PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Université Paris Dauphine

Mining Business Process Information from Email Logs for Business Process Models Discovery

Soutenue par
Diana AL JLAILATY
Le 15 Novembre 2019

École doctorale n°ED 543
Ecole Doctorale de Dauphine

Spécialité
Informatique

Composition du jury :

Salima BENBERNOU Professeur Université Paris Descartes	*Rapporteur, Présidente*
Walid GAALOUL Professeur Université Télécom SudParis	*Rapporteur*
Yehia TAHER Maître de conférences Université de Versailles Saint-Quentin-en-Yvelines	*Examinateur*
Dimitris KOTZINOS Professeur Université de Cergy Pontoise	*Examinateur*
Daniela GRIGORI Professeure Université Paris Dauphine	*Directrice de thèse*
Khalid BELHAJJAME Maître de conférences Université Paris Dauphine	*Co-Encadrant de thèse*

DAUPHINE UNIVERSITÉ PARIS | PSL

Acknowledgement

First and foremost, I would like to thank Prof. Daniela Grigori, Professor at Paris Dauphine University and the director of LAMSADE laboratory, for allowing me to do this thesis under her supervision. Thank you for your availability, advice and support through the last few years I spent in LAMSADE. I appreciate all the hard work we did together which helped me learn a lot. Thank you for setting me on the right path when I lost sight for what laid ahead and for always letting your door open for answering my questions. I would also like to express my deep gratitude for your carefulness and undeniable support not only on the professional level, but also on the personal level. I will always be grateful for you.

I would like to express my sincere gratitude to my second supervisor Dr. Khalid Belhajjame, Maître de Conférences at the Paris Dauphine University, for all his contributions and guidance throughout this thesis. Thank you for all your beneficial scientific advices, remarks and discussions. Your positivity and passion for research were always an inspiration for me to overcome whatever problems I may face.

I was honored that Prof Salima Benbernou, Professor at Paris Descartes University, had accepted to review my thesis. Thank you for being interested in my work, for your relevant comments in the pre-defense and in the final report, and for your pleasant communication.

It was my honor as well that Prof. Walid Gaaloul, Professor at Télécom SudParis, had accepted to review my thesis. Thank you for your precise reading of my thesis, for your interest in my work and for your relevant and useful remarks.

I would like to thank Prof Dimitris Kotzinos, Professor at the University of Cergy Pontoise, for his interest in my thesis and for agreeing to participate in my jury.

My gratitude goes to the faculty members of the LAMSADE lab for all of their scientific support, friendship, and encouragement. Particularly, I would like to thank Joyce El Haddad, Juliette Rouchier, and Furini Fabio for following my advancement during each year of my thesis.

Thank you for Prof. Mohamad Dbouk, Professor at the Lebanese University for encouraging me.

Thank you for Dr. Rafiqul Haque, CTO and Co-founder of Cognitus-Intelligencia, for all the discussions and help you provided me throughout these years.

My thanks then go to my colleagues at LAMSADE. Thank you for Amine, Justin, Marcel, Tom, Nathanael, Thomas, Olivier, Mahdi, Manel, Ons, Ian, Ioannis, Celine, Mehdi, Axel, Oussama, Amine Mouhoub, Hossein, Zahra, Saeed, Boris, Charles, Fabien, Anaëlle, Khalil, Pierre, Beatrice, George, Raja and all the others. Thank you to the sweetest Mariem, or Myriam as I like to call you, for your friendship, pure heart and continuous support. I am so lucky to get to know a trustful person like you and I hope our friendship will last for a lifetime.

Hiba, thank you for being there whenever needed. We started our theses together and we are finishing them together. We have walked together almost the same path. I wish you all the success and happiness in life because you really deserve it. Yassine, thank you for all the discussions we went through during the last few years, for all your on-point advices and for your kind personality.

My thanks also go to the out-of-university friends that became a family for me. Melodie, thank you for being by my side throughout the good and bad. I will always be grateful to you for all what you have done to support me when I needed it the most. Raya, thank you for your carefulness and thoughtfulness, your personality and attitude had been always a great motivation for me, few are the people like you. Layth, the man of the group, thank you for your sense of humor, positivity and motivational spirit, I feel so grateful for your continuous support and help. Sabine, we do not know each other from a long time, but you became very close to my heart, thank you for your positivity and kindness. Rabiaa, thank you for your supportive personality and for your advices. You were a real family to me. Thank you to Nour, Riham, Jomada, Ahmad Abdallah, Hajar, Maryse, Dima, Maryam, Amani, Ahmad Alaadein, and Ali Ahmad.

Dr. Yehia Taher, Maître de Conférences at the University of Versailles Saint-Quentin-en-Yvelines, I was supposed to write about you in the beginning of the acknowledgements as you are a member of the jury, but I preferred writing to you in the family section. I am doing this because you really had been the number one support from the first day I knew you. Starting from my Master's thesis till this moment, you were always by my side whenever I needed anything on the professional and personal level. Without your support, I would not have been here today. I am really grateful for everything you've done for helping me. Thank you for your endeared and charming personality.

I would like also to thank my cousins Nour, Nibal and Juana for their continuous support and encouragement. A big thank you to my precious brothers and sisters Mohamad, Hussein, Nahida and Lama for being always by my side and to my beautiful nieces and handsome nephews, I love you so much. Thank you, Nader, for being a real supportive brother to me. Thank you Samer, Ayat and Farah. Finally, I dedicate this thesis to my father and mother. I cannot find the suitable words that can express my gratitude to you. I hope that this achievement can make you happy and proud of me. Thank you for always supporting and loving me.

Abstract

Email or Electronic mail is considered one of the most popular uses of Internet. It is a mean that allows information exchange between entities possessing email accounts. It is undeniable that the email system occupies a significant role in today's modern business communication. Exchanged information in emails' texts are usually concerned by complex events such as meeting scheduling, organizing a conference, students applications etc... These complex events can be also considered as business processes in which the entities exchanging emails are collaborating to achieve the processes' final goals. Therefore, the flow of information in the sent and received emails constitute an essential part of such processes i.e. the tasks or the business activities. Such information can be harvested for understanding undocumented business processes of companies and institutions. Extracting information about business processes from emails can help in enhancing the email management for users. It can be also used in finding rich answers for several analytical queries about the employees and the organizations enacting these business processes.

Few are the researches that tackled the problem of extracting business-oriented information from email logs. Up to our knowledge, there exists no approach in the literature that is able to automatically elicit business processes from email logs. In other words, none of the previous works have fully dealt with the problem of automatically transforming email logs into event logs to eventually deduce the undocumented business processes. Towards this aim, we work in this thesis on a framework that induces business process information from emails. Our framework is able to extract the main attributes that constitute an event log from the emails' structured and unstructured content. The overall framework is composed of several components, each dealing with a part of the overall problem. In this thesis, we introduce approaches that contribute in the following: (1) discovering for each email the process topic it is concerned by, (2) finding out the business process instance that each email belongs to, (3) extracting business process activities from emails and associating these activities with metadata describing them, (4) improving the performance of business process instances discovery and business activities discovery from emails by making use of the relation between these two problems, and finally (5) preliminary estimating the real timestamp of a business process activity instead of using the email timestamp for that. Using the results of the mentioned approaches, an event log is generated which can be used for deducing the business process models of an email log. The efficiency of all of the above approaches is proven by applying several experiments on the open Enron email dataset.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 Context and Motivation

The recent development of communication in the cyberspace creates huge amounts of data that can be utilized for several goals. Email is by and large the first and the most popular professional communication and social medium [1]. It is a reliable, confidential, fast, free and easily accessible form of communication. With the ever increasing popularity of emails, it is very normal nowadays that people discuss specific issues, events or tasks using the available email management tools [21]. It provides users the ability to engage in multiple tasks or behaviors simultaneously such as project management, conference organization, meeting scheduling, etc... While its initial use focused on exchanging personal messages between individuals, emails have evolved from a mere communication system to a mean of organizing and coordinating the execution of complex activities (workflows) involving multiple individuals, storing information, or sharing and editing documents.

Due to the huge amounts of sent and received emails by a user on a daily basis, one of the user's core requirements is an efficient management of his/her emails. In recent years, email systems have been working on enhancing the user experience by developing tools for optimizing the management of the exchanged emails. For example, some email tools have helped the user in managing the tasks extracted from emails by organizing a to-do list which shows the tasks progress and deadlines. Other tools have helped the user in organizing the resources associated within the exchanged emails [8], [2], [64], [6].

Email has transformed over the years from a communication medium for simple message exchange, to a "habitat" [18] – an environment where users exchange emails for applying business processes. Exchanging emails becomes essential when applying tasks in organizational processes necessitates the involvement of multiple individuals. Assigning tasks, asking for more information, reporting results - all these activities are enacted via email messages. Therefore, such email messages necessarily contain process-related information that refer to the business process under execution.

A *Business Process* is composed of a set of activities that are applied in a specific sequence to achieve an organizational goal. Each business process is represented by a model i.e. *Business Process Model*. The model represents a series of related tasks or activities to be applied in a specific manner that result

[1] http://onlinegroups.net/blog/2014/03/06/use-email-for-collaboration/

in the desired output. A *business activity* is designed to perform a specific task or action that contributes to a business process. For example, consider the business process model about "meeting scheduling". The activities of this model may include "propose meeting", "refuse meeting", "postpone meeting", "confirm meeting" etc... Each organizational business process can have several occurrences which are called the *business process instances*. A process instance is a specific occurrence or execution of a business process model.

Therefore, the context of this thesis revolves around the analysis of the content of email logs from a business oriented point of view. In other words, instead of dealing with it individually or as a part of a thread, an email can be considered as a contributing entity in an organizational business process. In our context, we are not only interested in extracting valuable information from emails, but also we take into consideration that these information are valuable from a business process oriented point of view. In other words, extracted information should give an indication about the progress of an organizational business process that the email sender/receiver is concerned by.

However, email analysis from a Business Process Management (BPM) perspective has not been thoroughly studied in the literature. Some of the existing works allow the identification of email activities among a predefined set of activities [20], [11], [9]. The email analyzer developed by Van der Aalst [59] necessitates the user interference to extract a process instance from an email log. In [59], they also assume that the tasks names are always available in the email's subject in which they work on a predefined set of activities. Hence, until recently and up to our knowledge, none of the previous works has tackled the problem of extracting business process information from emails *automatically* and without any a priori knowledge for the goal of business process models discovery. Using the available email clients, a user is not able to follow the execution of a business process model. The reason behind this is the lack of the ability for email clients to handle the content of emails as a part of the progress of a business process or workflow.

Transforming email logs into event logs allows us to produce business process models using the available process mining tools. An event log is the dataset used by process mining tools to produce the corresponding business process model. In an event log, each event corresponds to an activity that is executed in the process, where multiple events (ordered by their timestamps) can be linked together as a process instance or case. Hence, an event log can be seen as a collection of cases and a case can be seen as a trace/sequence of events. The produced business process models can provide a clear overview on the processes and the activities in a user email log. Organizing emails as business processes allows the users being able to deal with an email as a part of a process which helps in managing their assigned tasks and tracking the overall progress of the process.

Since emails contain unstructured data such as texts, images, or documents, the main challenge in this case is how to extract such undocumented business process information and transform them into event logs. In other words, an e-mail message (particularly the email produced from a non automated system)

does not include any explicit information about the business activities it contains or its relations to a particular process instance or its relevance to one of the organization's business processes. In addition, the existing frameworks and tools do not provide an efficient approach to extract business process oriented information from the unstructured data of emails which allows the transformation of email logs into event logs for the sake of the discovery and management of the organizational business process. In this research, we tackle this problem by working on the extraction of business process information from email logs by analyzing the content of the emails and their relations.

1.2 Research Questions

Based on the above description of the research context, motivations and challenges, we define some research questions that this work will address. The research scope in this work concerns the the extraction of business process models from emails logs. Hence, the main research question can be summarized as the following: *Knowing that emails are categorized as unstructured data, how can we extract business process models from email logs?* In order to deduce business process models from an email log, the latter should be transformed into event logs. Thus, the main attributes of an event log (process identifier, process instance identifier, activity label, timestamp) are to be extracted from the emails. This research question can be further divided into the following research sub-questions:

- *Without having any a priori knowledge about the process topics or their number in an email log, how can we deduce what business process model topic each email is concerned by?* The answer to this question allows us to deduce the process identifier attribute of each email or in other words to which process topic it belongs.

- *Knowing that each process model has several executions or instances, how can we deduce to which business process instance an email belongs to?* The answer to this question allows us to deduce the business process instance identifier attribute of each email.

- *Without having any a priori knowledge about the activity types present in an email log, how can we deduce the activity type(s) that an email contains?* The answer to this question allows us to deduce the activity labels of the events extracted from the emails.

- *Knowing that the email timestamp is not always an accurate indicator about the email activities occurrences, how can we estimate the occurrence time of email activities?* The answer to this question allows us to deduce the timestamp attribute for each event or activity in an email.

- *What is the relation between emails business activities discovery and emails process instances discovery? Can the relational treatment between these*

two problems increase their performance efficiency? The answer to this question allows us to discover how activities in emails can give a good indication about their relation (whether the emails are related to the same process instance or not) and vice versa.

1.3 Contributions

To fulfill our research objectives, we have built a framework underlying multiple combined approaches. The framework is composed of several components where each component applies a specific approach that achieves a part of our research work. The contributions of our thesis work can be summarized as follows:

- We work on an approach that can find for each email the business process topic it belongs to. After pre-processing the email and transforming it into the amenable data format, analysis is applied on the email's body and subject to discover the business process topic it is concerned by. This approach depends mainly on unsupervised learning i.e. clustering. A study is provided that explains our choice of the similarity measurements and clustering techniques. We take into consideration in this approach that the system should have no a-priori knowledge about the topics contained in an email log. In addition, our approach is automatic in a way that users should not interfere or help in discovering the process topic of an email.

- A process instance discovery approach is introduced where we work on finding the business process instance an email belongs to. In this approach, we formulate a distance function that is the most efficient in terms of performance for clustering emails into business process instances. The distance function is defined in terms of a combination of some attributes. These attributes are actually extracted from the structured and unstructured content of an email. The efficiency of the distance computation depends crucially on the chosen attributes. Therefore, we present multiple combinations of email attributes and prove their validity using some examples and counter examples. On contrary to other existing works, this approach is automatic i.e a user effort is not demanded for choosing the attributes to be used in the distance computation. The efficiency of the results obtained in this phase is considered critical, especially that this phase is an intermediary step in the overall framework.

- We introduce an approach as a solution for extracting business activities from emails, and for annotating the elicited activities. Specifically, we start by the first hypothesis, where we build an approach that can extract a single activity per email. We then move to the second hypothesis where we present an approach that, using customized extractive business oriented summarization of emails and clustering of business-oriented sentences, discovers and labels one or multiple business activity types in an email. This is followed by automatically associating each activity type

with a set of metadata that describes it. A predefined set of activities is not always available since there is no a priori knowledge about the process topics available in the email log and there is no knowledge about the existing business activities. Moreover, according to our analysis of emails, we discover that remarkable number of emails contain more than one activity at a time. For this reason, in our approach for email business activities discovery, we overcome this by being able to extract one or more activities from a single email. In addition, in this approach, we work on specifying for each email activity a set of information or metadata which enrich and describe the corresponding activity. We tackle the challenge when several activities are present in one email where the information associated to an email such as attachments, URLs etc.. should be correctly connected to the email activities. This step in our approach opens a door for a high level analysis in which many analytical queries can be answered using to the extracted metadata.

- We propose an iterative relational approach that uses information about email business activities to identify emails of the same process instances and vice versa. In particular, we investigate (1) how information about email activities can assist with finding emails of the same process instances, and (2) how features of emails of the same process instances can assist with the classification of emails business activities.

- We propose a preliminary approach that can estimate the occurrence time of an event or email activity. In other words, the approach helps in the extraction of temporal relations between the email business process activities, the temporal expressions and the email timestamp. Thus, we address intra-relation identification between business process activities and/or temporal expressions mentioned in an email, and the relation identification between the email activities and the email sending time (timestamp).

- We provide a usecase that clarifies the steps of transforming an email log into an event log on a concrete example which explains the overall job and target of the thesis framework. The usecase is applied on an email log that contained emails revolving around multiple business topics and activities.

- The efficiency of all the above approaches is evaluated using multiple email folders from Enron email dataset [2]. This dataset was collected and prepared by the CALO Project [3] (A Cognitive Assistant that Learns and Organizes). It contains data from about 150 users, mostly senior management of Enron, organized into folders.

[2] https://www.cs.cmu.edu/ enron/
[3] http://www.ai.sri.com/project/CALO

1.4 Thesis Structure

Beside the introductory chapter (chapter 1), the thesis consists of 8 other chapters. In chapter 2, we provide a motivating scenario that clarifies the main challenges and motivations of our thesis work. In this chapter, we also show and briefly explain the overall framework of the this work. The literature is studied and discussed in chapter 3, where we present several methods, approaches and techniques that fall into the scope of our proposed approach.

The framework approaches are presented in chapters 4, 5, 6, 7, and 8. In chapter 4, we introduce the approach for email business process topic discovery. In this phase, the email log is pre-processed and transformed into the format accepted by the analysis tools. This is followed by applying the unsupervised learning on the pre-processed email log to discover the business process topic of each email. In chapter 5, a baseline process instances discovery approach is proposed where we choose a distance function that efficiently cluster emails according to the business process instances they belong to. In chapter 6, business process activities are extracted using a supervised learning approach. In addition, we introduce a method that associates each of the extracted activities with a set of metadata describing it. We then work in chapter 7 on the enhancement of the performance of the last two approaches by making use of the relation between both problems represented in chapters 5 and 6. A relational classification approach is proposed for this purpose. At the end, we work in chapter 8 on estimating the timestamp of the occurrence of an email activity. This is followed by a usecase that describes the application of the overall framework on an example email dataset. In each of these chapters, the steps of the approaches are detailed and evaluated using some experiments. Each approach is compared with other existing approaches.

Finally, the thesis contributions and results are summarized in chapter 9 and a brief overview on the future works are presented in that chapter.

CHAPTER 2

Motivating Scenario and Overall Framework

Contents

Figures

2.1 Motivating Example and Challenges

With the ever increasing popularity of emails, it is very normal nowadays that people discuss specific issues, events or tasks using the available email management tools [21]. For example, scheduling a meeting, organizing a conference, or applying for a travel grant. Although the available email management tools are considered as ubiquitous social media, they do not offer a support for process-oriented organization of emails. Mainly, all available email programs are designed to handle emails individually or as a part of a thread such as Outlook, Gmail, SendInBlue, Front... etc. It is widely recognized that professional emails are a valuable source of undocumented business oriented information. In fact, organizing emails as business processes has several advantages. Users will be able to deal with an email as a part of a process which helps in managing their assigned tasks and tracking the overall progress of the process.

Using the extracted information, email logs can be transformed into event logs. These event logs are amenable as an input for the business process mining techniques. While useful, existing proposals in the literature do not work on building a complete automatic framework that takes a raw email log and transforms it into an event log that is compatible to the available process mining techniques. Therefore, the target is to build business process models from email exchanges. For this reason, we work on transforming the email logs into event logs where we can find the main attributes of the events with their values.

The main attributes that should be present in an event log are:

1. Process Identifier (ProcessID): this attribute indicates to which process model an event belongs to. Knowing that each process model is concerned by a specific topic, the process identifier correlates each process model with its topic. To project this definition on our work, this attribute indicates to which process topic an email belongs to.

2. Process Instance Identifier: knowing that each process model has several executions in a log, each one of these executions is considered as a process

instance. Therefore, each set of events belong to one of these executions. Hence, an event will be associated to a process instance identifier.

3. Activity Type: each event represents an activity or a task that is applied. Each task has a specific name or label such as *confirm meeting*, *refuse application* etc...

4. Timestamp: it represents the time at which the event has occurred. This will help in specifying the sequence of occurrence of the events for modeling the business process.

Indeed, the elicitation of such kind of business information opens up the door to many business analytics by leveraging the emails to answering several analytical queries such as:

Q_1 What are the business activities executed by a specific employee? For example, a manager would like to know the productivity of a specific employee or to know the contribution of an employee in a specific process. (to identify time-consuming tasks that are not known to be assigned to him).

Q_2 How many times a user applied an activity? For example, an employee may wish to know how many times he/she applied for a travel grant during a specific period of time.

Q_3 What are the groups of people doing similar work? For example, a manager would like to know who are the people that apply similar types of activities. This may help in organizing working groups. An employee may wish to benefit from the experience of another employee that applied the same type of activity before.

Q_4 What is the average duration of a business process? This can be computed by averaging the time taken by all process instances of the same process model. This helps managers and employees to identify the expected duration of a specific process beforehand according to the previous executions of the same process.

Q_5 Which process instances take the longest time to be achieved? Normally, process instances should take similar periods of time. However, when a process instance takes a long duration for its achievement, this gives an alert about an abnormality. Knowing that there is a problem would help in identifying the reason behind it i.e identifying the reason behind the time delays. Overall time delays may be caused by partial time delays in several activities of the process or by a time delay in only one activity due to loops (an activity is repeated several times to be completed).

Q_6 How many instances are enacted in a specific period of time? This can also be related to the productivity of the enterprise. For example, a manager would like to know how many times a specific type of trading is enacted. It

may be also related to the interactions between employees. For example, someone may be interested to know how many meetings were organized for a specific group (or the number of organized events).

Q_7 Which process instances involve specific entities? For example, which mission funding application instance required the involvement of the department director? This question may help in identifying exceptional situations or inefficient process executions. This kind of query would help in mitigating similar problems that may occur in the future while applying the same type of process.

In addition to the already mentioned queries, building such process models provides the user a better understanding and management for his/her emails. Instead of dealing with emails separately or as threads, the user will be able to deal with them as parts of business processes. This provides the user a better organization of his/her processes which may span on a long time period or which may include a very big number of emails. The extracted models can be also used as a support for automation using Business Process Management (BPM) system.

We present a simplified example of an email log for clarifying the main goal of our work. For simplicity reasons, we use in this example the emails taken from a Ph.D student. An email inbox contains emails belonging to several processes that the student is directly or indirectly concerned by. As most of emails logs, this log contains some personal messages which are not interesting for our analysis. We exclude such kind of messages from the email log for a better efficiency of the analysis results. We are only concerned by emails that are business process oriented i.e. that contain business process activities or information about these activities. By spanning the student's email log, we realize that the emails are mainly concerned by mission demand applications such as applications to attend a conferences abroad or to get a refund for the summer school registration fees. We also found numerous emails about scheduling meetings with different entities (Ph.D student and the responsible people in the research lab or the supervisors).

Such kind of email logs is an example of a professional exchange of messages that would be helpful in our analysis. Applying the queries mentioned earlier in this section, a student may wish to know how many missions he has applied to during the scholar year. He may also want to compute the duration consumed to get an acceptance for a specific application. A student will also be able to organize his emails as a sequence of activities. He/she may learn from previous executions of a specific process activities to better execute current ones. All the above mentioned application helps in improving the experience of the student in using emails for managing his/her daily/monthly/yearly tasks.

Figure 2.1 includes an example of some emails of a Ph.D student. If we quickly look at these emails, we realize that they belong to different process topics or domains. Some emails are talking about scheduling a meeting, others about mission demand application. We realize also in this email log that emails

are separately treated. As we will explain in a later chapter, these emails are treated individually without taking into consideration any information about the threading relations between them since using threads may be a drawback in some cases and may reduce the efficiency of the obtained results. Each email is characterized by a set of different attributes. We extract for each emails its subject, sender, receiver(s) emails(s), the body of the email and its timestamp. In our work, the analysis of the emails crucially depends on the body and the subject of the email. It is obvious that in most cases the email's body and subject contain the most important information to be exchanged.

EmailID	Sender	Receiver	Subject	Timestamp	Body
1	diana.jlailaty@gmail.com	missionjc@dauphine.fr	mission demand	2016-04-19 09:51:00	Please find enclosed my mission application for the Summer School to be held in Urrugne from 5 to 10 June 2016
2	missionjc@dauphine.fr	diana.jlailaty@gmail.com	mission demand	2016-04-20 11:02:00	I note that you have not yet linked to the web page LAMSADE site. Your mission will be taken into account. Thank you to arrange and contact us again as soon as possible.
3	diana.jlailaty@gmail.com	missionjc@dauphine.fr	mission demand	2016-04-20 02:35:00	Thank you for considering my request. This is the link to my web page LAMSADE: http://lamsade.dauphine.fr/ djlailaty/ I am available for any further information.
4	missionjc@dauphine.fr	diana.jlailaty@gmail.com	mission demand	2016-04-20 03:08:00	Thank you Diana, There is an error in the web address, add to that In early WWW find your page. Mission signed the request is attached to this email. I invite you to visit the secretariat eleni with the mission request. You have been granted the amount of 550 euros.
5	diana.jlailaty@gmail.com	daniela.grigori@.dauphine.fr	meeting	2016-03-29 10:34:00	What time is the meeting today?
6	kbelhajj@googlemail.com	diana.jlailaty@gmail.com	meeting	2016-03-29 10:42:000	Is it scheduled for 11am in Daniela office.
7	diana.jlailaty@gmail.com	daniela.grigori@dauphine.fr	postpone the meeting	2016-03-29 10:47:00	Can the meeting today be on 2:00 pm instead of 1:30 pm? Because I want to attend phd defense and pot of my colleague.
8	daniela.grigori@dauphine.fr	diana.jlailaty@gmail.com	postpone the meeting	2016-03-29 10:52:00	For me, having the meeting in this time is Ok.
9	kbelhajj@googlemail.com	diana.jlailaty@gmail.com	postpone the meeting	2016-03-29 10:57:00	It is good for me also.
10	diana.jlailaty@gmail.com	daniela.grigori@dauphine.fr	set a meeting	2016-05-03 14:22:00	When will be our next meeting? I am available the whole week.
13	diana.jlailaty@gmail.com	daniela.grigori@dauphine.fr	set a meeting	2016-05-03 14:22:00	When will be our next meeting? I am available the whole week.
14	daniela.grigori@dauphine.fr	diana.jlailaty@gmail.com	set a meeting	2016-05-03 14:50:00	I am available tomorrow for the meeting, wednesday and friday.
15	kbelhajj@googlemail.com	daniela.grigori@dauphine.fr	set a meeting	2016-05-03 16:43:00	Is the meeting tomorrow 10h good for both of you?
16	kbelhajj@googlemail.com	diana.jlailaty@gmail.com	set a meeting	2016-04-22 17:50:00	What have you done with the mission application? Did you visit the secretariat Eleni for that? You should do this as soon as possible.
20	diana.jlailaty@gmail.com	missionjc@dauphine.fr	mission demand	2016-06-25 10:35:00	Please find attached my mission demand to grenoble conference and summer school.
21	missionjc@dauphine.fr	diana.jlailaty@gmail.com	mission demand	2016-06-26 11:20:00	Your mission will be taken into account. But before we confirm this, you need to specify the detailed costs. Thank you to arrange and contact us again as soon as possible.
22	diana.jlailaty@gmail.com	missionjc@dauphine.fr	mission demand	2016-06-27 14:05:00	Thank you for considering my demand. Please find attached the detailed cost of my trip.
23	missionjc@dauphine.fr	diana.jlailaty@gmail.com	mission demand	22016-06-28 15:01:00	Mission signed the request is attached to this email. I invite you to visit the secretariat eleni with the mission request. You have been granted the amount of 600 euros.

Figure 2.1: Example of email exchanges for a Ph.D student

As we have seen in figure 2.1, each email is described by a set of attributes extracted from the email log. These attributes and their values are used by the approaches conducted in this thesis where the final goal is to deduce event logs and consequently process models from email logs. The overall framework is divided into a set of approaches where each approach is responsible of obtaining a specific part of the final result. To transform the event log into an email log, the process ID, process instance ID and the activity labels should be defined for each email. Thus, each approach in our framework will be conducted to extract one of these attributes as a part of the event log.

The input email log can contain several process topics which the users and the analysts may have no apriori knowledge about them. This makes the task of defining a process ID for each email a non obvious one. In addition, an email may not contain any business process activities. It may also contain several business activities. Thus, it is non-trivial to obtain the number and labels of activities in an email. On the other hand, knowing that two emails may contain the same activities but belonging to different process instances, the analyst should be able using the structured and unstructured content of the email to deduce to which process instance an email belongs. These challenges and other will be detailed later in this chapter.

Therefore, to obtain the final results several approaches are enacted where each approach may or may not use as an input the output of a previous approach. Some approaches in this thesis are relational in an iterative manner such that the results of one approach are used in the other approach iteratively and vice versa (as long as the results of the first approach are changing, we use them as a part of the input of the second approach).

To give a brief overview on the main tasks of the approaches of the overall framework, we will start by an example of emails from an email log. This will help in clarifying the methods used to obtain the final results and the applications where we make use of the obtained results. In each email example, we show the main attributes: sender, receiver, subject and email body.

Let us consider the following three mails from an email log of a Ph.D student taken from figure 2.1:

From: diana.jlailaty@gmail.com
To: missionjc@dauphine.fr
Subject: travel grant application

Dear,
Please find attached the travel grant application containing all details for the summer school that is taking place in Paris from June 5th to June 10th 2018.
Thanks,
Diana

From: missionjc@dauphine.fr
To: diana.jlailaty@gmail.com
Subject: travel grant application

Dear Diana,
Your mission will be taken into account. However, please link to the webpage of LAMSADE site. It is important for signing your travel grant.
regards,

From: diana.jlailaty@gmail.com
To: missionjc@dauphine.fr
Subject: travel grant application

Dear,
Thank you for considering my request.
Here is the link to my LAMSADE webpage http://lamsade.dauphine.fr/djlailaty.
I am available for any further information
Thanks,
Diana

As we can see, these 3 emails are exchanged between a Ph.D student and the administration of the university. The student is requesting a travel grant from the university to attend a summer school. For simplicity reasons, we only show these 3 emails that demonstrate a part of the business process followed to achieve the final goal which is travel grant acceptance or refusal.

The examination of these emails reveals that each one of them contain 1 or more business activities which form a part of the business process. Since the student may apply to multiple grant applications, then his/her email log will contain emails belonging to different executions of the same process model. Therefore, we can find in one email log multiple emails containing the same set of activities but belonging to different process instances. In these example emails, we take 3 emails that are a part of the same process instance or execution of the process model topic: *travel grant application*.

As our final goal is to transform an email log into an event log and consequently deduce the business process models of the input email log, a solution is needed that can extract the business process information from an email log. Elaborating a solution that is able to deduce such kind of business information and consequently elicit business process models from email logs raises a number of challenges:

C_1 Not all emails in an email log are business oriented. Normally, most of email logs of students, professors, employees etc.. contain emails talking about personal issues. These emails are considered out of interest in our analysis. We are mainly concerned by emails that are business process-oriented. Such emails contain business process activities or information about applied activities. In fact, it is necessary to differentiate between business-oriented and non business-oriented emails in an email log. Otherwise, the personal emails will affect inconveniently the efficiency of the obtained results.

C_2 Neither the user nor the analyst would have an apriori knowledge about all the process topics tackled in an email log. The number and topics of the business processes in an email log may differ from one user to the other. Therefore, in order to have an efficient framework that can work well with all input email logs, the approaches conducted in this framework should be built on the fact that there is no apriori knowledge about the business process topics of an input email log. It is the job of these approaches to deduce for each email the process topic it belongs to.

C_3 Multiple emails in a email log may contain the same activities where each email belongs to a different business process instance. Each email log may contain several executions of the same process model. Therefore, it is essential to specify for each email the process instance it belongs to according to the email content and other attributes. Thus, each process instance in an email log includes a set of emails where each email contains a set of business activities. The approaches in this thesis should be able to specify to which process instance each email belongs.

C_4 Most of the previous works have tackled the problem of email activity discovery by specifying one activity or task for an email. However, in this real case, we assume that an email may encompass multiple activities. The above example emails prove the correctness of this assumption. For example, in the second email, the mentioned activities are: *link webpage* and *sign travel grant*. After solving the first challenge (excluding personal emails from an email log), we should consider that each email may contain more than one activity to be extracted as a part of the business process.

C_5 As a continuation of the previous challenges, email activities are associated with a set of information that we call metadata such as the actors performing an activity or an attachment or URL that come with an activity. However, as mentioned in the previous challenge, each email may contain multiple business activities. Therefore, it is a non trivial task to connect the metadata contained in an email on the set of activities an email contains. We should be able to associate to each email activity the correct set of information corresponding to it. We need a means to correlate the information abstracted from the resources associated with the emails.

2.2 Framework Overview

In this section, we will present the overall framework followed in this thesis. It is composed of three components. The first component: **Email Log Preprocessing** which is considered a preparatory phase that takes as an input a raw email log and prepares it for further analysis. Once the data is prepared in the first component, the transformation of an email log into an event log starts. This transformation is divided into two main components: **Process Topic Discovery** where each email is associated to a business process topic and **Process Models Discovery** which is a composite component consisting of several other components to output the business process models of an email log.

Figure 2.2 shows that our approach is a three-component process that given a set of emails as input, produces a set of process models. The emails can be supplied by an individual or an enterprise, for example, a student or a researcher, or an institution.

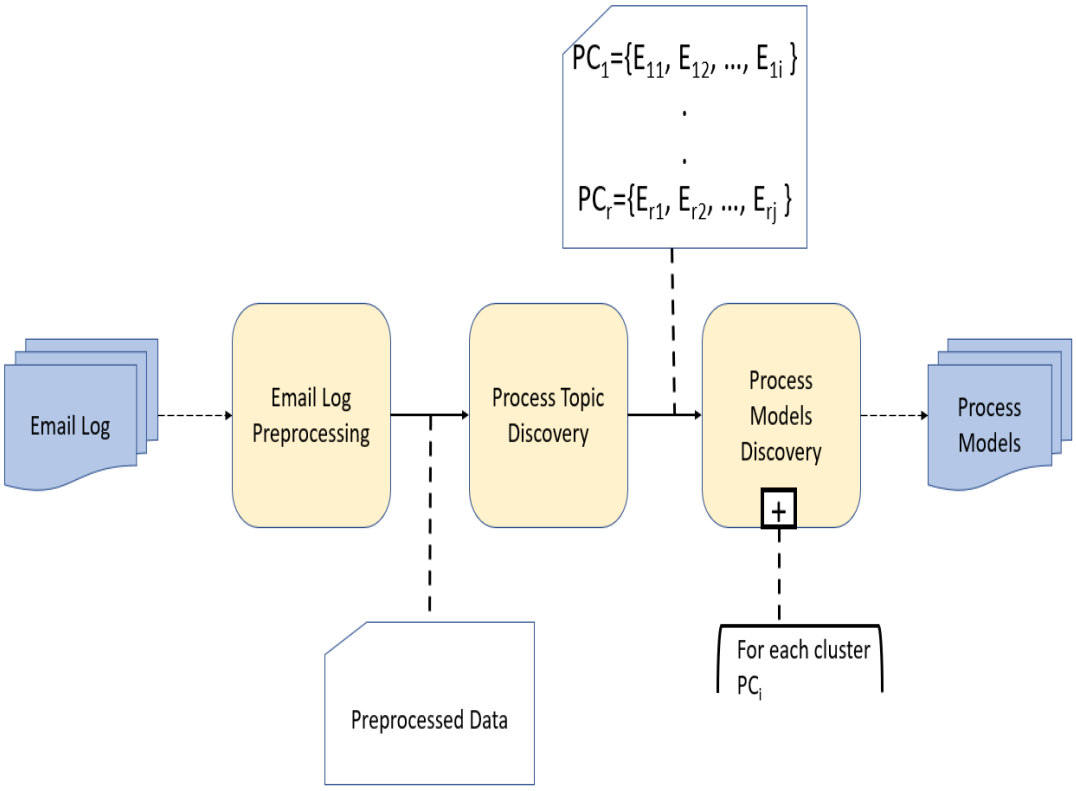

Figure 2.2: General framework overview

Starting with the first component: **Email Log Preprocessing**, the input data of this component is an email log. As described earlier, an email log is a set of emails exchanged between different entities (people, companies etc...) for a specific purpose such as scheduling a meeting, organizing a conference, purchasing an item etc.. Each email is represented by some attributes describing it: email subject, sender, receiver, email body, and email timestamp. Mainly, the email's main content is its body and subject texts, which are considered in the category of unstructured datatype. Knowing that structured data is well organized, follows a consistent order, is relatively easy to search and query, and can be readily accessed and understood by a person or a computer program, dealing with unstructured data, on the other hand, is challenging since it tends to be free-form, non-tabular, dispersed, and not easily retrievable; such data requires deliberate intervention to make sense of it. One must take considerable time to preprocess unstructured data with fixed fields so that they can be queried, quantified, and analyzed with data mining techniques. The email data should be cleansed and transformed into the format expected by the analysis

tools. Four main steps are applied in this component:

1. Data Cleansing
2. Data Representation
3. Features Selection
4. Verb-Nouns Extraction

Once the unstrutured email texts are preprocessed and prepared for analysis, the analysis component: **Process Topic Discovery** is enacted. Emails of an email log are concerned by different topics or what we call them *business process topics*. Every group of emails is exchanged for performing a process about a specific topic. In the second component (Process Topic Discovery phase), the main goal is to group emails according to their business process topics where each email is associated to a process identifier (ProcessID). The results of the applied approach in this component will also help in analyzing emails. Instead of dealing with an email log as a whole, we will be able to work on emails of different processes separately which reduces the complexity of the approaches applied later.

The results of the second component which is a set of clusters where each cluster PC_i contains emails $\{E_{i1}, E_{i2}, ..., E_{i3}\}$ belonging to the same process topic are used as an input for the third component: **Process Model Discovery**. The approach in this component is repeated on all process topic clusters. This component is a composite one. It is composed of several other sub-components as illustrated in Figure 2.3 where analytical approaches are applied in each sub-component to extract business process information from emails. Once these information are extracted for each process topic cluster, the email log is transformed into an event log hence the process models of an email log can be deduced.

The composite component is made up of 4 main phases as shown in 2.3: two of them are baseline approaches in which they only use the output of the previous component and are applied only once. The other two components are relational approaches in which they use the results of the baseline ones and are applied several times in an iterative manner as long as the results are changing.

Given a cluster of emails that belong to the same process model topic where each email is associated to a process identifier PiD, a *baseline* process instances discovery sub-component is enacted where such emails are first sub-grouped into preliminary groups each representing a process instance. The same input is provided to the *baseline* email process activities discovery sub-component where activity types discovery, labeling and metadata extraction takes place. The outputs of both baseline approaches are used as an input for the *relational* approaches. Relational email process instances discovery uses the results of the relational email process activities discovery and vice versa. We apply the relational approaches for checking whether using information from other sub-components can help in the improvement of the results of others i.e. better discovery of instances and activities in an email log.

