rc.de

***Abstract*—A fundamental requirement for the successful application of process mining are event logs of high data quality that can be constructed from structured data stored in organizations' core information systems. However, a substantial amount of data is processed outside these core systems, particularly in organizations doing consumer business with many customer interactions per day, which generate high amounts of unstructured text data. Although Natural Language Processing (NLP) and machine learning enable the exploitation of text data, these approaches remain challenging due to the required high amount of labeled training data. Recent advances in NLP mitigate this issue by providing pre-trained and ready-to-use language models for various tasks such as Natural Language Inference (NLI). In this paper, we develop an approach that utilizes NLI to derive topics and process activities from customer service conversations and that represents them in a standardized XES event log. To this end, we compute the probability that a sentence describing the topic or the process activity can be inferred from the customer's inquiry or the agent's response using NLI. We evaluate our approach utilizing an existing corpus of more than 500,000 customer service conversations of three companies on Twitter. The results show that NLI helps construct event logs of high accuracy for process mining purposes, as our successful application of three different process discovery algorithms confirms.**

***Index Terms*—Process Mining, Event Log Construction, Machine Learning, Natural Language Processing**

I. Introduction

Process mining aims to discover, monitor, and enhance business processes by processing event logs [1]. These logs are typically retrieved from information systems and constructed using various techniques [2]–[4]. Most existing approaches currently focus on constructing event logs from structured data stored in organizations' core information systems. However, a substantial amount of data (e. g., phone calls, emails [5], [6], and contracts) is processed outside those systems, which is one reason why business processes can deviate from expected behavior [7]–[9]. Frequently, these data are represented in unstructured formats [5], [6], [10] and offer untapped potential for process mining. For example, many organizations in consumer businesses record their customer interactions, among others, for legal reasons. At a high number of customer interactions using multiple communication channels, manual monitoring and enhancement of customer-centric processes seem infeasible. Nevertheless, organizations doing consumer businesses need to monitor the execution of their customer service since the customers' satisfaction with the service is of crucial importance for these organizations [11]. To this end, process mining provides a viable solution by utilizing event logs constructed from recordings of customer interactions [10].

To construct standardized event logs from textual data, related approaches already demonstrate the successful application of algorithms in the fields of Natural Language Processing (NLP) and machine learning [6], [10]. However, these event logs need to be of high data quality for a practical application of process mining [12]. Machine learning approaches typically require excessive training to achieve high data quality, and thus, a vast amount of labeled training data. However, recent advancements in the field of NLP research mitigated this issue, among others, by providing pre-trained language models for a multitude of NLP tasks [13]. One particular NLP task is Natural Language Inference (NLI). Given two sentences, referred to as the hypothesis and the premise, NLI determines whether the hypothesis can be inferred from the premise [14]. For example, the hypothesis "This sentence is an apology" can be inferred from the premise "We are sorry for the unpleasant experience". Combined with a pre-trained language model, Yin et al. [15] show the applicability of NLI for topic, emotion, and situation detection.

In this paper, we aim to utilize NLI to derive topics and process activities from customer service conversations that follow a question-answer pattern and represent them in an event log, assuming the presence of essential event log attributes (case ID, event ID, and timestamp) in the respective dataset. To this end, we compute the probability that a sentence describing the topic or the process activity can be inferred from the customer's inquiry or the agent's response using NLI. By embedding this concept into a reusable workflow, we develop an approach to represent customer service conversations as standardized IEEE XES event logs [16], which can then be imported and used by process mining applications, such as ProM [17] and Disco [18]. To evaluate our approach, we utilize an existing corpus of Twitter conversations [19] of AmazonHelp, AppleSupport, and SpotifyCares that comprises over 500,000 Tweets in total. As a benchmark, we chose a simple keyword-based approach that assigns a Tweet to a particular topic or process activity if it contains a corresponding keyword. The results show that using NLI with a hypothesis that precisely describes the topic or process activity of interest, almost all Tweets in representative samples of the dataset

could be assigned to the respective topics and activities with a Matthews correlation coefficient (MCC) greater than 0.72 after cross-validation without requiring extensive preprocessing. Furthermore, the constructed event logs emerged as suitable for process discovery, as our successful application of three different process discovery algorithms [20]–[22] confirms.

The remainder of this paper is structured as follows. In Section II, we introduce relevant concepts related to process mining and NLP. In Section III, we present our approach step by step and show how we determine the best NLI hypothesis and binary classification threshold for each topic and activity. We then evaluate our approach in Section IV, where we describe the dataset, our evaluation metrics, the results, and the process discovery application to our resulting event log. In Section V, we discuss our findings, followed by an overview of limitations and opportunities for future work in Section VI.

