Discovering Business Processes in CRM Systems by leveraging unstructured text data

Banziger, R.B., Basukoski, A. and Chaussalet, T.J.

https://dx.doi.org/10.1109/HPCC/SmartCity/DSS.2018.00257

Discovering Business Processes in CRM Systems by leveraging unstructured text data

Rolf B. Bänziger
Department of Computer Science
University of Westminster
London, United Kingdom
r.banziger@westminster.ac.uk

Artie Basukoski
Department of Computer Science
University of Westminster
London, United Kingdom
ABasukoski@westminster.ac.uk

Thierry Chaussalet
Department of Computer Science
University of Westminster
London, United Kingdom
chausst@westminster.ac.uk

***Abstract*—Recent research has proven the feasibility of using Process Mining algorithms to discover business processes from event logs of structured data. However, many IT systems also store a considerable amount of unstructured data. Customer Relationship Management (CRM) Systems typically store information about interactions with customers, such as emails, phone calls, meetings, etc. These activities are characteristically made up of unstructured data, such as a free text subject and description of the interaction, but only limited structured data is available to classify them. This poses a problem to the traditional Process Mining approach that relies on an event log made up of clearly categorised activities. This paper proposes an original framework to mine processes from CRM data, by leveraging the unstructured part of the data. This method uses Latent Dirichlet Allocation (LDA), an unsupervised machine learning technique, to automatically detect and assign labels to activities. This framework does not require any human intervention. A case study with real-world CRM data validates the feasibility of our approach.**

***Index Terms*—Process Mining, Process Discovery, Customer Relationship Management, CRM, Business Process Management, Latent Dirichlet Allocation.**

I. Introduction

Process Mining ([1], [2], [3]) describes a set of techniques used to extract knowledge from event logs from IT systems. The extracted knowledge is usually represented in the form of Process Models. Business Processes are discovered by analysing the event logs that modern IT systems store. A lot of research has been undertaken in discovering Business Processes from event logs generated by a variety of systems. Most research however used event logs in a structured format, where each event is classified to belong to a known category (also known as an event class).

However, many modern IT systems do also save plenty of unstructured data. Customer Relationship Management (CRM) systems are an example of such systems. CRM is deployed to maximise the value of a customer relationship.

CRM systems are commonly used to support Marketing, Sales and Service Processes of an Organisation. An ideal CRM system records any process instance as a case (e.g. a marketing campaign participation, a sales opportunity, a support case), related to the respective customer record. Any interactions with customers or colleagues, such as emails, phone calls, meetings, tasks, etc. are linked to these cases. In the CRM domain, these interactions are called activities. Activities are identified by participants, date and time and unstructured information like subject and body text. This provides a flexible way of supporting business processes, without the need to pre-define every possible step and implement them explicitly. The underlying business process may or may not be explicitly defined and a CRM system may implement the processes in more flexible or stricter ways ([4]).

There are some problems with not having formally defined business processes. There is no documentation, meaning executing the process relies on implicit knowledge. Training new colleagues is hard. People might work in different ways without realising it. Optimising the workflow will be difficult, as it is hard to get a real picture of how people work. Decisions might be taken on wrong or outdated information. Also, having a business process documentation is a requirement for quality management certifications such as ISO9000. If a business process is defined, it is important everyone is following the process. An organisation's management needs to monitor whether the real workflow conforms to the process documentation. If there is a mismatch, either the process documentation needs to be updated or people need more training.

In the case of a very strict implementation of a business process in a CRM system, the system will not allow the user to stray from the defined business process. Management will however still be interested in what process paths are usually taken, what are the throughput times, what decisions in the process account for the slowest and fastest process execution, etc.

