Chapter 8
Analysis of Multiple Business Perspectives

Abstract This chapter discusses analytical questions that arise at the interfaces between different BI perspectives, i.e., customer, production, and organization. In order to answer these questions, we introduce social network analysis as a new technique, adapt different analysis techniques introduced in the previous chapters to the analytical goals under consideration, and combine analysis techniques, for example, the combined application of cross-sectional analysis and process mining.

8.1 Introduction and Terminology

Let us first recall the different BI perspectives customer: production and organization as displayed in Fig. 8.1.

Chapters 5–7 introduce analytical techniques for cross-sectional, temporal, and process analysis. These techniques are mainly used to answer analytical questions regarding the customer and the production perspective. A possible example analysis question regarding the customer perspective answered by association analysis (cf. Sect. 6.5) reads as follows:

Which products were bought in combination with buying a nail?

An analytical question concerning the production perspective may be this:

Do uploads take place after the corresponding milestone deadline?

This question can be addressed based on process analysis methods such as process mining and compliance checking (cf. Sect. 7.5).

The organizational perspective and related questions have not been addressed in this book so far. We provide some insights on how organizational information is typically modeled and visualized in Chap. 4, Sect. 4.2.4, but analytical techniques dedicated to the organizational perspective have not been addressed yet.

In the case of the HEP use case (cf. Sect. 1.4.2), an analytical question regarding the organization could be:

Which actors work on the same task in higher education processes?

We can try to answer this question by using social network analysis (SNA). We will give an introduction into SNA in Sect. 8.2.1.

In addition to analytical questions that refer to a single perspective, questions that arise at the interfaces between the different perspectives can be of interest. An

W. Grossmann, S. Rinderle-Ma, *Fundamentals of Business Intelligence*,
Data-Centric Systems and Applications, DOI 10.1007/978-3-662-46531-8_8

Fig. 8.1 Overview of analysis techniques, cf. Chap. 1.3

example question arising at the interface between the organization and production perspectives in the HEP use case is

Which roles do participate in the higher education process?

In order to answer this question, a combined application of process mining and social network analysis can be employed as discussed in Sect. 8.2.2.

A question arising at the interface between the production and customer perspectives in the logistics use case (cf. Sect. 1.4.3) is:

For which containers did the vehicle return to the origin?

Section 8.3 introduces decision mining as a combined application of cross-sectional analysis and process mining that can be used to answer the above question.

Finally, an example question arising at the interface between the customer and organization perspectives is:

What is the opinion about the Oscar nominations of Jennifer Lawrence when compared to Sandra Bullock?[1]

This question can be tackled by applying text mining as a combination of data mining and social mining (cf. Sect. 8.4). More precisely, we can apply *opinion mining* (Sect. 8.4.5) which employs text mining techniques.

Section 8.5 focuses on evaluation and assessment methods for combined analysis techniques.

[1]Compare the "social mentions" on the Oscar website http://oscar.go.com/nominees.

8.2 Social Network Analysis and Organizational Mining

This section presents methods to analyze the organizational perspective (cf. Fig. 8.1). At first, an introduction to social network analysis is provided, followed by the application in the context of business processes.

8.2.1 Social Network Analysis

Social network analysis (SNA) aims at describing and analyzing the relationships between *social entities* such as actors and their roles or organizations. In addition, SNA is interested in finding "patterns and implications of these relationships" (see [37]), for example, starting from a certain person and including friends of friends. Relationships (or relations for short) can be of different types, for example, *working together* or *is friend of*. Two actors who are connected by a relation in the social network form a so-called *dyad*; three actors form a so-called *triad*. SNA provides the basis for answering questions from different areas such as business, social sciences, or biology [37].

In order to enable a meaningful and focused analysis, we have to identify which social entities are of interest and which kind of relation between these entities should be used. Both decisions depend on the data. If the data basis becomes very large, "the boundedness of social relations and the possibility of drawing samples" from possible social entities is crucial (see [33]). Think, for example, of Twitter or Facebook data, which is huge. The relationships between the entities may be defined explicitly or have to be derived from other relations in the database. In the case of social networks, the data often directly reveal a social relation between the entities, for example, "is friend of." Other data sources include more general relations between entities, such as "are involved in the same event," and the relationship has to be defined according to the precise formulation of the analytical goal.

The main steps of SNA are summarized in the following analysis template.

Template: Social Network Analysis

- **Relevant Business and Data:** A database containing information about social entities together with relations between these entities.
- **Analytical Goals:**
  - Visualization of the relations between the entities
  - Describing the relationships by summary measures
  - Finding patterns in the relationships
- **Modeling Task:** Generate from the data, first of all, a *data matrix* which defines a graph with the social entities as nodes and the relations between the entities as edges. This graph defines the analytical model and is henceforward called *sociogram* [33].

- **Analysis Task:** Analyze the sociogram using different metrics which allow quantification of the relationship between the entities.
- **Evaluation and Reporting Task:** Visualize the sociogram and represent the descriptive measures. Using an explorative approach, possibly supported by tools, the visualization can be the basis for the interpretation of the analysis results. It can be useful to visualize the sociograms together with different metrics in different layouts. In particular at the presence of large social networks or sociograms, a visualization should convey the analysis results in an understandable and interpretable way.