II. Theoretical Background and Related Work

A. Event Log Construction for Process Mining

Van der Aalst [1] divides process mining into three types: discovery, conformance checking, and enhancement. This distinction has been enhanced to ten use cases by the refined process mining framework that highlights the importance of providing operational support in addition to analyses on historical data [23]. All use cases have in common that event logs of high quality are a fundamental requirement to perform these process mining activities [12]. In practice, structured data stored in relational databases typically fulfill the prerequisite of high data quality. Therefore, existing approaches leverage modifications to the data to construct event logs for process mining [2]–[4]. However, applying process mining on these systems' databases does not unleash process mining's full potential since these systems handle structured data that result from already specified process models. Instead, the reasons why business processes deviate, fail, and need to be improved, are, among others, systems not providing required functionalities [9], exceptions [8], and workarounds [7]. Typically, the involved actors handle these issues outside the core systems and thereby create and process unstructured data.

Consequently, dealing with unstructured data, such as customer service conversations, is an ongoing issue in process mining research and has been explored using various approaches. One of the first attempts by van der Aalst and Nikolov [5] transforms tagged email messages to an email log and applies a process discovery algorithm to construct a process model. However, since tagging data requires high manual efforts in practice, there is a need for automatic approaches. To this end, Banziger et al. [10] present a framework to automatically discover events and activities from CRM systems using Latent Dirichlet Allocation (LDA). Although their approach yields promising results, LDA has inherent limitations when applied to short texts [24], such as questions and answers from conversations. Furthermore, it does not consider the structure of the analyzed sentences, and thus, its applicability in practice is limited. To overcome this issue, Jlailaty et al. [6] suggest representing texts using the word2vec representation in their approach for extracting activities from email logs. This representation closely depicts words to each other if they occur in a similar context [25] and commonly serves as input for most text-based machine learning models.

Other related literature further investigates event log construction using machine learning. Rebmann et al. [26] exploit image data to resolve ambiguities in the sensor data to construct an event log based on sensor data. Using a convolutional neural network, the authors recognize the scenery in which activities are executed. Similarly, Knoch et al. [27] use deep neural networks to extract assembly workers' movements. These movements, represented as trajectories, are subsequently clustered to identify work steps in the assembly process.

These advances in machine learning research let us conclude that there is untapped potential for constructing event logs of high accuracy from textual conversations. Thus, we further elaborate on approaches for text classification in the following.

B. Natural Language Inference

Due to the recent advancements in NLP, business process management research has been investigating NLP's capabilities for various purposes. For example, Friedrich et al. [28] developed an approach to automatically extract process models in Business Process Model and Notation (BPMN) by processing their textual descriptions, which generated 77% of the models correctly. Leopold et al. [29] propose another approach to automatically analyze textual process descriptions. In their work, the authors identify robotic process automation tasks and classify them as automated, manual, or as an interaction of a human with an information system. Concerning process mining, Baier et al. [30] show how to map events to activities of a given process model, thereby leveraging NLP for conformance checking.

The aforementioned approaches have in common that the textual descriptions require a not negligible amount of preprocessing before the actual NLP approach can be applied. However, the most recent developments in the field of NLP research resolve this limitation, for example, the Python library "transformers" [31], which internally applies pre-trained language models (e. g., Facebook's BART model [13]) for various NLP tasks, such as NLI. NLI is an NLP task that is suitable for text classification purposes. Given two sentences, referred to as the premise and the hypothesis, NLI determines whether the hypothesis can be inferred from the premise [14]. Combined with a pre-trained language model, Yin et al. [15] show the applicability of NLI on the examples of topic, emotion, and situation detection. The advantage of reusing pre-trained language models is that it does not require training a classifier, which typically involves vast amounts of labeled data. Consequently, this idea is referred to as Zero-shot classification [15]. Inspired by these results and the straightforward reusability of the approach implemented in the Python library "transformers" [31], we aim to leverage this approach to derive topics and activities from customer service conversations that follow a question-answer pattern to construct event logs suitable for process mining.

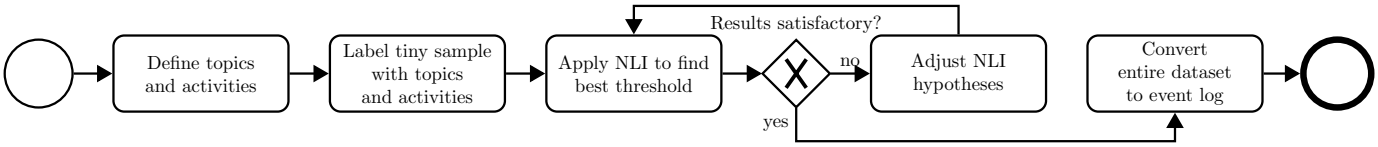

Fig. 1. Overview of the Proposed Approach

TABLE I
PROBABILITY COMPUTATION OF CUSTOMERS' QUESTIONS INQUIRING ABOUT THE DIFFERENT TOPICS

ID	This example is update	This example is battery	This example is refund	...
0	0.87	0.40	0.11	...
1	0.35	0.64	0.96	...
2	0.54	0.33	0.27	...
3	0.07	0.95	0.49	...
⋮	⋮	⋮	⋮	⋮

III. APPROACH

Fig. 1 provides an overview of our approach for constructing event logs from customer service conversations. Our approach automatically derives topics and activities using NLI and converts other essential event log attributes (case ID, event ID, and timestamp). Since customer service conversations follow a typical question-answer pattern, we map the customer's inquiry to one or many topics and the service agent's answer to one or more activities, which we both represent in the resulting event log in the standardized `case:concept:name` attribute. For a later distinction between the customers' inquiries and the agents' responses, the approach accordingly populates the standardized `org:resource` attribute in the event log.