CRM systems have been suggested as a target for Process Mining algorithms various times ([1], [2], [5], [6]), however, not much specific research has been undertaken in this area. Having every customer interaction documented means that CRM stores a trace of all customer-facing business processes. We can use the activities as a base for an event log, which in turn is used to discover business process models. One challenge in using activities is that CRM systems characteristically do not label activities with a classification of what this activity is about. Rather, the description of the activity is usually manually entered in free text subject and description fields. Subject and description are potentially different in every

instance and thus cannot be used as event classifiers. Most previous research for mining processes based on activities makes unrealistic assumptions (e.g. [7] assumes the subject contains the activity category) or requires human intervention to at least a certain extent ([8] use supervised machine learning to label activities which requires a manually labelled training set, [9] requires a user to label one exemplary activity of each automatically detected cluster).

This paper proposes an original framework to discover business process models from semi-structured data without the need for any human intervention in the process. We apply various pre-processing steps to CRM activities and then use Latent Dirichlet Allocation (LDA) to automatically classify and label all activities. After converting the activities to an event log, we use ProM [10] to further filter the event log and discover process models. We implemented this approach with R and tested it with real-world CRM data.

The contributions of this paper are:

- A framework to discover process models automatically from CRM data by exploiting the unstructured text data. It is worth noting that there may or may not be a known process underpinning the data. In both cases, it is worthwhile to discover a process model. A method to automatically label the steps of the discovered process model based on the most important keywords extracted from the text descriptions,
- a method to estimate the number of distinct process steps per activity type, and
- a use case that demonstrates the feasibility of this approach.

Section 2 gives an overview of related work. Section 3 describes our method in detail. Section 4 presents the results of the method applied to real-world CRM data and Section 5 summarises our conclusions.

II. Related Work

There is surprisingly little research into Process Mining with real-world CRM data. Mahendrawathi et al. [11] analyse the customer fulfilment process of a telecommunication company. They show that it is possible to discover the typical business process in CRM data, even when the examined business process is unstructured, and that the gained insight can be valuable to improve the business processes. The paper shows the steps required to create an event log from the CRM data and shows ways to analyse the data. However, their activities are already labelled with the event class. Hence this approach is not feasible for CRM implementations that do not label activities.

Laga et al. [12] are enriching existing business process models with communication activities. They add semantic information to an existing business process model. When a new activity is processed, this activity is compared to the enriched business process model and classified in accordance with the process model. The method is implemented in a CRM system. Their approach proves the feasibility of business process management techniques based on CRM activities. They do note that the traditional Process Mining techniques produce incomplete process models, as they only work on structured data, i.e. classified events and ignoring the semantics of communication activities.

We also draw from research that looks into extracting event logs from emails. This is similar to the problem of extracting event logs from CRM data, as email is nowadays one of the most important types of activities in CRM systems.

Van der Aalst and Nikolov [7] developed a tool to extract event logs from emails in Outlook. This tool assumes that event classes are represented in the subjects of the emails, either manually or automatically being added by a corporate process-aware system. CRM systems usually do not automatically tag activities with the event class, and our goal is to require no human intervention, so this approach is not feasible for our method.

Brander et al. [8] describe a method to leverage informal communication to discover business processes. They extract event logs from emails and personal tasks. A small training set is manually labelled. Supervised machine learning is then used to label the other activities. As this still requires manual intervention, this is not suitable for our goal of having a completely unsupervised process. It is also not certain how their method will perform when the training set is older than the examined emails and tasks. Business processes typically do change over time, so the training set will need to be labelled periodically.

Jlailaty et al. [13] present an interesting approach to extract an event log from personal emails. This approach uses k-means clustering of the subject and body of personal emails to assign emails to event classes. For each identified cluster, the user is then asked to label each cluster's medoid manually. This information is used as the event class label. The number of clusters is determined by the average number of emails per process trace. Any new emails are labelled using the 1-nearest-neighbour classifier. This is the research that comes closest to what we want to achieve. However, it still requires human intervention as the cluster's medoids must be labelled manually.

Hong & Moh [14] look at the problem of automatically organising emails by grouping them in similar topics. While their research is not aimed at extracting event logs from emails, it shows that Latent Dirichlet Allocation (LDA, [15]) can be used to effectively extract an email's topics. LDA is a topic modelling technique which assigns a latent topic distribution to each document in a corpus. It needs to be given the number of total topics. A fitted LDA model produces a matrix of documents and topics, specifying the probability of a document being of a certain topic. While determining the topic distribution per document, it also generates a word distribution for every document. This can be used to extract the n most relevant words for any topic, which gives a good human-readable topic label. We use LDA based topic modelling in our method to classify and label activities.