In order to apply the baseline and the relational approaches, we build algorithms that use different data mining techniques.

After applying the framework approaches, the business process information about the email log become available i.e. the process identifiers, process instance identifiers, and the activities labels (adding to it some other information like the timestamp of an activity). We can consider that the email log is transformed into an event log that is amenable to be an input to process mining tools that outputs process models.

Going back to our example emails about *travel grant application* process topic for a Ph.D student, if the above framework is applied we can deduce for each email a processID, processInstanceID, activity labels and their timestamps. If we do this for all emails of the same process topic, we can deduce the process model shown in figure 2.4. This process model describes the steps of the process followed in an institution by a student who wants to apply for a conference mission or a summer school grant.

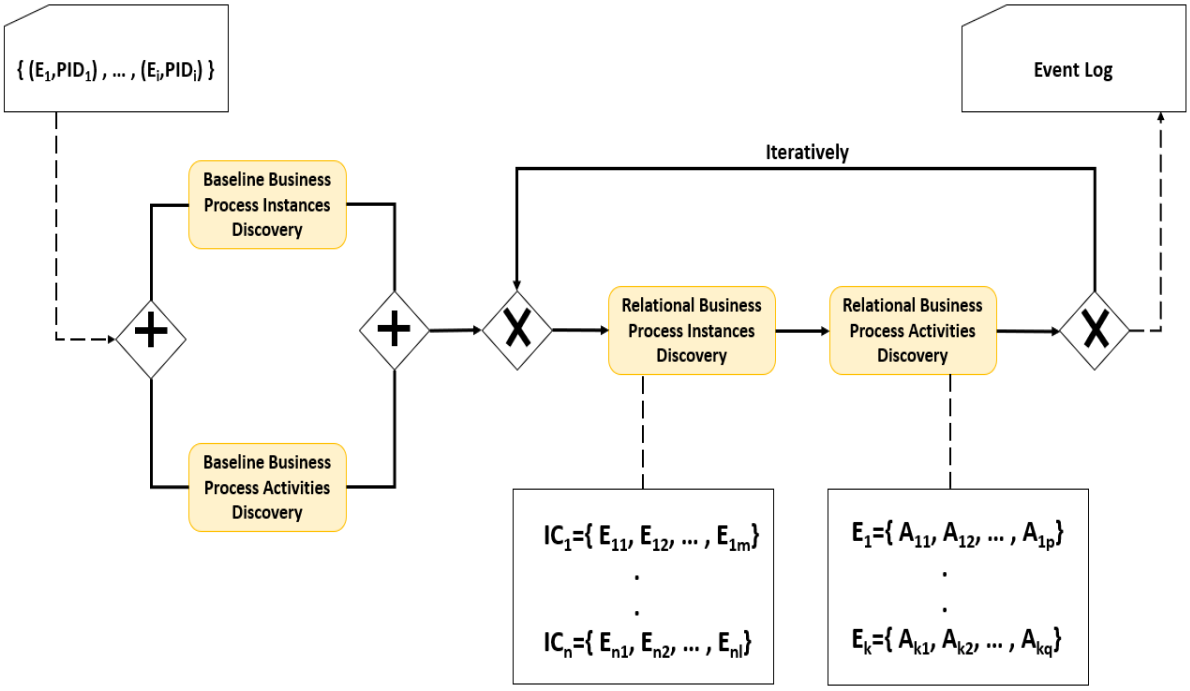

Figure 2.3: Sub-components of the composite component 3.

The initial input in figure 2.3 is a set of emails where each email E_i is associated to process identifier PID_i. This input is used by both the baseline approaches for process instances discovery and process activities discovery.

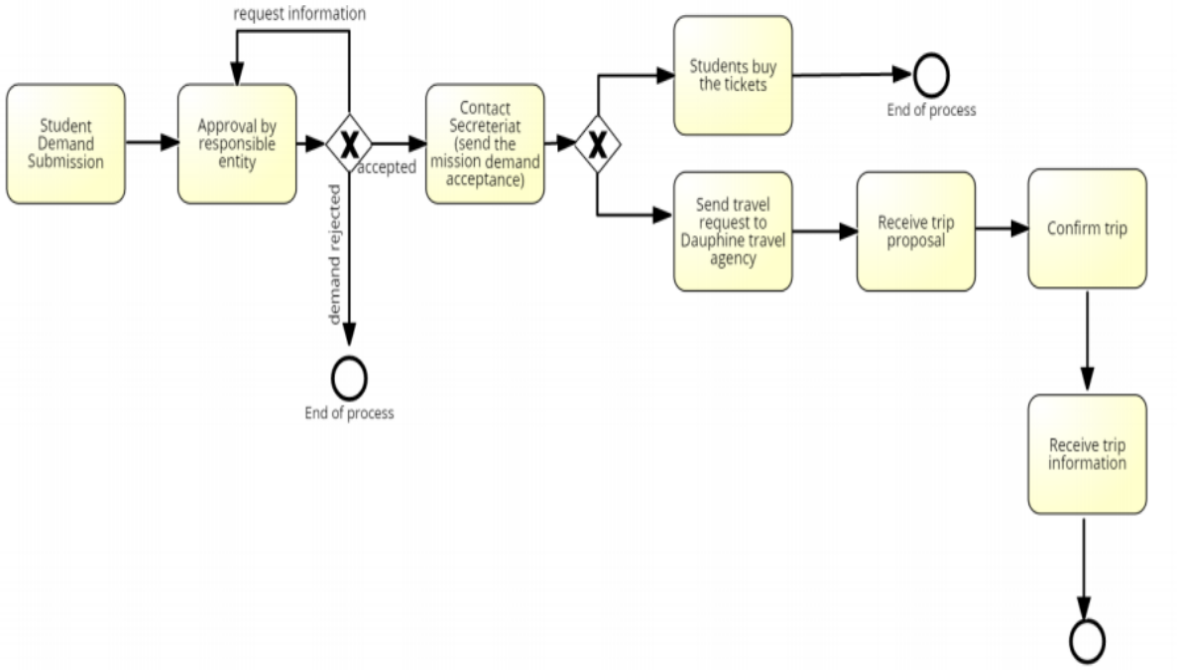

Figure 2.4: Example business process model for travel grant applications

Then, iteratively the relational approaches of instances and activities discovery are enacted. The relational process instances discovery phase produces clusters of emails where each cluster IC_i contains a set of emails belonging to the same process instance. Whereas the relational process activities discovery phase produces for each email E_i the set of business activity types A it contains. The extracted information (combined with the activity timestamp extraction) are collected to form the event log as an output.

CHAPTER 3

Background and State of the Art

Contents

Figures

The overall framework presented in chapter 2 includes multiple approaches which cover mainly 2 scientific fields. Specifically, we are concerned by (1) Email Management and (2) Business Process Management (BPM). The literature provides a huge number of works related to each of these two fields. However, until recently, very few are the works that combine both concepts in one framework. Email analysis from a BPM perspective has not been thoroughly studied in the literature. In this research, we combine these two concepts i.e. extracting business process information (business process models) by analyzing the content of the emails. In this chapter we will present several related works grouped into different categories. First, a study about *Email Management*: the commercial softwares, email foldering, email summarization and email task management. Then, we will present the related works concerned by *Process Model Discovery vs Email Analysis* that mostly revolver around: process instances discovery from emails and email process activities discovery.

3.1 Email Management

Email management is the process of collecting, storing and operating email data. The tools of email management are used to manage high volumes of inbound

and outbound electronic messages. Emails are rarely a standalone source of information, they usually contain pointers to further information such as attached files, links to webpages or references to other resources. Due to the remarkable amounts of valuable information that can be contained in email logs, email analysis and information extraction are considered essential for partly or fully "understanding" the email content. The goal behind email management is to be able to cope with the huge number of received and sent emails to ensure an efficient storage and manipulation.

There exist numerous commercial softwares that are used by businesses to provide customer support for email management. This type of softwares help agents to track and respond to email requests more easily. Another useful functionality is an efficient email receiving which helps to minimize spam. Other key features are data enhancement such as providing details about an email's author. The software may also help the user to understand and analyze the content of an email. Top email management solutions also offer email archiving and quick retrieval. There are some popular email management systems:

SendInBlue Email: is a tool that empowers companies to build and grow relationships using email campaigns, sophisticated transactional emails and automation of marketing workflows. It incorporates advanced email features, allowing users to know when recipients have opened their emails and adjust the content proactively to increase engagement.

Front: allows the user to bring all communication under the same roof by gathering messages into a consolidated inbox. Front also seamlessly integrates with many other popular software, including Asana, Salesforce, Intercom, Github, Trello and Slack, while the API allows development of custom integrations to cater to specific business requirements. The built-in analytics tools allow effective monitoring of performance and automatically track various performance metrics, including messages handled/user, time-to-reply and hourly messages handled.

Zoho Workplace: is a collaboration software that provides a complete set of solutions that not only helps the users collaborate but also create and communicate with their teams. It bundles its apps on email, document management, presentation, chat and other communication tools in one platform. These are: Zoho Mail, Zoho Connect, Zoho Chat, Zoho Writer, Zoho Show, Zoho Sheet, Zoho Docs, Zoho Sites and Zoho ShowTime.

Yesware: is an all-in-one sales toolkit. With this tool, the users can connect with prospects, monitor customer engagement, and complete deals, right from their inbox. The tool allows the users to track emails and work more efficiently, right from their Outlook or Gmail inbox. Understand their prospects by utiliz-

ing data on when they read their email, clicked on the link, and which email templates are the main features of this software.

Microsoft Outlook: is an email organizer with a built-in calendar and other tools that help people communicate and stay updated. This premium application from Microsoft also has a built-in Skype so users remain connected with their contacts through chat, voice and video without leaving the platform. With Outlook, users can remain focused on the most important things with an inbox where essential emails from one or more accounts are stored. This allows them to order and manage their priorities. Additionally, Outlook can integrate with a number of partner apps and services, extending its use and allowing users to manage different aspects of their lives within the software itself.

Gmail: is a free webmail system and email service from Google which users can access on a web browser, through its Android and iOS mobile app, or using third-party programs. Gmail uses conversation threading which stacks and organizes messages and conversations into threads, enabling users to access previous related messages they sent out. The webmail system automatically scans emails to check for attachments that may contain viruses or malware. It offers a spam filtering feature which automatically recognizes and tags spam messages, and stores them in a spam folder.

However, to be effective, email management tools must be able to accurately understand the content of email. It is not enough to manage emails by organizing their appearance such as threading or by associating the email software with a calendar to keep track of the to-do list etc.. To efficiently manage emails, it is essential to discover their topics and the tasks they contain which is absent in the previously presented systems. The related works of "Email Management" topic can be categorized into 2 main categories: (a) Email Foldering (b) Email Summarization.

3.1.1 Email Foldering

People use email to manage everyday work tasks, using the inbox as a task manager and their archives for finding contacts and reference materials. Users deliberately create folder structures or tags which helps in reducing the complexity of the inbox. Without folders, important messages may be overlooked when huge numbers of unorganized messages accumulate in an overloaded inbox [4, 13, 63].

We present here the works related to dividing emails into folders according to their topics or priorities which is categorized into supervised and non supervised approaches:

Among the supervised mining techniques we mention the works of:
GNUsmail Carmona-Cejudo et al. [8] is an open-source framework for online adaptive email classification, with an extensible text preprocessing module, based on the concept of filters that extract attributes from emails, and an

equally extensible learning module into which new algorithms, methods and libraries can be easily integrated. GNUsmail contains configurable modules for reading email, preprocessing text and learning. In the learning process, the email messages are read as the model is built because email messages are analyzed as an infinite flow of data.

GNUsmail is made up of two main modules: (1) Reading email and text preprocessing module, and (2) Learning Module. In (1), the reading email module can obtain email messages from different data sources, such as a local filesystem or a remote IMAP server. The text preprocessing module is a multi-layer filter structure, responsible for performing feature extraction tasks. GNUsmail performs a feature selection process using different methods for feature selection [23]. In (2), the learning module of GNUsmail offers three updateable classifiers from the WEKA [29] framework, and more can be easily added though.

Thus, GNUsmail incorporates a flexible architecture into which new feature extraction, feature selection, learning and evaluation methods can be incorporated. In the enhanced version of the framework, they incorporate recently proposed evaluation methods for online learning with concept drift. Such evaluation methods improve the prequential error measures by using mechanisms to reduce the effect of past examples, such as sliding windows and fading factors.

Yoo et al. [67] develop a personalized email prioritization method using a supervised classification framework. The goal is to model personal priorities over email messages, and to predict importance levels for new messages. They focus on analysis of personal social networks to capture user groups and to obtain rich features that represent the social roles from the viewpoint of a particular user. In this work, they also develop a novel semi-supervised learning algorithm that propagates importance labels from training examples to testing examples through message and user nodes in a personal email network. These methods together enable the analyst to obtain an enriched vector representation of each new email message, which consists of both standard features of an email message (such as words in the title or body, sender and receiver IDs, etc.) and the induced social features from the sender and receivers of the message. Using standard Support Vector Machines (SVMs) as the classifiers, the novel part of their SVM is the enriched representation of each input email message, especially in the part that represents the contact persons. They explore three different types of enriched features that are automatically induced based on personal social networks as follows:

- Clustering contact persons based on personal social networks to capture social groups among senders and recipients, which can be learned from personal email messages without importance labels (unsupervised learning).
- Measuring social importance of contacts to capture leadership levels of individual contacts. They claim that because personal social networks are different from user to user, using multi-dimensional metrics to characterize different users would lead to more efficient predictions than using any single metric alone.

- Semi-supervised importance propagation where the personal social network of each user can be used to propagate the importance scores from messages to contacts, then from contacts to messages, and repeat the propagation until all the scores are stabilized. They claim that this will leverage the transitivity of importance scores through personal social connections.

Figure 3.1 shows the clusters produced using the Newman clustering algorithm based on the email contact network of a user: nodes are the senders, and node sizes are adjusted to reflect the average importance of members in each cluster. In this figure, they shows us how they represent the average importance of contacts.

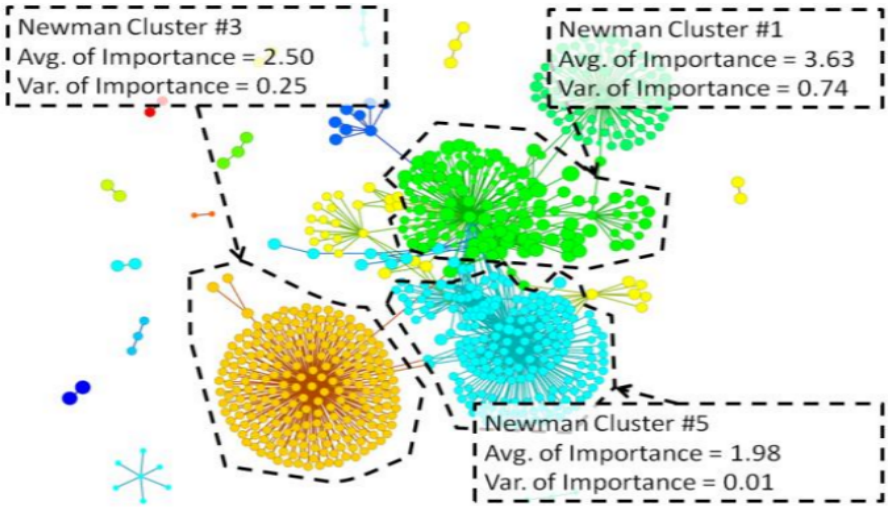

Figure 3.1: Clusters produced using the Newman clustering algorithm in [67].

In Koprinska et al. [39], the authors consider two e-mail classification tasks: automatic e-mail filing into folders and spam e-mail filtering. These tasks are formulated as supervised and semi-supervised machine learning problems. In a supervised setting, given a supervision in the form of a set of labelled training examples (e.g. e-mails labelled as belonging to different folders such as work, teaching etc, or as spam and non-spam), their goal is to build a classifier, that is used to predict the category of an unseen incoming e-mail. In a semi-supervised setting, the goal is the same but the learning is achieved with less supervision, i.e. using a small set of labelled e-mails, by taking advantage of the easily available unlabelled e-mails. In both supervised and semi-supervised setting for e-mail classification they study the application of Random Forest (RF), which is one of the recent ensemble techniques. RF is particularly suitable for classifying text documents as it is fast, easy to tune, and can handle large feature sets. They compare the learning behaviour of RF with other algorithms such as Support

Vector Machines (SVM), Naïve Bayes (NB) and Decision Trees (DT), and show that RF outperforms them in both settings.

In the work of Alsmadi et al. [2], a set of personal emails is used for the purpose of folder and subject classifications. Five classes are proposed to label the nature of emails users may have: Personal, Job, Profession, Friendship, and Others. They compare and evaluate two methods: Term frequency and WordNet (wordnet.princeton.edu), which are used for emails clustering and classification. The approach is made up of 3 main phases:

- Data collection stage: In their work, gmail personal emails are collected. General statistics about the emails' dataset is collected from Google report provided for Gmail accounts' users.

- Emails parsing and pre-processing: A MIME parser is used to parse information from the collected emails to generate a dataset that includes one record for each email with the following information parsed: Email file name, email body, from, subject, and sending date.

- Emails' dataset data mining: A tool is developed to further extract all text from all emails and calculate frequency of words. Five classes are proposed to label the nature of emails users may have: Personal, Job, Profession, Friendship, and Others. They tried also to use clustering to assist in classification. Rather than labeling emails manually by users, they cluster sets of emails based on some email aspects and then they pick a name for developed clusters to come up with an email classification scheme.

The goal of the work of Yang et al. [64], is to distinguish messages of interest from the huge amount of data received. An approach for intelligent email categorization has been proposed in this work using fast machine learning algorithms. The categorization is based on not only the body but also the header of an email i.e. metadata such as sender name, organization etc.... which improves the categorization capability. They adopt the RAINBOW [1], which performs statistical text classification, in their experiments where two fast learning algorithms are chosen and modified for their experimental studies: TFIDF classifier and Naïve Bayes classifier. They define corporate announcements, meeting, and spam categories as categories or classes. They label their email data manually on these three categories assuming that each message only belongs to one category. Over 43 features are defined for representing an email including the feature vector of the email body text, some additional features taken from the email's header, etc.... They compare experimental results on different set of feature combinations to deduce that Naïve Bayes provides better results in most cases.

Among the unsupervised mining techniques we mention the works of:

The work of Surendran et al. [54] illustrates the concept of email clustering that relates generally to document organization. Particularly, it relates to

[1] https://www.cs.cmu.edu/ mccallum/bow/rainbow/

automatic document organization through automatic discovery of user's topics of interest from his emails. They develop a system that facilitates organization of emails comprising a clustering component that clusters a plurality of emails and creates topics for emails by assigning key phrases extracted from emails within one or more clusters. An organization component utilizes the key phrases to organize documents. Furthermore, the organization component comprises a probability component that determines a probability that a document belongs to a certain topic. Figure 3.2 is a representative flow diagram illustrating a methodology for organizing items based upon key phrases extracted from emails within clusters of emails.

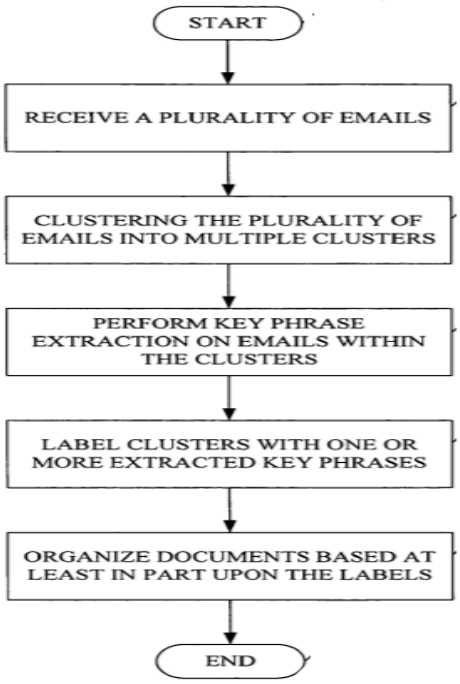

Figure 3.2: Methodology for organizing emails according to their key-phrases in [2].

Figure 3.3 shows the steps of the overall proposed approach of Patil et al. [15]. Their work focuses on implementing k-means clustering algorithm along with similarity measure (SMTP) Similarity Measure for Text Processing on email data set to categorize emails into different groups. SMTP considers presence and the absence of features and not only the occurrences of features in different emails. Once the data is extracted from the email, it is represented in some format. The most prevalent model for representation is the vector space

model. In the model every email message is represented by a single vector and each element as a token or feature. In such type of data the tokens are usually words or phrases. These tokens can be categorized mainly into three categories- Unigram, Bigram and Co occurrence. Once the term frequency is calculated, document vector is generated. For each email document, individual document vector is generated. Using document vector and similarity measure, similarity is calculated between email documents. The most similar documents are clustered together using clustering algorithm.

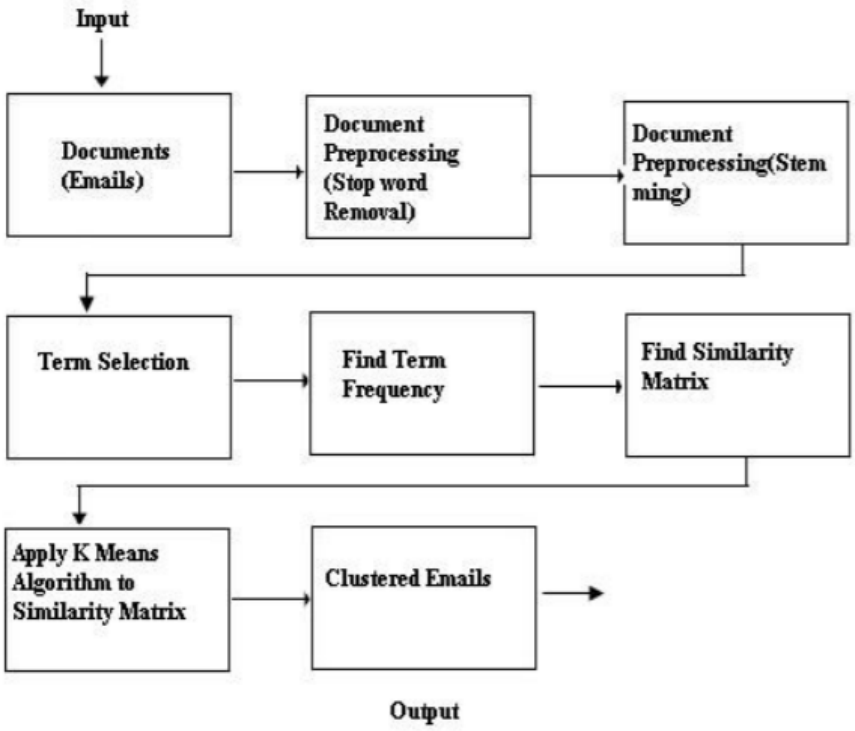

Figure 3.3: Overall steps of the proposed approach in [15].

In Maharana et al. [52], Lingo algorithm is utilized to cluster emails according to their topics. Lingo algorithm mainly depends on the frequent phrases and single words and cluster emails into topics accordingly. Lingo algorithm starts by preprocessing emails by extracting frequent phrases and single words as cluster label candidates. It then builds the term-document matrix using the stems of the label candidates. Next, they reduce the term-document matrix to the term-concept matrix according to the desired cluster count base threshold which is then followed by a matching step between the concepts and the candidate labels to get final clusters.

The work of Li et al. [41] presents a novel algorithm to cluster emails ac-

cording to their contents and the sentence styles of their subject lines. In their algorithm, natural language processing techniques and frequent itemset mining techniques are utilized to automatically generate meaningful generalized sentence patterns (GSPs) from subjects of emails which help in summarizing the subjects of a large number of similar emails which results in a semantic representation of the subject lines. They build an unsupervised approach which treats GSPs as pseudo class labels and conduct email clustering in a supervised manner. Their proposed algorithm improved the clustering performance, and it provided meaningful descriptions of the resulted clusters by the GSPs.

The core of the clustering algorithm in Turenne et al. [56] extracts terms co-occurring with a same set of other terms and matching a relational pattern (i.e. a graphical model). They assume that co-occurrences of terms occurring in fragments of texts (and repeated several times) can be significant of lexical semantic and conceptual associations. The main stages of the approach are : (1) Tagging terms using natural language preprocessing (terms are also truncated to root form using a small sets of suffixes). (2) Co-occurrence counting (ex: presence of a noun and a verb in a window). (3) Aggregating terms according to a measure of similarity. (4) Discriminating groups according to an overlapping factor of dissimilarity. They demonstrated that co-occurrences extracted from documents taking into account shared relations (patterns) can be efficient in a specific task such as document classification.

3.1.2 Email Summarization

Email is considered a useful case for summarization since most people receive a big number of emails each day. The volume of the received emails entails a great cost in terms of the time required to read, sort and archive the incoming data. By using email summarization in different ways, the process of managing one's email folders can be eased, thus it is a promising way to reducing the email triage. In addition, a generated summary can make it easier to access email on the small screen of a mobile device. Previous efforts in email thread summarization have adopted techniques developed for general multi-document text summarization and applied them to emails by including email specific elements. We can categorize summarization technqiues into two types: (a) Extractive and (b) Abstractive Summarization. In our work, we mainly concentrate on Extractive summarization. For this reason, we only mention related works about extractive summarization of emails.

Extractive Summarization

In extractive summarization, emails are summarized by extracting sentences from the email without applying any changes on them. In fact, this type of summarization lends itself well to machine learning. The text can be separated into sentences and then the problem becomes a classification task on sentences. Classification algorithms can be trained to select sentences for extraction given features for each sentence. In this way classification can be used for summa-

rization. The extracted sentences are considered the most important sentences in the email which give an indication about its content. In practice, sentence highlighting allows users to skim an email by reading the most important sentences.

The Clue Word Summarizer in Carenini et al. [6] is an approach which takes advantage of the email thread structure when creating summaries. Their goal is to provide a concise, informative summary of emails contained in a folder thus saving the user from browsing through emails one by one. Their approach have two novelties: using fragment quotation graph to try to capture an email conversation, and using clue words to try to measure the importance of a sentence in conversation summarization. A clue word from a node is a word that appears also in its parent node(s) and/or child node(s) in the quotation graph. A fragment quotation graph G = (V, E) is a directed graph, where each node u belongs to V is a text unit in the email folder, and an edge (u, v) means node u is in reply to node v. They specify the quoted fragments and build a graph corresponding to the email conversation. The edges of the graph are built according to the assumption that any new fragment is a potential reply to neighboring quotations, quoted fragments immediately preceding or following it. For each sentence, a clue score is computed by summing up the number of times that the words of the sentence occur in the parent or child nodes. The sentences with highest scores are selected.

The work of Carenini et al. [7] augments a new step to the previous work [6] where tan approach is proposed based on subjective opinions and integrate it with the graph-based ones. A cohesion measure is integrated together with the subjective opinions. The authors claim that sentences with subjective meanings are paid more attention than factual sentences (like decision making, providing advice or feedback). In order to asses the degree of subjectivity of a sentences, the frequency of words and phrases that are likely to bear subjective opinions is counted. The subjectivity of a sentence is computed by comparing it to a predefined set of known subjective words and opinions. This score is combined with the previously obtained scores using sentence quotation graph approach.

Wan and McKeown [61] worked on a more specific problem which was finding the topic of an email conversation. The approach uses term frequency vectors and SVD to find the different and most important topics. The topic was then summarized using the most descriptive sentence for the topic. Heuristics were finally used to fine tune the system.

The goal of the work in Lam et al. [40] is to improve users' interactions with their emails, by helping them prioritize new unread messages better and recall old read messages with better precision. They tackle the problem of using an existing software of email organization for well-authored formal documents to summarize email messages which lack the previous features. This gives a poor summarization of email messages. Their system preprocesses email messages using some heuristics to remove email signatures, header fields, quoted text from parent message, reporting names, dates and companies found in messages. The system exploits two aspects of email, thread reply chain and commonly found features to generate summaries. For the thread reply chain aspect, emails are

summarized only with their ancestors to provide the user with better context about the thread. For the commonly found features, names of people and companies or dates are extracted and added to the end of the summary which may give an indication to the user about the importance of the message.

The work of Zajic et al. [69] can be also mentioned in this context. The authors present two approaches for email summarization techniques: Collective Message Summarization (CMS) which applies a multi-document summarization approach and Individual Message Summarization (IMS) treats the problem as a sequence of a single document summarization task. They do not apply a purely extractive approach. Instead they employ linguistic and statistical methods to generate multiple compressions of sentences where they remove unimportant fragments of otherwise important sentences and then select from those candidates to produce a final summary. The selector chooses final sentences from the candidates based on features propagated from the sentence compression method, features of the candidates themselves, and features of the present summary state. Figure 3.4 represents the basic architecture of their summarization framework.

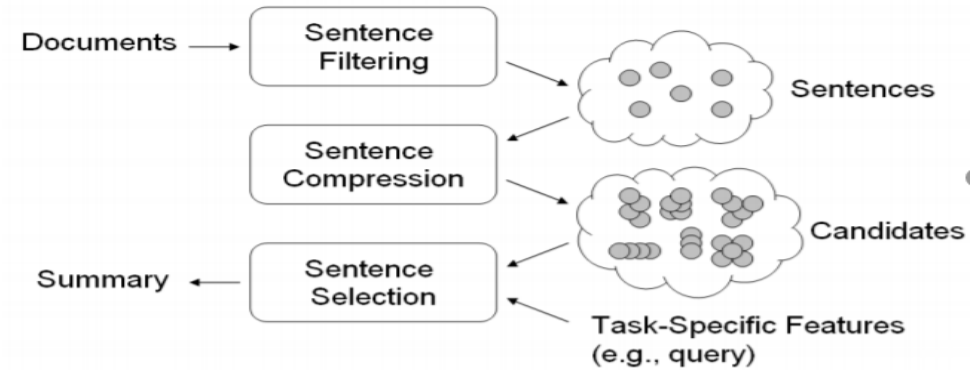

Figure 3.4: Basic architecture of the summarization framework in [69].

In Ulrich et al. [58], the authors represent a regression machine learning approach for email thread summarization. They compare different possible feature sets. They start with an already defined feature set that is used in Rambow et al. [49] and add to it speech acts, meta data labels and subjectivity levels. They show that regression-based classifiers perform better than binary classifiers because they preserve more information about annotator judgements. In their comparison between different regression-based classifiers, they found that Bagging and Gaussian Processes have the highest weighted recall.

In the work of Alam et al. [45], the authors claim that so far the attempts to perform true abstraction—creating abstracts as summaries have not been very successful. However, an approximation called extraction is more feasible. To create an extract, a system needs simply to identify the most important/topical/central topic(s) of the text and returns them to the reader. The

steps included in the methodology of their approach: (1) The system parses the user query provided by the user in natural language and finds the major parts in the string. (2) The system constructs the parse tree of the abstracted symbols. (3) Once the parse tree is generated, it will analyze the prioritization and the frequency of the abstracted symbols which will be documented in a table. (4) The user requirements will be then analyzed for summarization. (5) All sentences having the same keywords respective to the priority and user requirements will be extracted. So the system identifies the most occurring tokens in the text and the extract the top 5 sentences containing the identified tokens.

The work of Rambow et al. [49] focused on creating a set of sentence features by using proven text extraction features (e.g., the relative position of the sentence in the document) and adding email-specific features (e.g., the similarity of the sentence with the email Subject field). Extracted sentences are sent to a module that wraps these sentences with the names of the senders, the dates at which they were sent, and a speech act verb. The sentences are then sorted by the order in which they were sent Their results show that email-specific features significantly improved summarization.

3.1.3 Email task management

Several applications mainly consider the process of associating manually the email and their metadata such as attachments, links, and actors into activities.

In Bellotti et al. [3], one of the identified seven specific problems that participants in their studies experience with task management in email is collating related items and associated files and links. For this reason, they build TaskMaster, a system which recasts emails into Thrasks (thread + task), the interdependent tasks which comprise threads. They consider that the main element of interest is the task, not the message. In addition, they deal with attachments and links as first-class citizens. This feature opens up the intriguing possibility of being able to use Taskmaster as a bookmarking tool for favorite URLs such as our organization's phone list, or Google^{TM}. In this way Taskmaster feels less like a classic application and more like a general task-management environment, handling a variety of types of media.

In Gwizdka et al. [28], the TaskView interface is proposed for improving the effectiveness and efficiency of task information retrieval. They only work on improving the representation of emails using two email attributes: the time and the sender of an email. The authors claim that email messages containing other references are handled poorly in current email systems. This research examines how external representations of task information at the user interface can improve management and awareness of pending tasks that are encoded within email messages. In their work, they consider that users are forced to repeatedly review messages left in their inboxes, they cope with this constraint by employing a variety of strategies. TimeStore-TaskView interface is based on TimeStore [66] and uses the same graphical representation. In TimeStore-TaskView, tasks embedded in messages are represented by small icons on a two-dimensional grid with temporal and other attributes shown on the horizontal

and vertical axis, respectively. Other task attributes include sender, subject, or keywords extracted from the message body (user selectable). Navigation back and forward in time is provided. Displayed time period can be between one day and one year. The message body can be viewed by double clicking on the corresponding task icon.

3.2 Process Model Discovery vs Email Analysis

3.2.1 Process Instances Discovery from Emails

In the business process management field, a business process model is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal). The business process instance is a specific execution or case for the general process model. An instance describes an actual process which includes data, real actions, and specific decisions. In an event log, each event is identified by a process model identifier and a process instance identifier or case identifier. In the field of process mining ([60]), most of existing approaches consider that event logs contains case identifiers. A few exceptions are found in the field of service mining, where the problem is called event correlation mining. In the Business Process Management field, a huge number of researches worked on techniques for relating business process events into cases. As an example, such technique is presented in [47]. Their correlation process is based only on temporal relationship between events. An interactive process for event correlation is presented in [47] where the user inputs are taken into account for selecting interesting correlations. Event correlation conditions are discovered based on the value of common fields of events. In another work [51], MapReduce is used for scaling process event analysis approach. Their approach introduces efficient methods to partition an events log (aggregated from different data sources) across map-reduce cluster nodes in order to balance the workload related to atomic condition computations while reducing data transfers.

The report of Aalst et al. [59] presents the tool EMailAnalyzer which analyzes and transforms e-mail messages in MS Outlook to a format that can be used by process mining tools. They describe an approach for extracting the process event logs from the e-mail logs. They discover process instances using several options such as linked contact, thread, sender, receiver, single case (if all messages of the log belong to a single case), thread etc.... The user is asked to choose the options he/she considers suitable to extract a case. After selecting one of the options listed, the user can see the list of possible case names. The user can then edit the list by adding/deleting new case identifiers thus effectively filtering the log. They also try to identify the type of the activity or task. There are options to derive the identity of the task. It is assumed that the task name is included into the message subject. Then they try to derive the event type. By default EMailAnalyzer considers seven event types used by the ProM framework: “schedule”, “start”, “complete”, “suspend”, “resume”, “withdraw” and “abort”. Once the event log is deduced from the email log using the email

attributes and the user interference, process model can be derived using ProM framework.

We can mention the approach of Mavaddat et al. [42], they are trying to find out if by using the data in the email corpus of an organization in conjunction with the conversation for action and speech act theory they can create fragments of business process enactments to help process engineers understand organizational processes better. One of the challenges they concentrate on is to what extent the business processes are being carried out using email messages. Their first phase is Email categorization where the email corpus is divided into two different classes: business process related and non-business process related. After classifying the emails and finding the business process related emails, the second phase involves finding email threads inside the business process related emails. To perform this step, they use semantic similarity measurement. A refined version of Vector Space Model algorithm is used where each email is translated into a vector implemented in a sparse matrix and the semantic similarity is measured using the multiplication of vectors of emails. In the third phase, labelling speech acts is applied using the speech act theory [11] as this theory tries to define what the speaker intends to do by using words. By applying this process, the business process related set of emails turns into different threads, each representing one conversation network that is initiated by a role instance and has been continued until a mutual agreement. Figure 3.5 shows an overview of the main steps followed in their approach.

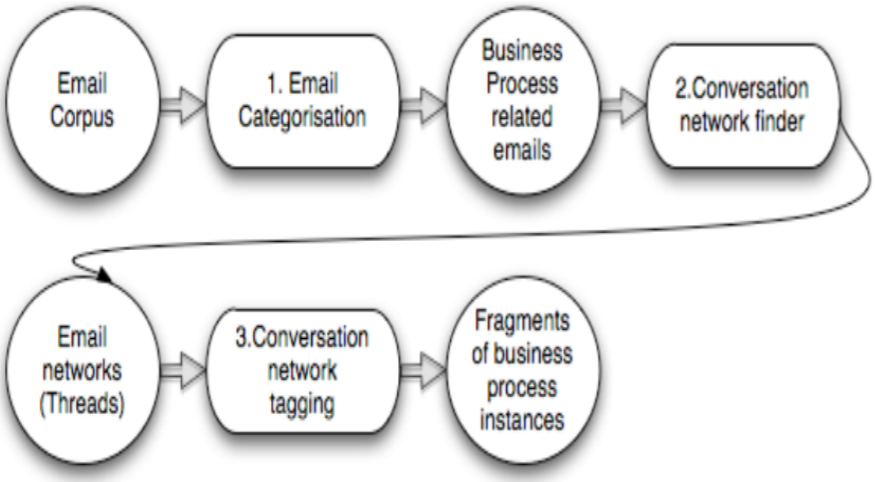

Figure 3.5: The approach framework in [42]

MailOfMine is proposed in Di Ciccio et al. [16]. The objective of their proposed approach, is to automatically build a set of workflow models that represent the artful processes laying behind the knowledge workers activities, on top of a collection of email messages. An advantage that they mention about their work is that they aim at addressing the open research challenge of

dealing with mixed logs, i.e., logs in which traces from multiple processes are present. Most of existing workflow mining approaches suppose to treat logs in which only traces of the same process are present. The authors describe their approach in three parts: (1) the preliminary steps, from the retrieval of email messages to the reconstruction of communication threads; they cluster retrieved messages into extended communication threads, i.e. flows of messages which are related to each other. Messages are then analyzed in order to identify key parts (important phrases). (2) the extraction of key parts, the activities and tasks detection, and the tasks definition. In this phase, the key parts are using with the clustering algorithm to identify matches between activities. (3) the final steps, from the activities definition to the final mined process extraction. In here, they use the emails timestamps to order activities. The applied steps allow them to deduce processes from the obtained tasks. Figure 3.6 shows the main phases of MailOfMine.