As Fig. 1 outlines, the utilization of the approach comprises four mandatory and one optional step. First, the domain-specific topics categorizing the customer's inquiry as well as the activities describing the service agent's response have to be defined. The topics can include, for example, issues related to the delivery of an order, a particular type of product, or the customer's account. In contrast, examples for the activities include requesting the customer number, apologizing for inconveniences, or asking for specific details.

In the second step, a tiny sample of the customers' inquiries and the service agents' responses are manually labeled using a binary encoding with the true topics and activities, respectively. Since an inquiry can comprise multiple topics and a response can comprise multiple activities, a message can be assigned to more than one category. A sample size of 100 pairs of customers' inquiries and agents' responses is sufficient to achieve a suitable accuracy when automatically assigning the remaining conversations later on due to the high accuracy of pre-trained language models (as discussed in Section II-B and as our results in Section IV-D indicate).

Afterward, the NLI algorithm is applied in the third step. As explained in Section II-B, NLI determines whether the hypothesis can be inferred from the premise [14]. In our approach, the customer's inquiry or the agent's response is the premise, whereas the hypothesis is a sentence that describes the topic of the inquiry or the activity in the response. In this step, we use the default hypothesis "This example is [topic/activity]" of the Python library "transformers" [31]. This library internally calls Facebook's BART model [13] and yields a probability for each combination of premise and hypothesis, as Table I shows. However, since the assignment of inquiries and responses to topics and process activities is a binary decision, we need to determine a robust decision threshold to either assign or not assign the inquiries and responses. To this end, we apply the cross-validation procedure illustrated in Table II. We implemented this procedure using the Python machine learning library "scikit-learn" [32]. For each combination of premise and hypothesis, the respective manually assigned labels from the second step and the probability computed previously are split into five disjoint lists. In each fold, four of the lists serve as the training set, whereas the remaining list serves as the test set and, thus, is used for independent validation. The folds are stratified, i. e., the original list's distribution is maintained across the individual lists. Table II shows the five test sets of the five-fold cross-validation procedure sorted by the probability in descending order. If a label is present less than five times, the number of folds can be reduced accordingly, or the label can be considered irrelevant, and thus, can be dropped. To determine the optimal decision threshold, for each candidate in the set $\{0.70, \ldots, 0.97, 0.980, \ldots, 0.999\}$, we assign all items in each fold's test set to the topic or action if the item's probability is greater or equal than the candidate. The optimal decision threshold for each topic or activity is the candidate that achieves the highest MCC across all folds, as depicted with the dividing line in Table II. We chose the MCC as a metric since it takes into account true and false positives and true and false negatives (i. e., all four dimensions of the binary classification confusion matrix), and neither depends on which class is the positive class (compared to the F1 score) nor is misleading on imbalanced datasets (compared to the accuracy measure) [33], [34].

The optional fourth step involves defining further NLI hypotheses that describe the defined topics and activities if the MCCs obtained in the previous step are not satisfactory. This step is optional since, in some cases, the default hypothesis achieves reliable results. However, other candidates, such as "The sentence is about [topic/activity]", "The customer asks about [topic]", and "Please provide/send ..." can lead to significant improvements for some topics and activities. The decision which results are satisfactory depends on the specific use case and remains to the user of our approach.

The remaining fifth step is constructing an event log that contains a trace with a corresponding case identifier for each conversation. Based on the hypotheses and thresholds

TABLE II
CALCULATION OF THE OPTIMAL DECISION THRESHOLD USING CROSS-VALIDATION

ID	Label	This example is battery
36	1	0.998
31	0	0.996
47	1	0.995
56	1	0.993
75	0	0.980
39	0	0.931
⋮	⋮	⋮

ID	Label	This example is battery
87	1	0.999
53	1	0.995
89	0	0.993
68	1	0.941
84	0	0.908
27	0	0.892
⋮	⋮	⋮

ID	Label	This example is battery
22	1	0.998
10	1	0.996
92	1	0.993
35	0	0.991
88	0	0.973
6	0	0.930
⋮	⋮	⋮

ID	Label	This example is battery
26	1	0.998
32	1	0.997
70	1	0.996
17	0	0.993
38	0	0.947
3	0	0.926
⋮	⋮	⋮

ID	Label	This example is battery
49	1	0.997
99	1	0.995
65	1	0.993
54	0	0.990
37	0	0.952
78	0	0.894
⋮	⋮	⋮

computed in the previous steps, NLI is applied to all messages for each conversation. If the computed probability is greater or equal to the threshold for the respective topic or activity, the message is classified accordingly. For each assigned topic or activity, an event is inserted into the trace, including the id of the message, the timestamp, the author (in the `org:resource` attribute), the text, and the assigned topic or activity (in the `case:concept:name` attribute). To this end, the Python library "PM4Py" [35] provides a function to export a classified dataset of events to a standardized IEEE XES event log [16], which can then be imported by process mining applications (e. g., ProM [17] or Disco [18]).