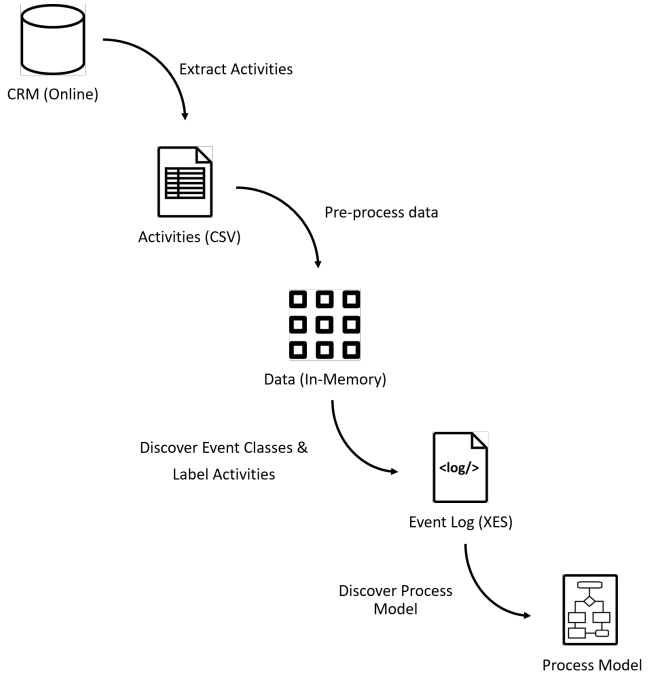

Fig. 1. CRM Process Mining framework.

III. Method

To discover process models from semi-structured CRM data, we propose a framework to mine CRM data for process models (cf. Fig. 1). This framework can be interpreted as a generic sequence of steps needed to be undertaken in all Process Mining projects where the goal is to mine semi-structured data. We start by describing the typical data structure of a CRM system. Then we describe each step of the framework in detail.

The first step is to select and export the cases and activities (our events) we want to examine. In this case, we extract them from our CRM system into a CSV file. Now we pre-process the data, for example, remove any HTML tags in emails and transform the data into a useable format. The next step is to identify event classes and label the activities. The data can now be converted to an event log in the XES format [16]. This event log is loaded by the tool ProM [10], which provides the necessary plugins to filter the event log and to discover process models.

A. Extract Activities

a) Data structure: In the first step, we need to extract the data we want to analyse. All CRM systems have a similar data model; a business process instance (a case) groups all related activities (cf. Fig. 2). Every case (typically, but not limited to, a marketing campaign, a sales opportunity or a support case) is identified by a unique Id and contains data about the (potential) customer, a descriptive title, status information, start and end dates, etc.

Activities represent an interaction with a customer of a case-related task. There are several different activity types: Emails,

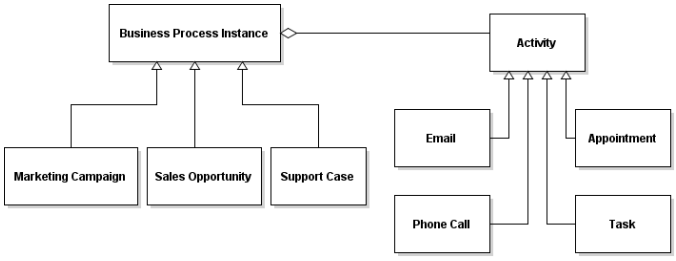

Fig. 2. CRM data model.

Appointments, Phone Calls and Tasks. We do artificially further refine these types by splitting emails into sent and received email and phone calls into incoming and outgoing calls. Activities are identified by a unique Id and contain subject and description text attributes and other information such as date, participants, etc. To create an event log out of CRM data, we use the case entity to represent a trace of a business process and the activities to represent the event belonging to a trace. We export all data to one big CSV file. An excerpt is shown in Table I. Note that the actual CSV table will contain more attributes.