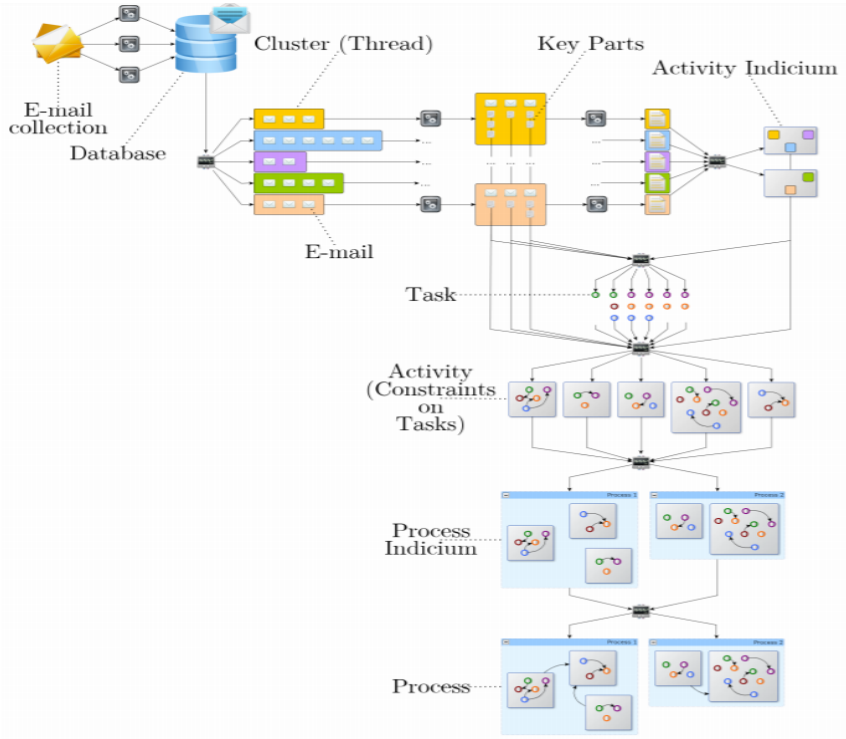

Figure 3.6: MailOfMine approach phases in [16]

The approach presented in Dredze et al. [17] in which emails are classified into process instances. The authors focus on the problem of classifying emails into instances, in order to automatically populate activities with the emails related to them. Their approach leverages two characteristics of instances: the

observation that instances connect groups of people together and the observation that activity-related email tends to center around particular topics. They introduce (1) the SimSubset and SimOverlap algorithms for email instance classification that compare the people involved in an instance against the recipients of a message; (2) the SimContent algorithm for email instance classification based on content similarity using iterative residual rescaling, a form of latent semantic indexing. They have shown empirically that the SimSubset and SimContent algorithms perform better on the dataset than the baseline approach of message threading and a naive Bayes classifier, and that a combined model that votes together the predictions of all three base learners performs better than any individual learner alone.

In this context, we can mention also the work of Khoussainov et al. [35]. They describe machine learning approaches to identify task and relations between individual messages in a task i.e. finding cause response relations between messages and for semantic message analysis i.e. how messages within a task relate to task progress. They exploit the relational structure of these two problems. The idea behind their approach is that related messages in a task provide a valuable context that can be used for semantic message analysis. Similarly, the activity related metadata in separate messages can provide relational clues that can be used to establish links between emails and group them into tasks. They propose an iterative synergetic approach based on relational classification, where task identification is used to assist semantic message analysis and vice versa.

3.2.2 Email Process Activities Discovery

One of our goals in this research is to recast emails into business activity centric resources. We describe an approach that is able to discover business process activities from emails. A business activity is a part of the business process model that we aim to discover from an email log. In this subsection, we present the works related to extracting activities or what are usually called "tasks" from emails.

In the work of Faulring et al. [20], they develop RADAR, a multiagent system with a mixed-initiative user interface designed to help office workers cope with email overload. RADAR agents observe experts to learn models of their strategies and then use the models to assist other people who are working on similar tasks. The agents' assistance helps a person to make a transition from the normal emails to a more efficient task-centric workflow. RADAR includes an Email Classifier that examines the content of each email for evidence that it contains any requests of the eight known task types [65]. The Email Classifier uses all available tokens and knowledge features of the email in one bag-of-words model and uses a regularized logistic regression algorithm they classify tasks contained within sentences of emails into a predefined set of classes. When it finds sufficient evidence for a given task type, it applies the label for that task type to the email. However, it cannot determine if an email contains multiple tasks of the same type. The evaluation of the RADAR 2.0 system showed that

the task-centric workflow enabled by the AI technologies helps users. However, the authors claim that he novice users lacked meta-knowledge about tasks such as task importance, expected task duration, and task ordering dependencies. An expert user with that knowledge should be able to make good decisions about which tasks to work on at any given time and which tasks to skip when time is limited. For this reason, they have built MCA, a new component of RADAR which provides guidance about the order in which to work on a set of tasks. The MCA learned a training model by passively observing experts performing tasks using the same user interfaces that test participants will later use. In the evaluation, they proved that with MCA participants earned higher scores than the average score of the Without MCA participants.

Cohen et al. [11] classify email according to the intent of the sender. They work on learning to classify email in this fashion, where each class corresponds to a verbnoun pair taken from a predefined ontology describing typical "email speech acts". Their ontology of nouns and verbs covering some of the possible speech acts associated with emails is summarized in figure 3.7. They assume that a single email message may contain multiple acts, and that each act is described by a verb-noun pair. To define the noun and verb ontology of figure 3.7, they examined email from several corpora (including their own inboxes) to find regularities, and then performed a more detailed analysis of one corpus. Based on the ideas from Speech Act Theory of Searle et al. [53], they define a set of "email acts" (e.g., Request, Deliver, Propose, Amend, Commit, Deliver, Refuse, Greet, Remind). The nouns in figure 3.7 constitute possible objects for the email speech act verbs. The nouns fall into two broad categories such as Information nouns that work with verbs Deliver, Remind, Request and Amend or Activity nouns that can be generally associated with email speech acts described by the verbs Propose, Request, Commit, and Refuse. Their experiments show that many categories of messages can be detected, with high precision and moderate recall, using existing text classification learning methods.

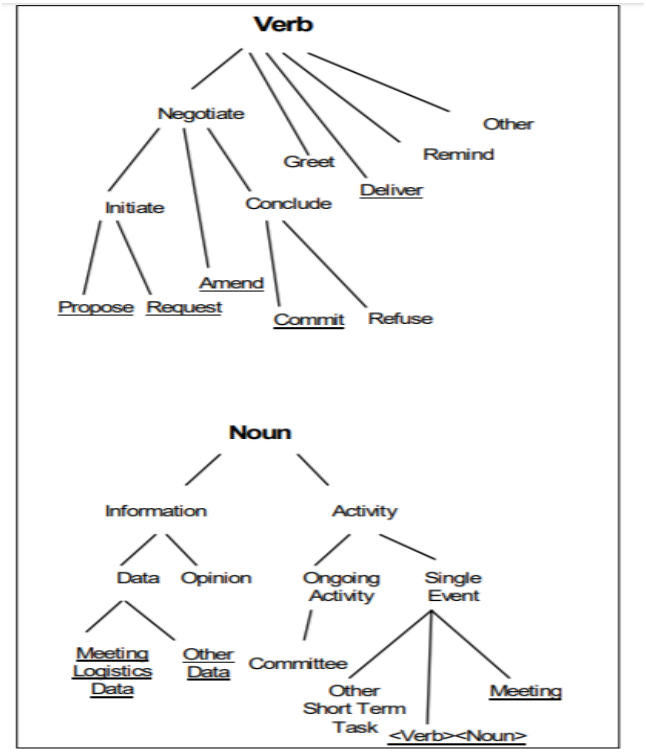

Figure 3.7: Ontology of speech acts.

SmartMail is described in Corston-Oliver et al. [12]. It is a prototype system for automatically identifying action items (tasks) in email messages. SmartMail presents the user with a task-focused summary of a message. The summary consists of a list of action items extracted from the message. The user can add these action items to their "to do" list. To do that, SmartMail first performs a superficial analysis of an email message to distinguish the header, message body (containing the new message content), and forwarded sections. Messages are broken into sentences, and a speech act is determined for each sentence by consulting a machine-learned classifier. Human annotators label the message body sentences, selecting one tag from the following set: Salutation, Chit-chat Task, Meeting Promise, Farewell, various components of an email signature (Name, Title, Affiliation, Location, Phone, Email, URL, Other), and the default category "None of the above". Each sentence in the message body is described by a vector of features. The features are of three types: properties of the message (such as the number of addressees, the total size of the message, and the number of forwarded sections in the email thread), superficial features and linguistic features. If the sentence is classified as a task, SmartMail performs additional linguistic processing to reformulate the sentence as a task description.

This task are reformulated and then presented to the user.

We mention also the work of Carvalho et al. [9]. The goal in this work is to discover whether or not email messages contain certain intents or email acts such as "propose a meeting" or "commit a task". They claim that analyzing n-grams instead of single words is more informative for the detection of email acts. In this work, they exploit the linguistic aspects of the problem by a careful combination of n-gram feature extraction and message preprocessing. After preprocessing the messages to detect entities, punctuation, pronouns, dates and times, they generate a new feature set by extracting all possible term sequences with the length of 1, 2, 3, 4, or 5 tokens. One possible way to extract the most important features, as mentioned in the paper, is using a feature selection method by computing the information gain score of each feature. They use SVM for classification. Their classification experiments provided them 26.4% drop in error rate compared to the work of Cohen et al. [11]. They finalize their work by introducing Ciranda: an open source package for Email Speech Act prediction. Ciranda provides an easy interface for feature extraction and feature selection, outputs the prediction confidence and allows retraining using several learning algorithms.

3.3 Techniques for Text Analytics

According to a study by the International Data Group (IDG), unstructured data is growing at an alarming rate of 62% per year. The same study also suggests that by 2022, close to 93% of all data in the digital world will be unstructured [2]. Text analysis, often used synonymously with text mining, is the process of analyzing chunk of unstructured data to find out undiscovered information and insights that can be leveraged for informed decision making and other processes. Text analysis is the automated process of obtaining information from text.

Text analysis tools are based on a complex process that consists of several concepts, such as statistics, machine learning, natural language processing and more. It also involves the use of many techniques. In this section, we will talk about the ones used in our work.

1. Information Extraction: the objective is to reconstruct a set of unstructured or semi-structured textual documents into a structured database.
    - The first step in the process of evaluation of unstructured data.
    - Involves tokenization and identification of named entities, key phrases and parts-of-speech.
    - Uses concept of pattern matching to find out predefined sequences if any within the data.
    - Identifies the relationship between entities and attributes.

[2] https://www.3rdisearch.com

In this work, we mainly use the Stanford Parser and the TF-IDF for the text preprocessing. A natural language parser i.e. The Stanford Parser [3] is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. This parser is a Java implemented package of probabilistic natural language parsers. The parser provides Universal Dependencies and Stanford Dependencies output as well as phrase structure trees. In our work, the Stanford Dependencies are utilized which provide a representation of grammatical relations between words in a sentence. They have been designed to be easily understood and effectively used by people who want to extract textual relations.

TF-IDF stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. In our work, we use tf-idf to select the most important words from messages. Instead of dealing with emails as a whole, we choose the words that are considered providing the important information about an email in a corpus of emails. This will enhance the efficiency of the further analysis applied on emails.

2. Clustering: the object is to bring together clusters of emails that have similar content or revolve around the same topic.

   - Generates multiple groups of emails known as clusters.
   - The content of documents in a specific cluster are very similar while that of documents in different clusters are not even remotely similar.
   - Differs from classification as it brings together documents without the use of any pre-defined categories as reference. This technique works on semantics - the principle on which semantic search engines work.
   - K-means and hierarchical clustering are the frequently used algorithms that bring good results.

   In our work, we are mainly interested by hierarchical clustering to cluster emails according to their content. Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each

[3] https://nlp.stanford.edu/software/lex-parser.shtml

cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other. Hierarchical clustering can be performed with either a distance matrix. The distance matrix contains the distances between all pairs of objects (emails) of the corpus. Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This continues until all the clusters are merged together.

To compute the similarity between objects, a distance metric should be defined. The choice of distance metric should be made based on theoretical concerns from the domain of study. That is, a distance metric needs to define similarity in a way that is sensible for the field of study.

3.4 Background: LSA and Word2vec

. In this work, we choose to try Latent Semantic Analysis (LSA) and Word2vec as similarity measurement methods in our clustering. They are two different approaches to generating corpus-based semantic embeddings. Corpus-based semantic embeddings exploit statistical properties of the text to embed words in vectorial space. Here we specifically analyze the capabilities of both models. We believe that word embeddings can bring new insights in email content analysis. We then provide a brief overview of both methods.

Semantic similarity plays a central role in how humans process knowledge, and serves as an organization principle for classifying objects, formulating concepts, and performing generalizations and abstractions [57]. A state of the art for assessing semantic similarities has been represented in the work of Christoph Lofi [10]. It states that the semantic similarity measurement approaches can be classified mainly into knowledge-based techniques which rely on a given ontology or taxonomy, and corpus-based approaches which use a large corpus of natural language text. As there is a lack of high quality ontologies in many domains, we orient our choice of similarity measurement techniques to corpus-based approaches which try to detect the similarity by applying statistics over large text corpora. Corpus-based techniques can be further classified into simple distributional approaches exploiting co-occurrences of words, and approaches based on dense vector representations which are usually the result of applying dimensionality reduction techniques to vector-based distributional approaches. Using such approaches is considered useful especially in cases where new domains and concepts are added continuously (highly adaptable). The central assumption of corpus-based techniques is the distributional hypothesis which claims that words that occur in similar contexts in a large text corpus also have similar or related semantics. So a straight-forward technique for exploiting the distributional hypothesis is representing words as high

dimensional vectors i.e dense vector representations. The quality of such high dimensional representation is often higher because it removes noises and generalizes concepts which is beneficial to many semantic tasks. According to the study done in [10], we favor corpus-based approaches (over knowledge-based approaches) which can approximate similarity and relatedness by analyzing a large natural language corpus. Among the corpus-based approaches, the dense vector representations is stated to have a very good performance, especially for the techniques which adopt factorization like LSA and the techniques which adopt the neural word embeddings like Word2vec.

Latent Semantic Analysis (LSA) Latent Semantic Analysis (LSA), as the name indicates, is the analysis of latent i.e hidden semantics from a corpus of documents. It is a method for analyzing documents and extracting underlying meanings or concepts out of these documents. If each word only means one concept and each concept is only described by one word, then LSA will simply map words to concepts. However, this is not the case in English language in which different words may have the same meaning and one word may have several meanings. LSA transforms the original data into different space so that two emails that are about the same concept are mapped together. This transformation is achieved by Singular value Decomposition (SVD) [24] of the original term-email matrix. Mathematically, let the original $tf\text{-}idf$ matrix be an $m \times n$ matrix X where m is the number of terms and n is the number of documents in the corpus. The LSA algorithm transforms X to an $r \times n$ matrix X' by generating a low-rank (dimension) approximation to X based on singular value decomposition (SVD) of X [48]. We can deduce from the SVD factorized matrix a special list of vectors v_1, v_2, ..., v_n so that every document can be written as a linear combination of v_i. Using these obtained linear combinations, LSA looks at patterns of words co-occurrences. These words co-occurrences will be linked to concepts. For example, the terms "available" and "tomorrow" will be discovered to co-occur under the same concept of scheduling a meeting.

Word2vec Dating back to 1986, Hinton [30] has proposed distributed representation of words. The main idea is to map each word into a k-dimensional digital vector and determine the semantic similarity between words by computing distances between the obtained vectors. Word2vec uses this idea and represents each word as a digital vector [68]. Word2vec is a computationally-efficient method for learning high-quality distributed vector representations (called word embeddings). Representing words as unique, discrete ids leads to data sparsity, and usually means that we may need more data in order to successfully train statistical models. In contrast to that, the word embeddings can capture a large number of precise syntactic and semantic word relationships [43]. Word embeddings

are represented in a Vector Space Model (VSM) which represents words as points in the space where semantically similar words are mapped to nearby points. While word embeddings are not fully understood yet, they show promising results for similarity measurement tasks [10]. The semantics of embeddings seem to go far beyond simple similarity as also the more complex concept of relational similarity can be covered.

Word2vec is not a single monolithic algorithm. It comes in two distinct flavors: Continuous Bag of Words (CBOW) and Skip-gram, each with two different training models. Both architectures describe how the neural network learns the underlying word representations for each word. In CBOW model, context is represented by multiple words for a given target word. Skip-gram model reverses the use of target and context words. The task of CBOW is to predict a word given its context words. However, Skip-gram's task is to predict the context given a word. Figure 3.8 shows the model structure of CBOW and Skip-gram [37].

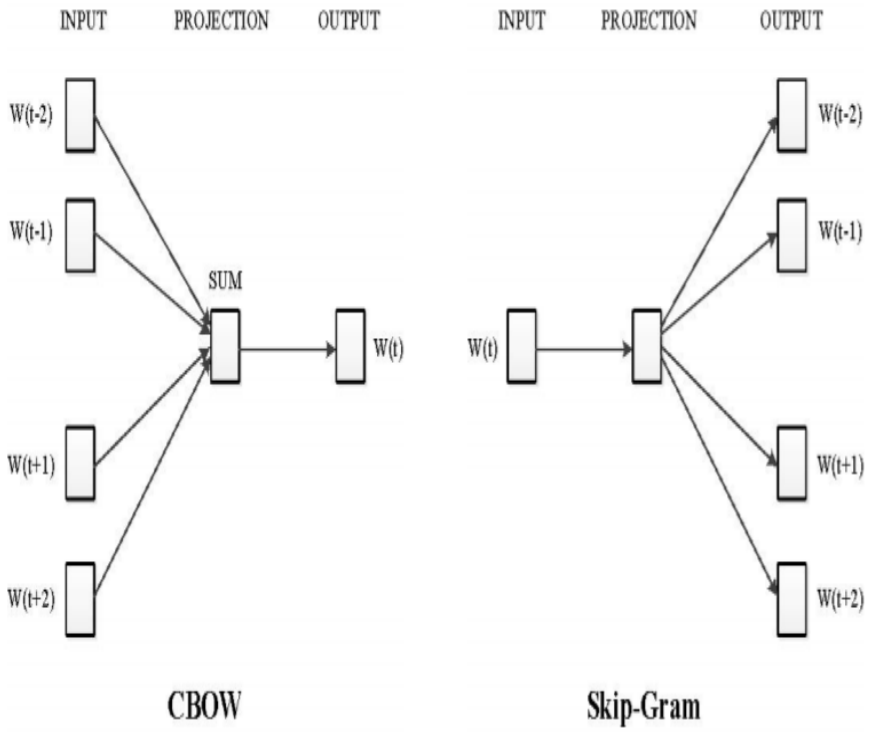

Figure 3.8: Word2vec architectures: CBOW and Skip-Gram [25]

After selecting a distance metric, it is necessary to determine from where distance is computed. For example, it can be computed between the two most similar parts of a cluster (single-linkage), the two least similar bits of

a cluster (complete-linkage), the center of the clusters (mean or average-linkage), or some other criterion. Many linkage criteria have been developed. As with distance metrics, the choice of linkage criteria should be made based on theoretical considerations from the domain of application [4].

3. Classification: its objective is to assign one or more categories to an unstructured text document.

    - Works on an input-output principle wherein the system is given inputs regarding the pre-defined categories under which the data in the new documents is to be classified.

    The choice of the classification modeling technique to be used is mainly based on the input dataset. Firstly, we should indicate whether our data is linearly separable or not. Two data subsets are said to be linearly separable if there exists a hyperplane that separates the elements of each set in a way that all elements of one set resides on the opposite side of the hyperplane from the other set [5]. However, in some scenarios data cannot be linearly separated and thus non-linear techniques should be applied. Generally speaking, in machine learning and before applying any classification technique, it is essential to study the data we are dealing with so that we can decide which classification algorithm to be used. There exist several techniques that help in indicating whether the data is linearly separable or not such as domain knowledge, data visualization, linear programming, computational geometry, etc. For simplicity sake, we use the data visualization technique which visualizes the features values as a scatter matrix containing the data visualization of all combinations of features. Regarding our dataset, the scatter matrix shows that the data points are non linearly separable. Figure 3.9 visualizes the dataset in terms of some features in the analyzed dataset features. The figure shows that the data values can not be linearly separable by a straight line. For this reason, using non linear classification algorithm is essential.

[4] https://www.displayr.com/what-is-hierarchical-clustering/

[5] http://www.tarekatwan.com/index.php/2017/12/methods-for-testing-linear-separability-in-python/

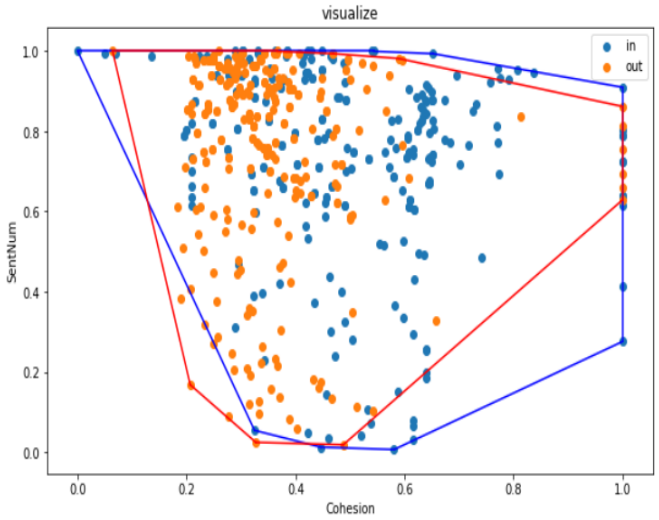

Figure 3.9: Visualization of the data values of the some features.

Multiple non linear classification techniques are used in this work for the purpose of classifying emails and their contents into different categories according to different targets.

- Gaussian Naive Bayes: which is a member of the family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. These classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.
- Neural Network: consists of an artificial network of functions, called parameters, which allows the computer to learn, and to fine tune itself, by analyzing new data. Each parameter, sometimes also referred to as neurons, is a function which produces an output, after receiving one or multiple inputs. Those outputs are then passed to the next layer of neurons, which use them as inputs of their own function, and produce further outputs. Those outputs are then passed on to the next layer of neurons, and so it continues until every layer of neurons have been considered, and the terminal neurons have received their input. Those terminal neurons then output the final result for the model.
- Linear Discriminant Analysis: a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear clas-

sifier, or, more commonly, for dimensionality reduction before later classification.

- K-Nearest Neighbor: In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression: In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.
- Decision Trees: Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.
- Gradient Boosting: Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.
- Support Vector Machines: support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

CHAPTER 4

Business Process Topics Discovery From an Email Log

Contents

Figures

A *Business Process* is composed of a set of activities that are applied in a specific sequence to perform a specific organizational goal. Business processes should have purposeful goals, be as specific as possible and have consistent outcomes. Each business process is represented by a model i.e. *Business Process Model*. The model represents a series of related tasks to be applied in a specific manner that result in the desired output. The models typically show business actions and links, in the sequence from end to end. We define a *Business Process Topic* as the subject in which the business process is concerned by. Each email log contains several business process topics. The topics and their quantities differ from an email log to another according to the domain of the email users and their jobs. An email user applies several organizational processes by exchanging emails with different entities. These processes are of different topics such as *meeting scheduling, event organization, travel grant application*, etc...

As mentioned earlier in this thesis, one of the main attributes of an event in an event log is the ProcessID i.e. to which process model the event belongs. In other words, which process topic the email is concerned by. Thus, to start building the event logs from email logs, it is considered an important step to refer each email to a specific business process topic (or ProcessID). In our work, we assume that each email belongs to a single business process topic. Therefore, the email log will be grouped into different sets of emails, where each group represents one of the process topics in the email log. Using the results of this component, we can enact the other components of our overall framework. In the other components, we work on each set of emails belonging to a unique process topic without taking into consideration other emails of other topics.

In order to do this, we use unsupervised machine learning techniques. In particular clustering techniques accompanied with semantic and non-semantic

distance measurement techniques are applied to compare their efficiency. This step is preceded by a very essential step for an efficient application of text mining and machine learning which is the preprocessing step. These techniques and their applications are clarified in a usecase at the end of this chapter. Therefore, in this chapter, we elaborate our approach of process topic discovery in emails by demonstrating the following: We first start by explaining the email preprocessing step in section 4.1 which will be used also in different chapters as a common step. In this thesis, each time we are applying any analysis on the email text, particularly, the subject and the body of the email, we start by preprocessing these texts. We provide an overview on the clustering approach for process topic discovery in section 4.2. The used clustering technique and its application in our work on email texts are detailed. The approach is evaluated in section 4.3. The experimental settings are detailed followed by a usecase to better understand the application of the mentioned techniques for our purpose.

4.1 Email Log Preprocessing

We start our work with a set of emails belonging to an email user. We use this email log as an input for our framework. An email log is a set of emails exchanged between different entities (people, companies etc...) for a specific purpose such as scheduling a meeting, organizing a conference, purchasing an item etc.. Each email is represented by some attributes describing it: email subject, sender, receiver, email body, and email timestamp. Mainly, the email's main content is its body and subject texts, which are considered in the category of unstructured datatype. This type of data should be prepared before any analytical work.

Generally, text data requires intelligent algorithms to retrieve relevant information from the repositories. These kinds of algorithms and techniques are categorized under the text mining field. Text mining is not a new concept, it is evolved from data mining and all the data mining algorithms can be applied on the textual data. The difference between the two is that data mining is applied on structured data and relational data whereas textual mining deals with texts that are considered as unstructured or semi-structured data. Actually to make this possible, the data is to be converted into semi structured or structured format so the data mining machine learning algorithms can be applied easily. This conversion of data is done by applying the preprocessing step. The preprocessing of the text data is an essential step as there we prepare the text data to be ready for the mining. If we do not apply this step, then data would be very inconsistent and could not generate good analytics results. Therefore, we consider that applying data preprocessing on raw emails as an important starting step in our thesis work.

We aim at extracting some features from emails that can efficiently represent them for further analysis. We process the emails separately without having any knowledge about the threading relations between them. A thread starts with an original email which is followed by a sub-sequence of replies between

peers allowing them to keep track of past conversations. We deal with emails regardless of their threading relations for two main reasons: (1) some email management systems do not collect emails in the form of threads, thus, we try to generalize our work to be able to accept all types of information (2) one process topic may span on multiple threads i.e. email users may start with a process topic in $thread_1$ and then continue this process in $thread_2$; and one thread may include multiple processes i.e. email users may work on different process topics in one thread (email with the same subject title while drifting the body content topics).

Since the unstructured textual data is mainly contained in the subjects and the bodies of the emails, in this step we work on preprocessing them. We apply different preprocessing phases that will serve in the different approaches explained in later chapters.

4.1.1 Data Cleansing

The text has to be as clean as possible before we can convert it into any other data representation. The majority of available text data is highly unstructured and noisy in nature – to achieve better insights or to build better algorithms, it is necessary to play with clean data.

In this section, therefore we discuss about the possible noise elements and how we could clean them step by step. After retrieval of data, the data is ready for the preprocessing. We apply the following steps to cleanse the email texts. These steps are applied on the email's subject and body.

1. We eliminate irrelevant data from the texts such as the punctuation, numbers, white spaces. These characters can have a negative effect as they are not important in the applied analysis. We remove such characters to ensure that the efficiency of the obtained results is not affected by any noises.

2. We transform cases. This step is useful in normalizing the text. The text would get converted into a single case either to uppercase or lowercase. We convert capital letters into small letters for a better comparison between documents (knowing that the comparison between strings is sensitive to capital and small letters).

3. We filter the stop words such as: the, is, at, which, on, in.. which have a confusing effect in our analysis without providing any additional information. These words do not contain any important meanings and would not be helpful in our analysis.

4. We apply stemming on the words of the texts. In this step the words are stemmed into their roots, all the suffixes are removed such as: responded to response, eaten to eat and plurals such as: women to woman, men to man and horses to horse. Although stemming sometime causes the loss of the meaning of the actual word but still works well in most of the cases.

It is important to mention that we keep the initial words in storage to get back to them later. In later chapter, we use the tense of a verb to be able to extract some temporal features from email. Therefore, stemming in this case would be inconvenient.

Once the email text data is cleansed, it can be represented in a manner that is accepted by the analysis tools. For this reason, the data representation step is considered an important one as it transforms texts into a standard form.

4.1.2 Data Representation

Knowing that text data is highly unstructured, its representation is not a trivial step. In this step, our goal is to reformat the cleansed data to be compatible with the data required for some analysis. The pre-processing of a document gives us a document with a bag of words only on which we cannot apply the algorithms directly. We need to convert this bag of words into term vector. The term vector gives a numeric values corresponding to each term appearing in a document which is very helpful in feature selection.

There are three main ways for converting terms into vector: Term Frequency, Term Occurrences and Term Frequency-Inverted Document Frequency (TF-IDF). The most useful and popular one is TF-IDF [50]. This gives the higher weight to the important terms and lower weight to the unimportant terms. Its vector value lies in between of 0 to 1. 0 means that the term has no importance in the context of the documents in which we are looking for and 1 means the terms are relevant. Typically, the TF-IDF weight is composed of two terms: the first computes the normalized Term Frequency (TF), as known as, the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

When the TF-IDF values of all words in the email log are obtained, the emails' body texts can be converted into a matrix M. The rows of M represent the emails IDs and the columns of M represent all the terms occurring in all email texts. Each value v_{ij} in each entry of M identifies the TF-IDF value of each word w_j in the email e_i. 4.1 shows a small sample of the TF-IDF matrix obtained.

Term Email ID	account	add	address	amazon	amount
1	0.00	0.00	0.00	0.14	0.00
2	0.16	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.21
4	0.00	0.21	0.00	0.00	0.00
5	0.00	0.00	0.14	0.16	0.00

Table 4.1: Example of TF-IDF matrix

4.1.3 Feature Selection

Despite removing the stop words and replacing each term by its stem in the preprocessing phase, the number of words in a weight matrix created in the text representation phase is still very large. Therefore, the feature selection phase is applied for reducing the dimensionality of the feature set by removing the irrelevant features. The goal of this phase is improving the efficiency of analysis accuracy as well as reducing the computational requirements. Feature selection is performed by keeping the features with highest score according to the features importance [55].

We work on selecting the attributes from our data which are most relevant to the problem we are working on. The choice of good features along with a suitable clustering method is very important for partitioning a given corpus into clusters of emails of the same process topic. Feature selection is popular in supervised learning. For supervised learning, feature selection algorithms maximize some function of predictive accuracy. If we are given class labels for the business process topics, it is natural that we keep only the features that are related to or lead to these classes. But in our case, the unsupervised learning, we are not given class labels. Which features should we keep? Why not use all the information we have? The problem is that not all features are important. Some of the features may be redundant, some may be irrelevant, and some can even misguide clustering results. In addition, reducing the number of features increases the comprehensibility and ameliorates the problem that some unsupervised learning algorithms do not work properly with high dimensional data. The goal of feature selection for unsupervised learning is to find the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion [19]. There are three general classes of feature selection algorithm:

1. Filter Methods: Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable.

2. Wrapper Methods: Wrapper methods consider the selection of a set of

features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy. The search process may be methodical such as a best-first search, it may be stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features.

3. Embedded Methods: Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods. Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients).

To efficiently apply feature selection on the available data, we choose the filter approach for its efficiency and simplicity. It basically pre-selects the features (in our case the terms), and then applies the selected feature subset to the clustering algorithm. As mentioned before, filters select features by ranking them according to certain scoring schemes. Unneeded features are identified and eliminated. Given the email-term occurrence matrix M, we filter and select the features to be included in our analysis. Using the obtained TF-IDF values, we can specify the most important features (terms) in our dataset. Mainly, the terms having an average TF-IDF in all emails greater than a threshold t are kept and the others are eliminated. The best threshold is specified by sensitivity analysis. The matrix M is then updated by removing the columns corresponding to the eliminated words. We call the updated matrix M'. Therefore, each email now contains only the words that are considered important in our mining.

4.1.4 Verb-Noun Pairs Extraction

We are interested in identifying whether a sentence contains a business process activity or not. We extract these business process activities in chapter 6, hence, we consider that the verb-noun pairs are the most indicative element in a sentence for that purpose. We consider that the verb-noun pairs are likely to be candidates of such activities. Therefore, we work on extracting such pairs from emails sentences to apply the study on them. We first start by text segmentation i.e. instead of dealing with the text as a whole, we segment it into multiple sentences by using the sentence tokenization methods. Sentence tokenization is the process of dividing written text into meaningful units, such as words or sentences. We then use for verb-noun extraction the Stanford Parser which outputs grammatical relations in the new Universal Dependencies representation [14]. An example is the sentence: "I would like to confirm the meeting on Thursday", the verb-object pair "confirm meeting" can be considered as a process activity in the meeting scheduling process. Multiple verb-noun pairs can be extracted for each email. For example: "I already paid the money. I received

the receipt" contains 2 verb-noun pairs: (1) paid money and (2) received receipt. Both pairs can be considered as candidates for being business process activities.

4.2 Clustering Emails According to their Process Topics

In this section, our objective is to group the emails into clusters according to what process model they are concerned by. If we fetch the emails of a researcher, we can detect that his/her emails are concerned by different processes such as scheduling a meeting or organizing a conference, etc.. We aim in this section to separate the emails of different processes into separate clusters. Each cluster is then associated to a process identifier (ProcessID). Hence, each email of the cluster is provided the corresponding ProcessID. Beside that this step identifies the ProcessID for each email, it also helps in analyzing emails. Instead of dealing with an email log as a whole, we will be able to work on emails of different processes separately which makes the analysis and mining less complex.

Classification and clustering are the two types of learning methods which characterize objects into groups by one or more features. These types appear to be similar, but there is a difference between them in context of data mining. The prior difference between classification and clustering is that classification is used in supervised learning technique where predefined labels, in our case the process topics, are assigned to instances, on the contrary, clustering is used in unsupervised learning where similar instances are grouped, based on their features or properties. Since in our case the class labels are missing i.e. we do not have any apriori knowledge about the business process topics available in the email log, we decided to use unsupervised machine learning methods which is the clustering as it does not require such knowledge.

Clustering is the grouping or segmenting of a set of objects, in our case the emails, into subsets or clusters, in which objects in one cluster are more related to one another than those of different clusters. There are two major methods for clustering: K-means and hierarchical clustering. Taking into consideration our goal from clustering, we choose the hierarchical clustering, as k-means clustering would have the following disadvantages:

- K-means does not provide us the hierarchy structure. In our approach we need to obtain a hierarchy of emails with different levels in order to be able to find clusters (relative to process models) and sub-clusters (relative to activity types).
- K-means requires as input the number of clusters, while we have no information about the number of clusters we need to obtain.
- K-means is sensitive to cluster center initialization.

For the above mentioned drawbacks, we decide to choose hierarchical clustering to group emails into clusters of business process topics. In particular, we choose

the agglomerative clustering which is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also known as AGNES (Agglomerative Nesting). The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram. Hierarchical clustering does not require any constraints or specifications before enactment. Agglomeratively (using the complete linkage), the clusters are fused together according to the chosen similarity measurement technique. We try two different methods for similarity measurement between the selected features of emails. We justify the choice of these two techniques in section 3.

1. Similarity measurement using Latent Semantic Analysis (LSA) method [22].

2. Similarity measurement using Word2vec method [44].

It is important to note that, only the email's body is included in the similarity calculation. We justify this by the fact that the body of an email contains the most important information about the email's context. In some cases, users may exchange emails in the same thread under the same subject phrase but the contents of the messages drift to another topic. The email subject may be confusing sometimes because the email content may not conform with it, while its body always reflects its real context. For ensuring that the subject attribute will not have a dominating effect on the distance value, we check if the words of the subject are compatible or similar to the words of the email body. If they are similar, we calculate the similarity between two emails using a weighted sum of the emails's subjects and bodies. Otherwise, we exclude the subjects from our measurements. We avoid the fact that the subject of an email may be misleading in some cases.

For the first clustering trial, we use LSA to map words occurring in emails into concepts. LSA is applied as follows:

1. We construct a matrix X whose rows are associated with emails, and the columns with terms contained in the emails. Each cell $c_{i,j}$ of the matrix contains the TF-IDF value of the occurrence frequency of term j in email i.

2. The matrix X is then submitted to singular value decomposition (SVD) which gives as a result three matrices D, T (orthonormal) and S (diagonal), such that $X = DST$. Most of the columns of D, S and T are removed in a way that the matrix $X' = D'S'T$ is the least best fit approximation of X.

3. The number of concepts (process topics) contained in the corpus can be deduced from the above obtained matrices, where each term is ranked relative to each concept. A vector of ranks will be associated to every term.

4. Each email will be subsequently ranked, relative to each concept. Email ranking depends on its terms ranks. Each email is now associated to a vector of ranks.

5. The similarity degree between two emails can be calculated as the dot product between their LSA rank vectors.

6. The similarities between emails will be fed to the hierarchical clustering algorithm which results a multi-level hierarchy of emails.

For the second clustering trial, Word2vec, specifically CBOW, is utilized for the similarity calculation between emails. Its steps are applied as follows:

1. We load to our system an already trained Word2vec model on 1 billion words where each target word of the model is represented in the VSM by a vector of 300 context words.

2. For each email e_i, we get the vectors of all its words $\{v_1, v_2, v_3, ..., v_n\}$.

3. We then calculate the average vector V_i for the whole email by averaging the values of all vectors v_1, v_2, v_3,..., v_n.

4. Using these vectors the similarity between emails in the corpus will be computed.

5. The similarities between emails will be fed to the hierarchical clustering algorithm which results a multi-level hierarchy of emails.

For each of the above trials, we use the resulting multi-level hierarchical representation of emails to deduce how they are grouped. According to our visualization study and the results analysis, we apply several cuts on the obtained dendrogram or hierarchy to obtain the one that provides the best clustering results. We choose to cut the hierarchy in a way such that emails belonging to same process model are clustered together. We obtain the best cut according to the experimental evaluation of the clustering results. The output of this cut is a set of clusters $\{PC_1, PC_2, PC_3, ..., PC_n\}$, where each cluster PC_i contains a set of emails related to the same process model topic. PC_i and subsequently the emails contained in it will be associated to a ProcessID.

4.3 Experiments and Results

In this section, we introduce in details about how we experiment our approach. We first start by describing the experimentation settings and then show the obtained experimentation results. We start with a usecase that better explains the followed approach. We compare the clustering quality results when using LSA and Word2vec for this usecase. The approach is also applied on an email log from Enron dataset. The results of Enron email log clustering will be used in the rest of the thesis.