IV. EVALUATION AND RESULTS

A. Evaluation Strategy

To evaluate our approach, we compare the performance of the best NLI hypothesis (referred to as "NLI - final" in the following) and the default NLI hypothesis ("NLI - default") for each topic or process activity to the performance of a simple keyword-based classifier ("Keyword"). The latter looks up if a message (converted to lowercase) contains the given keyword. In this case, it assigns the message to the respective activity or topic. For example, if a customer's inquiry contains the word "deliver", the message is assigned to the topic "delivery". Similarly, if a customer service agent's response contains "?", the message is assigned to the activity "Investigate issue". We claim the keyword-based classifier as a suitable benchmark since it is comparatively simple to implement and strains less computational complexity than NLI. After assessing the performance of the approach, we then show the applicability of the constructed event logs by applying process discovery using three discovery algorithms.

B. Data and Preprocessing

We utilized an existing corpus of Twitter conversations [19] that consists of almost three million Tweets to and from the customer support accounts of 108 companies. Furthermore, it has been asserting itself as suitable in recent academic publications (e. g., [36] or [37]). We exemplarily chose the Tweets to and from AmazonHelp, AppleSupport, and SpotifyCares to ensure a comprehensive evaluation.

We preprocessed the dataset as follows. First, we filtered for conversations that involve exactly one company since only in these cases it is feasible to automatically decide which company is responsible for resolving the customer's inquiry. Second, we removed all conversations in a non-English language. For this purpose, we invoke Facebook's Python library "fastText" [38], [39], which provides language identification using a pre-trained model. Third, we deducted all Tweets that cannot be considered as conversations since the company did not reply to the customer's inquiry. Fourth, to improve the classification algorithm's accuracy, we applied spelling correction to all inbound Tweets using the Python library "pyspellchecker". Our final datasets for AmazonHelp, AppleSupport, and SpotifyCares consist of 288,828, 231,683, and 88,774 Tweets, respectively.

We applied these steps since the specific dataset requires a suitable preprocessing before transforming it to a useful event log. However, depending on the origin of the dataset, other means of preprocessing might be suitable since the overall approach, as outlined in the following, is not limited to a particular type of dataset (e. g., Twitter data) and can, for example, also convert transcripts of customer service calls.

C. Approach Instantiation

Following our developed approach from Section III (Fig. 1), we first defined exemplary topics and process activities by analyzing samples of 200 inbound and 100 outbound Tweets. In the second step, we labeled them accordingly. To account for human errors in the labels, three of the authors checked and agreed on the labeled dataset. We labeled twice as many inbound Tweets as outbound Tweets since the inbound Tweets turned out to cover a broader range of topics, and in many cases, they could also stand alone without a response although the respective company answered them. Furthermore, our sample of inbound Tweets only contains Tweets that mark the beginning of the conversations since we observed that the customers usually describe their inquiry in the first message. The first decision point of the underlying support process seems to be of crucial interest from a process mining perspective (compared to decision points later on in the process). For example, depending on whether the customers describe an issue with their phone's battery or an issue with their computer's software, it is quite likely that different subprocesses handle their issues. Thus, we achieve to mine this central decision point reliably by labeling more inbound Tweets. After applying NLI with the default hypothesis "This example is [topic/activity]" in the third step, we investigated further

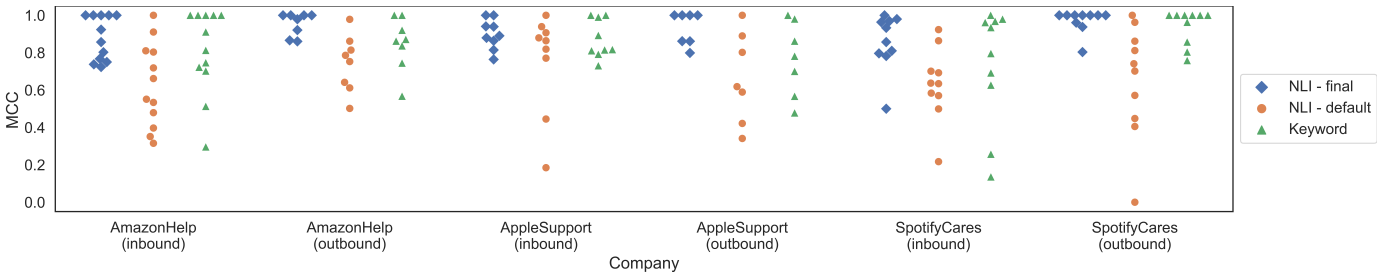

Fig. 2. Classification Results for Inbound and Outbound Tweets of AmazonHelp, AppleSupport, and SpotifyCares.

hypotheses in the fourth step to improve the results for the topics and activities for which the default hypothesis did not yield satisfactory results. Following an iterative procedure, we included between 4 and 38 further combinations, such as "The sentence is about [topic/activity]", "The customer asks about [topic]", and "Please provide/send ...". Based on the results, we chose the NLI hypothesis with the highest MCC and the computed optimal threshold. In the last step, we constructed a standardized XES event log [16] using "PM4Py" [35]. The detailed results for all hypotheses and companies, as well as the resulting event logs for the entire datasets, can be found in our GitHub repository (https://github.com/kechtel/ericsson).