TABLE I
Exported CRM data

Case	Activity			
Id	**Id**	**Subject**	**Date**	**Type**
1	1	CRM Issues	05/01 15:25	Received Email
	2	RE: CRM issues	06/01 09:42	Sent Email
	3	RE: CRM issues	06/01 11:29	Received Email
2	4	User access problem	05/01 16:12	Received Email
	5	Connect to the client's system and diagnose…	05/01 17:04	Task

b) Data selection: When extracting cases and activities, we have to first decide what cases we want to analyse. We need to decide what business process we want to analyse. We cannot mix Support Cases with Sales Opportunity cases, as they clearly represent different business processes. We also need to take care that we don't mix Sales Opportunity cases from two departments that have an inherently different business process. Otherwise, we would end up with a "spaghetti model" (a term coined by van der Aalst in [2] to describe hard-to-read process models with little or no structure). We also should be careful to only extract completed cases, as incomplete cases will distort the resulting process model. As business processes change over time, we need to further restrict the selection to only include cases from a certain time frame, e.g. a year. Finally, we might also add selection criteria based on the actual case data. So it might be we're only interested in support cases concerning a certain product range or sales opportunities that have been won (or lost).

B. Pre-processing data

Before the activities can be labelled, the data needs to be cleaned up and transformed so it is viable for further processing. We strip any HTML tags from the subject and description fields and combine these fields into one text field, as we're not concerned about the differentiation between subject and description. We experimented with treating the text in the subject field as more important than the text in the body but found that most subjects, especially in emails are in practice not very descriptive. For example, most of the email just repeated the subject of the original email of a conversation. We also remove any mentioning of the customer name from the text, as the customer name does not add any information value and is considered noise. We remove the activity owner for the same reason. We also remove any URLs. Finally, the activity type is refined by including any directional information, i.e. emails are divided into sent and received emails and phone calls are divided into incoming and outgoing phone calls. No such transformation is necessary for appointments and task.

C. Discover event classes and label activities

To discover event classes and label activities, we use Latent Dirichlet Allocation (LDA). LDA was introduced by Blei et al. [15] and is a generative statistical model that allows to automatically detect latent topics of documents in a corpus. A topic is represented in the model as a probabilistic distribution over a set of words. LDA also assigns a topic distribution to each document. This means that a document can have multiple topics, i.e. one email could be 60% about organising a meeting and 40% about product features. To fit the LDA model, two inputs are required: the corpus and the number of topics.

We fit a separate LDA model for each activity type (Sent emails, received emails, tasks, etc.). The combined subject and description fields of each activity make up our corpus. Before fitting the model, any numbers and a standard list of English stop words are removed from all documents. Any non-alphabetic character, including the hyphen, is used to separate words. All words are stemmed.

When dealing with emails, a lot of them will have the same standard disclaimer. This information is not important for the labelling of the events. Instead, it only adds noise in the LDA model. There are also many other words that appear often, like the company name, greetings, etc. All these words do not add any relevant information. We thus want to eliminate such words. To do this, we calculate the Inverse Document Frequency (idf) for every term t in our corpus (1).

$$idf_t = \log \frac{N}{df_t}. \tag{1}$$

N stands for the number of documents in the corpus, df_t is the number of documents where the term t occurs. As next step, we calculate the Term frequency-Inverse Document Frequency (Tf-Idf) matrix ((2)). $tf_{t,d}$ stands for the number of occurrences of term t within document d.

$$tf{-}idf_{t,d} = tf_{t,d} \times idf_t \tag{2}$$

The tf-idf score for a term t in document d is highest when t occurs often in d, but not in other documents. It is lowest when t occurs in many or all documents [17].