4.3.1 Experiments

We applied the approach described in this chapter on two different email logs. One is formed of 250 emails taken from a Ph.D student. Usually the professional emails of a Ph.D student revolve around processes such as organizing conferences, organizing meetings, mission demands or refunds etc... The other is of 628 emails taken from the Enron email dataset which is also about different process topics such as trading, recruitment, meetings etc.... We use Python programming language for doing our testing. The input of the framework is an email log where each email is described by a set of attributes: sender, receiver, subject, body and timestamp. The emails are first preprocessed by removing stopwords, whitespaces, punctuation, numbers, by converting letters to lower case and by stemming. This has been done by utilizing the Natural Language Toolkit (NLTK) which is an open source Python library for Natural Language Processing. NLTK is a powerful tool for extracting and manipulating texts. The TF-IDF matrix is then deduced and accordingly feature selection is applied in which the unimportant terms are eliminated. Verb-noun pairs are extracted using the Stanford parser which is a natural language parser that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb.

Similarities between emails are computed using both methods LSA and Word2vec in which we obtain a similarity matrix for each method used. Hierarchical clustering is then applied on the two similarity matrices separately. This results a multi-level emails hierarchy for each method. We cut the hierarchy to obtain clusters containing emails related to the same process model. The best cut is chosen by trial and error analysis (the cutting points that are explored are mainly those where the gap between two successive similarities is largest.)

For deploying the LSA method and for using hierarchical clustering algorithm, we use the powerful and rich Scikit-learn package developed by David Cournapea in 2007 which provides a range of supervised and unsupervised learning algorithms via a consistent interface. To use Word2vec tool, we imported the Gensim python package. We load a 3.4 GB Word2vec model containing all vectors of 1 billion words trained on Google news corpus. The results of clustering and sub-clustering using Word2vec are compared to the traditional Latent Semantic Analysis (LSA) method.

4.3.2 Usecase

We apply our framework on the input email logs datasets. We first expose the dataset to the pre-processing phase where data is cleansed, best features are selected, and represented in matrix amenable for analysis. Figure **??** shows a small portion of the matrix that will be an input for the clustering phase.

	Terms					
Docs	account	add	address	afternoon	amazon	amount
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.1682018	0.0000000	0.0000000	0.0000000	0.0000000	0.1416126
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
4	0.0000000	0.2136564	0.1416126	0.0000000	0.0000000	0.1416126
5	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

Figure 4.1: A small portion of the preprocessed data matrix(the TF-IDF values of the first 6 words in 5 emails.

These emails are then clustered using the hierarchical clustering algorithm by comparing LSA and Word2vec for similarity measurements between emails. Clustering quality evaluation metrics (Rand-Index, Precision, Recall, F-measure and Purity) show that Word2vec provides better clustering results. From the first cut, we obtain 5 main clusters. When fetched, these clusters seem to revolve around the topics: (1) meeting scheduling, (2) items purchase, (3) applying for a mission grant, (4) mission refund by the research center, (5) accommodation application. In the Enron email dataset, we obtain clusters about trading, recruitment, meetings, discussion, transportation, retail, projects.

4.3.3 Results

In this subsection, we will show the numerical results of the clustering of emails using the similarities obtained by LSA and Word2vec on both datasets. We compute the rand-index, precision, recall, F-measure, and the purity as clustering evaluation metrics. We manually fetch the results and compare them to the correctly clustered data.

1. $Recall = \frac{TP}{TP+FN}$
2. $Accuracy = \frac{TP+TN}{N}$
3. $Precision = \frac{TP}{TP+FP}$

4. $F - measure = 2\frac{precision * recall}{precision + recall}$

5. $Purity = \frac{1}{N}\sum_{i-1}^{k} \max_j c_j \cap t_j$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative, N is the total population, k is the number of clusters, c_j is a cluster, and and t_j is the classification which has the max count for cluster c_j

Tables 4.2 and 4.3 below show the different metrics values for the process topic clustering on both email logs.

	LSI	Word2vec
Rand Index	0.34	0.68
Precision	0.42	0.63
Recall	0.44	0.41
F-measure	0.42	0.5
Purity	0.58	0.64

Table 4.2: A comparison of process model clustering quality when applying different similarity measurement methods on Ph.D student email log.

	LSI	Word2vec
Rand Index	0.31	0.71
Precision	0.45	0.60
Recall	0.36	0.42
F-measure	0.45	0.49
Purity	0.53	0.59

Table 4.3: A comparison of process model clustering quality when applying different similarity measurement methods on Enron email log.

We realize that Word2vec provides better results in almost all clustering quality evaluation metrics, which proves its efficiency in discovering similar texts. The good clustering results of Word2vec shows the importance of using word vectors and word context vectors. Therefore, it is believed that prediction based model (such as Word2vec) capture similarity in a better manner.

4.3.4 Discussion

In traditional document clustering technique, the terms (words) of the documents are considered as features; however, the semantic relation among these terms of documents is not taken into consideration. Due to this, problems like synonymy and polysemy, ambiguity, high dimensionality, etc. take place. There

are several ways to solve this problem that takes place due to use of traditional approach. Different ways to solve the problem include the use of Latent Semantic Indexing (LSI or LSA) or Word2vec as we have done in this chapter.

The previous existing works have suggested several approaches for organizing emails according to their topics, folders or priorities. In our research, we treat some of their limitations that are discussed in the following:

Different works (supervised and unsupervised techniques) that are discussed in the State of the Art chapter 3 apply foldering or topic clustering and classification according to predefined set of topics. They study the email messages features and identify accordingly the topic that the email belongs to. However, this can not be practically applied all the time. Analysts may not know the set of topics available in the email log. For this reason, we work on not being limited to a specific number of folders or topics. In our approach for finding email process topics, there is no a priori knowledge on the number or types of topics available in the email log. This enhances the generality of our approach. In other words, we discover topics of emails during analysis. This eliminates the limitation that may happen when new topics are added to the email logs (new email exchanges). In addition to the automatic identification of topics, the user interference is not needed in this phase. In contrary to other works that ask the user to specify some key phrases or words that may help in the topic discovery of the email, our work is purely automatic and unsupervised. This will reduce the effort needed by the user or analyst to perform this phase.

The analysis in our work is purely dependent on the content of the message. Other works include other information such as the threading relations in their analysis. We claim that such kind of data may not be accurate enough for an efficient analysis. Although it can give some indications in some cases, however this kind of information may be misleading in some cases. In an email thread, the subject of the messages may stay constant during the whole conversation knowing that the message content may drift to another subject. Thus, working on the messages' content is more guaranteed for an efficient analysis. In the following, we compare different approaches for business process topic discovery from emails according to some criteria:

a) The system should have no a-priori knowledge about the topics contained in an email log.

b) Users should not interfere or help in the analysis.

c) Email topic identification should be dependent mainly from the email content since it contains the real information about the topic of the email.

d) Using semantic similarity between emails which gives more information about emails than key-phrases occurrences.

A comparison between the different previously mentioned related works is provided in table 4.4 to check whether their approaches cover the demanded requirements or not.

Requirement \ Related Work	[8]	[67]	[39]	[2]	[64]	[54]	[15]	[52]	[41]	[56]	our approach described in Chapter 4
a	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
b	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
c	No	No	No	No	No	No	No	No	No	No	Yes
d	-	-	-	-	-	No	No	No	No	No	Yes

Table 4.4: Requirements comparison for email topic discovery between different approaches and our approach.
The work described in this chapter have been published in [32].

CHAPTER 5

Business Process Instances Discovery From an Email Log

Contents

Figures

Process mining is an analytical discipline for discovering, monitoring, and improving business processes by extracting knowledge from event logs readily available in today's information systems [1]. In process mining, there exist the main terms: *business process model* and the *business process instance*. A process instance is a specific occurrence or execution of a business process model. For example, if making a cake is a process, the recipe is the process model. A process instance occurs each time a person makes a cake using this recipe. In business process management, each process model can be executed by applying different sequences of activities respecting the conditions and rules of the model. Different executions of the process model are called the process instances. In fact, when email users exchange emails to perform a specific activity in a process, they are actually enacting an activity in a process instance. Although activities in all process instances should be either the same or a subset of the activities in the process model, executions of activities in different process instances may include different information. This what makes a process execution or occurrence distinguished from the others. Therefore, in the context of process mining, each process model can be applied several times.

This can be applied for discovering business process models from email logs. Each email log can contain processes of different topics. Let us suppose that in an email log, the "meeting scheduling" process topic exists. An employee may exchange emails with two different entities for scheduling different meetings. Thus, multiple email exchanges take place for this purpose. These email exchanges represent the different executions or occurrences or what is called the *process instances* of the general process model which is "scheduling a meeting". This means that in the same email log, we may have two emails containing the same activity but belonging to two different process instances. Emails including

[1] https://www.celonis.com/process-mining/what-is-process-mining

the same activities but belonging to different process instances can be distinguished by activities information they contain. For example, if a user is sending two emails about two different meetings, each email will contain a different set of information such as the location, the people in the meeting, the subject of the meeting, etc...

In the previous chapter 4, we work on grouping emails according to what process topic they belong to. Therefore, each email is associated to a ProcessID i.e. to the process topic it is concerned by. As our overall goal is to transform email logs into event logs, we aim to move deeper in emails to discover to which process instance an email belongs to. Thus, we associate to each email a process instance identifier (processInstanceID). Beside transforming email logs into event logs, discovering business process instances in emails can have many other applications. As mentioned in chapter 2, business process instances discovery in emails is useful in itself for answering several analysis questions. We recall the queries concerned in process instances discovery in the following:

1. What is the average duration of a business process? This can be computed by averaging the time taken by all process instances of the same process model. This helps managers and employees to identify the expected duration of a specific process beforehand according to the previous executions of the same process.

2. Which process instances take the longest time to be achieved? Normally, process instances should consume similar periods of time. However, when a process instance takes a long duration for its achievement, this gives an alert about an abnormality. Knowing that there is a problem, may help in identifying the reason behind it i.e identifying the reason behind the time delays. Overall time delays may be caused by partial time delays in several activities of the process or by a time delay in only one activity due to loops (an activity is executed several times to be completed).

3. How many instances are enacted in a specific period of time? This can also be related to the productivity of the enterprise. For example, a manager would like to know how many times a specific type of trading is enacted. It may be also related to the interactions between employees. For example, someone may be interested to know how many meetings were organized for a specific group (or the number of organized events).

4. Which process instances involve specific entities? For example, which mission funding application instance required the involvement of the department director? This question may help in identifying exceptional situations or inefficient process executions. This kind of query would help in mitigating similar problems that may occur in the future while applying the same type of process.

In this chapter, we work on analyzing email texts for identifying for each email the process instance it belongs to. This is mainly done by choosing attributes and features from email texts that help in distinguishing between emails of

different process instances and in grouping of the same process instances. The problem here is that this kind of information are not explicitly available in emails. There are no predefined tags or attributes associated to each email containing such information. Therefore, in this chapter, we work on extracting process instance related information from email logs.

We demonstrate the process instance discovery phase in section 5.1 where we work on choosing the best distance function for clustering emails into process instances. The distance function consists of a combination of attributes. The efficiency of the distance computation depends crucially on these attributes. The experimentation setting and results are described in section 5.2.

5.1 Baseline Process Instances Discovery

In this thesis, we apply two ways for discovering business process instances in email logs. The first one is a baseline approach in which its techniques depends only on the intrinsic features of an email. In the second approach, we use intrinsic and extrinsic features which will be explained in chapter 7. In this section, we describe the baseline approach for business process instances discovery from email logs. We start by choosing the features that will be used by the distance function. We justify our choice of these features and explain their effect on the accuracy of the clustering results. We do this by representing some examples and counter examples.

As the overall framework shows, our work starts by email log preprocessing and business process topics discovery for each email. Before applying the current phase, each email is associated to a ProcessID. The emails and their processIDs are considered as an input for the composite component represented in figure 2.2. For discovering business process instances from email logs, we start from the previously obtained process topic clusters $\{PC_1, PC_2, PC_3, ..., PC_n\}$, where each cluster PC_i represents a business process topic. Each cluster is composed of a set of emails which we consider as related to the same process model topic. To enact this sub-component, we use as an input each of the obtained clusters separately. Therefore, in this section, we aim to deduce for each business process topic cluster, the set of process instances it contains. In other words, at the end of this chapter, each email will be assigned to a ProcessID and a ProcessInstanceID.

5.1.1 Defining an Appropriate Distance Function

It is well known that obtaining good clustering results that best serves the analyst target, is crucially dependent from the distance function used by the clustering technique. In our case, our target is to sub-cluster emails of the same process topic cluster into process instances. Therefore, the defined distance function should take into consideration additional features that help in this sub-clustering phase. Instead of only dealing with the semantic similarity between emails as done in the process topic discovery phase, we work in this approach

on utilizing structured and unstructured features in an email to calculate the distances between emails.

In order to separate emails belonging to the same process model into different process instances, we apply a sub-clustering step on the already obtained process topics clusters. We illustrate the steps of this phase and the distance function definition by using an example which is concerned by applications for missions funding.

Example Suppose that one of the obtained clusters contains emails about all applications of Ph.D students for "missions funding" process topic. Emails are taken from the administrative secretary responsible for accepting or rejecting applications in the university or the laboratory where the Ph.D student works. This cluster contains all emails sent and received by the secretary for the process of student mission applications. Several applications of different students or even same student but with different missions such as conferences in different countries or summer schools, will have similar sequences of emails with some slight differences. For example, for a mission application for student A, some documents may be demanded different from those of the mission application of student B. Therefore, we may have in the same business process topic cluster, two different sequences of activities of the same business process. These are the so-called process instances where each instance represents a specific execution of the general process model. Figure 5.1 shows an example of two process instances of the same business process topic.

For grouping emails belonging to the same process instance, we need to define a distance function that best serves in relating emails of the same process execution. An email is characterized by specific attributes: sender, receiver, body, subject and timestamp. Some of these attributes contain structured information and other contain unstructured information. We make use of both structured and unstructured information to build the distance function. We choose to represent an email by combining some of its attributes.

First, we consider the body and the subject of an email as we assume that they hold most of its important information. However, we consider that it is useless to include the whole body and subject texts. Some features are extracted from these texts that can help in identifying emails of the same instance. Emails of the same process instance are supposed to revolve around some common names. Take as an example the emails exchanged between a student and the secretary for applying for a funding to attend the SCC 2017 conference in Hawaii. Most of these emails bodies and subjects will include the named entities "SCC" or "Hawaii" or "Honolulu". We claim that these named entities can be helpful in discovering which emails are related to the same process execution (same mission application). We may assume that all emails holding the same named entities as belonging to the same business process instance. However, in some cases named entities will not be sufficient to distinguish instances.

Counter Example 1 The cluster concerned by mission applications of Ph.D students may contain two emails about attending the same conference SCC by the same student but in two different years. It happens that the same student may apply for the same conference in two consecutive years. The following example emails clarify the counter example idea.

Email 1

***From**: diana.jlailaty@gmail.com (student A)*
***To**: missionjc@dauphine.fr*
***Subject**: travel grant application*
***Timestamp**: 2015-05-03 13:45*

Dear,
Please find attached all documents needed for my application in SCC conference of this year.
Thanks,
Diana

Email 2

***From**: diana.jlailaty@gmail.com (student A)*
***To**: missionjc@dauphine.fr*
***Subject**: travel funcding application*
***Timestamp**: 2016-05-15 17:23*

Dear,
Kindly, find the SCC conference mission application attached to this mail..
Thanks,
Diana

If we try to find emails of the same process instance using only the named entities, the above two emails will be considered to belong to the same execution (both of them have only SCC as a named entity) which is not the case. Each email is sent for a different mission application in two different years so they belong to two different process instances. Thus, we decide to add another attribute which gives an indication about the time of sending an email. The timestamp of an email will be useful in such situation. Since **Email 1** and **Email 2** are sent in two different and far timings, their timestamps will help in separating them into two different instances. Adding the timestamp attribute to the distance estimation function will provide more accuracy in finding emails of the same execution. However, in some rare cases named entities combined

with the timestamp of an email will not be sufficient for separating emails into instances.

Counter Example 2 The cluster concerned by mission application of Ph.D students may contain two emails for the same mission application in the same year for two different students. It may happen that two students are applying to the same conference in the same year. This will make it impossible for a distance function consisting only of the named entities and the timestamps to separate the two emails into different process instances.

Email 3

***From**: diana.jlailaty@gmail.com (student A)*
***To**: missionjc@dauphine.fr*
***Subject**: travel grant application*
***Timestamp**: 2016-05-03 16:56*

Hello,
Please find attached all documents needed for my application in SCC 2017 conference taking place in Hawaii.
Thanks,
Diana

Email 4

***From**: hiba.alili@gmail.com (student B)*
***To**: missionjc@dauphine.fr*
***Subject**: travel funcding application*
***Timestamp**: 2016-05-04 10:12*

Dear,
Kindly, find the SCC 2017 Hawaii conference mission application attached to this mail..
regards,
Hiba

The above two emails are sent in close timestamps. In addition, they both contain similar named entities. In this case, if we just depend on named entities and timestamps of emails, **Email 3** and **Email 4** will be considered belonging to the same process instance which is not the case. This drives us to choose an additional attribute that will play the role in separating such emails. The sender/receiver attribute will help to distinguish emails belonging to different instances. In the case of **Email 3** and **Email 4**, the senders will be different

(*student.A@dauphine.fr* and *student.B@dauphine.fr*). Accordingly, both emails will be separated into two different clusters.

Starting from these remarks, we define the distance function as follows: we first define the similarity function and then derive the distance function.

$$Sim(E_{i_j}, E_{i_k}) = w_1 \times Sim(NE_{i_j}, NE_{i_k}) + w_2 \times Sim(T_{i_j}, T_{i_k}) + w_3 \times Sim(SR_{i_j}, SR_{i_k}) \tag{5.1}$$

Therefore the distance funtion is:

$$Distance(E_{i_j}, E_{i_k}) = 1 - Sim(E_{i_j}, E_{i_k}) \tag{5.2}$$

where E_{i_j} and E_{i_k} are two different emails j and k in the same process model cluster C_i. (NE_{i_j}, T_{i_j}, SR_{i_j}) and (NE_{i_k}, T_{i_k}, SR_{i_k}) are the named entities of the subjects and bodies, timestamps and sender/receiver of emails E_{i_j} and E_{i_k} respectively. Weights w_1, w_2, w_3 ($w_1 + w_2 + w_3 = 1$) represent the relative importance of named entities, timestamps and sender/receiver of emails, respectively.

For each cluster separately, we extract the named entities from each email's body and subject. The number of common named entities n_{jk} is computed between all pairs of emails. Let n_{max} and n_{min} be the maximum and minimum, respectively of these numbers. The similarity of named entities for two emails i, j is computed as follows:

$$Sim(NE_{i_j}, NE_{i_k}) = (n_{jk} - n_{min})/n_{max} - n_{min} \tag{5.3}$$

The value of Sim(NE_{i_j},NE_{i_k}) is obtained by normalizing n_{jk} into the range [0,1]. As n_{jk} increases, the similarity between the two emails increases.

The timestamp difference t_{jk} is computed between all pairs of emails. Let t_{max} and t_{min} be the maximum and minimum, respectiviely of these numbers. The value of Sim(T_{i_j}, T_{i_k}) is deduced by normalizing t_{jk} into a value of range [0,1] as follows:

$$Sim(T_{i_j}, T_{i_k}) = 1 - (t_{jk} - t_{min})/t_{max} - t_{min} \tag{5.4}$$

As the time difference increases, the similarity decreases.

We also check whether an email's sender(s)/receiver(s) is the same or a part of the sender(s)/receiver(s) of another email. If yes, the value of Sim(SR_{i_j}, SR_{i_k}) will be equal to 1, otherwise 0.

$$Sim(SR_{i_j}, SR_{i_k}) = \begin{cases} 1, & \text{if sender(s)/receiver(s) of } E_{i_j} \text{ and } E_{i_k} \text{ overlap} \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

This normalization is applied because the three attributes provide values of different ranges. The named entities and timestamps similarities are provided the highest importance (w_1 and w_2 respectively). A smaller weight (w_3) is given to the sender/receiver attribute. We justify this by the fact that named entities and timestamp features can distinguish between emails of different instances (or can relate emails of the same instance) in most cases. The sender/receiver attribute is added only for the rare cases in which the formers do not help ($w_1 > w_3$, $w_2 > w_3$).

5.1.2 Clustering Emails Into Process Instances

Using the above distance function, we calculate distances between all pairs of emails. Accordingly, hierarchical clustering is applied where we get the emails distributed on a hierarchical structure. We tried several cuts on the obtained hierarchy. We choose the one which provides the best clustering quality (according to clustering quality measures mentioned in the experimentation section). Each of the obtained clusters contains emails belonging to the same process instance. Every cluster is provided a process instance identifier and as a result, each email of this cluster will be labeled by a process instance ID.

5.1.3 Analyzing Business Process Instances

Reaching this step, each email in an email log is associated to a ProcessID i.e. to what process topic it belongs and to a ProcessInstanceID i.e. to which process instance it belongs. Therefore, the analytical queries can be answered using the available information. For each instance, we can calculate the following:

- its duration as the difference between the maximum and minimum of the timestamps of the emails belonging to it
- number of emails exchanged
- set of involved persons and its cardinality
- minimum, maximum and average elapsed time between two consecutive emails based on the difference of their timestamps
- starting time (minimum of the timestamps of the emails belonging to it)

Based on the above information, we can combine the obtained results and calculate some statistics for all instances:

- minimum, maximum and average process duration
- minimum, maximum and average number of emails exchanged
- minimum, maximum and average number of of involved persons
- minimum, maximum and average elapsed time between two consecutive emails
- instance arrival rate for different time periods

This enriched log is stored in a database that can be used to query or to mine process behavior patterns.

5.2 Experiments and Results

In this section, we will explain in details the conducted experimental settings and results. A brief usecase will be provided to illustrate the baseline approach for process instances discovery. Finally, we will provide the results of the applied experiments. The same prototype implementations settings are used as in section 4.3 since the techniques used are the same for both phases.

5.2.1 Usecase

For the testing phase, we use an email log of 240 emails taken from a Ph.D student. We then apply the approach on different subsets from Enron email log. We choose different process topic clusters to apply the approach on. Starting with the emails of the Ph.D student, usually they revolve around processes such as organizing conferences, organizing meetings, mission demands or refunds etc...

These emails that are clustered using the hierarchical clustering according to business process models by applying the approach of chapter 4 are then clustered according to the business process instances they belong to. By applying the process topic clustering step, we obtain 5 main clusters. When fetched, these clusters seem to revolve around the topics: (1) meeting scheduling, (2) items purchase, (3) applying for a mission grant, (4) mission refund by the research center, (5) accommodation application. If we take the fourth cluster: mission refund by the research center, the sub-clustering results in different process instance: (1) XXX summer school mission application, (2) YYY conference mission application , (3) ZZZ conference mission application, etc... Figure 5.1 shows two examples of the obtained process instances for the cluster of missions applications.

During our analysis, we found that some instances have an important numbers of emails, compared with the average number. A further analysis revealed that these instances contains loops, i.e., multiple emails were sent and received to achieve a specific activity. For example, we discover a loop on the "approve" activity due to multiple documents demanded by the administrative entity of the research center. Such loops decreases the performance (efficiency) of the process models in achieving business goals. A preliminary idea for our future work, is to build a recommendation system that predicts the activity to be applied as a next step. When predicting the activity, the system will recommend the best actions to be done to avoid loops. An example on our usecase is to recommend all the needed documents for a faster grant approval.

Similarly, we apply the same approach on an Enron email log of 250 emails which is only concerned by the process topic "recruiting". If we visualize the obtained instances, we get instances such as the one shown in figure 5.2.

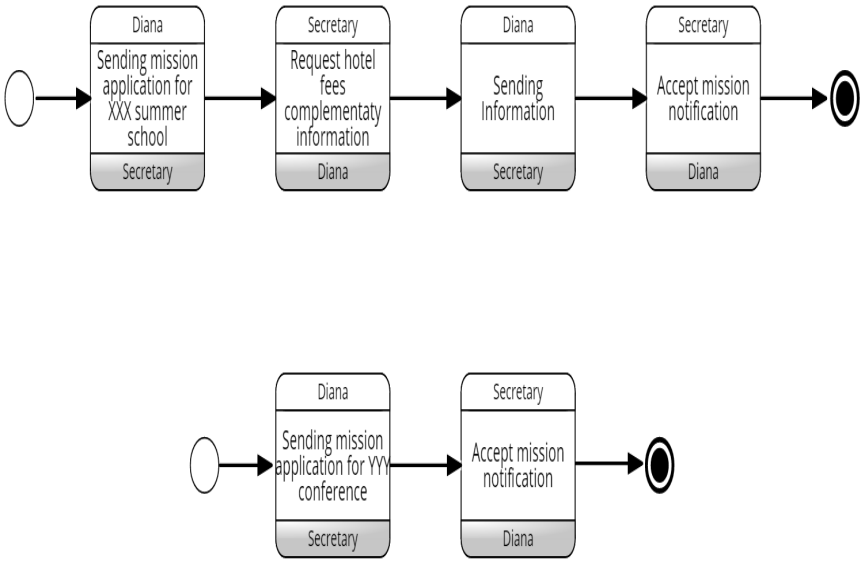

Figure 5.1: Example of process instances for mission funding application.

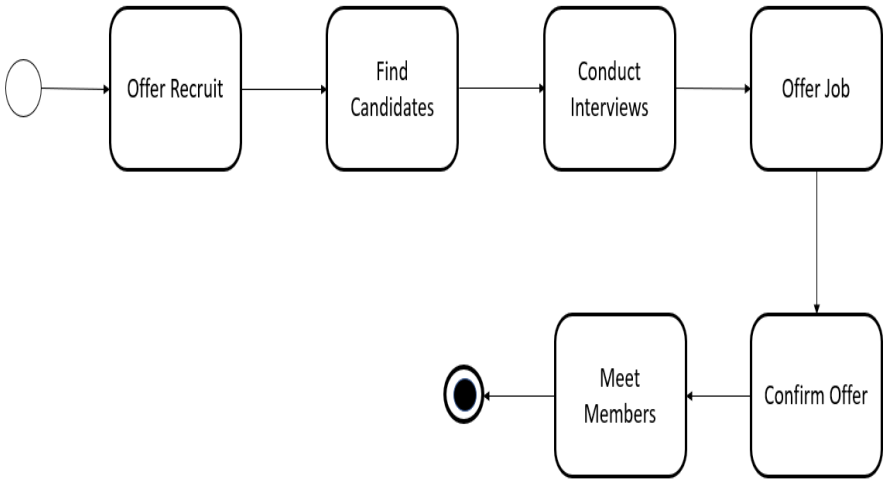

Figure 5.2: Example of a process instance for recruiting process topic.

5.2.2 Clustering Quality Measurement

In this subsection, we evaluate the quality of the sub-clustering of emails applied using the chosen distance function. We compute the rand-index, precision, recall, F-measure, and the purity. The results are provided for both email logs.

After applying the process topic clustering step, although we had good clustering results, some mis-clustered emails existed. In order to evaluate the second step, we applied some corrections to the mis-clustered emails (we transfered each mis-clustered emails into its suitable process model cluster). This helps in experimenting the efficiency of the defined distance function for email-process instance clustering.

As mentioned in section 5.1, we tried several weights for the three used attributes in the defined distance function. We discovered that the best values of the weights are w_1=0.45, w_2=0.4, w_3=0.15. The named entities and timestamp attributes are given similar weights, but greater than the weight provided to the sender/receiver attribute.

Table 5.1 and 5.2 below shows the different metrics values for the clustering according to process instances.

	Process Instance Clustering
Rand Index	0.84
Precision	0.87
Recall	0.74
F-measure	0.79
Purity	0.85

Table 5.1: Clustering quality metrics results for the mission application process topic

Evaluation Metrics	Process Instance Clustering
Accuracy	0.75
Precision	0.77
Recall	0.64
F-measure	0.69

Table 5.2: Clustering quality metrics results for the recruiting process topic

5.2.3 Discussion

Previous existing approaches tackle the problem of identifying business process instances from emails mainly by either taking into consideration the threading relation between emails as an indication for instances identification or by allowing the user to choose the email attribute(s) that helps in relating emails into

instances. However, these approaches are not always efficient. Threads may not always be a good indication for emails belonging to the same process instance. As mentioned before, emails maintained in the same thread may drift to a new subject different from the initial one. In other words, users may deal with two or more different processes inside one thread. This denies the fact that threads are always a good indication for relating emails belonging to the same process instance. On the other hand, it is not a practical solution to ask the user to choose the attribute(s) for email process instances discovery. The efficiency of such approaches depends on the user's choices which require additional efforts. The user may not know what are the best attributes to be chosen for obtaining efficient results. To overcome this problem, we apply in the approach described in this chapter a study showing multiple combinations of email attributes for distance measurement for process instances discovery. This reduces the user effort and ensures the efficiency of the instances identification (especially that this step is considered an intermediary step in the overall framework). In the following, we compare different approaches for process instances discovery from emails according to some criteria:

a) No user interference to choose the attributes that serve in emails process instances discovery.

b) Dealing with the whole email content, without taking into consideration the threading relation between emails or just dealing with a single email attribute.

A comparison between the different previously mentioned related works in provided in table 5.3 to check whether their approaches cover the demanded requirements or not.

Requirement \ Related Work	[59]	[42]	[16]	[17]	[35]	our approach described in chapter 5
a	No	Yes	Yes	Yes	Yes	Yes
b	No	No	No	No	No	Yes

Table 5.3: Requirements comparison for process instances discovery between different approaches and our approach.
The work described in this chapter have been published in [31].

CHAPTER 6

Business Process Activities Discovery From an Email Log

Contents

Figures

A *Business Process Model* is composed of a set of *Business Activities* enacted in a specific sequence to achieve a business goal. *Business Process Activities* are designed to perform specific tasks or actions that contribute to a business process. For example, consider the business process model about "meeting scheduling". The activities of this model may include "propose meeting", "refuse meeting", "postpone meeting", "confirm meeting" etc... Another example is about the activities of the business process model "travel grant application". This process may contain activities such as "send application", "request information", "accept grant", "refuse grant" etc...

In the context of emails, email logs have been revealed to contain a significant number of activities that are exchanged between different entities such as enterprises, employees etc... that are collaborating to achieve the corresponding business goals. More recently, many researches [36] have discussed how email is transforming into a “habitat”, the central place from which work is received, managed, and delegated in organizations. Therefore, email activities discovery has occupied a huge part of the literature of the email analysis and management.

Obvious benefits of email include efficiency, convenience, and cost. Each email is sent for the aim of requesting, canceling, confirming a specific task or set of tasks. To recall, our main target is to transform email logs into event logs. One of the main attributes in the event log is the activity label. Therefore, in this chapter, we work on extracting activity labels from email logs.

To facilitate email management, a number of research proposals have been made, see e.g., [5, 62, 12]. For example, Corston et al. [12] exploit emails to identify actions (tasks) in email messages that can be added to the user's "to-do" list. While useful, current email management tools lack the ability to recast emails into *into business activity centric resources*, that can be considered as a part of a business process model. To overcome the limits of related work, we need to extract from the emails, the event log expected by the process mining tools where each event is represented by an identifier of the activity type, process model and process instance. In previous chapters, we addressed the problem of identifying for each email the ProcessID and ProcessInstanceID.

There are two main hypotheses for the extraction of business process activities from email logs. The first hypothesis is that each email contains one and only activity. This hypothesis is applicable in many emails, however, it is not always true. Therefore, we propose the second hypothesis in which we assume that an email can contain 0, 1 or more activities. To allow for the extraction of business activities from emails, and therefore cater for the evaluation of analytic queries, we propose first a preliminary work [32] described in section 6.1 that adopts the first hypothesis such that each email revolves around only one activity. While the usefulness of such a proposal was empirically validated, it also underlined some limitations of the method we use for extracting activities. In particular, each email is associated with a single business activity. However, in practice, an email can be associated with 0 or multiple activities. For this reason, we suggest the second hypothesis and accordingly build an advanced approach. The extraction of business process activities from emails is considered important for answering several analysis queries such as:

1. What are the business activities executed by a specific employee? For example, a manager would like to know the productivity of a specific employee or to know the contribution of an employee in a specific process. (to identify time-consuming tasks that are not known to be assigned to him).

2. How many times a user applied an activity? For example, an employee may wish to know how many times he/she applied for a travel grant during a specific period of time.

3. What are the groups of people doing similar work? For example, a manager would like to know which people apply similar types of activities. This may help in organizing working groups. An employee may wish to benefit from the experience of another employee that applied the same type of activity before.

Another goal for business process activities discovery in emails is to perform analytics on the information associated with the extracted activities. Therefore, annotations describing the activities need to be extracted as well. Activities annotations are a set of information that describe activities. Usually, these information can be found in the text of the emails and in the resources associated with an email such as the attachments or links. In fact, the extraction and analysis of such information help answering some other queries such as the kind of documents that can be attached to an email activity or web pages are concerned by an activity? This may help in producing fewer exchanges of emails on the same activity type (attach a bill or receipt for confirming payment activity).

We propose in this chapter, a solution for extracting business activities from emails, and for annotating the elicited activities. Specifically, the following contributions are elaborated in this chapter:

1. We first start by the first hypothesis, where we build an approach that can extract a single activity per email.
2. We then move to the second hypothesis where we present an approach that, using customized extractive business oriented summarization of emails and clustering of business-oriented sentences, discovers and labels one or multiple business activity types in an email.
3. We automatically associate each activity type with a set of metadata that describes it.
4. We evaluate our approach on a set of data from Enron email log.

This chapter is organized as follows: (1) We present our preliminary approach and its disadvantages in section 6.1. (2) We present the phases of the improved approach for multiple business activities discovery in section 6.2. (3) We finally evaluate the improved approach compared to the preliminary one in section 6.3.

6.1 Single Activity Per Email Approach (Preliminary Approach)

Taking into consideration the motivations for extracting activities from email logs described in the introduction of this chapter, we build a baseline approach composed of 3 main phases. Figure 6.1 shows the main phases of this approach. In contrast to the improved approach that will be described in section 6.2 which work on the emails themselves without previous grouping, the preliminary approach is a continuation of the process model topic clustering phase explained in chapter 4. Back to what was explained in that chapter, according to our visualization study, we first choose to cut the hierarchy in a way such that emails belonging to same process model are clustered together. This is considered as a

high-level cut. The output of the high-level cut is a set of clusters {PC_1, PC_2, PC_3, ..., PC_n}, where each cluster PC_i contains a set of emails related to the same process model topic. PC_i and subsequently the emails contained in it will be associated to a ProcessID.

Having the clusters in hand, we need to identify for each cluster the set of activities applied in its emails. To do this, we apply a sub-clustering phase in which the emails of each cluster are sub-grouped according to the activity types they belong to. In order to obtain for each cluster (i.e process model) the set of activities mentioned in it, we apply a low level cut on the same hierarchy. In other words, we extract the sub-clusters for each of the already obtained topic clusters. Thus, we have a set sub-clusters {AC_1, AC_2, AC_3, ..., AC_n} representing activity types. We try multiple cuts on multiple levels to deduce those which provide the best clustering and sub-clustering quality (according to quality evaluation metrics).

The third phase, activity labeling, is explained in 6.2.4. This step is common between both the preliminary and the improved approach.

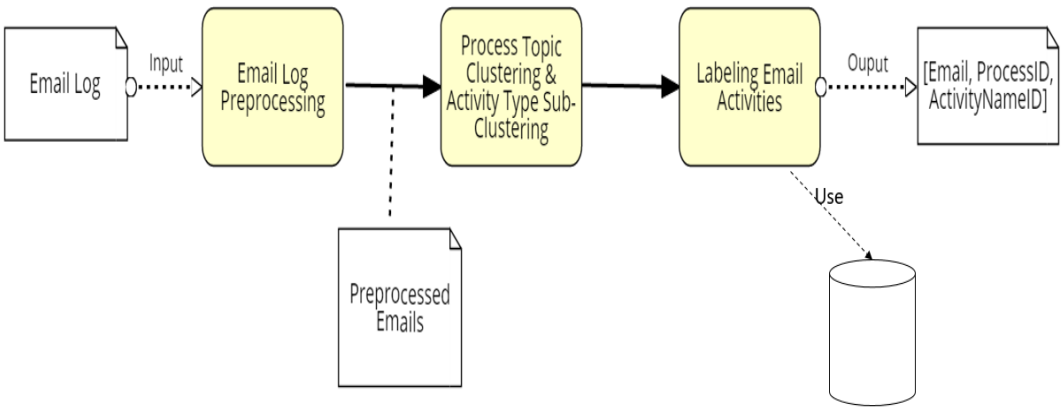

Figure 6.1: Preliminary Approach

Limitations of the preliminary approach Emails are first clustered according to what process topic they belong to and then sub-clustered according to their activity types. This means that each email will be associated to one and only activity type. However, this is not always applicable in all cases. Some emails may not contain any business activities and others may contain more than one business activity. This is considered a drawback in our proposed base-

line approach which motivated us to work on the improvement elaborated in the next section.

6.2 Fine Granularity Activity Discovery Approach (Improved Approach)

6.2.1 Motivation and Approach Overview

Taking into consideration the limitations that appeared in the baseline approach described in section 6.1, we build a complete new approach that works on the emails themselves and not the clusters of emails. The email below is an example that proves the limitation of the preliminary approach. In this motivating scenario, we provide an example that emphasizes the motivations and challenges of our work. Consider the following email taken from an email log of an employee:

From: thomas.myers@enron.com
To: bus.all-hou@enron.com
Subject: Manual Wire and Same Day Payment Authorization

Dear all,
I am pleased to announce that it was an effective day (8/16). We authorized signing manual wires. Same day Tom Myers transfer payments to Georgeanne Hodges. Attached is the list of payments. As always Wes Colwell will also continue to have signing authority.
Thanks,
Tom

Manual examination of the above email reveals that it encompasses the following activities: *"Authorize signing manual wires"*, *"Transfer payments"*, *"Keep signing authority"*. Note, however, that using existing solutions, e.g., our previous solution, will yield a single business activity, namely *"Transfer payments* since the email is clustered with other emails concerned in the payment activity. However, the activity *"Transfer payments"* is one of multiple business activities mentioned in the email. Our previous solution led to the loss of many business information that helps in building the overall process model.

The above shows the need for a solution able to elicit potentially multiple activities from a single email. We need to discover email business activities with a finer granularity. We note also that emails are sometimes associated with resources, e.g., attachment files, web links, actors... For example, the above email is associated with an attachment file containing the list of payments. Such resources can be harvested to derive metadata that can be utilized to annotate elicited activities. The derived annotations can be utilized to help better understanding of the activities and to facilitate their management, e.g., their indexing and search.