D. Results

The swarm plot in Fig. 2 provides an overview of the results by plotting the MCCs of the three classifiers explained in Section IV-A for the inbound and outbound Tweets of AmazonHelp, AppleSupport, and SpotifyCares. The blue diamonds, orange dots, and green triangles represent the distribution of the MCCs for classifying the topics (for inbound Tweets) and the activities (for outbound Tweets) using the "NLI - final", "NLI - default", and "Keyword" approach, respectively. The distributions lead us to the following conclusions: First, on average, the results achieved using the final NLI hypothesis outperform the other approaches. In 55 of 56 cases, the MCC is greater than 0.72, implying a high performance in all four dimensions of the binary classification confusion matrix. Therefore, we conclude a feasible approach for the construction of event logs. Second, in some cases, the "NLI - default" and, more particularly, the "Keyword" approach already achieve highly accurate results. For example, our labeled datasets indicate that customers having an issue with a particular product, name that product explicitly in their inquiry. Another example is when the customer support agent provides an URL to the customer. These URLs all start with "https://t.co" due to the Twitter URL shortening feature. Third, the "NLI - default" hypothesis ("This example is [...]") completed with the keyword has the highest variance in MCC among all approaches. We trace this observation back to some grammatically incorrect hypotheses. For example, "This example is what's happening" achieves an MCC of 0 on the outbound Tweets of Spotify, whereas the keyword approach using "what's happening" as well as the final hypothesis "This example asks to describe what's happening" achieves an MCC of 1.

Table III provides further insights into the performance of our approach on the example of 100 outbound Tweets of AppleSupport. As apparent from Fig. 2, the example is representative of the remaining five datasets of AppleSupport (inbound), AmazonHelp (inbound and outbound), and SpotifyCares (inbound and outbound). We provide these datasets in the same format in the GitHub repository of this project. Furthermore, Table III lists concrete activities that we derived from the sample, including the frequency of each activity in the sample. We also list other standard binary classification evaluation metrics next to the MCC for comparison with related approaches.

The examples listed in Table III again show how "NLI - final" outperforms the other two approaches by achieving a score of 1 in all evaluation metrics for four of seven activities. An analysis of the remaining five datasets yields similar results. Comparing "NLI - default" to the "Keyword" approach, both exhibit a high variance in their evaluation metrics and classify different activities better. The "Keyword" approach tends to accomplish satisfactory results when the sentence describing the activity contains the keyword exactly and without any ambiguities. However, the "NLI - default" approach obtains better results when activities are described using several wordings. For example, the "NLI - default" approach exploits the fact that "restarting a device" means the same as "turning a device off and on again", which is represented in current language models.

E. Process Discovery Application

To evaluate whether our constructed event logs are suitable for process mining purposes, such as process discovery, we imported the constructed log of AppleSupport again using "PM4Py" [35] and applied the Alpha Miner [20], the Inductive Miner [21], and the Heuristics Miner [22]. Before applying the miners, we filtered the event log for traces in which the customer's request could be assigned to one of our topics defined in Section IV-C. Next, to reduce the vast variants, we exemplarily filtered the event log for the five most frequent variants. Fig. 3 visualizes the resulting process maps portrayed as Petri nets of the three applied process discovery algorithms. Although the Petri nets' visualizations differ among the algorithms, the Petri nets reflect the same process model. The models reveal that when a customer inquired about the topic "iPhone", the customer service agent provided an URL, regardless of whether the customer's issue could also be

TABLE III
CLASSIFICATION RESULTS FOR 100 OUTBOUND TWEETS OF APPLESUPPORT

Activity	Frequency	Approach	Hypothesis/Keyword	Optimal Threshold	MCC	Accuracy	Balanced Accuracy	F1
Request DM	56	NLI - final	The sentence is about DM	0.83	1.00	1.00	1.00	1.00
		NLI - default	This example is DM	0.85	1.00	1.00	1.00	1.00
		Keyword	dm		0.98	0.99	0.99	0.99
Provide URL	80	NLI - final	The example provides a https://t.co	0.86	1.00	1.00	1.00	1.00
		NLI - default	This example is https://t.co	0.80	0.80	0.92	0.95	0.95
		Keyword	https://t.co		1.00	1.00	1.00	1.00
Request iOS version	18	NLI - final	This example asks for iOS version	0.986	0.86	0.96	0.89	0.88
		NLI - default	This example is ios	0.81	0.59	0.87	0.81	0.67
		Keyword	ios		0.78	0.93	0.91	0.82
Request device model	9	NLI - final	The example requests the device model	0.985	1.00	1.00	1.00	1.00
		NLI - default	This example is device	0.95	0.42	0.89	0.74	0.48
		Keyword	device		0.48	0.91	0.75	0.53
Investigate issue	56	NLI - final	This example requests details	0.991	0.80	0.90	0.90	0.91
		NLI - default	This example is question	0.78	0.62	0.81	0.81	0.83
		Keyword	?		0.57	0.73	0.76	0.68
Request device restart	4	NLI - final	This example is about restart	0.87	1.00	1.00	1.00	1.00
		NLI - default	This example is restart	0.74	0.89	0.99	0.99	0.89
		Keyword	restart		0.70	0.98	0.75	0.67
Refer to specialist team	3	NLI - final	The example mentions a team	0.999	0.86	0.99	0.99	0.86
		NLI - default	This example is team	0.88	0.34	0.82	0.91	0.25
		Keyword	team		0.86	0.99	0.99	0.86