We then remove all words from our corpus with a tf-idf score of less than the median of all tf-idf scores. Our experiments have shown that this vastly improves the quality of the labels.

a) Number of topics: To fit the LDA model to the corpus, the number of topics n, needs to be specified. As the best value for n is not known, it needs to be estimated. If n is too small, we will end up with separate activities sharing the same label (event classifier). In the extreme case, we'll have only one label per activity type. The process model will be simpler and easier to read, but at the same time, important information might be missing. If n is estimated too big, similar activities will be shown as different events in the resulting process model and we end up with a large number of distinct activity types, making the model hard to read. It is possible for n to assume any value between 1 (all activities are of the same event) and the total number of activities in the corpus (any single activity is of a different event class and there are no similar events within and between cases). While the specific number of n depends on the actual process, high values will be rather unlikely, if we are looking at process instances of the same process. We initialize the number of topics with the average number of activities per case.

We do not want to specify n for each individual activity type. Instead, we look at the distribution of activities of a certain type (e.g. tasks) per case and define a quantile k, which is used to determine n. Consider Fig. 3, where we show the number of Tasks per Case. If we set $k = 50\%$, n will be 1. Note that the value of n will be different for other activity types.

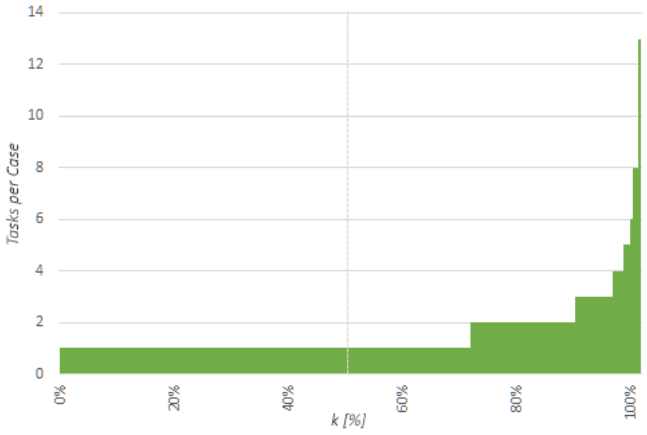

Fig. 3. Number of Tasks per Case.

b) Labelling activities: Once the LDA model is fitted, we know the topic distribution for each activity. Our experiments

have shown that there is usually one predominant topic for any activity. Thus, the most-likely topic is used to label the activity. As LDA also assigns a distribution of words to any topic, we take the five most probable words to label a topic. As we applied stemming to the corpus before, we need to "un-stem" the words to make them comprehensible. We create a mapping of all un-stemmed words in the corpus and their stemmed representation. Now we replace each stemmed word in the topic label with the most frequent un-stemmed version of it. This label is now appended to the activity type (e.g. sent email), resulting in a human-readable label (e.g. "Appointment: meeting discussion marketing software crm").

c) Writing the event log: As the last step in this phase, the labelled events need to be exported as an event log. We use the XES format [16]. This format lets us save any set of attributes for cases and events. We can not only include the minimally necessary attributes for discovering process models, but we can also add other attributes, such as the status of a case (e.g. won, lost, ongoing) or the product group a support case is related to, etc. This information can, for example, be used to filter the cases further.

To save the data in the XES format, we export it as a normal CSV file. We then use the tool XESame (cf. [18]) to transform the CSV file into an XES file.

D. Discover Process Models

To discover the process model, the event log is fed into a Process Mining Tool, such as ProM [10]. If needed, the event log may be pre-processed. We can filter the event log, e.g. only select certain event classes or only select certain cases. Sometimes it is necessary to add artificial start and end events, as this can improve the readability of the resulting process model significantly and is also a requirement for some process mining algorithms, such as the heuristics miner.

The most common perspective when mining event logs is the control-flow perspective, meaning Process Mining algorithms will generate a control-flow model, such as a petri net, a heuristics net, a BPMN diagram etc. from the event log. The most commonly used algorithms per [19] are Fuzzy Miner [20] and Heuristics Miner [5]. Also noteworthy is the α-algorithm that produces a Petri Net [13]. While this is a very simple algorithm, it does not deal very well with noise and incomplete event logs and does not produce very good results in practice. De Weerdt et al [21] found that the Heuristics Miner delivers the best results in mining real life processes. However, they did not include Fuzzy Miner in their studies.