Elaborating a solution that is able to elicit multiples activities form an email, and to augment such activities with annotations, raises a number of challenges that need to be addressed.

C_1 An email does not provide only information on business activities, it also provides information on non-business oriented activities. We, therefore, need to be able to distinguish between business and non-business activities (non-business oriented: *announce an effective day*; and business-oriented: *authorize signing manual wires, transfer payments, have signing authority*)

C_2 Not all emails contain information about business activities. We, therefore, need to be able to identify and discard non-relevant emails.

C_3 We need to identify all the emails describing multiple occurrences of the same activity (e.g., all the emails about "transfer payments"). This is challenging, as emails are free text and can use different words to describe the same activity.

C_4 An email may encompass multiple business activities, and as such associating the appropriate metadata with each activity is not obvious. We, therefore, need a means to correlate the information that can be extracted and abstracted from the resources associated with the emails (such as attached files, links and actors) with their corresponding business activities.

Approach Overview

We address the challenges we have just described using the approach depicted in Figure 6.2. It takes as input an email log and produces the set of business activity types it contains, associated with their corresponding metadata.

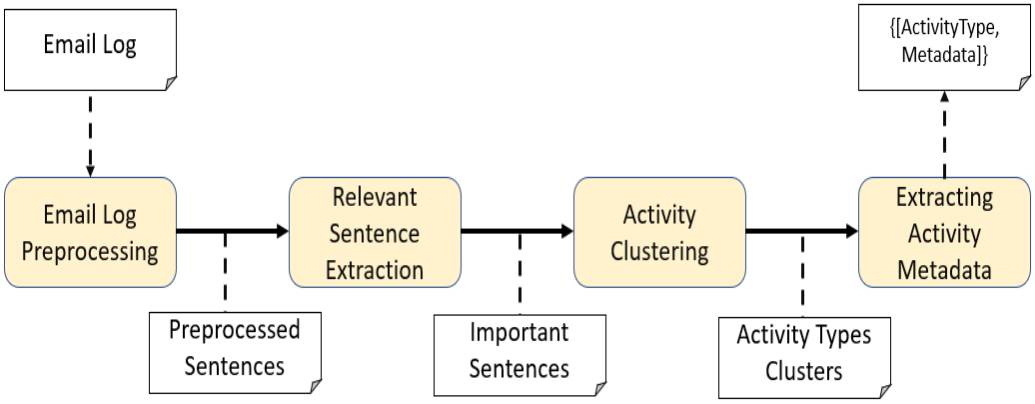

Figure 6.2: Approach Steps

We start our approach by the emails preprocessing step which is already described in section 4.1. We cleanse and transform the emails into a format that is compatible for the analysis goals. The preprocessed sentences of the emails will be then an input for the following phases.

6.2.2 Relevant Sentence Extraction

Following the preprocessing step comes the relevant sentence extraction phase. Not all the sentences that compose an email are about business activities. Some sentences may be talking about personal issues or greetings. We, therefore, need a means for identifying the sentences in the email that provide information about business activities such as the verb-noun pair representing the business activity or some other information characterizing the activity such as names of actors, locations, documents, links etc... We refer to such sentences by *relevant sentences*. To identify relevant sentences in an email, we use a classification technique that associates each email sentence with one of the following labels *Relevant* or *Non-Relevant*. This problem of sentence classification can be related to the problem known in the literature as extractive summarization [27], given that it reduces an input text by discarding non-informative sentences.

While machine learning classification techniques have been thoroughly investigated in the literature, it is widely recognized that there is no *one-size-fits-all* solution. Indeed, there are two parameters that considerably affects the outcome of classification: i) The features used to characterize the population, in our

case email sentences, and ii) the classification algorithm used. In the remaining of this section, we discuss these parameters. In doing so, we present arguments that justify our choices.

Sentence Feature Extraction

We define the following features that characterize the sentences as relevant and business oriented. We divide the features into two categories: *Syntactic Features* and *Semantic Features*.

1. **Syntactic Features:**

   - F_1**: Position of a sentence:** This feature provides the position of the sentence in an email. Usually, the most important sentences in an email are the ones that are located in the middle. The sentences located in the beginning and the end of an email are more likely to be introductory or ending phrases with no significant amount of information.
   - F_2**: Length of the sentence:** Usually short sentences are not as important as long ones [1]. We assume that as the number of words increases in a sentence, the amount of information it holds increases too.

2. **Semantic Features:**

   - F_3**: Number of named entities:** The existence of named entities such as names of people (actors), locations, names of conferences, etc. gives an indication about the possibility that this sentence contains an activity (which is performed by the mentioned actors or in the specified location).
   - F_4**: Cohesion of a sentence with the centroid sentence vector:** We consider the centroid sentence as the entity holding the overall information about an email. Email sentences with higher cohesion (higher similarity value) with the centroid sentence are considered of a higher importance. Using the word embeddings techniques, each email sentence is converted into a numerical vector (by averaging the numerical vectors of its words). The centroid sentence vector is deduced by calculating the average of all the numerical vectors of the email sentences.
   - F_5**: Dissimilarity with greetings and ending phrases:** The greeting and ending phrases are considered to be of no importance for our analysis. Thus email sentences with higher similarity with these phrases are likely to be less important in our analysis. The dissimilarity is computed based on the distance between each sentence and an already defined set of email greetings and ending phrases.

- F_6**: Similarity with the process-oriented activities:** A business process is a flow of business activities connected towards the achievement of some business goals. A business activity is a constituent element of the business process which performs a part of the overall process goal. This feature estimates the likelihood of the sentence to contain a process-oriented activity. Usually, activities in the emails are written in the form of verb-object pairs such as "confirm meeting", "cancel meeting", "send payment", "receive receipt", etc. So we estimate the probability that the verb-noun pair of a sentence is, in fact, a business activity. We calculate it based on the semantic similarity between the verb-noun pairs in a sentence and a collection of process-oriented activities. The set of process-oriented activities is extracted from an external repository of process models including different topics such as purchasing or selling items, insurance agreements, incident management, money transfer, meeting organization, etc. Accordingly, we choose the process activities that are the most similar to the existing verb-noun pairs.

It is important to note that all word and sentence semantic similarity calculations applied in this work are done using Word2vec model [26]. More precisely, Word2vec model characterizes each word by a numerical vector. By averaging the numerical vectors of the words of a sentence, we obtain the sentence's feature vector. Word2vec is a computationally-efficient method for learning high-quality distributed vector representations (called word embeddings).

What is missing in our training data is the label or class of each sentence. We label our training data manually. An expert decides whether the sentence is meaningful from a business-oriented perspective (it contains a business activity or any information about an activity) or not. By having our training data built, we are ready now to train our model.

Classification

In order to ensure we obtain the most efficient classification results, we compared 7 different *non-linear* classification techniques:

a) Neural Networks (NN)

b) Non Linear Discriminant Analysis (NLDA)

c) K-Nearest Neighbor (KNN)

d) Decision Tree (DT)

e) Gradient Boosting Classifier (GBC)

f) Gaussian Naive Bayes (GNB)

g) Support Vector Machine (SVM).

In the experimentation section 6.3, we deduce the classification technique which is the most efficient method in our case. The trained model is used to classify the sentences of emails as important or not from a process-oriented perspective. For each email sentence, its feature vector is obtained and fed to the trained model which decides whether the sentence is relevant or not.

Going back to our example, the relevant sentences are the following:

- $Sentence_3$ We authorized signing manual wires.
- $Sentence_4$ Same day Tom Myers transfer payments Georgeanne Hodges.
- $Sentence_5$ Attached list payments.
- $Sentence_6$ As always Wes Colwell also continue have signing authority.

We can see that the extracted sentences are either the sentences containing the business activities (sentences 3, 4, and 6) or sentences containing annotations about activities (sentence 5)

6.2.3 Activity Types Discovery

The business activities elicited in the previous steps can be further processed to organize them by their *activity types*. Organizing business activities by types has several applications, e.g., they can be used for indexing activities, searching for them and performing analytic queries over them. For example, a company manager may be interested in knowing the average number of business activities of type "shipping goods" that are processed by month.

To organize activities by type, we use clustering techniques. Since we do not have a priori knowledge about the number of activity types, hierarchical clustering is chosen as it does not require predefining this constraint. Hierarchical clustering is applied to sentences containing process oriented verb-noun pairs (i.e. activities). The similarity between two sentences is calculated using the cosine similarity between the Word2vec vectors of their verb-noun pairs (i.e. activities). For each verb-noun pair, an average numerical feature vector is obtained. The distance function used for clustering is:

$$Sim(a_i, a_j) = CosineSim(vn_i, vn_j) \tag{6.1}$$

where vn_i and vn_j are the verb-noun pairs of the activities

According to our visualization study, we apply a cut on the obtained cluster hierarchy. We try multiple cuts on multiple levels to deduce those which provide the best clustering quality. This phase will give as a result a set of clusters where each cluster contains sentences from different emails but with the same activity type ($\{AC_i\}$).

6.2.4 Activity Labeling

After clustering activities contained in emails, our goal now is to deduce their labels. In other words, we should provide labels for the obtained clusters $\{AC_i\}$. For each cluster, we choose the top N verb-noun pairs mentioned in the activity cluster(for example N can be equal to 3). Then one of these verb-noun candidates can be chosen by an expert as a label for the cluster.

6.2.5 Extracting Activity Metadata

The aim in this phase is to enrich the activity types with metadata that provides the user with further analytical capabilities. For example, a manager may want to know what are roles of the employees that participate in an activity type. Another application would be suggesting for a user the document type that should be attached to an email containing a specific activity type. Using the classified relevant sentences obtained in the previous phase, we apply the two following steps: (1) extracting information about each activity type instance in each cluster AC_i and (2) aggregating the extracted information of these activity instances to formulate the metadata for the whole cluster i.e. for the activity type represented by the cluster.

Extracting Information of each Activity Instance

As explained earlier in step 3, email activities are clustered into groups and are labeled by activity types. Thus, all activity instances of the same type will have the same label. Therefore, different names of activity instances of the same type are unified by a single label.

In this step, we extract/infer for each activity instance a set of attributes representing information about the activity. Each activity instance associated to an activity type will be also associated to a metadata describing it. For each activity, the following attributes are extracted when applicable:

- **Organizational role of the people exchanging the email:** We make here the *reasonable* assumption that different occurrences (instances) of the same activity type are exchanged between people of similar organizational roles. The roles of people are deduced from the organizational database. For example, in a company, each email address of an employee will be associated with his/her role in a predefined knowledge-base or directory such as administrator, engineer, etc...

- **Named Entities:** Such as names of enterprises, names of locations, conferences etc.. For example, a named entity would be the name of the enterprise where an activity is applied such as the name of the purchasing company.

- **Actors:** The actors are the people performing the activity described in the email. Actors may not always be the senders or the receivers of the

email. The senders/receivers are people who may be interested to know information about the activity (people involved in the activity).

- **Attachments:** We extract the attached documents, their types and the information they contain such as a bill, document to be filled, filled document, data document, etc.
- **Web pages descriptions:** These are webpages associated to an activity. We mainly extract their domains and descriptions.

To find the values for the attributes, we apply the following steps:

1. Since in each email we may have multiple activities, we need to correlate each activity with the sentence(s) containing information about it. This is done by using:
   - The semantic similarity between the words of the verb-noun pair (the activity) and the words of the other relevant sentences. Sentences including information about an activity are likely to contain words similar to the activity name. For example, suppose we have an email containing the following activity sentence *"We received your payment"* and information sentence *"The receipt of your payment is attached to this email"*. Both sentences contain words about *"receiving"* and *"payment"*. Using the semantic similarity measurements, we can detect the correlation between both sentences.
   - The farness between the activity sentence and information sentence. We suppose that correlated sentences are likely to be close to each other.
2. For each activity, we parse the sentence containing the activity and its correlated information sentences to extract the values of the attributes. Check the experimentation subsection 6.3.4 for detailed explanation on values extractions.

6.2.6 Eliciting Activity Metadata

After extracting information for each activity instance in the cluster, we aim to aggregate this information to get a generalized definition of the metadata describing an activity type.

For the actors and named entities attributes, we keep the information as they are (multiple people of different roles may be involved in one activity type). We just combine the values obtained from different activity instances.

For the attachment type, we check for the most occurring phrases in the text or the title or the name of the attachment to be associated with the activity type.

Regarding the weblinks, we concatenate the different descriptions obtained for different web links.

6.3 Experiments and Results

In this section, we describe in details the experiments we did to validate our approach. We verify the efficiency of the 7 trained models for the classification of relevant sentences, the clustering for activity discovery comparing it with the results of the baseline approach, and for activity metadata extraction.

6.3.1 Experimentation Settings

Dataset:

In order to do our experimentation, we use the Enron dataset [38] which contains several folders belonging to several employees from the enterprise Enron. Each email from the log loaded from a chosen folder can be divided into a set of attributes: sender, receiver, subject, body, and timestamp. We extract the content of the emails from the Enron files. For the training phase, we choose an email folder containing 628 emails, which include in total 5124 sentences.

Implementation packages and tools:

As implementation settings, we use python packages to implement the described approach. The preprocessing step is applied using the Natural Language Toolkit (NLTK) [1] which is an open source Python library for Natural Language Processing. NLTK is a powerful tool for extracting and manipulating texts. In the second phase, we extract the features values for each sentence. Specifically, for the extraction of named entities, we use the tokenization and chunking functions of NLTK. For the cohesion feature, we compute the numerical vectors for the centroid sentences. For this purpose, we use the Word2vec model. We import the Gensim python package in which we load a 3.4 GB Word2vec model containing all vectors of 1 billion words trained on Google news corpus. The same tool is used to calculate the dissimilarity between greetings/ending phrases and the sentences. Finally, for computing the similarity between verb-noun pairs of sentences and the process-oriented activities, we apply two steps:

1. Building the process-oriented activities set: we obtain a process model repository containing about 4000 models of different domains from Signavio. Using this repository we extracted about 21000 business process activities.

2. Extracting the verb-noun pairs from each sentence: we use the natural language processor called Stanford Parser[2] which is able to extract all the parts of speeches from a sentence. For our work, we need the verb-noun (or verb-object) pairs of each sentence.

[1] https://www.nltk.org/
[2] https://nlp.stanford.edu/software/lex-parser.shtml

After computing the values of the features of 5124 sentences, we manually annotate labels for each sentence. The labels refer to whether the sentence should be included or excluded from the analysis. The expert chooses 1 as a label if the sentence contains business process oriented activities or information about the activities, 0 otherwise.

We then train 7 different classification models: Non-linear discriminant analysis, neural network, k-nearest neighbor, decision tree, gradient boosting classifier, Gaussian naive Bayes, and support vector machine. The most efficient trained model is used later for the classification of sentences. All training algorithms are loaded in python using the Sklearn package.

6.3.2 Classification

To test the performance of the obtained classifiers, we choose a subset of Enron dataset containing 50 emails with 724 sentences. The sentences are transformed into the defined features to be used as an input for the trained models. To check which classifier has the best prediction efficiency, we measure the prediction efficiency using classifiers evaluators:

1. $Recall = \frac{TP}{TP+FN}$
2. $Accuracy = \frac{TP+TN}{N}$
3. $Precision = \frac{TP}{TP+FP}$
4. $F - measure = 2\frac{precision*recall}{precision+recall}$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative, N is the total population. Table 6.1 below show the different classification evaluators values for the 7 different classifiers.

Evaluator	NN	NLDA	KNN	DT	GBC	GNB	SVM
Recall	0.68147	0.67934	0.63393	0.78164	0.78408	0.67099	0.67320
Accuracy	0.69952	0.70316	0.66134	0.78892	0.80333	0.70306	0.69760
Precision	0.69041	0.70217	0.65639	0.78661	0.79098	0.67793	0.67101
F-measure	0.68351	0.68246	0.63773	0.78643	0.78312	0.67402	0.67290

Table 6.1: Binary classifier performance evaluation

Using these results, we deduce that the Gradient Boosting Classifier (GBC) provides the best values which means it is the most efficient classification model for our goal. The sentences classified by the GBC are going to be clustered in the next step.

Figure 6.3 shows a bigrams cloud of the verb-noun pairs or in other words the business activity instances of the tested emails.

Figure 6.3: Bigrams Cloud for Business Activity Instances

6.3.3 Clustering

In order to efficiently measure the performance of the clustering, we choose only the sentences that are correctly positively classified (True Positives). We use the Scipy [3] package to apply the hierarchical clustering on the activities of the positively classified sentences. We obtained 7 clusters where each cluster represents an activity type. The obtained activity types are: "Fill Document", "Modify Data", "Suggest Meeting","Open Discussion", "Modify Rule", "Modify Capacity", and "Provide Product".

The results of the activity discovery step are compared to the results of a preliminary approach. In the preliminary approach, we apply multi-level clustering to obtain email topics (first level) and activity types (second level). We suppose that each email contains only one activity type. Each email is characterized by a numerical feature vector computed as the average of the Word2vec feature vectors of all its words. These feature vectors are then used to calculate the similarity between emails. Applying this approach as a baseline, we obtain only two clusters of the activities "Modify Data" and "Suggest Meeting".

Obtaining multiple activity types in one email proves that our new approach that first extracts relevant sentences and then clusters process oriented sentences provides more accurate activities discovery than our preliminary approach where

[3] https://www.scipy.org/

clustering is applied on the whole email bodies.

Table 6.2 below shows the different clustering evaluation metrics values for the two approaches.

Clustering Metrics	Current Approach	Baseline Approach
Purity	0.86	0.69
Rand-Index	0.81	0.65
Precision	0.918	0.63
Recall	0.87	0.57

Table 6.2: Clustering performance evaluation

6.3.4 Metadata Extraction

The main information we extract for each activity are:

- **Organizational role of the people exchanging the email:** The role of sender/receiver(s) could be obtained from the organizational database or directory of the enterprise. We have built a directory for the email addresses contained in the testing emails. We associate to each email address a specific role such as administrator, director, engineer, etc. For each email (containing the activities) we identify the roles of the sending and receiving entities. This kind of database is usually available in enterprises and organizations.

- **Named Entities:** such as locations, organizations names. We parse email sentences for such names using the Stanford Parser.

- **Actors:** Actors are people who perform a specific activity. We associate each activity type to the set of actors performing it. We fetch the sentence containing the activity and its correlated information sentences for Named Entities of type "Person".

- **Attachments types:** We build (simulate) a thesaurus containing names and extensions of attachments associated to their subjects or type. For example, the activity instances of type "Modify Data" are associated with attachments of names "Template.xls" and "Doc.xls". Thus we add these names to the thesaurus associated with document type as "excel data document to be edited (add or delete)". This thesaurus always enriched and used to extract types for similar attachment names.

- **Web pages descriptions:** We use web scraping [4] in python to automate keywords extraction of each website mentioned in the sentences. This tool loads the HTML file of a website. We extract words from website's text that are most similar to the website title words.

[4] https://data-lessons.github.io/library-webscraping/04-lxml/

Table 6.3 shows an example on activity instances belonging to the same activity type "Modify Data" with their corresponding information.

Table 6.3: Activity instances of the same activity cluster (same activity type) and their associated information

Activity Name	Sender/Receiver role	Attachment	URL description	Actors Information	Named Entities (NE)
Modify Data	Administrator	Template.xls	Global innovation economy	No actor	Enron
Edit Data	Administrator	Template.xls	Insights on economy	Jennifer Stewart: Administrator	Enron
Add Information	Engineer	Doc.xls	No Description	No actor	No NE

The set of values of the metadata attributes can be inspected by the user. After applying data aggregation, we obtain for the activity type "Modify Data" the following information as metadata:

- Sender/Receivers Role: Administration, Engineering
- Attachment Type: Excel data document to be edited (add or delete).
- Web-page description: Global innovation and insights about Economy.
- Actor: Jennifer Stewart, Administrator
- Named Entities: Enron

6.3.5 Discussing Some Analytical Questions

Each activity type is now correlated with a set of attributes which facilitates answering some analytical questions. Discussing these analytical questions is considered a base for further future information retrieval. We provide in the following some analytical questions that can be answered using the results that we obtain in our improved activity discovery approach:

Q_1 **What are the business activities executed by a specific employee? (to identify time-consuming tasks that are not known to be assigned to him).**
Each activity type is correlated to a set of actors which allows us to specify all the business activities an employee applies. For example, the employee Jennifer Stewart is responsible for the execution of the activity "Edit Data".

Q_2 **What are the groups of people doing similar work? (they apply similar activity types).**
People involved in an activity are the people exchanging emails about this activity which can be deduced from the sender/receiver(s) part of the email. For example, the group of people who are involved in the activity "Modify Data" are: Stephen Allen, Tony B, Herb Caballero, Kenneth, Roger Raney, Henry Van, Linda Adels, Paul Duplachan

Q_4 **What kind of documents are sent as email attachments for a specific activity?**
Each activity type is characterized by an attribute describing the attachments usually associated with it. Since we didn't have the contents of the attachments of Enron dataset, we only can extract information from the name and the type of the attachments of a specific activity.

Q_5 **What domains of web pages or links are used for a given activity?**
Using the obtained keywords describing the links and web pages associated with activities, a user can deduce their domains.

Q_6 **Give information about people that usually apply an activity.**
Each activity type is associated with a set of actors with their information. Figure 6.4 is an example graph about the relations between actors (circular nodes) regarding a specific activity (rectangular node) where the relations are represented by edges. For example, the actor Jennifer S. applies the activity "Edit Data". We can see the Jennifer S. and Carrie R. apply the same activity with the same set of people. The number on the edges represents the number of times Jennifer S. sent emails about the activity "Edit Data" to the other employees.

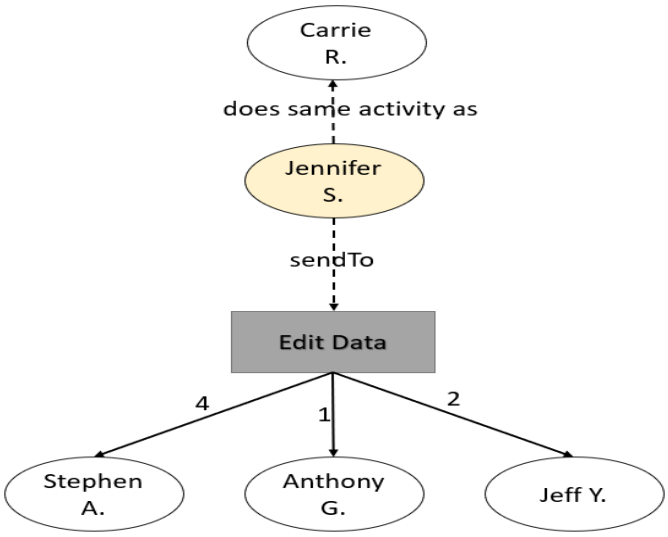

Figure 6.4: An information graph for the actor Jennifer S. applying the activity "Edit Data"

6.3.6 Discussion

To compare the approach applied in this chapter with existing works and discuss the differences, we categorized the existing works into three categories. Our approach first starts by extracting relevant email sentences, then extracts email activities and finally associate activities to metadata. Therefore, we divide the discussion as follows:

Relevant Sentences Extraction This is also known as Email Summarization in the literature. Several researches have tackled the extractive summarization of texts particularly emails. Extractive summarization of emails in previous works is limited to the goal of discovering the email sentences that best represent the email. For example, some existing approaches use parsing methods which choose some clue words or by asking the user to identify some requirements which help in the important sentences extraction. However, in our work for extracting important sentences from emails, we build our assumption on the fact that the user has no a priori knowledge about the topics of emails. For this reason, we provide the users a tool that automatically extracts important sentences from their emails. Once the features of the email sentences are identified, our trained classification model can be used to automatically identify the sentences that are important for our analysis. Furthermore, all previous existing approaches extract important sentences either from a general point of view or from the user's point of view. However, in our work we go more deeply to identify sentences that are important for our analysis from a business oriented point of view. Most of our defined sentences features are dedicated for a business oriented classification of email. This will cater to deduce sentences that are important for our goal i.e. extraction of business oriented information from emails. To summarize, for an efficient extractive email summarization from a process oriented perspective, the following criteria are considered as important:

a) Important sentences should be extracted automatically without the interference of the user.

b) Extracted sentences should be interesting from a business oriented point of view.

A comparison between the different previously mentioned related works in provided in table 6.4 to check whether their approaches cover the demanded requirements or not.

Requirement \ Related Work	[6]	[7]	[61]	[40]	[69]	[58]	[45]	[49]	our approach described in 6.2.2
a	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
b	No	No	No	No	No	No	No	No	Yes

Table 6.4: Requirements comparison for email summarization between different approaches and our approach.

Email Activities/Tasks Extraction While useful, current emails management tools lack the ability to recast emails into *into business activity centric resources*. In the existing works tackling the extraction of activities from emails there are two limitations: (1) They have predefined set of activity types that they tend to discover in emails in which each email will be associated to one of them. In the speech acts discovery approaches, an already defined set of speech acts is predefined to classify email verb-nouns accordingly; (2) In some of the approaches for extracting business email activities, each email is associated to only one activity. However, these two facts may affect negatively the business activity identification task. A predefined set of activities is not always available since there is no a priori knowledge about the process topics available in the email log and thus there is no knowledge about the existing business activities. Moreover, according to our analysis of emails, we discovered that remarkable number of emails contain more than one activity at a time. For this reason, in our approach described in this chapter, we overcome this by not being limited to a specific set of predefined activities. In addition, we extract multiple business process activities from a single email. In the following, we compare different approaches for business activities discovery from emails according to some criteria:

a) No predefined set of activities should be used to identify email activities. A user would not be able to identify tasks present in their email log.

b) Dealing with multiple activities in one email. An email may contain several activities (not only a single activity or task).

c) Extracted activities should be business oriented.

A comparison between the different previously mentioned related works in provided in table 6.5 to check whether their approaches cover the demanded requirements or not.

Requirement \ Related Work	[20]	[11]	[12]	[9]	our approach described in 6.2.3
a	No	No	No	No	Yes
b	No	Yes	Yes	No	Yes
c	No	No	No	No	Yes

Table 6.5: Requirements comparison for email process activities discovery between different approaches and our approach.

Email Activities Metadata Extraction In the existing approaches, authors develop tools such as TaskMaster [3] that help users manage URLs and files associated to tasks in emails. Up to our knowledge, non of the related approaches is able to automatically associate activities of emails to a high level metadata. In the approach presented in this chapter, we work on specifying for each email activity a set of information or metadata which enrich and describe the corresponding activity. We tackle the challenge when several activities are present in one email where the information associated to an email such as attachments, URLs etc.. should be correctly divided on the email activities. This step in our approach opens a door for a high level analysis in which many analytical queries can be answered using to the extracted metadata. In the following, we compare different approaches for metadata extraction from emails according to some criteria:

a) Automatically associating activities of emails to their metadata.

b) Coping with the fact that email metadata should be associated to multiple activities of the same email.

A comparison between the different previously mentioned related works is provided in table 6.6 to check whether their approaches cover the demanded requirements or not.

Requirement \ Related Work	[3]	[28]	our approach described in 6.2.5
a	No	No	Yes
b	No	No	Yes

Table 6.6: Requirements comparison for email task management between different approaches and our approach.

The preliminary approach of this chapter have been published in [32]. The rest of the work in this chapter have been published in [33] and [34].

CHAPTER 7

Relational Activities-Instances Discovery

Contents

As mentioned earlier, our goal is to help the user manage his/her emails from a process-oriented perspective. Instead of dealing with emails or threads of emails, he/she manages the received messages as business processes or workflows. Such processes are represented as related activities grouped into process instances manifested in messages. Towards this goal, we already described approaches in this work that extract business process activities from email messages and relate them into business process instances. We have mentioned that, on one hand, the elicitation of business activities from a set of emails in itself opens up the door to perform some activity analytics such as identifying the business activities executed by specific employees or the number of times a user applied an activity. On the other hand, discovering business process instances from email logs is useful for performing several analysis queries such as obtaining the average duration of a specific business process, calculating the number of process instances enacted each month, deducing the process instances consuming the longest period of time to be achieved, or identifying the process instances which involve specific entities (for example actors) etc...

Previous approaches (including our work) have tackled the two problems (extraction of email activities and email process instances discovery) separately by considering the structured and non-structured contents of the emails. We have proposed an approach that discovers and labels email activities only by using its content.

Up to our knowledge, related works about the extraction of business process (1) activities and (2) instances from email logs follow non relational approaches i.e. they deal with each learning approach as a separate problem. In other words, the classification approaches for extracting process activities and identifying email instances have been applied in an independent manner. However, it is a fact that relational data sets a special opportunity for improving classification. The opportunity exists if, when two objects are related, inferring something about one object can help you infer something about the other [46].

Applying this fact on our context, activity related information available in separate email messages can provide relational clues that can be used to build links between emails grouping them into business process instance. In particular, we mean by activity related information, the information associated to each email activity such as the attachments, URLs, actors (people applying the activity) etc.. For example, suppose we have email e_1 containing activity a_1 with activity information $info_1$, and email e_2 containing activity a_2 with activity information $info_2$. Particularly, e_1 contains activity "Send Draft" with an attachment of type "Report" including specific named entities and information about a topic T, and e_2 contains activity "Review Draft" with an attachment of type "Report" including specific named entities and information about the same topic T. Our aim is to check if the information associated to a_1 and a_2 can help in relating e_1 and e_2 into the same business process instance.

Similarly, we claim that related emails (emails belonging to the same process instance) can provide valuable context for improving the extraction of email business activities. For example, suppose we know that e_1, e_2, e_3 and e_4 are related to the same process instance where e_1 includes the activity "Send Application", e_2 includes the activity "Request Documents" and e_4 includes the activity "Accept Application". Using the relational approach and the information contained in the related emails, can we deduce that e_3 probably includes the activity "Send Documents"?

Here comes the importance of the iterative relational approach. Relational data offer a unique opportunity for improving the classification accuracy of statistical models [46]. We start by applying initial iterations for (1) email process activities classification and (2) email log process instances discovery where the inferences with high confidence of these two baseline approaches are fed back as input data in the subsequent iterations to strengthen the inferences accuracy.

Therefore, the contributions in this chapter can be summarized as follows:

- We propose an iterative relational approach that uses information about email business activities to identify emails of the same process instances and vice versa. In particular, we investigate (1) how information about email activities can assist with finding emails of the same process instances, and (2) how features of emails of the same process instances can assist with the classification of emails business activities.

- We evaluate the efficiency of our approach using Enron dataset compared to the baseline non-relational approaches.

The chapter is divided into several sections. In section 7.1, we decompose the problem into 4 different problems where two of them are baseline approaches previously treated in chapters 5 and 6. The relational process instances and activities discovery are elaborated in sections 7.2 and 7.3 respectively. The overall iterative algorithm and the experiments are described in section 7.4.

7.1 Problem Decomposition

There are multiple ways to approach a classification problem. One way is to ignore related objects (in our case objects are the emails content) and build classifiers based only on the properties of an object in isolation. Another approach uses information in related objects and dynamically updates these information as predictions about related objects change. Iterative classification uses the latter way by applying a classification approach in a dynamic way to fully leverage the relational structure. Our approach is based on the idea that process activities information in separate emails can be used to group emails into process instances. Similarly, messages belonging to the same process instance can provide a valuable context for emails activities discovery. Therefore, we claim that an iterative relational treatment of these two problems would improve the efficiency of the obtained results compared to the separate treatment.

In a non relational approach, we use baseline methods to discover email process activities and to relate emails into process instances. In baseline text analysis approaches, we would use the content of the messages (structured and non-structured content) in isolation to find emails belonging to the same process instances and to label emails activities.

However, in the relational approach for email process instances discovery, we can use both the email content and the information about email activities. For example, suppose a candidate email is about purchasing an item, call it item P and another candidate email is the confirmation of the purchase of item P. Both emails will contain metadata about P with names of the selling company or the name of the buyer or attachments about the item characteristics or the bill. The usage of these information about the activities "Purchase Item" and "Confirm Purchase" can help in relating both emails into the same process instance. So our relational classification model can infer that if an email e_1 contains activity "Purchase Item" associated with metadata $info_{PurchaseItem}$ and an email e_2 contains an activity "Confirm Purchase" with metadata $info_{ConfirmPurchase}$ where $info_{PurchaseItem}$ is similar to $info_{ConfirmPurchase}$, then e_1 and e_2 are likely to belong to the same process instance.

On the other hand, information about discovered process instances can be used to help in extracting email business process activities. If we re-use the above example knowing that e_1 and e_2 belong to the same process instance I where e_2 is a response to e_1. Taking into consideration that e_1 contains the activity "Purchase Item" and that e_2 is the email following e_1 in the process instance I, we build a classification model that (using old occurrences of similar emails as e_1 and e_2) predicts that e_2 is likely to contain the activity "Confirm Purchase".

Therefore, we decompose the overall problem into four different phases :

- **Phase 1**: Extracting process activities with their metadata from emails using their content (i.e. without using information about emails process instances).
- **Phase 2**: Identifying process instances from emails using a similarity

computation (i.e. without using any information about email process activities).

- **Phase 3**: Using extracted email activities and their metadata to improve the identification of process instances in emails.
- **Phase 4**: Using links between emails belonging to the same process instances to improve the extraction of emails process activities.

In our iterative relational classification approach, we start by solving **Phase 1** and **Phase 2** using baseline methods dealing with each problem separately. The results of **Phase 1** and **Phase 2** are then used to solve **Phase 3** and **Phase 4**. Iteratively, the dynamically changing inferences (with high confidence) deduced from **Phase 3** is fed back to **Phase 4** to improve the accuracy of its results and vice versa. Algorithm 1 shows the main steps of our iterative relational classification approach.

1. Baseline approach for email activities extraction with their metadata (**Phase 1**).
2. Baseline process instances discovery from an email log (**Phase 2**).

for *Iteration* $i = 1....K$ **do**

3. Use extracted email activities and their metadata to identify process instances (**Phase 3**).
4. Use discovered process instances for email process activities extraction (**Phase 4**).

end

Algorithm 1: Iterative Relational Classification Approach.

In chapters 5 and 6, we have worked on approaches that tackle phases 2 and 1 respectively. In chapter 5, we have built a baseline approach that discovers business process instances from email logs using the intrinsic features of the emails. Similarly, in chapter 6, we have built a baseline approach that extracts business process activities from email logs. In the following sections, we will use the results of chapters 5 and 6 to launch the approaches solving phases 3 and 4.

7.2 Discovering Email Process Instances Using Email Activities (Phase 3)

In **Phase 2**, we work on identifying business process instances in an email log using only the content of the email. In this phase, we aim to exploit the relational classification concept in the process instances discovery approach. Unlike phase 2, this approach does not only depend on the structured and non structured content of an email, but also makes use of the links between email business activities information. As explained before, we claim that using information about email process activities can help in relating emails of the same process instance. Our goal is to prove the positive impact of the relational approach on the accuracy of emails process instances discovery phase.

To recall, the goal in this phase is to group emails according to what process instance they belong to. For checking if two emails are related to the same process instance using email activities, a relational classification model is built for this purpose. The input of the classification model is a feature vector representing a pair of emails and the output is 1 if the emails belong to the same process instance, and 0 otherwise. To train the classification model, we need to produce a training dataset from the available email log.

7.2.1 Training the Classification Model

The purpose behind training the classification model is that it would be able to check if a pair of emails belong to the same business process instance or not. So the classification model should accept information about pairs of emails as an input. Therefore, the training dataset is prepared by defining for each pair of emails (e_i, e_j) in the training email log a vector representing the relation between the activities metadata of e_i and e_j such that e_j occurs after e_i (to ensure that we take all combinations of emails into consideration and avoid repetitions).

As described in **Phase 1**, each email contains a set of business process activities where each email activity is associated to a metadata describing it. The activity and its metadata are going to be used to define the feature vectors representing the relations between pairs of emails. In order to build and label the training dataset, we first start by applying the baseline email activities extraction approach described in **Phase 1** where the training email log E is associated to the set of the activity types it contains A=$\{a_1$,a_2,...,$a_k\}$, such that each email $e \in E$ contains activities of types that belong to A. Each activity type instance is associated to some information or metadata describing it. Algorithm 3 represents the main step for defining the training feature vectors of E. Algorithm 2 which is used in algorithm 3 shows that for each email pair (e_i , e_j) in E , a feature vector is added to the training dataset with its labels. Each feature vector represents the relation between activities of e_i and activities of e_j by calculating the semantic similarity between the attributes of the activities metadata of both emails.

To label these feature vectors, we use the results of the baseline approach in **Phase 2**. We exploit the emails which are clustered with **high confidence** (true positives and true negatives). If e_i and e_j are true positives, we label their feature vector by 1. If they are true negatives, we label their feature vector by 0. Once the training dataset is ready, we train a classification model iM in which we try several non-linear classification techniques (mentioned in the experiments section) to deduce the one with the most efficient results.

Result: Feature vectors of all pairs of activities of (e_i,e_j)
Input: Pair of emails (e_i,e_j);
Input: Activities set A (in E);
Let $\{act_i\} \in A$= set of activities in e_i;
Let $\{act_j\} \in A$= set of activities in e_j;
for *a in* act_i **do**
  for *b in* act_j **do**
    $info_1$=metadata(a);
    $info_2$=metadata(b);
    sim=Similarity($info_1$,$info_2$);
    Add sim to the feature vector fv of the pair of emails (e_i,e_j);
    if (e_i, e_j) belong to the same process instance according to the highly confident results of **Phase 2**, label fv by 1, 0 otherwise;
  end
end
Algorithm 2: Defining and labeling the feature vectors for a pairs of emails.

Result: Classification model for instances identification: iM
Input: Training email log E;
for e_i *in* E **do**
  for e_j *in* E **do**
    if *timestamp*(e_i)<*timestamp*(e_j) **then**
      Define the feature vector of (e_i,e_j) and its label by applying algorithm 2;
      Add the obtained feature vector to the training dataset;
    end
  end
end
Train the classification model iM using the training dataset;
Return iM;
Algorithm 3: Training the relational classification model for email process instances discovery.

7.2.2 Testing the Classification Model

Once iM is trained, it can be used for testing email logs to identify emails belonging to the same process instance. For an incoming testing email log, first, the baseline approach for extracting activities (**Phase 1**) is applied. Using the emails activities and their metadata and as described in algorithm 3, a set of feature vectors is obtained for each pair of emails in the testing email log. These feature vectors are fed to trained model iM which will output 1 if the two emails are related or 0 otherwise.

Emails classified as related are grouped together. Therefore, this phase re-

sults in groups of related emails, where each group represents the emails belonging to the same process instance. Consider that the obtained set of instances is I=$\{I_1$,I_2,..., $I_n\}$.