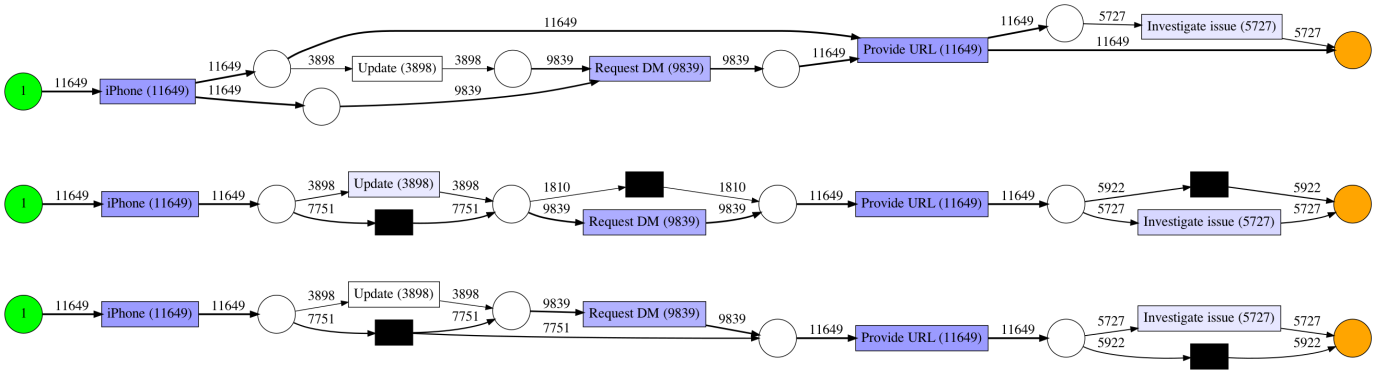

Fig. 3. Process Maps Discovered by Applying the Alpha Miner [20] (top), Inductive Miner [21] (center), and Heuristics Miner [22] (bottom) of "PM4Py" [35] to the Five Most Frequent Variants of the AppleSupport Dataset

assigned to the topic "update". However, in 84 percent of cases, the agent asked the customer to send a direct message, and in 49 percent of cases, the agent further investigated the customer's issue.

V. DISCUSSION

The evaluation of our approach reveals how NLI using a pre-trained language model and a hypothesis that precisely describes the topic or process activity of interest helps construct event logs of high accuracy for process mining purposes. Almost all Tweets in our samples for AmazonHelp, AppleSupport, and SpotifyCares could be assigned to the respective topics and activities with an MCC greater than 0.72 without requiring extensive preprocessing as comparable NLP-based approaches. In terms of binary classification evaluation, a high MCC indicates a low proportion of false positives and false negatives, as well as a high proportion of true positives and true negatives, independent of which class is the positive class [34].

A noteworthy aspect is that we obtained these results without training a classifier, which usually requires a vast amount of labeled data. Instead, we exploited Zero-shot classification capabilities [15] using a pre-trained language model [13]. The labeling of 300 Tweets for each dataset was necessary to compute the optimal binary classification threshold, as this classifier outputs a probability for each input. Based on this tiny sample, we successfully converted more than 200,000 Tweets to an event log suitable for process mining purposes, as shown by the process discovery algorithms in Section IV-E. However, a larger sample size can mitigate the consequences of human errors in the labels and increase the probability that the randomly drawn sample is representative of the entire dataset. Due to the individual inference for each topic and activity, another advantage compared to some supervised

approaches is that adding further topics and activities does not affect the quality of inferring the existing topics and activities.

Finding a suitable NLI hypothesis for a particular topic or activity remains challenging due to the NLP pipeline's black-box nature that computes the probabilities. For example, the hypothesis "This example is about prime" achieved an MCC of 0.91 on the inbound Tweets of AmazonHelp. In contrast, the default hypothesis "This example is prime" achieved a significantly worse MCC of 0.53. These results are in line with the findings of Yin et al. [15], who concluded that the definition of a particular topic is not suitable for a hypothesis compared to including the topic itself or the topic together with its definition. For example, for all three companies, most of the hypotheses for the activity "Refer to specialist team" did not achieve satisfactory results since the hypothesis defined the activity, but the corresponding Tweets did not always contain the word "team". Furthermore, the suitability of a hypothesis depends on the specific task (e. g., topic, emotion, and situation detection) and the extent to which the topic or activity is a common word [15]. Therefore, on the example of product names, such as "Mac" and "iCloud", it proved more challenging to find a suitable hypothesis in contrast to finding a suitable hypothesis for an apology.