IV. Experiment

We validate the proposed framework by testing it with real-world CRM data. The data set is provided by a Microsoft Dynamics CRM consulting company and is extracted from their internal CRM system. We selected all completed support cases created between 1st of January 2015 and 31 March 2015. The resulting data set contains 427 support cases with a total of 3277 activities. We anonymised the data.

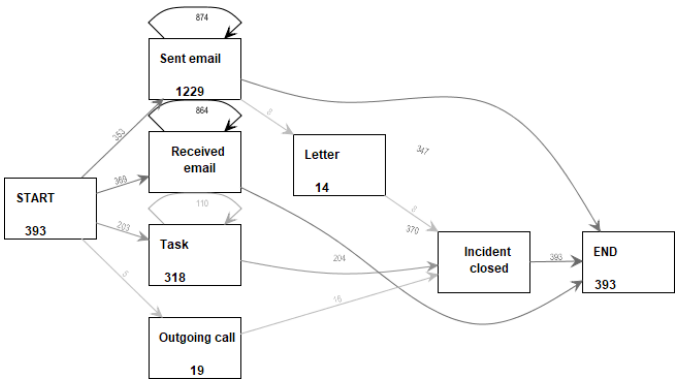

Fig. 4. Discovered Process Model without applying our method.

The data is exported and pre-processed as described above. We use the *topicmodels* package for *R* to calculate the LDA model [22]. We did increase the number of the topics k discovered per activity type, as we found that the support process is very heterogeneous and contains a lot of different activities. We set the k to the third quantile of the number of activities per case, instead of the average. After loading the event log into ProM, we added artificial start and end events to every case and discovered a process model by using the heuristics miner [5] on the event log.

First, we ran the heuristics miner plugin on the event log without applying the steps described in our framework. The resulting process model is shown in Fig. 4. We see that this process model provides a very high-level view of the business process, with a small number of event classes. When we apply our pre-processing steps, we expect to get a more detailed process model that gives more insight into the business process.

We applied our framework multiple times, each time with a different value for the parameter k, that controls into how many segments the activities are split. We found there is a stark increase in the number of segments n towards the higher end of k across all activity types (cf. Fig. 5). We found that setting $k = 0.95$ gives a good compromise between details and readability of the resulting model with our data.

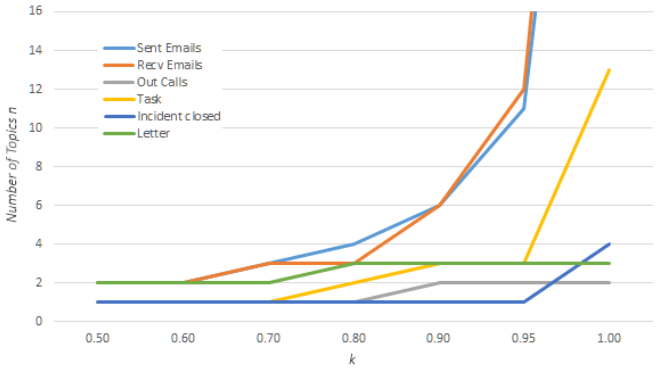

Fig. 5. Distribution of topics over k.