7.3 Discovering Email Activities Using Process Instances (Phase 4)

Unlike in **Phase 1**, in this phase, the classification of email business process activities does not only depend on the content of the email itself, but also on the fact that an email is related to other emails forming a process instance. According to the relational classification, we assume that emails belonging to the same process instance can give an indication that helps in extracting business process activities from an email. Therefore, here, our study depends on two main factors: the content of the message and the properties of the relations between emails of the same process instance. Our goal is to study whether the features obtained using the identification of email process instances can improve the email activities extraction.

Again, the problem here is treated as a supervised learning task in which the goal is to build a classification model that can identify the process activities of an email. However, in this phase, the feature vector for training the model is different from that of **Phase 1**. The training dataset is built using the activities of an email and its neighbor emails (emails of the same process instance).

7.3.1 Training the Classification Model

A business process instance is composed of a set of emails, where each one contains a set of activities. Each email of a process instance can be seen as a result of the previous message and cause of the later one. These relations between emails of the same process instance are translated into feature vectors for training the relational classification model. To apply the email activities relational discovery, a training dataset is built and labelled to train the classification model. We consider that, for each message in the training email log, we have an initial identified set of activities $\in A$ annotated by an expert. Algorithm 4 explains the steps of defining the emails feature vectors for the training dataset.

We start by applying the baseline process instance discovery approach in **Phase 2** to identify initial email instances eI by using the true positives and true negatives of the obtained results. The algorithm loops over all instances of the process instances set eI in the email log. Each process instance $I_i \in eI$ is made up of a set of emails. For each email $e_{ij} \in I_i$, its predecessor emails $pred_{ij}$ and ancestor emails ans_{ij} are identified. The feature vector fv_{ij} of each email e_{ij} represents the occurrences of the activities of A in the emails of $pred_{ij}$ and ans_{ij}. Thus, the length of fv_{ij} is equal to length of A. The feature vector is built in a way such that if an activity $a_k \in A$ occurs in any of the emails in $pred_{ij}$ or in ans_{ij}, we update fv_{ij} by adding 1 in its k^{th} position. Using this algorithm, each email e_{ij} is associated to a vector fv_{ij} representing the

occurrences of activities of A in the neighboring messages.

Result: Training feature vectors
Input: Email process instances: eI;
Input: All email log activities: A;
for *instance* I_i *in* eI **do**
  for *email* e_{ij} *in* I_i **do**
    Initialize feature vector fv_{ij} of e_{ij} by 0's representing A s.t. length(fv_{ij})=length(A);
    Identify the predecessors $pred_{ij}$ of e_{ij} according to I_i;
    Update fv_{ij} for e_{ij}: add 1 in fv_{ij_k} if a_k of A occurs in in ans_{ij} or $pred_{ij}$;
  end
end

Algorithm 4: Building the training feature vectors of the email log for activity discovery using email process instances

To finalize our training dataset, each of the obtained feature vectors using algorithm 4, should be labeled. In the training dataset, each email is associated to a set of known activities from A. The class label for each email in this training dataset is a vector (a multi-label class). For each email e_{ij}, the label vector represents the occurrences of activities of A in e_{ij}. Thus, the vector is updated such that 1 is added in the k^{th} position if $a_k \in A$ occurs in e_{ij}. Using these training feature vectors and their labels, we train the classification model aM. In fact, we tried several non-linear classification methods to check which one provides the best classification results (they are mentioned in the experiments section).

7.3.2 Testing the Classification Model

Once aM is trained, it can be used for future email logs to enhance the extraction of process activities from them. For evaluating the performance of the classification model aM, we use a testing dataset. The testing emails should belong to the same process topic as that used for training the classification model. Suppose aM is trained on emails about the process topic "Meeting Scheduling", the testing emails should belong to the topic "Meeting Scheduling" as well.

The testing email log contains a set of emails where each email is identified to which process instance I it belongs and what initial activities it contains. Using these information, we obtain for each email a feature vector by following the steps of algorithm 4. Once the features vectors of the training email log are ready, they are used then as an input for the classification model aM. The model aM outputs for each input feature vector of email e, a multi-label class or vector representing the set of activities in the corresponding email e.

7.4 Iterative Relational Classification Approach

Our goal in this work is to check whether the relational classification technique helps in improving the classification accuracy for both (1) Process Instances Discovery in Email Logs, and (2) Extraction of Business Process Activities from Emails. As explained in the **Phase 3** and **Phase 4**, we utilized information about email business activities for the identification of process instances and vice versa. In this section, we apply phases 3 and 4 in an iterative fashion to dynamically update the attributes of some objects as inferences are made about related objects.

We start by applying the baseline approaches for business process activities extraction and for business process instances discovery in emails as explained in **Phase 1** and **Phase 2**. Using their results, we launch the algorithms of **Phase 3** to discover whether using information about email activities improves the identification of related emails. Once the results of **Phase 3** are obtained, we launch the algorithm of **Phase 4** which take as an input the results of **Phase 3** to check if these information can help in better extracting email business activities.

Iteratively, we propagate the high accuracy results of **Phase 3** to apply **Phase 4** and vice versa. Multiple iterations are applied until the accuracy of **Phase 3** and **Phase 4** are no more improving.

7.4.1 Experiments and Results

In this part, we describe in details the settings and the results of the relational approach experimentation. We also show the difference between the results of the baseline approaches and the iterative relational classification approach.

As for the experimentation settings, we use some of the available python packages. The Natural Language Toolkit (NLTK) package is utilized, it is an open source Python library for Natural Language Processing (NLTK package is used for entities extraction. Its basic technique for entity detection is chunking which segments and labels multi-token sequences [1]). In addition, we use the powerful and rich Scikit-learn package developed by David Cournapea in 2007 which provides a range of supervised and unsupervised learning algorithms via a consistent interface. For the Word2vec tool, we import the Gensim python package. We load a 3.4 GB Word2vec model containing all vectors of 1 billion words trained on Google news corpus.

As mentioned earlier in algorithm 1 describing the relational approach, we start our experiments by applying phases 1 and 2 and then iteratively loop over phases 3 and 4. We extract from the Enron email dataset, an email log E of 300 emails concerned by the business process topic *recruiting* using the process topic discovery approach described in chapter 4. If we quickly span the email log, we realize that there are business activities in the email texts such as "conduct interviews", "offer recruit", "find candidates" etc... We divide the

[1] http://www.nltk.org/book/ch07.html

email dataset into two parts. A part consisting of 250 emails used for training the classification models through out our experiments (name it $E_{training}$). The other part consisting of 50 emails used for testing the obtained classification models (name it $E_{testing}$).

Starting with the steps of **Phase 1**, we use the python packages mentioned above to prepare the training feature vectors for $E_{training}$ by parsing the training email log and extracting information from its content. The obtained feature vectors are labeled and then used to train the classification model in **Phase 1**. To test the accuracy of the obtained model, the feature vectors of the emails of $E_{testing}$ are extracted and fed to the trained model to obtain the results of the model. In fact, we try 7 different classification techniques (1) Gradient Boosting (GB), (2) Neural Networks (NN), (3) Non-Linear Discriminant Analysis (NLDA), (4) K-Nearest Neighbor (KNN), (5) Decision Trees (DT), (6) Gaussian NB (GNB), (7) Support Vector Machines (SVM). The results are shown in table 7.1.

Evaluator	NN	NLDA	KNN	DT	GDC	GNB	SVM
Recall	0.68147	0.67934	0.63393	0.78164	0.78408	0.67099	0.67320
Accuracy	0.69952	0.70316	0.66134	0.78892	0.80333	0.70306	0.69760
Precision	0.69041	0.70217	0.65639	0.78661	0.79098	0.67793	0.67101
F-measure	0.68351	0.68246	0.63773	0.78643	0.78312	0.67402	0.67290

Table 7.1: Binary classifier performance evaluation

Phase 2 is then applied on the dataset E. We start by extracting information from the email texts and calculating the similarities between pairs of emails using the obtained similarity function. Hierarchical clustering is then applied on the calculated similarities in which we get as a result a hierarchy that can be cut on different levels. We apply several cuts to deduce the one that provides the biggest number of True Positives and True Negatives clustered emails. The clustering results are shown in table 7.2.

Evaluation Metrics	Process Instance Clustering
Accuracy	0.75
Precision	0.77
Recall	0.64
F-measure	0.7

Table 7.2: Clustering quality metrics results

Once the baseline approaches are applied, we launch the relational classification approaches. We first start by **Phase 3**. To train the model of **Phase 3**, we use $E_{training}$, in which for each email the set of the activities it contains and their metadata are specified by applying **Phase 1**. Using these activities

and the metadata set, we obtain the feature vector for each pair of emails in $E_{training}$. To label this dataset, we use the true positives and true negatives obtained in the results of **Phase 2**. Once the initial training dataset is available, the classification model iM of **Phase 3** is trained.

To train the classification model in **Phase 4**, we use the email activities obtained with high confidence from **Phase 1**. The feature vector with its multi-label class vector for each email is defined as explained earlier. The training dataset trains the classification model aM accordingly.

Once all models are trained, we start by getting the email activities and their metadata for $E_{testing}$ using the baseline approach of **Phase 1**. According to the algorithms of **Phase 3**, we obtain the feature vectors of the emails pairs in $E_{testing}$. These vectors are fed to the trained model iM in which it outputs 1 if the emails belong to the same process instance and 0 otherwise. Thus, emails belonging to the same process instances are grouped together.

The results of **Phase 3** for $E_{testing}$ are used in **Phase 4**. In other words, the obtained instances of the previous phase are used here to define the testing feature vectors of $E_{training}$ as described earlier in **Phase 4**. These vectors are fed to aM which outputs for each email the set of activities it should contain. Each activity is again associated with its metadata and the results of this phase are fed again to **Phase 3**.

Phases 3 and 4 are iteratively conducted using the results of each other until the accuracy of the obtained results is no more changing (positively) (until no changes in the process instances or no new email activities are discovered). We try training the models on several classification techniques: (1) Gradient Boosting (GB), (2) Neural Networks (NN), (3) Non-Linear Discriminant Analysis (NLDA), (4) K-Nearest Neighbor (KNN), (5) Decision Trees (DT), (6) Gaussian NB (GNB), (7) Support Vector Machines (SVM). Therefore, in each iteration, we take the results of high confidence of **Phase 3** (according to the classification evaluation metrics) and input it to **Phase 4** and vice versa. We evaluated the results of the iterative relational classification on each of the models using some metrics: **Accuracy**, **Precision**, **Recall**, and **F-measure**. The obtained results over several iterations of **Phase 3** are shown in the tables 7.3, 7.4, 7.5 and 7.6, and the results of the multiple iterations of **Phase 4** are shown in tables 7.7, 7.8, 7.9 and 7.10

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.867707	0.867704	0.866024	0.866041	0.865211	0.331971	0.862711
Iteration 2	0.890159	0.883510	0.877687	0.784353	0.891826	0.316183	0.876024
Iteration 3	0.907659	0.904326	0.886853	0.896013	0.903499	0.294488	0.884360
Iteration 4	0.915118	0.920937	0.901798	0.905128	0.913451	0.560685	0.894308

Table 7.3: Evaluating the **Accuracy** of **Phase 3**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.853333	0.759524	0.726667	0.257744	0.764286	0.149046	0.766667
Iteration 2	0.906349	0.910775	0.683849	0.858398	0.884444	0.199869	0.905556
Iteration 3	0.893784	0.888425	0.700000	0.922222	0.816093	0.182415	0.937500
Iteration 4	0.880343	0.865407	0.725160	0.846111	0.8404450	0.249827	0.855556

Table 7.4: Evaluating the **Precision** of **Phase 3**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.289997	0.331347	0.289555	0.186854	0.301057	0.638209	0.172750
Iteration 2	0.425625	0.499188	0.402743	0.266371	0.539924	0.642985	0.279838
Iteration 3	0.421161	0.524926	0.400323	0.265527	0.524926	0.630338	0.304278
Iteration 4	0.880343	0.865407	0.725160	0.846111	0.840450	0.649827	0.85556

Table 7.5: Evaluating the **Recall** of **Phase 3**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.432137	0.463705	0.405445	0.310334	0.430029	0.426042	0.487555
Iteration 2	0.568435	0.642940	0.503235	0.399756	0.650852	0.329419	0.425448
Iteration 3	0.560576	0.657556	0.508770	0.407132	0.628556	0.304404	0.452869
Iteration 4	0.615970	0.711635	0.588275	0.396229	0.703586	0.388708	0.440936

Table 7.6: Evaluating the **F-measure** of **Phase 3**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.7282	0.7435	0.6153	0.7282	0.7269	0.1705	0.6269
Iteration 2	0.7358	0.7500	0.6307	0.7500	0.7589	0.0.1589	0.6538
Iteration 3	0.7038	0.7282	0.6205	0.7256	0.7256	0.1756	0.6179
Iteration 4	0.7435	0.7653	0.6346	0.7551	0.7435	0.1666	0.6423

Table 7.7: Evaluating the **Accuracy** of **Phase 4**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.7495	0.7637	0.6743	0.7760	0.7593	0.1792	0.6532
Iteration 2	0.7748	0.7882	0.7096	0.7907	0.8014	0.1861	0.6766
Iteration 3	0.7401	0.7727	0.6809	0.7650	0.7702	0.1862	0.6473
Iteration 4	0.7618	0.7936	0.7089	0.7844	0.7618	0.1772	0.6659

Table 7.8: Evaluating the **Precision** of **Phase 4**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.7333	0.7484	0.6666	0.7717	0.7435	0.2365	0.6230
Iteration 2	0.7410	0.7551	0.6993	0.7602	0.7692	0.2538	0.6371
Iteration 3	0.7141	0.7602	0.6638	0.7410	0.7576	0.2371	0.6166
Iteration 4	0.7435	0.7769	0.7026	0.7602	0.7435	0.2352	0.6358

Table 7.9: Evaluating the **Recall** of **Phase 4**.

Evaluator	GB	NN	NLDA	KNN	DT	GNB	SVM
Iteration 1	0.7277	0.7431	0.6565	0.7623	0.7380	0.1537	0.6256
Iteration 2	0.7427	0.7568	0.6859	0.7602	0.7726	0.1548	0.6388
Iteration 3	0.7175	0.7564	0.6611	0.7461	0.7538	0.1582	0.6200
Iteration 4	0.7397	0.7730	0.6951	0.7651	0.7519	0.1521	0.6376

Table 7.10: Evaluating the **F-measure** of **Phase 4**.

Comparing the results of **Phase 1** shown in table 7.1 to the results of **Phase 4** shown in the tables 7.7 to 7.10, we realize that there is a light enhancement in the evaluation metrics values. Comparing the results of **Phase 2** shown in table 7.2 to the results of **Phase 3** shown in tables 7.3 to 7.6, we realize that there is a remarkable enhancement generally in all evaluation metrics (with few exception). No enhancements occurred after the 4^{th} iteration. Finally, we can conclude that our combined iterative classification algorithm was able to simultaneously improve performance on both email activities discovery and emails process instances discovery. The results provide an empirical evidence in favour of the proposed approach.

7.4.2 Discussion

In this chapter, we treated two problems using iterative relational classification approach: (1) The problem of discovering process instances, (2) The problem of discovering email activities. We claimed that information about emails process instances can help in better identifying email activities and vice versa. Iteratively, we propagate the high accuracy results of each phase to improve the results of the other phase.

In fact, our work is inspired by the work of Khoussainov et al. [35]. They describe machine learning approaches to identify task and relations between individual messages in a task i.e. finding cause response relations between messages and for semantic message analysis i.e. extracting metadata about how messages within a task relate to task progress. They exploit the relational structure of these two problems. The idea behind their approach is that related messages in a task provide a valuable context that can be used for semantic message analysis. Similarly, the activity related metadata in separate messages can provide relational clues that can be used to establish links between emails and group them into tasks. In this paper, we map their work on the business process management field. We first extract process activities and instances independently from a business-oriented point of view. A limitation in their approach is that they deal with 5 speech acts which are the 5 most frequent verbs: “Propose”, “Request”, “Deliver”, “Commit”, and “Amend”. In our work, we overcome this limitation by allowing a dynamic definition of activities. We start by a set of activities that we have already extracted from the training email log. This set is updated as soon as new activities are introduced in a new email log. When new

activities arrive, the classification model is trained accordingly. This is considered more practical as it is very possible to receive new business activities in our email exchanges. We prove our assumption by experimenting the algorithms on a folder from Enron email dataset and compare the results with the baseline approaches.

CHAPTER 8

Deducing Business Process Models from an Email Log

Contents

Figures

In the previous chapters, we have worked on the extraction of business process oriented information from email logs. The goal behind that is to transform email logs into event logs that can be used as an input for process mining techniques that deduce the corresponding business process models. In this chapter we aim to complete the transformation of the email log into an event log by adding the timestamp attribute which indicates the occurrence time of an activity.

In order to clarify the steps and approaches applied in all previous chapters and sections, we describe a usecase that starts from the preprocessing phase reaching the business process model discovery phase. The example is applied on a folder containing emails from Enron email log. This folder contains emails of different process topics in which we separate them as described earlier in this thesis. We continue the example with one of the process topics reaching the business process model discovery phase.

8.1 Temporal Feature Extraction

Reaching this step, we have discovered, for each email, the set of business process activities it contains and to which process model they belong to. Each process activity is also associated to a business process instance. Hence, if we consider each email business process activity as an event, each event is associated to an activity label, business process model (ProcessID), and business process instance (ProcessInstanceID). To recall, the main attributes of an event log are Process identifier, process instance identifier, activity label and timestamp. Therefore, what is missing in the event log extracted from the email log is the timestamp attribute.

Each email is exchanged in a specified timestamp. The timestamp of an email is considered one of the structured information about an email. Each exchanged email contains a set of business process activities. As a first hypothesis, one can consider that the timestamp of an activity is the same as that of the email it belongs to. However, this is in most cases not true. An email may contain business activities that were already applied, in progress activities, or activities that will be applied in the future. Therefore, an email timestamp can not be considered as an indication about the time of the activity occurrence. For this reason, we need a method that using the email timestamp and the unstructured temporal expressions in an email can deduce an approximation about the time of the activity occurrence. Hence, we build a simple method which is able to approximate the real timestamp value of a business activity by parsing the email text to extract temporal information for each activity.

In this section, we focus on the extraction of temporal relations between the email extracted business process activities, the email temporal expressions and email timestamp. In other words, we address intra-relation identification between business process activities and/or temporal expressions and the relation identification between the activities and the email sending time.

Therefore, we work on the discovery of the temporal relation:

1. Between the email activities and the email timestamp: the objective is to temporally locate an email activity according to the email timestamp in which it occurs. To be accurate, we can divide the relation between the activity and the email timestamp into different categories. Possible categories are *Before*: in which the activity occurs before the email sending time, *Overlap* in which the activity occurs at the time the email is sent and *After* in which the activity will occur after the email sending time.

2. Between the email activities themselves: the objective here is to extract the intra temporal relations between the email activities using the email temporal expressions.

Once these temporal relations are extracted, we will be able to deduce for each email activity, using the email timestamp and the discovered temporal relation, an approximation for the real email activity occurrence time which can be added to the event log. In the approach here, we work only on the intra temporal relations between business activities. We exclude the inter temporal relations from our study i.e. the relation between the business activities of different emails. In fact, our main goal from this approach is to try to order the activities of a process instance. Therefore, we get the real timestamp order between the activities of one email and we keep the original timestamp order between activities of different emails.

8.1.1 Main Steps of the Intra-Temporal Relations Discovery between Email Activities

As a first step, we start by temporally locating an email activity according to the email timestamp. Each activity is annotated by a verb-noun pair. What is interesting for us in this step is the verb of the activity. In order to find the temporal relation between an email activity and the email timestamp, we check the tense of the activity verb. It is assumed that people writing an email in a proper English language (especially the professional ones), will use the correct tenses. In other words, activities that were applied in the past are mentioned in the past tense, ongoing activities are mentioned in the present tense and activities that will be applied in the future are mentioned in the future tense.

In the preprocessing phase we apply the stemming to return all verbs and nouns to their original form. Therefore, to fetch the tenses of the activities verbs, we go back to the un-preprocessed emails to check the tense in which they are originally mentioned. Part of speech discovery methods of the Stanford Parser, are used to detect the tense of the activity verb. Once the tense is detected, we are able to categorize it into one of the following categories: (1) *Before*, (2) *Overlap*. and (3) *After*. If the verb tense is *past*, then the activity was applied *Before* the sending time of the email, if the verb tense is *present*, then the activity time *Overlaps* with the email sending time, and if the verb tense is *future*, then the activity is going to be applied *After* sending the email.

Thus, for each email we have now three main clusters or categories of email activities. The business activities that occurred in the *past* and those of the *present* and others in the *future*. Suppose that the original email timestamp that contains these business activities is t. The business activities in the category *Before* are associated to a timestamp $t - \beta$, the activities in the category *Overlap* are left with the timestamp t, and the activities in the category *After* are associated to a timestamp $t + \beta$, where β is a small time duration that can differentiate between activities of different tenses, respecting the timestamps of emails of the same process instance. In other words, suppose we have two emails of the same process instance. Email e_1 with timestamp t_1 is sent before e_2 of timestamp t_2. If e_2 contains an activity that occurs in the past, we associate it with a timestamp $t_2 - \beta$, where $t_2 - \beta$ is still greater than t_1. Reaching this step, we have an initial classification of the email activities according to their occurrence order.

However, the email activities of the same category, also occur in a specific order. We may have two or more email activities that are mentioned in the past and that are provided a timestamp $t - \beta$. Using this timestamp only, we will face a problem ordering the activities of the same category. Therefore, we need a method that is able to deduce intra-temporal relations between activities of the same tense. To do that, we need to analyze the unstructured text of an email to find the temporal expressions in the sentences of the corresponding email to deduce the order of the occurrence of the activities of the same tense.

We start by defining the set of temporal expressions that may occur in an English text. These expressions are divided into two categories: the phrases that

are concerned by the past, name it P, and the phrases that are concerned by the future, name it F. For example, the expressions about the past are *before, earlier, formerly, in the past, not long ago, long ago, once, preceding, previously, prior, yesterday etc...*, and the expressions about the future are *after, after a few days, after a while, consequently, following, later, second, third, soon, then, tomorrow etc...*

As mentioned before, our main goal is to temporally order the email activities in one process instance. For each email, we already categorized the email activities as *Before*, *Overlap*, and *After*. Obviously, the initial order is that the activities of *Before* category occur before those of *Overlap* and *After* categories. We check now the email activities of each category separately. In fact, we check the email activities of the same category pairwisely. Suppose we want to identify the order of two activities a_1 and a_2 that belong to the same tense category. We start by parsing and analyzing the email sentences containing these activities. We try to cope with most of the cases. There are two main cases:

1. **Case 1** The two activities (a_1,a_2) are contained in the same email sentence: if two activities are mentioned in the same sentence, we first start by checking if the sentence is simple, compound or composite sentence.

   (a) If the sentence is simple: We fetch the sentence to find temporal expressions. In case the temporal expression belongs to P and is between a_1 and a_2, then a_1 likely has occurred before a_2. For example, suppose we have the sentence: *Jeff has signed the document before sending the bill.* We conclude that the activity "sign document" occurs before "send bill". On contrary, if the temporal expression belongs to F, we conclude that a_1 occurs after a_2.

   (b) If the sentence is compound or composite, we divide it into two separate sentences. For example, suppose we have the sentence: *Jeff sent the bill, but before he has signed document.*. When the sentence is divided into two different sentences, then the two activities a_1 and a_2 are separated into two different sentences, thus, we apply the same procedure as in Case 2.

2. **Case 2** The two activities are contained in two different sentences. We fetch the second sentence for the temporal expression it contain (that is usually at the beginning of the sentence). For example, suppose we have the following two sentences: *Jeff sent the bill.* and *But first he has signed the document.* If the temporal expression belongs to P, then the activity mentioned in the second sentence occurs before the activity mentioned in the first sentence. On contrary, if the temporal expression belongs to F, then the activity mentioned in the first sentence occurs before the one mentioned in the second sentence.

We apply the same procedure on all pairs of activities in each of the categories *Before*, *Overlap* and *After*. We deduce an approximate order for their activities belonging to one category. For example, suppose we apply the above procedure

on the email activities of the category *Before* (which are given an initial timestamp $t - \beta$). We temporally order the activities of this category by adding a short time period to their timestamps. Suppose we have a_1 that occurs before a_2 which in its turn occurs before a_3 in the *Before* category. Their timestamps will be updated to $t-\beta+\epsilon$, $t-\beta+2\epsilon$, and $t-\beta+3\epsilon$ respectively. It is important to ensure that the timestamp given to the latest activity in the category *Before*, does not exceed the timestamp of the earliest activity in the category *Overlap*. Similarly, we apply the same process on the other categories and update their initial timestamps. In this way, we get for each email an approximation of the real timestamp order of its activities. This is considered a preliminary work just to allow more business analysis on emails.

8.2 Deducing the Business Process Models

As explained earlier in the introduction of this thesis, our main goal is to transform email logs into event logs that can be used as an input for process mining tools to produce business process models contained in the emails. Therefore, we work in the previous chapters and sections on extracting undocumented business process information from the unstructured texts of emails. We extract process identifiers, process instances identifiers, activity labels and finally we approximate the timestamp of each activity occurrence. Thus, we consider that an email log is transformed into an event log characterized by the mentioned attributes.

In the following usecase, we will describe the steps of transforming an email log into an event log on a concrete example which will clarify the overall job and target of the general framework described in chapter 2 of this thesis.

8.2.1 Usecase

The usecase is applied on an Enron folder that contains emails for an Enron employee. This folder contains an email log in which its emails revolve around several topics. To clarify the idea, we show examples of the emails of the chosen folder. These emails will be used to explain the usecase and all the applied steps after. For simplicity, we will only show the important information of an email.

In the figures 8.1 and 8.2 below, we show a few number of example emails from the Enron folder we applied the usecase on. If we manually skim and scan the emails in the figures, we find out that they are mainly concerned by three different process topics: Recruiting, E-Trading or Internet Trading, and Meeting Scheduling.

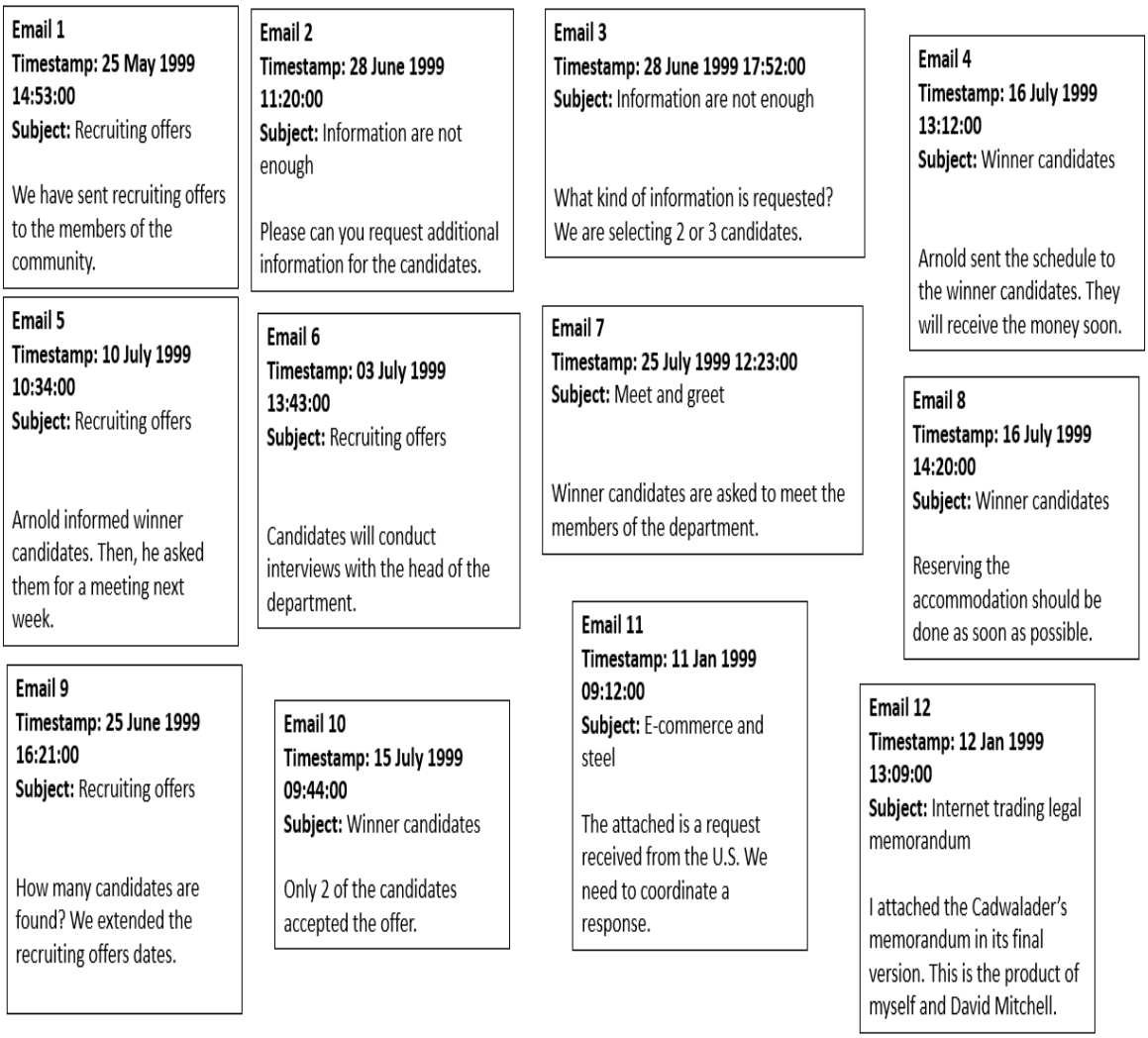

Figure 8.1: Example emails from an Enron email folder.

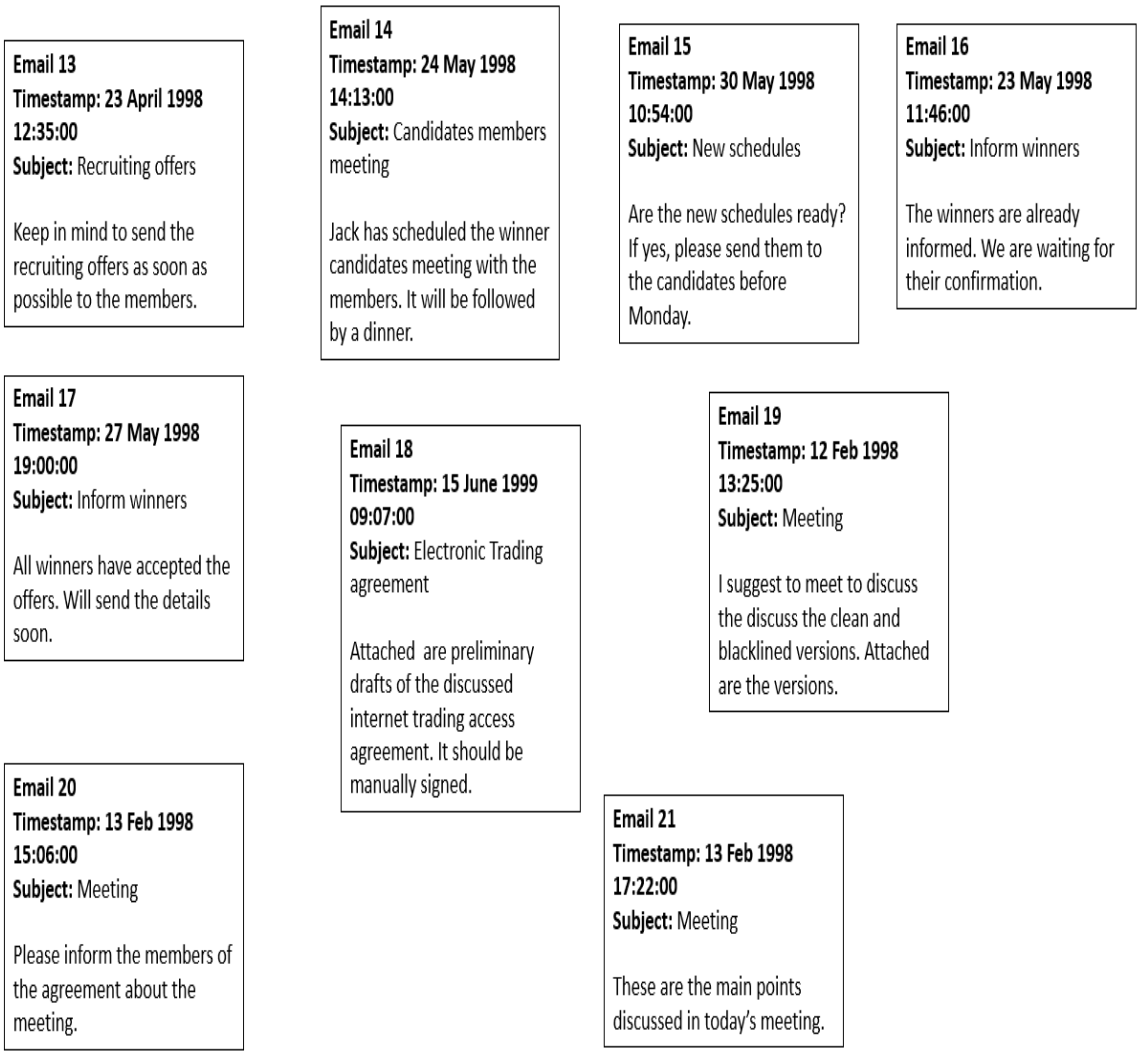

Figure 8.2: Example emails from an Enron email folder.

As a first step, we start by the approach applied in chapter 4. Emails are preprocessed by applying the cleansing, stemming, stop words removal etc... An example of the preprocessed emails is shown in figure... The preprocessed emails are then fed to the process topic discovery approach, where emails are clustered according to what process topics they belong to according to the information in their texts. Figure 8.3 shows the main clusters of the example emails. Thus, we obtain 3 separate clusters representing the process topics of the emails.

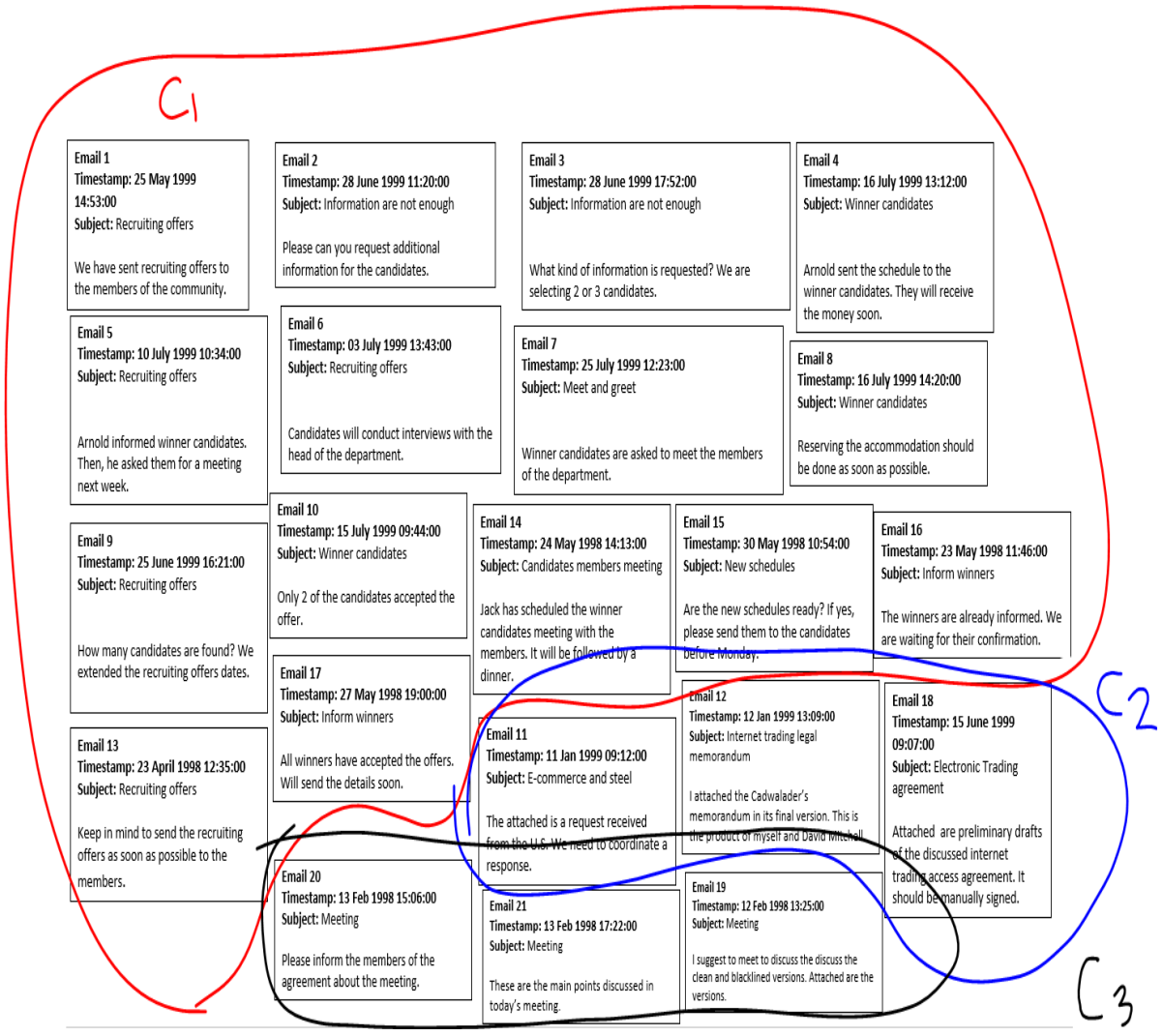

Figure 8.3: Three main clusters.