Nevertheless, some topics and activities in our data could be inferred with a comparable accuracy using more straightforward keyword-based approaches, for example, Tweets containing a specific sequence, such as a URL pattern. In case this straightforward approach fails, NLI using the default hypothesis of, for example, the "transformers" library [31], can significantly improve the accuracy. If none of these two simple approaches yields satisfactory results, defining further hypotheses describing the topics and activities seems promising. For a practical instantiation of our approach, we recommend starting with evaluating the two simple approaches on a labeled sample and refining the hypothesis if an improvement of the results is desired. The examples in our GitHub repository and the examples in Table III might serve as a starting point.

VI. Conclusion

This paper presents an approach to represent customer service conversations as a standardized IEEE XES event log [16], which is suitable to be imported by most common process mining applications. Our results reveal that NLI with a hypothesis that precisely describes the topic or process activity of interest achieves the highest performance compared to two more straightforward baseline approaches. Due to the underlying model's ability to understand natural language, NLI outperforms other linguistic approaches that do not exploit pre-trained models' capabilities, such as memorizing which words occur in a similar context. Although we demonstrated and evaluated our approach on the example of written conversations on Twitter, the approach enables further use cases. For example, it is also suitable to convert transcripts of customer service calls and conversations within organizations (i. e., if employees are considered as internal customers).

Our contribution to research and practice is threefold. First, we show how NLI using a pre-trained model enables the construction of event logs of high data quality. In contrast to a supervised machine learning approach, we do not require vast amounts of labeled training data to achieve high data quality, thus mitigating the complexity of applying process mining to unstructured text data. Instead, labeling 300 Tweets per dataset turned out as sufficient and seems feasible for an instantiation in practice. Second, we demonstrate our approach's applicability using real-world data of the customer support Twitter accounts of three companies. The constructed event logs emerged as suitable for process discovery, as our successful application of three different process discovery algorithms [20]–[22] confirms. Third, the implementation of the approach, including the Twitter data, is publicly available on GitHub and can be reused by the community.

Nevertheless, we also encounter limitations of our approach. First, we limited our analysis of the customers' issues to their first message. However, a deeper analysis can lead to more insights into the discovered processes. Such an analysis can, for example, compare how the process behaves in case the customer replies that a restart of the device (if requested by the customer service agent) was successful or not. Second, as our approach involves manually identifying the topics and activities of interest, we assume practitioners have the required domain knowledge about the dataset that should be represented as an event log. If this assumption does not hold, the upstream application of other algorithms for topic extraction seems desired for a comprehensive overview of possible topics and activities of interest. Consequently, since we do not know the actual underlying process models and, notably, the topics and activities that might interest the companies, our evaluation remains narrowed to investigating the topics and activities we could identify during screening the samples. Therefore, the discovered process models are likely to represent only a small excerpt of the actual underlying support process. Third, the NLP pipeline's black-box nature can impede finding a suitable hypothesis for a particular topic or activity, as discussed in the previous section.

Due to its reusability, our presented approach can serve as the foundation for future work in the field of process mining. First, particularly at the intersection of process mining and NLP research, our approach enables the evaluation of chatbots' ability to learn business processes encompassed in the textual training data. Second, further language models can be examined to improve the accuracy of the approach and mitigate the complexity of finding a suitable NLI hypothesis. For example, OpenAI's GPT-3 language model, which was trained with 175 billion parameters, significantly surpasses most existing language models without requiring fine-tuning for a specific task [40]. Third, since most pre-trained models are limited to a certain text length, using our approach for longer texts, such as emails, requires sliding window approaches. Fourth, a more thorough evaluation could contrast the different related supervised and unsupervised approaches for constructing event logs from conversations.

REFERENCES

[1] W. M. P. van der Aalst, *Process mining: Discovery, conformance and enhancement of business processes*. Berlin, Heidelberg: Springer, 2011.

[2] R. Andrews, C. G. J. van Dun, M. T. Wynn, W. Kratsch, M. K. E. Röglinger *et al.*, "Quality-informed semi-automated event log generation for process mining," *Decision Support Systems*, vol. 132, p. 113265, 2020.

[3] D. Calvanese, M. Montali, A. Syamsiyah, and W. M. P. van der Aalst, "Ontology-driven extraction of event logs from relational databases," in *Business Process Management Workshops*. Cham: Springer International Publishing, 2016, pp. 140–153.

[4] S. Schönig, A. Rogge-Solti, C. Cabanillas, S. Jablonski, and J. Mendling, "Efficient and customisable declarative process mining with SQL," in *Advanced Information Systems Engineering*. Cham: Springer International Publishing, 2016, pp. 290–305.

[5] W. M. P. van der Aalst and A. Nikolov, "Mining e-mail messages: uncovering interaction patterns and processes using e-mail logs," *International Journal of Intelligent Information Technologies*, vol. 4, no. 3, pp. 27–45, 2008.

[6] D. Jlailaty, D. Grigori, and K. Belhajjame, "Mining business process activities from email logs," in *2017 IEEE International Conference on Cognitive Computing (ICCC)*, Honolulu, HI, USA, 2017, pp. 112–119.