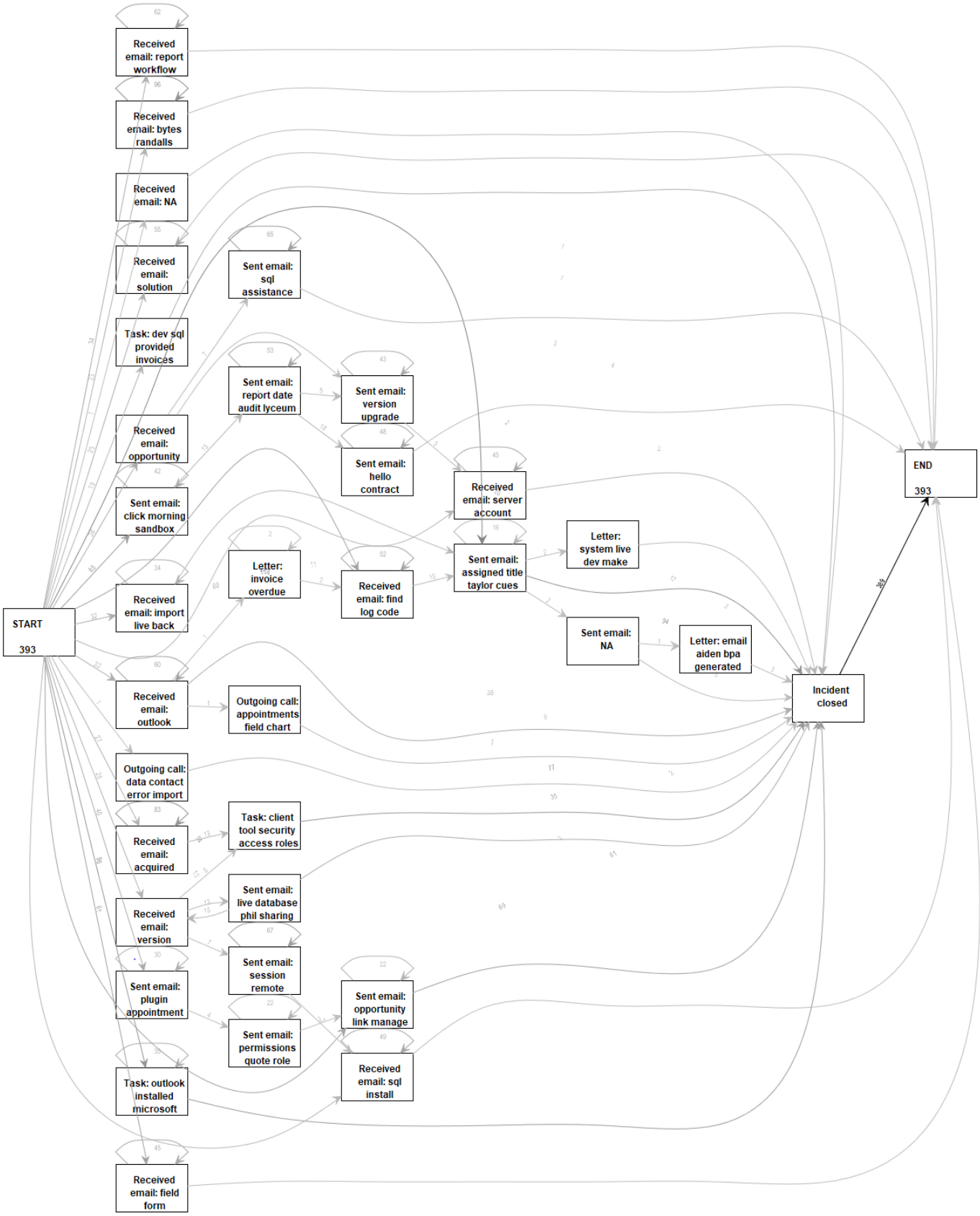

Fig. 6. The Process Model discovered with our method shows more details.

Fig 6 shows the discovered process model. While the actual model is not very clearly readable, it is obvious that it is more detailed (we're not discussing the actual process here). There are much more activity types shown in the discovered process model. We also see that there are new paths shown which couldn't be discovered before. For example, there is a loop between the event "Received email: version" and "Sent email: live database phil sharing". In the previously discovered process model, there was no sequence between any received and sent emails at all. Comparing the new process model with the actual event log, we also see that the paths shown in this process model reflect the event log better.

V. Conclusions and future work

In this paper, we developed a framework for mining process models from CRM data. The result of our experiment shows that the proposed method produces more detailed process models than existing methods when analysing CRM data with unstructured components. Our approach mainly relies on classifying events by leveraging text data using LDA models. Our experiment also shows that the discovered process is highly unstructured, something that is typical for CRM data.

In future work, we plan to validate the event classification results by using a human-tagged event log and comparing the resulting process model with the model generated by our method and with models generated using other text clustering techniques. We also plan to verify our results by validating the generated process models with domain experts and we will analyse larger datasets in the future.