After applying the process topic discovery phase, each email is associated to a processID. Let us suppose that the processID for Recruiting topic is 1, for E-Trading topic is 2 and for Meeting Scheduling topic is 3. We choose to continue the usecase with the cluster C_1 containing emails about Recruiting. Hence, all emails contained in this clusters are associated to a processID = 1. We then move to the next two phases which are mainly concerned by process activity discovery and process instances discovery. We first apply the baseline approaches of these two phases which are then used in the relational approaches as explained in chapter 7. It does not matter the order of the baseline approaches to start with. Therefore, we work on discovering baseline instances that is explained in chapter 5. We then discover business process activities in emails using the baseline approach of activity discovery explained in chapter 6. Once the baseline approaches results are obtained, we launch the relational approach for process instances and activities discovery as explained in chapter 7. The activities obtained from the recruiting emails are the following:

- Email 1: send recruiting offers
- Email 2: request additional information
- Email 3: select candidates
- Email 4: send schedule
- Email 5: inform winners, schedule meeting
- Email 6: conduct interviews
- Email 7: meet members
- Email 8: reserve accommodation
- Email 9: find candidates, extend date
- Email 10: accept offer
- Email 13: send recruiting offers
- Email 14: schedule meeting
- Email 15: send schedules
- Email 16: inform winners, wait confirmation
- Email 17: accept offer

using these activities and other information, emails are grouped into instances. The results are shown in figure 8.4. Reaching this step, each email activity is associated with a processID, processInstanceID, and an activity label. We then approximate the activities timestamps by applying the method explained in section 8.1. The overall result of all the email activities transformed into an event log is shown in the event log represented in table 8.1.

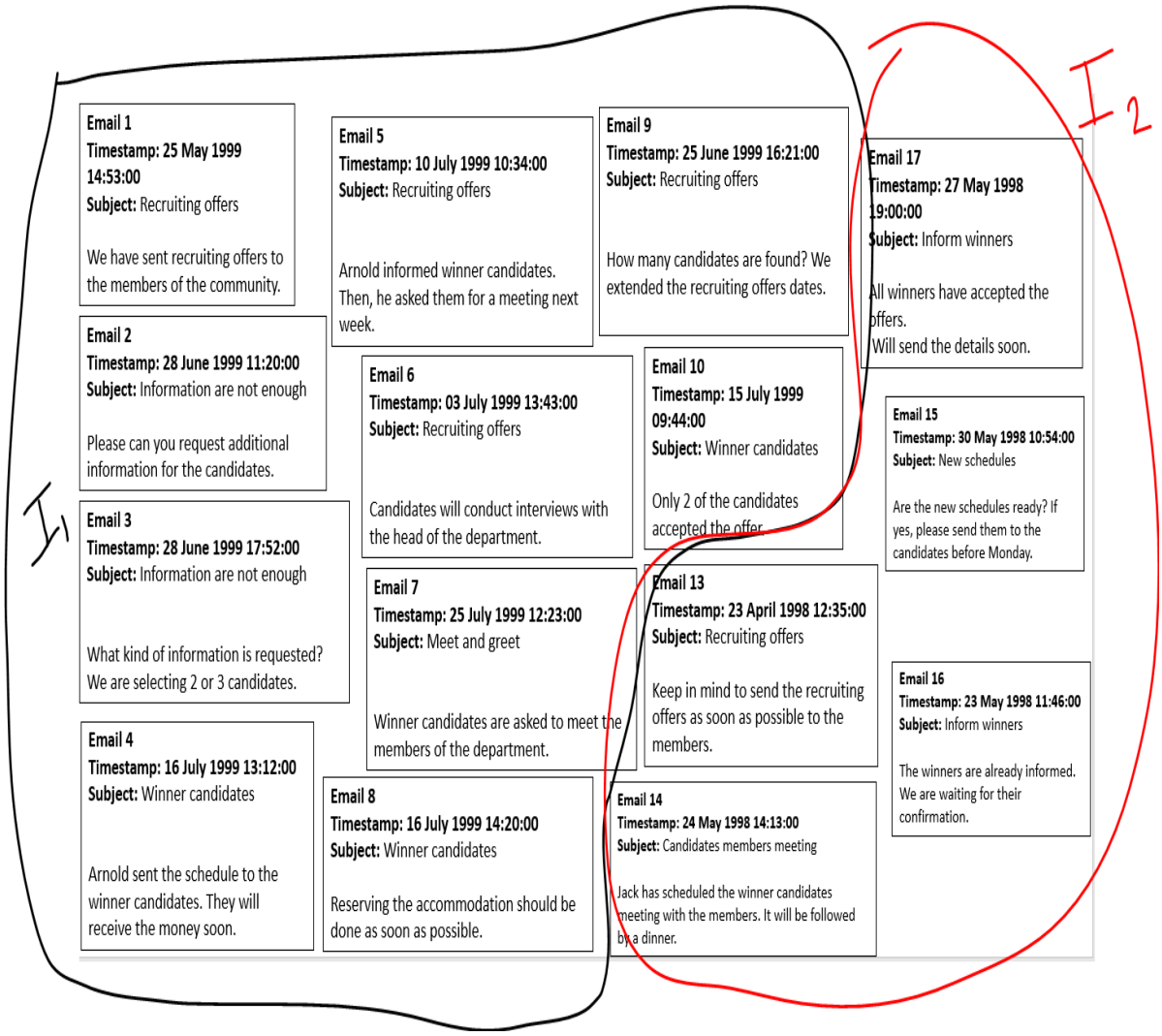

Figure 8.4: Discovered instances of the Recruiting emails.

Activity Label	EmailNum	ProcessID	ProcessInstanceID	Timestamp
send recruiting offers	1	1	1	t_1
request additional information	2	1	1	t_2
select candidates	3	1	1	t_3
send schedule	4	1	1	t_4
inform winners	5	1	1	t_5
schedule meeting	5	1	1	$t_5 + \beta$
conduct interviews	6	1	1	t_6
meet members	7	1	1	t_7
reserve accommodation	8	1	1	t_8
find candidates	9	1	1	t_9
extend date	9	1	1	$t_9 - \beta$
accept offer	10	1	1	t_{10}
send recruiting offers	13	1	2	t_{11}
schedule meeting	14	1	2	t_{12}
send schedules	15	1	2	t_{13}
inform winners	16	1	2	$t_{14} - \beta$
wait confirmation	16	1	2	t_{14}
accept offer	17	1	2	t_{15}

Table 8.1: Extracted event log.

Since the event log is ready, we are able to input it to a process mining tool that discovers for us the corresponding process model. For our example, the process model of the Recruiting topic is shown in figures 8.5 and 8.6. In that way, we have reached our goal of transforming an email log into an event log to discover the undocumented business process models.

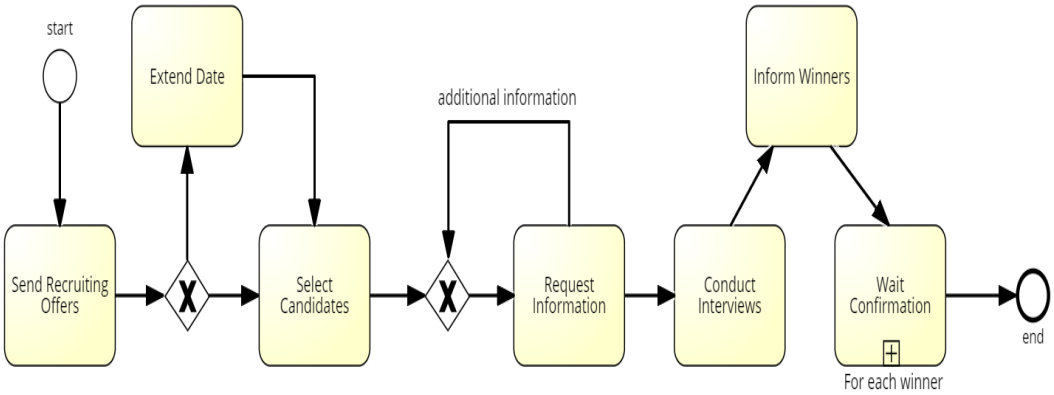

Figure 8.5: Business process model for Recruiting in Enron.

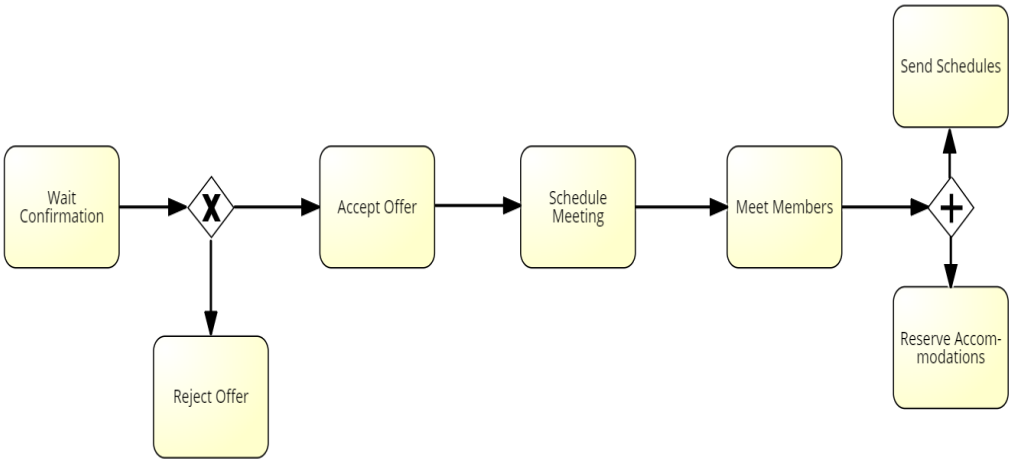

Figure 8.6: Composite process "Wait Confirmation" for each candidate.

CHAPTER 9

Conclusions and Future Work

9.1 Conclusions

While its initial use focused on exchanging personal messages between individuals, emails have evolved from a mere communication system to a mean of organizing complex activities (workflows), storing information, sharing and editing documents, and coordinating the execution of complex tasks involving multiple individuals. Because of its wide use in personal, but most importantly, professional contexts, email represents a valuable source of information that can be harvested for understanding, reengineering and repurposing undocumented business processes of companies and institutions.

Taking this into consideration, we work in this thesis on the extraction of business process information from email logs. We transform email logs into event logs where the latter can be used as an input for the traditional process mining tools to discover business process models. The produced business process models provide a clear overview on the processes and the activities in a user email log. Organizing emails as business processes allows the users to deal with an email as a part of a process which helps in managing their assigned tasks and tracking the overall progress of the process. In addition, organizing email logs as processes facilitates and enriches the analysis that can be applied on emails.

Throughout the chapters of this thesis, we describe the main approaches that constitute the framework presented in this work. We assign to each component of this framework an approach that is in charge of a specific part of the overall target. Each of the described approaches outputs some results that can be susceptible for querying and analysis. Together, the results collected from all approaches allow achieving the final goal i.e. business process models discovery from email logs. The components of the framework are mainly concerned by: (1) business process topic discovery for emails, (2) business process instances discovery for emails, (3) business activities and metadata extraction from emails, (4) relational learning between process instances discovery and business activities discovery, (3) preliminary estimation of the activity occurrence timestamp. This has enabled the extraction of the main event log attributes from an email log. The deduced event log in its turn allows the discovery of the business process models.

Based on our literature study applied on the existing approaches and their drawbacks, we have built a framework that is composed of different approaches and methods. We can summarize the contributions of these approaches as follows:

- We introduce an approach that discovers the business process topic that an email is concerned by. This approach uses the unsupervised learning for this purpose. Emails are clustered according to their process topics using semantic similarity measurement between emails. The advantage of this approach is that it does not have any a priori knowledge about the process topics present in an email log and that it does not require the interference of the user for this purpose. This approach allows the identification of each email by a process ID.

- We propose an approach that discovers the business process instance that an email belongs to. After the specification of the process topic that an email is concerned by, this approach uses also unsupervised learning techniques to specify the business process instance that the email belongs to. In this approach, a combination of structured and unstructured data are extracted from emails and used for specifying clusters of emails as process instances. This approach allows the identification of each email by a process instance ID.
- We also work on an approach that extracts business process activities from email logs. This approach uses supervised classification techniques and tools to specify the business activities that an email contains. Without having any a priori knowledge about the quantity or types of activities contained in an email log, this approach is able to discover and differentiate between personal and business activities. In addition, a set of metadata is extracted for each of the discovered business activities allowing further analysis on emails and their activities. This approach allows the identification of the activity labels or types available in an email log.
- We introduce an approach that enhances the performance of the process instances discovery phase and business activities discovery phase by dealing with both problems in an iterative relational manner. We study how the outcome of instances discovery phase can be used in the business activities extraction phase to improve its results and vice versa.
- A preliminary temporal information extraction phase is proposed. Knowing that the timestamp of an email do not always give a good indication about the real email activity enactment, we work in this approach on the estimation of the real time of the activities' enactment using the email temporal expressions.
- The performance of the proposed approach are evaluated and studied by applying experiments on these approaches using email folders from the Enron email dataset.

9.2 Future Work

In fact, our achieved work opens up doors for multiple further researches. There exist several potential perspectives based on the obtained results. One of the perspectives is to build a recommendation system, that learns from previously obtained business process models from email logs. This recommendation system can recommend to the user, upon receiving an email, the potential activities to be applied next based on the past received emails. It may also recommend to the user the types of documents that can be attached to the email to be sent or some candidate weblinks that may be helpful for the receiver of the email etc... Such recommendation system helps in optimizing the business process enactment in terms of performance and the time duration it takes to be completed.

Another important potential work is to allow the incremental learning in our framework. Email users receive emails on a daily basis. These emails may contain new business process topics and activities. Therefore, our applied approaches would not be helpful if we only depend on the business processes learnt from past email exchanges. For this reason, incremental learning is considered important in this case. As new business process topics and activities are received and detected, our learnt models should be incrementally updated. This allows our models to be dynamic, up-to-date and coping with all the changes that may occur in the future.

For the user to be able to use this framework, a user interface is needed. For now, our framework can be enacted by individuals who are aware of how the program works. However, in order to present our framework as a product, it is crucial to build a user interface that allows users and analysts from different domains to make use of all available characteristics and tools. This thesis can be considered as step forward in building a product that helps users managing their emails in a brand-new way that is not actually available in today's email management tools.

Résumé en Français

Le courrier électronique est considéré comme l'une des utilisations les plus populaires de l'internet. C'est un moyen qui permet l'échange d'informations entre des entités possédant des comptes de courrier électronique. Il est indéniable que le système de courrier électronique occupe une place importante dans la communication métier moderne d'aujourd'hui. Les informations échangées dans les textes des courriers électroniques sont généralement concernées par des événements complexes tels que la programmation de réunions, l'organisation d'une conférence, les candidatures d'étudiants, etc. Ces événements complexes peuvent également être considérés comme des processus commerciaux dans lesquels les entités qui échangent des courriers électroniques collaborent pour atteindre les objectifs finaux des processus. Ces événements complexes peuvent également être considérés comme des processus métiers dans lesquels les entités qui échangent des courriers électroniques collaborent pour atteindre les objectifs finaux du processus. Par conséquent, le flux d'informations dans les courriers électroniques envoyés et reçus constitue une partie essentielle de ces processus, c'est-à-dire les tâches ou les activités métiers. Ces informations peuvent être récoltées pour comprendre les processus métiers non documentés des entreprises et des institutions. L'extraction d'informations sur les processus commerciaux à partir de courriers électroniques peut contribuer à améliorer la gestion des courriers électroniques pour les utilisateurs. Elle peut également être utilisée pour trouver des réponses riches à plusieurs questions analytiques sur les employés et les organisations qui mettent en œuvre ces processus.

Le développement récent de la communication dans le cyberespace crée d'énormes quantités de données qui peuvent être utilisées à plusieurs fins. Le courrier électronique est, dans l'ensemble, le premier et le plus populaire des moyens de communication professionnels et sociaux [1]. C'est une forme de communication fiable, confidentielle, rapide, gratuite et facilement accessible. Avec la popularité croissante du courrier électronique, il est très normal de nos jours que les gens discutent de questions, d'événements ou de tâches spécifiques en utilisant les outils de gestion du courrier électronique disponibles [21]. Il permet aux utilisateurs de s'engager dans plusieurs tâches ou comportements simultanément comme la gestion de projets, l'organisation de conférences, la programmation de réunions, etc... Alors que son utilisation initiale était axée sur l'échange de messages personnels entre individus, le courrier électronique a évolué d'un simple système de communication à un moyen d'organiser et de coordonner l'exécution d'activités complexes (workflows) impliquant plusieurs individus, de stocker des informations ou de partager et d'éditer des documents.

En raison de l'énorme quantité de courriers électroniques envoyés et reçus par un utilisateur au quotidien, l'une des principales exigences de l'utilisateur est une gestion efficace de ses courriers électroniques. Ces dernières années, les systèmes de courrier électronique ont travaillé à l'amélioration de l'expérience des utilisateurs en développant des outils pour optimiser la gestion des courriers électroniques échangés. Par exemple, certains outils de courrier électronique ont aidé l'utilisateur à gérer les tâches extraites des courriers électroniques en

[1] http://onlinegroups.net/blog/2014/03/06/use-email-for-collaboration/

organisant une liste de tâches qui indique l'avancement des tâches et les délais. D'autres outils ont aidé l'utilisateur à organiser les ressources associées dans les courriels échangés : [8], [2], [64], [6].

Le courrier électronique s'est transformé au fil des ans, passant d'un simple moyen de communication pour l'échange de messages à un "habitat" [18], un environnement où les utilisateurs échangent des courriers électroniques pour appliquer des business processes. L'échange de courriers électroniques devient essentiel lorsque l'application de tâches dans les processus organisationnels nécessite la participation de plusieurs personnes. Attribuer des tâches, demander plus d'informations, rendre compte des résultats - toutes ces activités sont effectuées par le biais de messages électroniques. Par conséquent, ces messages électroniques contiennent nécessairement des informations liées au processus qui se réfèrent au processus en cours d'exécution.

Un *Business Process* est composé d'un ensemble d'activités qui sont appliquées dans un ordre spécifique pour atteindre un objectif organisationnel. Chaque processus d'entreprise est représenté par un modèle, c'est-à-dire *Business Process Model*. Le modèle représente une série de tâches ou d'activités connexes à appliquer d'une manière spécifique qui aboutissent au résultat souhaité. Un *Business Activity* est conçu pour exécuter une tâche ou une action spécifique qui contribue à un business process. Prenons par exemple le modèle de processus relatif à la "programmation des réunions". Les activités de ce modèle peuvent inclure "proposer une réunion", "refuser une réunion", "reporter une réunion", "confirmer une réunion", etc. Chaque processus organisationnel peut avoir plusieurs occurrences qui sont appelées les *Business Process Instances*. Une instance de processus est une occurrence ou une exécution spécifique d'un modèle de business process.

Le contexte de cette thèse s'articule donc autour de l'analyse du contenu des journaux d'e-mails d'un point de vue orienté vers les métiers du processus. En d'autres termes, au lieu de le traiter individuellement ou comme une partie d'un fil de discussion, un courriel peut être considéré comme une entité contributive dans un processus métier organisationnel. Dans notre contexte, nous ne sommes pas seulement intéressés par l'extraction d'informations précieuses à partir de courriers électroniques, mais nous prenons également en considération le fait que ces informations sont précieuses d'un point de vue orienté vers les processus métiers. En d'autres termes, les informations extraites doivent donner une indication sur l'évolution d'un processus métier organisationnel qui concerne l'expéditeur/le destinataire du courrier électronique.

Cependant, l'analyse du courrier électronique du point de vue de la gestion des processus métiers (BPM) n'a pas fait l'objet d'une étude approfondie dans la littérature. Certains des travaux existants permettent d'identifier les activités de courrier électronique parmi un ensemble prédéfini d'activités : [20], [11], [9]. L'analyseur de courrier électronique développé par Van der Aalst [59] nécessite l'intervention de l'utilisateur pour extraire une instance de processus d'un journal de courrier électronique. Dans [59], ils supposent également que les noms des tâches sont toujours disponibles dans l'objet du courriel dans lequel ils travaillent sur un ensemble prédéfini d'activités. Ainsi, jusqu'à récemment et à

notre connaissance, aucun des travaux précédents ne s'est attaqué au problème de l'extraction d'informations sur les processus métier à partir de courriels *automatiquement* et sans aucune connaissance a priori dans le but de découvrir des modèles de processus métier. En utilisant les clients de messagerie disponibles, un utilisateur n'est pas en mesure de suivre l'exécution d'un modèle de processus métier. La raison en est que les clients de messagerie ne sont pas en mesure de traiter le contenu des courriels dans le cadre de la progression d'un processus métier.

La transformation des journaux d'e-mails en journaux d'événements nous permet de produire des processus métiers en utilisant les outils d'extraction de processus disponibles. Un journal d'événements est l'ensemble de données utilisé par les outils d'exploration des processus pour produire le modèle de processus métier correspondant. Dans un journal d'événements, chaque événement correspond à une activité qui est exécutée dans le processus, où plusieurs événements (ordonnés par leur horodatage) peuvent être liés entre eux comme une instance ou un cas de processus. Ainsi, un journal d'événements peut être considéré comme une collection de cas et un cas peut être considéré comme une trace/séquence d'événements. Les processus métiers produit peut fournir une vue d'ensemble claire des processus et des activités dans un journal de courrier électronique de l'utilisateur. L'organisation des courriers électroniques en tant que processus métiers permet aux utilisateurs de traiter un courrier électronique dans le cadre d'un processus qui les aide à gérer les tâches qui leur sont assignées et à suivre l'évolution générale du processus.

Comme les courriels contiennent des données non structurées telles que des textes, des images ou des documents, le principal défi dans ce cas est de savoir comment extraire ces informations non documentées sur les processus commerciaux et les transformer en journaux d'événements. En d'autres termes, un message électronique (en particulier le courrier électronique produit à partir d'un système non automatisé) ne contient aucune information explicite sur les activités qu'il contient ou sur ses relations avec une instance de processus particulière ou sa pertinence pour l'un des processus métiers organisationnels.

En outre, les cadres et outils existants ne permettent pas d'extraire efficacement des informations orientées métiers des données non structurées des courriers électroniques qui permettent de transformer les journaux de courriers électroniques en journaux d'événements pour la découverte et la gestion des processus métiers organisationnels. Dans cette recherche, nous abordons ce problème en travaillant sur l'extraction d'informations relatives aux processus métiers à partir des journaux de courriels en analysant le contenu des courriels et leurs relations.

Sur la base de la description ci-dessus du contexte, des motivations et des défis de la recherche, nous définissons certaines questions de recherche que ce travail abordera. Le champ de recherche de ce travail concerne l'extraction des processus métiers des journaux de courriels. Ainsi, la principale question de recherche peut être résumée comme suit : *Sachant que les courriels sont classés comme des données non structurées, comment pouvons-nous extraire des modèles de processus métier à partir de journaux de courriels ?* Afin de déduire

les processus métiers à partir d'un journal de courrier électronique, ces derniers doivent être transformés en journaux d'événements. Ainsi, les principaux attributs d'un journal d'événements (identificateur de processus, identificateur d'instance de processus, étiquette d'activité, horodatage) doivent être extraits des courriers électroniques. Cette question de recherche peut être subdivisée en plusieurs sous-questions de recherche:

- Sans avoir une connaissance a priori des sujets de processus ou de leur nombre dans un journal de courrier électronique, comment pouvons-nous déduire quel sujet de modèle de processus métiers chaque courrier électronique est concerné ? La réponse à cette question nous permet de déduire l'attribut d'identification du processus de chaque courriel ou, en d'autres termes, à quel sujet de processus il appartient.

- Sachant que chaque modèle de processus comporte plusieurs exécutions ou instances, comment pouvons-nous déduire à quelle instance de processus métier appartient un courrier électronique ? La réponse à cette question nous permet de déduire l'attribut d'identification de l'instance de processus métier de chaque courrier électronique.

- Sans avoir une connaissance a priori des types d'activité présents dans un journal de courrier électronique, comment pouvons-nous déduire le(s) type(s) d'activité qu'un courrier électronique contient ? La réponse à cette question nous permet de déduire les étiquettes d'activité des événements extraits des courriels.

- Sachant que l'horodatage d'un courriel n'est pas toujours un indicateur précis de l'occurrence des activités d'un courriel, comment pouvons-nous estimer le temps d'occurrence des activités d'un courriel ? La réponse à cette question nous permet de déduire l'attribut d'horodatage pour chaque événement ou activité dans un courriel.

- Quelle est la relation entre la découverte d'activités de courrier électronique et la découverte d'instances de processus de courrier électronique ? Le traitement relationnel entre ces deux problèmes peut-il augmenter l'efficacité de leurs performances ? La réponse à cette question nous permet de découvrir comment les activités dans les courriers électroniques peuvent donner une bonne indication de leur relation (que les courriers électroniques soient liés à la même instance de processus ou non) et vice versa.

Pour atteindre nos objectifs de recherche, nous avons construit un cadre qui sous-tend de multiples approches combinées. Ce cadre est composé de plusieurs éléments, chacun d'entre eux appliquant une approche spécifique qui permet de réaliser une partie de notre travail de recherche. Les contributions de notre travail de thèse peuvent être résumées comme suit:

- Nous travaillons sur une approche qui permet de trouver pour chaque courriel le sujet de processus métier auquel il appartient. Après avoir prétraité le courrier électronique et l'avoir transformé en format de données exploitable, l'analyse est appliquée au corps et au sujet du courrier électronique pour découvrir le sujet de processus métier qui le concerne. Cette approche repose principalement sur l'apprentissage non supervisé, c'est-à-dire le regroupement. Une étude est fournie qui explique notre choix des mesures de similarité et des techniques de mise en grappes. Nous prenons en considération dans cette approche que le système ne doit pas avoir de connaissances a priori sur les sujets contenus dans un journal de courrier électronique. En outre, notre approche est automatique de telle sorte que les utilisateurs ne doivent pas interférer ou aider à découvrir le sujet de traitement d'un courriel.

- Une approche de découverte d'instance de processus est introduite, dans laquelle nous travaillons à la recherche de l'instance de processus métier à laquelle appartient un courriel. Dans cette approche, nous formulons une fonction de distance qui est la plus efficace en termes de performance pour regrouper les courriels en instances de processus métier. La fonction de distance est définie en termes de combinaison de certains attributs. Ces attributs sont en fait extraits du contenu structuré et non structuré d'un courrier électronique. L'efficacité du calcul de la distance dépend de manière cruciale des attributs choisis. Par conséquent, nous présentons de multiples combinaisons d'attributs de courrier électronique et prouvons leur validité à l'aide d'exemples et de contre-exemples. Contrairement à d'autres travaux existants, cette approche est automatique, c'est-à-dire qu'un effort de l'utilisateur n'est pas nécessaire pour choisir les attributs à utiliser dans le calcul de la distance. L'efficacité des résultats obtenus dans cette phase est considérée comme critique, d'autant plus que cette phase est une étape intermédiaire dans le cadre général.

- Nous introduisons une approche comme solution pour extraire les activités des courriels et pour annoter les activités sollicitées. Plus précisément, nous commençons par la première hypothèse, où nous construisons une approche permettant d'extraire une seule activité par courrier électronique. Nous passons ensuite à la deuxième hypothèse, où nous présentons une approche qui, en utilisant une synthèse extractive personnalisée des courriels axée sur les activités métiers et le regroupement de phrases axées sur les activités métiers, découvre et étiquette un ou plusieurs types d'activités métiers dans un courriel. Ensuite, chaque type d'activité est automatiquement associé à un ensemble de métadonnées qui le décrit. Un ensemble prédéfini d'activités n'est pas toujours disponible car il n'y a pas de connaissance a priori sur les sujets de processus disponibles dans le journal des courriels et il n'y a pas de connaissance sur les activités métiers existantes. De plus, selon notre analyse des courriels, nous découvrons qu'un nombre remarquable de courriels contiennent plus d'une activité à la fois. C'est pourquoi, dans notre approche de la découverte des activités métiers

par courrier électronique, nous surmontons ce problème en étant capables d'extraire une ou plusieurs activités d'un seul courrier électronique. En outre, dans cette approche, nous travaillons à la spécification, pour chaque activité de courrier électronique, d'un ensemble d'informations ou de métadonnées qui enrichissent et décrivent l'activité correspondante. Nous relevons le défi lorsque plusieurs activités sont présentes dans un même courriel, où les informations associées à un courriel telles que les pièces jointes, les URL, etc. Cette étape de notre approche ouvre la porte à une analyse de haut niveau dans laquelle il est possible de répondre à de nombreuses questions analytiques en utilisant les métadonnées extraites.

- Nous proposons une approche relationnelle itérative qui utilise les informations sur les activités métiers de messagerie pour identifier les courriels des mêmes instances de processus et vice versa. En particulier, nous étudions (1) comment les informations sur les activités de courrier électronique peuvent aider à trouver les courriers électroniques des mêmes instances de processus, et (2) comment les caractéristiques des courriers électroniques des mêmes instances de processus peuvent aider à la classification des activités de courrier électronique.

  Les objets liés changent. La classification itérative utilise ce dernier moyen en appliquant une approche de classification de manière dynamique pour exploiter pleinement la structure relationnelle. Notre approche est basée sur l'idée que les informations sur les activités de processus contenues dans des courriels séparés peuvent être utilisées pour regrouper les courriels en instances de processus. De même, les messages appartenant à la même instance de processus peuvent fournir un contexte précieux pour la découverte des activités des courriels. Par conséquent, nous affirmons qu'un traitement relationnel itératif de ces deux problèmes améliorerait l'efficacité des résultats obtenus par rapport au traitement séparé.

  Dans une approche non relationnelle, nous utilisons des méthodes de base pour découvrir les activités de traitement des courriels et pour relier les courriels aux instances de traitement. Dans les approches d'analyse de texte de base, nous utiliserions le contenu des messages (contenu structuré et non structuré) de manière isolée pour trouver les courriels appartenant aux mêmes instances de processus et pour étiqueter les activités des courriels.

  Cependant, dans l'approche relationnelle pour la découverte des instances de processus de courrier électronique, nous pouvons utiliser à la fois le contenu du courrier électronique et les informations sur les activités du courrier électronique. Par exemple, supposons qu'un courriel candidat concerne l'achat d'un article, appelez-le article P et un autre courriel candidat est la confirmation de l'achat de l'article P. Les deux courriels contiendront des métadonnées concernant P avec le nom de la société vendeuse ou le nom de l'acheteur ou des pièces jointes concernant les caractéristiques de l'article ou la facture. L'utilisation de ces informa-

tions sur les activités "Achat d'un article" et "Confirmation de l'achat" peut aider à relier les deux courriels dans la même instance de processus. Ainsi, notre modèle de classification relationnelle peut déduire que si un courriel e_1 contient l'activité "Achat d'article" associée aux métadonnées $info_{PurchaseItem}$ et un courriel e_2 contient l'activité "Confirmation d'achat" avec les métadonnées $info_{ConfirmPurchase}$ où $info_{PurchaseItem}$ est similaire à $info_{ConfirmPurch}$, alors e_1 et e_2 sont susceptibles d'appartenir à la même instance de processus.

D'autre part, les informations sur les instances de processus découvertes peuvent être utilisées pour aider à extraire les activités des processus métiers de courrier électronique. Si nous réutilisons l'exemple ci-dessus en sachant que e_1 et e_2 appartiennent à la même instance de processus I où e_2 est une réponse à e_1. En tenant compte du fait que e_1 contient l'activité "Achat d'un article" et que e_2 est le courriel qui suit e_1 dans l'instance de processus I, nous construisons un modèle de classification qui (en utilisant d'anciennes occurrences de courriels similaires comme e_1 et e_2) prédit que e_2 est susceptible de contenir l'activité "Confirmer l'achat".

Nous décomposons donc le problème global en quatre phases différentes :

- **Phase 1** : Extraire les activités de processus avec leurs métadonnées des courriels en utilisant leur contenu (c'est-à-dire sans utiliser les informations sur les instances de processus des courriels).
- **Phase 2** : Identifier les instances de processus à partir de courriels en utilisant un calcul de similarité (c'est-à-dire sans utiliser d'informations sur les activités de processus de courriel).
- **Phase 3** : Utilisation des activités de courrier électronique extraites et de leurs métadonnées pour améliorer l'identification des instances de processus dans les courriers électroniques.
- **Phase 4** : Utiliser les liens entre les courriers électroniques appartenant aux mêmes instances de processus pour améliorer l'extraction des activités de processus des courriers électroniques.

Dans notre approche de classification relationnelle itérative, nous commençons par résoudre **Phase 1** et **Phase 2** en utilisant des méthodes de base traitant chaque problème séparément. Les résultats de la **Phase 1** et de la **Phase 2** sont ensuite utilisés pour résoudre la **Phase 3** et la **Phase 4**. Itérativement, les déductions changeant dynamiquement (avec un degré de confiance élevé) déduites de la **Phase 3** sont renvoyées à la **Phase 4** pour améliorer la précision de ses résultats et vice versa.

- Nous proposons une approche préliminaire qui permet d'estimer le temps d'occurrence d'un événement ou d'une activité de courrier électronique. En d'autres termes, cette approche permet d'extraire les relations temporelles entre les activités du processus de gestion des courriels, les expressions temporelles et l'horodatage des courriels. Ainsi, nous abordons

l'identification intra-relationnelle entre les activités de processus métier et/ou les expressions temporelles mentionnées dans un courriel, et l'identification de la relation entre les activités de courriel et le moment d'envoi du courriel (horodatage).

- Nous fournissons un cas d'utilisation qui clarifie les étapes de transformation d'un journal de courrier électronique en journal d'événements sur un exemple concret qui explique le travail global et l'objectif du cadre de thèse. Le cas d'utilisation est appliqué à un journal de courrier électronique qui contenait des courriers électroniques tournant autour de multiples sujets et activités métiers.

  Therefore, we work on the discovery of the temporal relation:

  1. Between the email activities and the email timestamp: the objective is to temporally locate an email activity according to the email timestamp in which it occurs. To be accurate, we can divide the relation between the activity and the email timestamp into different categories. Possible categories are *Before*: in which the activity occurs before the email sending time, *Overlap* in which the activity occurs at the time the email is sent and *After* in which the activity will occur after the email sending time.
  2. Between the email activities themselves: the objective here is to extract the intra temporal relations between the email activities using the email temporal expressions.

- L'efficacité de toutes les approches susmentionnées est évaluée à l'aide de plusieurs dossiers de courrier électronique provenant de l'ensemble de données de courrier électronique d'Enron [2]. Cet ensemble de données a été collecté et préparé par le projet CALO [3] (Un assistant cognitif qui apprend et s'organise). Il contient des données provenant d'environ 150 utilisateurs, pour la plupart des cadres supérieurs d'Enron, organisées en dossiers.

En utilisant les informations extraites, les journaux de courrier électronique peuvent être transformés en journaux d'événements. Ces journaux d'événements peuvent être utilisés comme données d'entrée pour les techniques d'extraction des processus métiers. Bien qu'utiles, les propositions existantes dans la littérature ne permettent pas de construire un cadre automatique complet qui prend un journal de courrier électronique brut et le transforme en un journal d'événements compatible avec les techniques d'extraction de processus disponibles. Par conséquent, l'objectif est de construire des modèles de processus métiers à partir des échanges de courriers électroniques. Pour cette raison, nous travaillons à la transformation des journaux de courriels en journaux d'événements où nous pouvons trouver les principaux attributs des événements avec leurs valeurs.

[2] https://www.cs.cmu.edu/ enron/
[3] http://www.ai.sri.com/project/CALO

Les principaux attributs qui devraient être présents dans un journal d'événements sont les suivants:

1. Identificateur de processus (ProcessID) : cet attribut indique à quel modèle de processus appartient un événement. Sachant que chaque modèle de processus est concerné par un sujet spécifique, l'identificateur de processus met en corrélation chaque modèle de processus avec son sujet. Pour projeter cette définition sur notre travail, cet attribut indique à quel sujet de processus appartient un courriel.

2. Identificateur d'instance de processus : sachant que chaque modèle de processus comporte plusieurs exécutions dans un journal, chacune de ces exécutions est considérée comme une instance de processus. Par conséquent, chaque ensemble d'événements appartient à l'une de ces exécutions. Ainsi, un événement sera associé à un identificateur d'instance de processus.

3. Type d'activité : chaque événement représente une activité ou une tâche qui est appliquée. Chaque tâche a un nom ou une étiquette spécifique comme *confirmer la réunion*, *refuser la demande* etc...

4. Horodatage : il représente l'heure à laquelle l'événement s'est produit. Cela permet de spécifier la séquence d'apparition des événements pour la modélisation du processus métier.

En effet, l'obtention de ce type d'informations métiers ouvre la porte à de nombreuses analyses métiersen exploitant les e-mails pour répondre à plusieurs questions analytiques telles que:

Q_1 Quelles sont les activités métiers exercées par un employé spécifique ? Par exemple, un directeur aimerait connaître la productivité d'un employé spécifique ou connaître la contribution d'un employé dans un processus spécifique. (pour identifier les tâches qui prennent du temps et dont on ne sait pas si elles lui sont attribuées).

Q_2 Combien de fois un utilisateur a-t-il appliqué une activité ? Par exemple, un employé peut souhaiter savoir combien de fois il a fait une demande de bourse de voyage pendant une période donnée.

Q_3 Quels sont les groupes de personnes qui font un travail similaire ? Par exemple, un directeur aimerait savoir qui sont les personnes qui exercent des types d'activités similaires. Cela peut aider à organiser des groupes de travail. Un employé peut souhaiter bénéficier de l'expérience d'un autre employé qui a déjà appliqué le même type d'activité.

Q_4 Quelle est la durée moyenne d'un processus métier ? Elle peut être calculée en calculant la moyenne du temps pris par toutes les instances d'un même modèle de processus. Cela permet aux responsables et aux employés d'identifier à l'avance la durée prévue d'un processus spécifique en fonction des exécutions précédentes du même processus.

Q_5 Quelles sont les instances de processus qui prennent le plus de temps à réaliser ? Normalement, les instances de processus devraient prendre des périodes de temps similaires. Cependant, lorsqu'une instance de processus prend beaucoup de temps pour être réalisée, cela donne une alerte sur une anomalie. Le fait de savoir qu'il y a un problème aide à en identifier la raison, c'est-à-dire à identifier la raison des délais. Les retards globaux peuvent être causés par des retards partiels dans plusieurs activités du processus ou par un retard dans une seule activité en raison de boucles (une activité est répétée plusieurs fois pour être achevée).

Q_6 Combien de cas sont promulgués dans une période donnée ? Cela peut également être lié à la productivité de l'entreprise. Par exemple, un directeur aimerait savoir combien de fois un type de commerce spécifique est mis en place. Cela peut également être lié aux interactions entre les employés. Par exemple, une personne peut être intéressée de savoir combien de réunions ont été organisées pour un groupe spécifique (ou le nombre d'événements organisés).

Q_7 Quelles instances de processus impliquent des entités spécifiques ? Par exemple, quelle instance de demande de financement de mission a nécessité l'intervention du directeur du département ? Cette question peut aider à identifier les situations exceptionnelles ou les processus inefficaces. Ce type de question permettrait d'atténuer les problèmes similaires qui pourraient survenir à l'avenir en appliquant le même type de processus.