[7] S. Alter, "Theory of workarounds," *Communications of the Association for Information Systems*, vol. 34, pp. 1041–1066, 2014.

[8] S. Rinderle and M. Reichert, "Data-driven process control and exception handling in process management systems," in *Advanced Information Systems Engineering*. Berlin, Heidelberg: Springer, 2006, pp. 273–287.

[9] U. M. König, A. Linhart, and M. Röglinger, "Why do business processes deviate? Results from a Delphi study," *Business Research*, vol. 12, pp. 425–453, 2019.

[10] R. B. Banziger, A. Basukoski, and T. Chaussalet, "Discovering business processes in CRM systems by leveraging unstructured text data," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Exeter, UK, 2018, pp. 1571–1577.

[11] M. Wolfinbarger and M. C. Gilly, "eTailQ: dimensionalizing, measuring and predicting etail quality," *Journal of Retailing*, vol. 79, no. 3, pp. 183–198, 2003.

[12] W. van der Aalst, A. Adriansyah, A. K. Alves de Medeiros, F. Arcieri, T. Baier *et al.*, "Process mining manifesto," in *Business Process Management Workshops*. Berlin, Heidelberg: Springer, 2012, pp. 169–194.

[13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 7871–7880.

[14] B. MacCartney and C. D. Manning, "Modeling semantic containment and exclusion in natural language inference," in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, Manchester, UK, 2008, pp. 521–528.

[15] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3914–3923.

[16] IEEE, "IEEE standard for eXtensible Event Stream (XES) for achieving interoperability in event logs and event streams," *IEEE Std 1849-2016*, pp. 1–50, 2016.

[17] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The ProM framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets*. Berlin, Heidelberg: Springer, 2005, pp. 444–454.

[18] C. W. Günther and A. Rozinat, "Disco: Discover your processes," in *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012)*, Tallinn, Estonia, 2012, pp. 40–44.

[19] Thought Vector. (2017) Customer support on Twitter. (Licensed under CC BY-NC-SA 4.0). [Online]. Available: https://www.kaggle.com/thoughtvector/customer-support-on-twitter

[20] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.

[21] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs - a constructive approach," in *Application and Theory of Petri Nets and Concurrency*. Berlin, Heidelberg: Springer, 2013, pp. 311–329.

[22] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. Alves de Medeiros, "Process mining with the HeuristicsMiner algorithm," 2006.

[23] W. van der Aalst, *Process Mining: Data Science in Action*. Berlin, Heidelberg: Springer, 2016.

[24] J. Tang, Z. Meng, X. Nguyen, Q. Mei, and M. Zhang, "Understanding the limiting factors of topic modeling via posterior contraction analysis," in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, vol. 32, Beijing, China, 2014, pp. 190–198.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations (ICLR)*, Scottsdale, AZ, USA, 2013.

[26] A. Rebmann, A. Emrich, and P. Fettke, "Enabling the discovery of manual processes using a multi-modal activity recognition approach," in *Business Process Management Workshops*. Cham: Springer International Publishing, 2019, pp. 130–141.

[27] S. Knoch, S. Ponpathirkoottam, and T. Schwartz, "Video-to-model: Unsupervised trace extraction from videos for process discovery and conformance checking in manual assembly," in *Business Process Management*. Cham: Springer International Publishing, 2020, pp. 291–308.

[28] F. Friedrich, J. Mendling, and F. Puhlmann, "Process model generation from natural language text," in *Advanced Information Systems Engineering*. Berlin, Heidelberg: Springer, 2011, pp. 482–496.

[29] H. Leopold, H. van der Aa, and H. A. Reijers, "Identifying candidate tasks for robotic process automation in textual process descriptions," in *Enterprise, Business-Process and Information Systems Modeling*. Cham: Springer International Publishing, 2018, pp. 67–81.

[30] T. Baier, C. Di Ciccio, J. Mendling, and M. Weske, "Matching events and activities by integrating behavioral aspects and label analysis," *Software and Systems Modeling*, vol. 17, no. 2, pp. 573–598, 2018.

[31] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, October 2020, pp. 38–45.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 6, 2020.

[34] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Mining*, vol. 14, no. 13, 2021.

[35] A. Berti, S. J. van Zelst, and W. M. P. van der Aalst, "Process mining for Python (PM4Py): Bridging the gap between process- and data science," in *Proceedings of the ICPM Demo Track 2019, co-located with 1st International Conference on Process Mining (ICPM 2019)*, Aachen, Germany, Jun 2019.

[36] M. Hardalov, I. Koychev, and P. Nakov, "Towards automated customer support," in *Artificial Intelligence: Methodology, Systems, and Applications*. Cham: Springer International Publishing, 2018, pp. 48–59.

[37] M. Misuraca, G. Scepi, and M. Spano, "A network-based concept extraction for managing customer requests in a social media care context," *International Journal of Information Management*, vol. 51, p. 101956, 2020.

[38] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou *et al.*, "FastText.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain, April 2017, pp. 427–431.

[40] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan *et al.*, "Language models are few-shot learners," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, Online, December 2020, pp. 1877–1901.