While the method has been tested with CRM data, it is basically feasible for any event data with unstructured text data and few pre-defined event classes, e.g healthcare diagnostics data.

References

[1] IEEE Task Force on Process Mining, "Process Mining Manifesto," *Business Process Management Workshops*, pp. 169–194, 2011.

[2] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1 ed., 2011.

[3] W. M. P. Van Der Aalst and C. W. Günther, "Finding Structure in Unstructured Processes: The Case for Process Mining," *Proceedings - 7th International Conference on Application of Concurrency to System Design, ACSD 2007*, no. Acsd, pp. 3–12, 2007.

[4] V. Kumar and W. Reinartz, *Customer Relationship Management*. Springer Texts in Business and Economics, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[5] A. J. M. M. Weijters, W. M. P. Van Der Aalst, and A. K. A. D. Medeiros, "Process Mining with the Heuristics Miner Algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, pp. 1–34, 2006.

[6] J. Yin, B. Cao, S. Deng, and Z. Wu, "Process Discovery from the Log of Business Rule Engine," *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL*, vol. 8, pp. 5277–5293, jul 2012.

[7] W. M. P. Van Der Aalst and A. Nikolov, "EMailAnalyzer: An E-Mail Mining Plug-in for the ProM Framework," *BPM Center Report BPM-07-16*, no. August, pp. 1–26, 2007.

[8] S. Brander, K. Hinkelmann, B. Hu, A. Martin, U. V. Riss, B. Thönssen, and H. F. Witschel, "Refining Process Models through the Analysis of Informal Work Practice," in *Business Process Management: 9th International Conference, BPM 2011, Clermont-Ferrand, France, August 30 - September 2, 2011. Proceedings* (S. Rinderle-Ma, F. Toumani, and K. Wolf, eds.), pp. 116–131, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[9] D. Jlailaty, D. Grigori, and K. Belhajjame, "A framework for mining process models from email logs," 2016.

[10] H. Verbeek, J. Buijs, B. Van Dongen, and W. M. van der Aalst, "ProM 6: The process mining toolkit," *Proc. of BPM Demonstration Track*, vol. 615, pp. 34–39, 2010.

[11] E. Mahendrawathi, H. M. Astuti, and A. Nastiti, "Analysis of Customer Fulfilment with Process Mining: A Case Study in a Telecommunication Company," *Procedia Computer Science*, vol. 72, pp. 588–596, 2015.

[12] N. Laga, M. O. Kherbouche, and P.-a. Masse, "Communication-Based Business Process Task Detection - Application in the CRM Context," in *2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 1–8, IEEE, sep 2016.

[13] D. Jlailaty, D. Grigori, and K. Belhajjame, "Multi-level clustering for extracting process-related information from email logs," *Proceedings - International Conference on Research Challenges in Information Science*, pp. 455–456, 2017.

[14] H. Hong and T. S. Moh, "Effective topic modeling for email," in *Proceedings of the 2015 International Conference on High Performance Computing and Simulation, HPCS 2015*, 2015.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation David," *Journal of Machine Learning Research2*, vol. 3, no. Jan, pp. 993–1022, 2003.

[16] C. W. Günther and E. Verbeek, "Xes standard definition," 2014.

[17] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008.

[18] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "XES, XESame, and ProM 6," vol. 72 of *Lecture Notes in Business Information Processing*, pp. 60–75, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[19] J. Claes and G. Poels, "Process Mining and the ProM Framework: An Exploratory Survey," in *Business Process Management Workshops*, vol. 132, pp. 187–198, 2013.

[20] C. W. Günther and W. M. van der Aalst, "Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics," *Business Process Management - Lecture Notes in Computer Science*, vol. 4714, pp. 328–343, 2007.

[21] J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens, "A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs," *Information Systems*, vol. 37, no. 7, pp. 654–676, 2012.

[22] B. Grün and K. Hornik, "topicmodels: An R Package for Fitting Topic Models," *Journal of Statistical Software*, vol. 40, no. 1, pp. 1–30, 2011.