En plus des requêtes déjà mentionnées, la construction de tels modèles de processus permet à l'utilisateur de mieux comprendre et gérer ses courriels. Au lieu de traiter les courriers électroniques séparément ou comme des fils de discussion, l'utilisateur pourra les traiter dans le cadre de processus métiers. Cela lui permet de mieux organiser ses processus qui peuvent s'étendre sur une longue période ou qui peuvent inclure un très grand nombre de courriers électroniques. Les modèles extraits peuvent également être utilisés comme support pour l'automatisation à l'aide du système de gestion des processus d'entreprise (BPM).

Nous présentons un exemple simplifié de journal de bord pour clarifier l'objectif principal de notre travail. Pour des raisons de simplicité, nous utilisons dans cet exemple les courriels d'un étudiant en doctorat. Une boîte aux lettres électronique contient des courriels appartenant à plusieurs processus qui concernent directement ou indirectement l'étudiant. Comme la plupart des journaux de courriels, ce journal contient des messages personnels qui ne sont pas intéressants pour notre analyse. Nous excluons ce type de messages du journal des courriels pour une meilleure efficacité des résultats de l'analyse. Nous ne sommes concernés que par les messages électroniques qui sont orientés vers les processus métiers, c'est-à-dire qui contiennent des activités de processus métiers ou des informations sur ces activités. En couvrant le journal des e-mails de l'étudiant, nous nous rendons compte que les e-mails sont principalement concernés par les demandes de mission telles que les demandes de participation à une conférence

à l'étranger ou de remboursement des frais d'inscription aux cours d'été. Nous avons également trouvé de nombreux courriels concernant la programmation de réunions avec différentes entités (doctorant et responsables du laboratoire de recherche ou superviseurs).

Ce type de journaux de courrier électronique est un exemple d'échange professionnel de messages qui serait utile à notre analyse. En appliquant les requêtes mentionnées plus haut dans cette section, un étudiant peut souhaiter savoir à combien de missions il a postulé pendant l'année scolaire. Il peut également vouloir calculer la durée consommée pour obtenir une acceptation pour une candidature spécifique. Un étudiant pourra également organiser ses courriels comme une séquence d'activités. Il peut tirer des enseignements des précédentes exécutions d'activités d'un processus spécifique afin de mieux exécuter les activités actuelles. Toutes les applications mentionnées ci-dessus contribuent à améliorer l'expérience de l'étudiant dans l'utilisation des courriels pour la gestion de ses tâches quotidiennes, mensuelles et annuelles.

Chaque courriel est décrit par un ensemble d'attributs extraits du journal des courriels. Ces attributs et leurs valeurs sont utilisés par les approches menées dans le cadre de cette thèse dont l'objectif final est de déduire des journaux d'événements et par conséquent de traiter des modèles à partir des journaux de courriels. Le cadre général est divisé en un ensemble d'approches où chaque approche est responsable de l'obtention d'une partie spécifique du résultat final. Pour transformer le journal d'événements en un journal de courrier électronique, l'ID du processus, l'ID de l'instance du processus et les étiquettes d'activité doivent être définis pour chaque courrier électronique. Ainsi, chaque approche dans notre cadre sera menée pour extraire un de ces attributs comme partie du journal d'événements.

Le journal d'entrée des courriels peut contenir plusieurs sujets de processus que les utilisateurs et les analystes peuvent ne pas connaître a priori. La tâche consistant à définir un identifiant de processus pour chaque courriel n'est donc pas évidente. En outre, un courriel ne peut contenir aucune activité de processus métier. Il peut également contenir plusieurs activités métiers. Il n'est donc pas aisé d'obtenir le nombre et les intitulés des activités dans un courriel. D'autre part, sachant que deux courriels peuvent contenir les mêmes activités mais appartenant à des instances de processus différentes, l'analyste doit pouvoir utiliser le contenu structuré et non structuré du courriel pour déduire à quelle instance de processus appartient un courriel.

Par conséquent, pour obtenir les résultats finaux, plusieurs approches sont adoptées, chacune pouvant ou non utiliser comme intrant le résultat d'une approche précédente. Certaines approches de cette thèse sont relationnelles de manière itérative de sorte que les résultats d'une approche sont utilisés dans l'autre approche de manière itérative et vice versa (tant que les résultats de la première approche changent, nous les utilisons comme une partie de l'entrée de la deuxième approche).

Comme notre objectif final est de transformer un journal de courrier électronique en un journal d'événements et, par conséquent, de déduire les modèles de processus métiers du journal de courrier électronique d'entrée, il faut une so-

lution qui puisse extraire les informations de processus métiers d'un journal de courrier électronique. L'élaboration d'une solution capable de déduire ce type d'informations métiers et, par conséquent, d'obtenir des modèles de processus métiers à partir des journaux de courrier électronique pose un certain nombre des défis:

C_1 Tous les courriels d'un registre ne sont pas orientés vers les métiers. Normalement, la plupart des journaux de courrier électronique des étudiants, des professeurs, des employés, etc. contiennent des courriers électroniques traitant de questions personnelles. Ces courriels sont considérés comme étant sans intérêt dans notre analyse. Nous sommes principalement concernés par les courriels qui sont orientés vers les processus métiers. Ces courriels contiennent des activités de processus métiers ou des informations sur des activités appliquées. En fait, il est nécessaire de faire la différence entre les courriels à caractère professionnel et ceux à caractère non professionnel dans un journal de courrier électronique. Dans le cas contraire, les courriers électroniques personnels affecteront de manière gênante l'efficacité des résultats obtenus.

C_2 Ni l'utilisateur ni l'analyste n'auraient une connaissance a priori de tous les sujets de processus abordés dans un journal de courrier électronique. Le nombre et les sujets des processus métiers dans un journal de courrier électronique peuvent varier d'un utilisateur à l'autre. Par conséquent, afin de disposer d'un cadre efficace pouvant fonctionner correctement avec tous les journaux de courrier électronique entrants, les approches menées dans ce cadre doivent être fondées sur le fait qu'il n'existe pas de connaissance apriori des sujets des processus métiers d'un journal de courrier électronique entrant. Ces approches doivent permettre de déduire pour chaque courrier électronique le sujet de processus auquel il appartient.

C_3 Plusieurs courriels dans un journal de courrier électronique peuvent contenir les mêmes activités, chaque courriel appartenant à une instance de processus métier différente. Chaque journal de courrier électronique peut contenir plusieurs exécutions du même modèle de processus. Il est donc essentiel de spécifier pour chaque courriel l'instance de processus à laquelle il appartient en fonction du contenu du courriel et d'autres attributs. Ainsi, chaque instance de processus dans un journal de courrier électronique comprend un ensemble de courriers électroniques où chaque courrier électronique contient un ensemble d'activités métiers. Les approches de cette thèse devraient permettre de spécifier à quelle instance de processus chaque courriel appartient.

C_4 La plupart des travaux précédents ont abordé le problème de la découverte de l'activité des courriels en spécifiant une activité ou une tâche pour un courriel. Cependant, dans ce cas réel, nous supposons qu'un courriel peut englober plusieurs activités. Les exemples de courrier électronique ci-dessus prouvent la justesse de cette hypothèse. Par exemple, dans le

deuxième courriel, les activités mentionnées sont : *lien vers la page web* et *signer la bourse de voyage*. Après avoir résolu le premier défi (en excluant les courriels personnels d'un journal de bord), nous devons considérer que chaque courriel peut contenir plus d'une activité à extraire dans le cadre du processus métier.

[C_5] Dans le prolongement des défis précédents, les activités de courrier électronique sont associées à un ensemble d'informations que nous appelons métadonnées, comme les acteurs qui exécutent une activité ou une pièce jointe ou une URL qui accompagne une activité. Toutefois, comme mentionné dans le défi précédent, chaque courriel peut contenir plusieurs activités métiers. Par conséquent, il est non négligeable de relier les métadonnées contenues dans un courriel à l'ensemble des activités qu'il contient. Nous devrions être en mesure d'associer à chaque activité de courrier électronique l'ensemble d'informations correctes qui lui correspond. Nous avons besoin d'un moyen de corréler les informations extraites des ressources associées aux courriels.

Le cadre général suivi par cette thèse est composé de trois éléments. La première composante : **Email Log Preprocessing** qui est considéré comme une phase préparatoire qui prend comme entrée un journal de courrier électronique brut et le prépare pour une analyse plus approfondie. Une fois les données préparées dans le premier composant, la transformation d'un journal de courrier électronique en un journal d'événements commence. Cette transformation est divisée en deux composantes principales : **Process Topic Discovery** où chaque courriel est associé à un sujet de processus métier et **Process Models Discovery** qui est un composant composite composé de plusieurs autres composants pour produire les modèles de processus métier d'un journal de courrier électronique.

En commençant par le premier volet : **Email Log Preprocessing**, les données d'entrée de ce composant sont un journal des e-mails. Comme décrit précédemment, un journal de courrier électronique est un ensemble de courriers électroniques échangés entre différentes entités (personnes, entreprises etc...) dans un but précis tel que la programmation d'une réunion, l'organisation d'une conférence, l'achat d'un article etc. Chaque courriel est représenté par certains attributs qui le décrivent : objet du courriel, expéditeur, destinataire, corps du courriel et horodatage du courriel. Le contenu principal du courriel est constitué du corps et de l'objet du courriel, qui sont considérés comme des types de données non structurées. Savoir que les données structurées sont bien organisées, suivent un ordre cohérent, sont relativement faciles à rechercher et à interroger, et peuvent être facilement accessibles et comprises par une personne ou un programme informatique, traiter des données non structurées, en revanche, est difficile car elles ont tendance à être de forme libre, non tabulaires, dispersées et difficilement récupérables ; de telles données nécessitent une intervention délibérée pour leur donner un sens. Il faut un temps considérable pour prétraiter les données non structurées avec des champs fixes afin qu'elles puissent être interrogées, quantifiées et analysées à l'aide de techniques d'exploration de données. Les données du courrier électronique doivent être net-

toyées et transformées dans le format attendu par les outils d'analyse. Quatre étapes principales sont appliquées dans ce volet :

1. Data Cleansing
2. Data Representation
3. Features Selection
4. Verb-Nouns Extraction

Une fois que les textes des courriels non structurés sont prétraités et préparés pour l'analyse, la composante analyse : **Process Topic Discovery** est promulguée. Les courriels d'un journal de courrier électronique sont concernés par différents sujets ou ce que nous appelons *sujets de processus d'entreprise métiers*. Chaque groupe de courriels est échangé pour effectuer un processus sur un sujet spécifique. Dans le deuxième volet (phase de découverte des sujets de processus), l'objectif principal est de regrouper les courriels en fonction de leurs sujets de processus métier où chaque courriel est associé à un identificateur de processus (ProcessID). Les résultats de l'approche appliquée dans cette composante aideront également à analyser les courriels. Au lieu de traiter un journal de courrier électronique dans son ensemble, nous pourrons travailler sur des courriers électroniques de différents processus séparément, ce qui réduit la complexité des approches appliquées par la suite.

Les résultats du deuxième composant, qui est un ensemble de grappes où chaque grappe PC_i contient des courriels $\{E_{i1}, E_{i2}, ..., E_{i3}\}$ appartenant au même sujet de processus, sont utilisés comme entrée pour le troisième composant : **Process Model Discovery**. L'approche de ce composant est répétée pour tous les groupes de thèmes de processus. Ce composant est un composant composite. Il est composé de plusieurs autres sous-composants dans lesquels des approches analytiques sont appliquées dans chaque sous-composant pour extraire des informations sur les processus métiers à partir de courriers électroniques. Une fois ces informations extraites pour chaque groupe de sujets de processus, le journal des courriers électroniques est transformé en un journal des événements, ce qui permet de déduire les modèles de processus d'un journal des courriers électroniques.

La composante composite est composée de 4 phases principales : deux d'entre elles sont des approches de base dans lesquelles elles n'utilisent que les résultats de la composante précédente et ne sont appliquées qu'une seule fois. Les deux autres composantes sont des approches relationnelles dans lesquelles elles utilisent les résultats des composantes de base et sont appliquées plusieurs fois de manière itérative tant que les résultats changent.

Étant donné un groupe de courriels qui appartiennent au même sujet de modèle de processus où chaque courriel est associé à un identificateur de processus PiD, un sous-composant de découverte d'instances de processus *baseline* est mis en place où ces courriels sont d'abord sous-groupés en groupes préliminaires représentant chacun une instance de processus. La même entrée est

fournie à la sous-composante de découverte des activités de processus de courrier électronique *baseline* où la découverte des types d'activité, l'étiquetage et l'extraction des métadonnées ont lieu. Les résultats des deux approches de base sont utilisés comme données d'entrée pour les approches *relationnelles*. La découverte des instances de processus de courrier électronique relationnel utilise les résultats de la découverte des activités de processus de courrier électronique relationnel et vice versa. Nous appliquons les approches relationnelles pour vérifier si l'utilisation d'informations provenant d'autres sous-composantes peut contribuer à l'amélioration des résultats des autres, c'est-à-dire à une meilleure découverte des instances et des activités dans un journal de courrier électronique.

Afin d'appliquer la ligne de base et les approches relationnelles, nous construisons des algorithmes qui utilisent différentes techniques d'exploration de données.

Après l'application des approches du cadre, les informations sur les processus d'entreprise concernant le journal des e-mails deviennent disponibles, c'est-à-dire les identificateurs de processus, les identificateurs d'instance de processus et les étiquettes d'activités (en y ajoutant d'autres informations comme l'horodatage d'une activité). On peut considérer que le journal des courriels est transformé en un journal des événements qui peut être utilisé pour alimenter les outils d'extraction de processus qui produisent des modèles de processus.

Le cadre général comprend des approches multiples qui couvrent principalement 2 domaines scientifiques. Nous sommes plus particulièrement concernés par (1) la gestion du courrier électronique et (2) la gestion des processus métiers (BPM). La littérature fournit un grand nombre d'ouvrages relatifs à chacun de ces deux domaines. Cependant, jusqu'à récemment, très peu d'ouvrages combinent les deux concepts dans un même cadre. L'analyse du courrier électronique du point de vue de la BPM n'a pas fait l'objet d'une étude approfondie dans la littérature. Dans cette recherche, nous combinons ces deux concepts, c'est-à-dire l'extraction d'informations sur les processus métiers (modèles de processus métiers) en analysant le contenu des courriers électroniques. Dans ce chapitre, nous présenterons plusieurs travaux connexes regroupés en différentes catégories. Tout d'abord, une étude sur *Email Management* : les logiciels métiers, le pliage des courriels, le résumé des courriels et la gestion des tâches des courriels. Ensuite, nous présenterons les travaux connexes concernés par *Process Model Discovery vs Email Analysis* qui tournent principalement autour de : la découverte d'instances de processus à partir de courriers électroniques et la découverte d'activités de processus de courrier électronique.

La gestion du courrier électronique est le processus de collecte, de stockage et d'exploitation des données de courrier électronique. Les outils de gestion du courrier électronique sont utilisés pour gérer des volumes importants de messages électroniques entrants et sortants. Les courriers électroniques sont rarement une source d'information autonome, ils contiennent généralement des pointeurs vers des informations complémentaires telles que des fichiers joints, des liens vers des pages web ou des références à d'autres ressources. En raison de la quantité remarquable d'informations précieuses que peuvent contenir les jour-

naux de courrier électronique, l'analyse des courriers électroniques et l'extraction d'informations sont considérées comme essentielles pour "comprendre" partiellement ou totalement le contenu des courriers électroniques. L'objectif de la gestion du courrier électronique est de pouvoir faire face à l'énorme quantité de courriers électroniques reçus et envoyés afin d'assurer un stockage et une manipulation efficaces. Les questions analytiques mentionnées ci-dessus peuvent être répondues comme suit :

Q_1 **Quelles sont les activités métiers exercées par un employé spécifique ? (pour identifier les tâches qui prennent du temps et dont on ne sait pas si elles lui sont attribuées).**
Chaque type d'activité est corrélé à un ensemble d'acteurs, ce qui nous permet de spécifier toutes les activités métiers qu'un employé exerce. Par exemple, l'employée Jennifer Stewart est responsable de l'exécution de l'activité "Modifier les données".

Q_2 **Quels sont les groupes de personnes qui font un travail similaire ? (ils appliquent des types d'activités similaires).**
Les personnes impliquées dans une activité sont les personnes qui échangent des courriels sur cette activité, ce qui peut être déduit de la partie expéditeur/récepteur du courriel. Par exemple, le groupe de personnes qui participent à l'activité "Modifier les données" est : Stephen Allen, Tony B, Herb Caballero, Kenneth, Roger Raney, Henry Van, Linda Adels, Paul Duplachan

Q_4 **Quel type de documents sont envoyés en pièces jointes à un courriel pour une activité spécifique ?** Chaque type d'activité est caractérisé par un attribut décrivant les attachements qui lui sont habituellement associés. Comme nous n'avions pas le contenu des pièces jointes de l'ensemble de données Enron, nous ne pouvons extraire des informations que du nom et du type des pièces jointes d'une activité spécifique.

Q_5 **Quels domaines de pages web ou de liens sont utilisés pour une activité donnée ?** En utilisant les mots clés obtenus décrivant les liens et les pages web associés aux activités, un utilisateur peut déduire ses domaines.

Il existe de nombreux logiciels commerciaux qui sont utilisés par les entreprises pour fournir une assistance à la clientèle pour la gestion des courriers électroniques. Ce type de logiciels aide les agents à suivre et à répondre plus facilement aux demandes de courrier électronique. Une autre fonctionnalité utile est la réception efficace des courriels, qui permet de réduire le spam. D'autres fonctionnalités clés sont l'enrichissement des données, comme la fourniture de détails sur l'auteur d'un courriel. Le logiciel peut également aider l'utilisateur à comprendre et à analyser le contenu d'un courriel. Les meilleures solutions de gestion de courrier électronique offrent également l'archivage et la récupération rapide des courriers électroniques.

Les gens utilisent le courrier électronique pour gérer les tâches quotidiennes, en utilisant la boîte de réception comme gestionnaire de tâches et leurs archives pour trouver des contacts et des documents de référence. Les utilisateurs créent délibérément des structures de dossiers ou des étiquettes qui contribuent à réduire la complexité de la boîte de réception. Sans les messages importants peuvent être négligés lorsque des le nombre de messages non organisés s'accumule dans un boîte de réception surchargée [4, 13, 63].

Le courrier électronique est également considéré comme un cas utile pour la synthèse puisque la plupart des gens reçoivent un grand nombre de courriers électroniques chaque jour. Le volume des courriels reçus entraîne un coût important en termes de temps nécessaire pour lire, trier et archiver les données entrantes. En utilisant le résumé de courrier électronique de différentes manières, le processus de gestion des dossiers de courrier électronique peut être facilité, ce qui constitue un moyen prometteur de réduire le triage des courriers électroniques. En outre, un résumé généré peut faciliter l'accès au courrier électronique sur le petit écran d'un appareil mobile. Les efforts précédents en matière de résumé des fils de discussion ont adopté des techniques développées pour le résumé général de textes multi-documents et les ont appliquées aux courriels en incluant des éléments spécifiques aux courriels. Nous pouvons classer les techniques de résumé en deux catégories : (a) Résumation extractive et (b) Résumation abstraite. Dans notre travail, nous nous concentrons principalement sur la résumation extractive. Pour cette raison, nous ne mentionnons que les travaux connexes sur la synthèse extractive des courriels.

Dans la synthèse extractive, les courriels sont résumés en extrayant des phrases du courriel sans y appliquer de modifications. En fait, ce type de résumé se prête bien à l'apprentissage machine. Le texte peut être séparé en phrases et le problème devient alors une tâche de classification sur les phrases. Les algorithmes de classification peuvent être formés pour sélectionner des phrases à extraire en fonction des caractéristiques de chaque phrase. De cette façon, la classification peut être utilisée pour la synthèse. Les phrases extraites sont considérées comme les phrases les plus importantes du courriel qui donnent une indication sur son contenu. En pratique, la mise en évidence des phrases permet aux utilisateurs de parcourir un courriel en lisant les phrases les plus importantes.

Dans le domaine de la gestion des processus métiers, un modèle de processus d'entreprise est un ensemble d'activités ou de tâches connexes et structurées qui produisent un service ou un produit spécifique (servent un objectif particulier). L'instance de processus métier est une exécution ou un cas spécifique du modèle de processus général. Une instance décrit un processus réel qui comprend des données, des actions réelles et des décisions spécifiques. Dans un journal d'événements, chaque événement est identifié par un identificateur de modèle de processus et un identificateur d'instance de processus ou de cas. Dans le domaine de l'extraction de processus ([60]), la plupart des approches existantes considèrent que les journaux d'événements contiennent des identificateurs de cas. Quelques exceptions se retrouvent dans le domaine de l'extraction de services, où le problème est appelé extraction de corrélation d'événements.

Dans le domaine de la gestion des processus d'entreprise, un grand nombre de recherches ont porté sur les techniques permettant de relier les événements des processus d'entreprise aux cas. À titre d'exemple, une telle technique est présentée dans [47]. Leur processus de corrélation est basé uniquement sur la relation temporelle entre les événements. Un processus interactif de corrélation d'événements est présenté dans la [47] où les entrées de l'utilisateur sont prises en compte pour sélectionner des corrélations intéressantes. Les conditions de corrélation des événements sont découvertes sur la base de la valeur des champs d'événements communs. Dans un autre ouvrage [51], MapReduce est utilisé pour la mise à l'échelle de l'approche d'analyse des événements de processus. Leur approche introduit des méthodes efficaces pour partitionner un journal d'événements (agrégés à partir de différentes sources de données) entre map-reduce cluster et réduire les nœuds de grappe afin d'équilibrer la charge de travail liée aux calculs des conditions atomiques tout en réduisant les transferts de données.

En conclusion, alors que son utilisation initiale était axée sur l'échange de messages personnels entre individus, le courrier électronique a évolué d'un simple système de communication à un moyen d'organiser des activités complexes (flux de travail), de stocker des informations, de partager et de modifier des documents et de coordonner l'exécution de tâches complexes impliquant plusieurs individus. En raison de sa large utilisation dans des contextes personnels, mais surtout professionnels, le courrier électronique représente une source précieuse d'informations qui peuvent être récoltées pour comprendre, réorganiser et réorienter les processus métiers non documentés des entreprises et des institutions.

En tenant compte de cela, nous travaillons dans cette thèse sur l'extraction d'informations sur les processus métiers à partir de journaux de courrier électronique. Nous transformons les journaux de courriels en journaux d'événements où ces derniers peuvent être utilisés comme intrants pour les outils traditionnels d'extraction de processus afin de découvrir des modèles de processus métiers. Les modèles de processus métiers produits fournissent une vue d'ensemble claire des processus et des activités dans un journal de courrier électronique de l'utilisateur. L'organisation des courriers électroniques en tant que processus métier permet aux utilisateurs de traiter un courrier électronique dans le cadre d'un processus qui les aide à gérer les tâches qui leur sont assignées et à suivre l'évolution globale du processus. En outre, l'organisation des journaux de courriers électroniques en tant que processus facilite et enrichit l'analyse qui peut être appliquée aux courriers électroniques.

Tout au long des chapitres de cette thèse, nous décrivons les principales approches qui constituent le cadre présenté dans cet ouvrage. Nous attribuons à chaque composante de ce cadre une approche qui est en charge d'une partie spécifique de la cible globale. Chacune des approches décrites produit des résultats susceptibles d'être interrogés et analysés. Ensemble, les résultats collectés de toutes les approches permettent d'atteindre l'objectif final, c'est-à-dire la découverte de modèles de processus métiers à partir des journaux de courriers électroniques. Les composantes du cadre sont principalement concernées par : (1) la découverte de sujets de processus métier pour les courriels, (2) la décou-

verte d'instances de processus métier pour les courriels, (3) les activités métier et l'extraction de métadonnées des courriels, (4) l'apprentissage relationnel entre la découverte d'instances de processus et la découverte d'activités métier, (3) l'estimation préliminaire de l'horodatage de l'occurrence de l'activité. Cela a permis d'extraire les principaux attributs du journal des événements à partir d'un journal de courrier électronique. Le journal d'événements déduit permet à son tour la découverte des modèles de processus métiers.

En fait, le travail que nous avons accompli ouvre la voie à de multiples recherches supplémentaires. Il existe plusieurs perspectives potentielles basées sur les résultats obtenus. L'une des perspectives est de construire un système de recommandation, qui tire des enseignements des modèles de processus métiers obtenus précédemment à partir des journaux de courriers électroniques. Ce système de recommandation peut recommander à l'utilisateur, à la réception d'un courriel, les activités potentielles à appliquer ensuite sur la base des courriels reçus précédemment. Il peut également recommander à l'utilisateur les types de documents qui peuvent être joints à l'e-mail à envoyer ou certains liens internet candidats qui peuvent être utiles pour le destinataire de l'e-mail, etc. Un tel système de recommandation permet d'optimiser l'exécution des processus d'entreprise en termes de performance et de durée.

Un autre travail potentiel important est de permettre l'apprentissage progressif dans notre cadre. Les utilisateurs du courrier électronique reçoivent des courriels quotidiennement. Ces courriels peuvent contenir de nouveaux sujets et activités relatifs aux processus de gestion. Par conséquent, nos approches appliquées ne seraient pas utiles si nous ne dépendions que des processus métiers appris lors des échanges de courriers électroniques passés. C'est pourquoi l'apprentissage progressif est considéré comme important dans ce cas. Au fur et à mesure que de nouveaux sujets et activités de processus métier sont reçus et détectés, nos modèles d'apprentissage devraient être mis à jour progressivement. Cela permet à nos modèles d'être dynamiques, à jour et de faire face à tous les changements qui pourraient survenir à l'avenir.

Pour que l'utilisateur puisse utiliser ce cadre, une interface utilisateur est nécessaire. Pour l'instant, notre cadre peut être mis en œuvre par des personnes qui connaissent le fonctionnement du programme. Toutefois, afin de présenter notre cadre comme un produit, il est crucial de construire une interface utilisateur qui permette aux utilisateurs et aux analystes de différents domaines d'utiliser toutes les caractéristiques et tous les outils disponibles. Cette thèse peut être considérée comme un pas en avant dans la construction d'un produit qui aide les utilisateurs à gérer leurs e-mails d'une toute nouvelle manière qui n'est pas réellement disponible dans les outils de gestion d'e-mails actuels.

Bibliography

[1] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: A brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

[2] Izzat Alsmadi and Ikdam Alhami. Clustering and classification of email contents. *Journal of King Saud University-Computer and Information Sciences*, 27(1):46–57, 2015.

[3] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taskmaster: recasting email as task management. *PARC, CSCW*, 2, 2002.

[4] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 345–352. ACM, 2003.

[5] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, Ian Smith, and Rebecca E Grinter. Quality versus quantity: E-mail-centric task management and its relation with overload. *Human-computer interaction*, 20(1):89–138, 2005.

[6] Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM, 2007.

[7] Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. Summarizing emails with conversational cohesion and subjectivity. *Proceedings of ACL-08: HLT*, pages 353–361, 2008.

[8] José M Carmona-Cejudo, Manuel Baena-García, Rafael Morales Bueno, João Gama, and Albert Bifet. Using gnusmail to compare data stream mining methods for on-line email classification. In *WAPA*, pages 12–18, 2011.

[9] Vitor R Carvalho and William W Cohen. Improving email speech acts analysis via n-gram selection. In *Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech*, pages 35–41. Association for Computational Linguistics, 2006.

[10] LOFI Christoph. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. *Information and Media Technologies*, 10(3):493–501, 2015.

[11] William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. Learning to classify email into speech acts. In *In Proceedings of Empirical Methods in Natural Language Processing*, 2004.

[12] Simon Corston-oliver, Eric Ringger, Michael Gamon, and Richard Campbell. Task-focused summarization of email. In *IN PROCEEDINGS OF THE TEXT SUMMARIZATION BRANCHES OUT ACL WORKSHOP*, 2004.

[13] Laura A Dabbish, Robert E Kraut, Susan Fussell, and Sara Kiesler. Understanding email use: predicting action on a message. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 691–700. ACM, 2005.

[14] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[15] Yashwant Dongre Deepa Patil. A clustering technique for email content mining. *International Journal of Computer Science and Information Technology (IJCSIT)*, 7(3), 2015.

[16] Claudio Di Ciccio, Massimo Mecella, Monica Scannapieco, Diego Zardetto, and Tiziana Catarci. Mailofmine–analyzing mail messages for mining artful collaborative processes. In *International Symposium on Data-Driven Process Discovery and Analysis*, pages 55–81. Springer, 2011.

[17] Mark Dredze, Tessa Lau, and Nicholas Kushmerick. Automatically classifying emails into activities. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 70–77. ACM, 2006.

[18] Nicolas Ducheneaut and Victoria Bellotti. E-mail as habitat: an exploration of embedded personal information management. *interactions*, 8(5):30–38, 2001.

[19] Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

[20] Andrew Faulring, Brad Myers, Ken Mohnkern, Bradley Schmerl, Aaron Steinfeld, John Zimmerman, Asim Smailagic, Jeffery Hansen, and Daniel Siewiorek. Agent-assisted task management that reduces email overload. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 61–70. ACM, 2010.

[21] Danyel Fisher and Paul Moody. *Studies of automated collection of email records*. Institute for Software Research, University of California, Irvine, 2002.

[22] Peter W Foltz. Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, & Computers*, 28(2):197–202, 1996.

[23] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

[24] George W Furnas, Scott Deerwester, Susan T Dumais, Thomas K Landauer, Richard A Harshman, Lynn A Streeter, and Karen E Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 465–480. ACM, 1988.

[25] Giuseppe Futia, Antonio Vetro, Alessio Melandri, and Juan Carlos De Martin. Training neural language models with sparql queries for semi-automatic semantic mapping. *Procedia Computer Science*, 137:187–198, 2018.

[26] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[27] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[28] Jacek Gwizdka. Taskview: design and evaluation of a task-based email interface. In *Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research*, page 4. IBM Press, 2002.

[29] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[30] Geoffrey E Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[31] Diana Jlailaty, Daniela Grigori, and Khalid Belhajjame. Business process instances discovery from email logs. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 19–26. IEEE, 2017.

[32] Diana Jlailaty, Daniela Grigori, and Khalid Belhajjame. Mining business process activities from email logs. In *Cognitive Computing (ICCC), 2017 IEEE International Conference on*, pages 112–119. IEEE, 2017.

[33] Diana Jlailaty, Daniela Grigori, and Khalid Belhajjame. Email business activities extraction and annotation. In *International Workshop on Information Search, Integration, and Personalization*, pages 69–86. Springer, 2018.

[34] Diana Jlailaty, Daniela Grigori, and Khalid Belhajjame. On the elicitation and annotation of business activities based on emails. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 101–103. ACM, 2019.

[35] Rinat Khoussainov and Nicholas Kushmerick. Email task management: An iterative relational learning approach. In *CEAS*, 2005.

[36] ChanMin Kim. Using email to enable e3 (effective, efficient, and engaging) learning. *Distance Education*, 29(2):187–198, 2008.

[37] Jan Hajič Hans Uszkoreit António Branco Kiril Simov, Petya Osenova. In *The Workshop on Deep Language Processing for Quality Machine Translation (DeepLP4QMT)*. Springer, 2016.

[38] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer, 2004.

[39] Irena Koprinska, Josiah Poon, James Clark, and Jason Chan. Learning to classify e-mail. *Information Sciences*, 177(10):2167–2187, 2007.

[40] Derek Scott Lam. *Exploiting e-mail structure to improve summarization*. PhD thesis, Massachusetts Institute of Technology, 2002.

[41] Hua Li, Dou Shen, Benyu Zhang, Zheng Chen, and Qiang Yang. Adding semantics to email clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 938–942. IEEE, 2006.

[42] Matin Mavaddat, Ian Beeson, Stewart Green, and Jin Sa. Facilitating business process discovery using email analysis. In *The First International Conference on Business Intelligence and Technology*. Citeseer, 2011.

[43] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[44] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[45] Mohit Kakkar Mubashir Alam. Email summarization-extracting main content from the mai. *International Journal of Innovative Research in Computer and Communication Engineering*, 3, 2015.

[46] Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[47] Shaya Pourmirza, Remco M. Dijkman, and Paul W. P. J. Grefen. Correlation mining: Mining process orchestrations without case identifiers. In *Service-Oriented Computing - 13th International Conference, ICSOC 2015, Goa, India, November 16-19, 2015, Proceedings*, pages 237–252, 2015.

[48] Feng Qian, Abhinav Pathak, Yu Charlie Hu, Zhuoqing Morley Mao, and Yinglian Xie. A case for unsupervised-learning-based spam filtering. In *ACM SIGMETRICS Performance Evaluation Review*, volume 38, pages 367–368. ACM, 2010.

[49] Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 105–108. Association for Computational Linguistics, 2004.

[50] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[51] Hicham Reguieg, Boualem Benatallah, Hamid R. Motahari Nezhad, and Farouk Toumani. Event correlation analytics: Scaling process mining using mapreduce-aware event correlation discovery techniques. *IEEE Trans. Services Computing*, 8(6):847–860, 2015.

[52] Maharana Sangeeta, Mohite Minal, and Wadekar Pornima. Email clustering using lingo algorithm. *International Journal of Computer Science Trends and Technology (IJCST)*, 2, 2014.

[53] John R Searle. A classification of illocutionary acts. *Language in society*, 5(01):1–23, 1976.

[54] Arungunram C Surendran, Erin L Renshaw, and John C Platt. Automatic organization of documents through email clustering, July 27 2010. US Patent 7,765,212.

[55] Kabita Thaoroijam. A study on document classification using machine learning techniques. *International Journal of Computer Science Issues (IJCSI)*, 11(2):217, 2014.

[56] Nicolas Turenne. Learning semantic classes for improving email classification. In *Proceedings of Text Mining and Link Analysis Workshop*, 2003.

[57] Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

[58] Jan Ulrich, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. Regression-based summarization of email conversations. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[59] Wil MP van der Aalst and Andriy Nikolov. Emailanalyzer: an e-mail mining plug-in for the prom framework. *BPM Center Report BPM-07-16, BPMCenter. org*, 2007.

[60] Wil MP Van der Aalst, Boudewijn F van Dongen, Joachim Herbst, Laura Maruster, Guido Schimm, and Anton JMM Weijters. Workflow mining: A survey of issues and approaches. *Data & knowledge engineering*, 47(2):237–267, 2003.

[61] Stephen Wan and Kathy McKeown. Generating overview summaries of ongoing email thread discussions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 549. Association for Computational Linguistics, 2004.

[62] Steve Whittaker, Victoria Bellotti, and Jacek Gwizdka. Email and pim: Problems and possibilities. *Communications in ACM*, 2007.

[63] Steve Whittaker and Candace Sidner. Email overload: exploring personal information management of email. *Culture of the Internet*, pages 277–295, 1997.

[64] Jihoon Yang and Sung-Yong Park. Email categorization using fast machine learning algorithms. In *International Conference on Discovery Science*, pages 316–323. Springer, 2002.

[65] Yiming Yang, Shinjae Yoo, Jian Zhang, and Bryan Kisiel. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 98–105. ACM, 2005.

[66] Kelvin S Yiu, Ronald Baecker, Nancy Silver, and Byron Long. A time-based interface for electronic mail and task management. *Advances in human factors ergonomics*, 21:19–22, 1997.

[67] Shinjae Yoo, Yiming Yang, Frank Lin, and Il-Chul Moon. Mining social networks for personalized email prioritization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 967–976. ACM, 2009.

[68] Y Yuan, L He, L Peng, and Z Huang. A new study based on word2vec and cluster for document categorization. *Journal of Computational Information Systems*, 10(21):9301–9308, 2014.

[69] David M Zajic, Bonnie J Dorr, and Jimmy Lin. Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, 44(4):1600–1610, 2008.

RÉSUMÉ

Les informations échangées dans les textes des courriels sont généralement concernées par des événements complexes ou des processus métier dans lesquels les entités qui échangent des courriels collaborent pour atteindre les objectifs finaux des processus. Ainsi, le flux d'informations dans les courriels envoyés et reçus constitue une partie essentielle, les activités métier de l'entreprise. L'extraction d'informations sur les processus métier à partir des courriels peut aider à améliorer la gestion des courriels pour les utilisateurs. Il peut également être utilisé pour trouver des réponses riches à plusieurs questions analytiques sur les employés et les organisations. Aucun des travaux précédents n'a résolu le problème de la transformation automatique des journaux de courriels en journaux d'événements pour éventuellement en déduire les processus métier non documentés. Dans ce but, nous travaillons dans cette thèse sur un framework qui induit des informations de processus métier à partir d'emails. Nous introduisons des approches qui contribuent à ce qui suit : (1) découvrir pour chaque courriel le sujet de processus qui le concerne, (2) découvrir l'instance de processus métier à laquelle appartient chaque courriel, (3) extraire les activités de processus métier des courriels et associer ces activités aux métadonnées qui les décrivent, (4) améliorer la performance de la découverte des instances de processus métier et des activités métier en utilisant la relation entre ces deux problèmes, et enfin (5) estimer au préalable la date/heure réelle d'un activité métier. En utilisant les résultats des approches mentionnées, un journal d'événements est généré qui peut être utilisé pour déduire les modèles de processus métier d'un journal de courriels. L'efficacité de toutes les approches ci-dessus est prouvée par l'application de plusieurs expériences sur l'ensemble de données de courriel ouvert d'Enron.

MOTS CLÉS

Email Mining, Gestion des Processus Métier, Fouille de Données, Extraction des Activités Modèles

ABSTRACT

Exchanged information in emails' texts is usually concerned by complex events or business processes in which the entities exchanging emails are collaborating to achieve the processes' final goals. Thus, the flow of information in the sent and received emails constitutes an essential part of such processes i.e. the tasks or the business activities. Extracting information about business processes from emails can help in enhancing the email management for users. It can be also used in finding rich answers for several analytical queries about the employees and the organizations enacting these business processes. None of the previous works have fully dealt with the problem of automatically transforming email logs into event logs to eventually deduce the undocumented business processes. Towards this aim, we work in this thesis on a framework that induces business process information from emails. We introduce approaches that contribute in the following: (1) discovering for each email the process topic it is concerned by, (2) finding out the business process instance that each email belongs to, (3) extracting business process activities from emails and associating these activities with metadata describing them, (4) improving the performance of business process instances discovery and business activities discovery from emails by making use of the relation between these two problems, and finally (5) preliminary estimating the real timestamp of a business process activity instead of using the email timestamp. Using the results of the mentioned approaches, an event log is generated which can be used for deducing the business process models of an email log. The efficiency of all of the above approaches is proven by applying several experiments on the open Enron email dataset.

KEYWORDS

Email Mining, Business Process Management, Text Mining, Business Activities Extraction