Modeling Task

The basis for SNA is the representation of the *social network* as sociogram. From the graph structures introduced in Sect. 2.3, the most important ones for the context are summarized in the following overview.

Model Structures for Social Network Analysis

- *Undirected graphs:* As introduced in Sect. 2.3, an undirected graph G is defined as $G = (V, E)$ with set of nodes V and set of undirected edges E.
- *Directed graphs:* Opposed to undirected edges, directed edges establish a relation that reflects a causal relation or a relation that is directed from one to another entity.
- *Weighted graphs:* It can be also useful to assign weights to the edges in the graph, i.e., a weight $w(e)$ expressing some kind of quantitative measure for the relation.
- *Connected subgraphs:* Special connected subgraphs might be of interest. A subgraph consisting of two nodes (with or without relations between them) describes a dyad and a subgraph consisting of three nodes of interest, a triad.

In the case of undirected graphs, an edge (v_1, v_2) between the nodes $v_1, v_2 \in V$ means that the relation between the entities associated with nodes v_1 and v_2 exists in both directions. For example, the entity associated with v_1 works with the entity associated with v_2 and vice versa. In the case of directed graphs, the interpretation of a directed edge (v_1, v_2) may be that the entity associated with v_1 hands over work to the entity associated with v_2. Weighted graphs are useful for quantifying the strength of a relationship. For example, a directed edge $e = (v_1, v_2)$ with assigned weight $w(e) = 3$ could mean that the entity associated with v_1 has handed over work to the entity associated with v_2 for three times.

With respect to the representation of the graph, one can use either an adjacency matrix or an incidence matrix defined in the modeling task by extracting information from the existing data about the business. The main challenge in modeling is the selection of the entities and the definition of the social relations. Let us illustrate the modeling task by building the sociogram in connection with the higher education use case.

HEP Use Case: Definition of a Sociogram

In the teaching processes of the HEP project, different actors are involved, i.e., students, lecturers, tutors, and the system (learning platform). Assume that we are interested in the following question:

Which actors work together in higher education processes?

In order to answer the question, we decide that the nodes of the sociogram will reflect the actors. For establishing the relations in the sociogram recall that the HEP data is provided as process logs. From these logs, we can find out, for example, which actors worked on the same tasks. Consider the following log fragment consisting of entries of 4 activity executions. The `Originator` field contains the information which actor performed an activity denoted by the `WorkflowModelElement`, e.g., actor `person001-lecturer` performed activity `Evaluate presentation 1`.

```
<AuditTrailEntry>
   <WorkflowModelElement>Evaluate presentation 1</WorkflowModelElement>
   <EventType>complete</EventType>
   <Timestamp>2008-10-29T23:59:00.000+01:00</Timestamp>
   <Originator>person001-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
   <WorkflowModelElement>Evaluate presentation 1</WorkflowModelElement>
   <EventType>complete</EventType>
   <Timestamp>2008-10-28T23:59:59.000+01:00</Timestamp>
   <Originator>person003-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
   <WorkflowModelElement>plus</WorkflowModelElement>
   <EventType>complete</EventType>
   <Timestamp>2008-11-12T00:00:00.000+01:00</Timestamp>
   <Originator>person003-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
   <WorkflowModelElement>plus</WorkflowModelElement>
   <EventType>complete</EventType>
   <Timestamp>2008-11-05T00:00:00.000+01:00</Timestamp>
   <Originator>person004-lecturer</Originator>
</AuditTrailEntry>
```

Based on the log fragment, we can see that activity `Evaluate presentation 1` is performed by actors `person001-lecturer` and `person003-lecturer` and activity `plus` by actors `person003-lecturer` and `person004-lecturer`. The corresponding incidence matrix and sociogram are shown in Fig. 8.2. The sociogram includes undirected edges between nodes `person001-lecturer` and `person003-lecturer` as well as `person003-lecturer` and

Data Matrix:

	Evaluate presentation 1	plus
person001-lecturer	1	0
person002-lecturer	0	0
person003-lecturer	1	1
person004-lecturer	0	1

Sociogram:

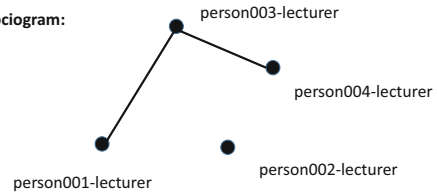

Fig. 8.2 HEP Data: sociogram with three entities (actors) and relations "working on same task" for process activities `evaluate presentation 1` and `plus`

`person004-lecturer`. Details on the HEP project can be found on the homepage of the book:
www.businessintelligence-fundamentals.com

Analysis Task

For the analysis of sociograms, different metrics can be calculated that reflect different social relations. We can, for example, determine whether a social network is *dense*, i.e., many interactions between the entities within the network take place. Other metrics refer to single entities or groups of entities instead of the entire network. For a single entity, it can be of interest whether it is a *central* entity within the network, i.e., many interactions occur with involvement of the entity. In this book, we cannot cover all measures on sociograms that are used for SNA. We will introduce intuitive measures that enable an understanding of the basic principles of SNA and its application in organizational mining (cf. Sect. 8.2.2). For an overview on further measures, we refer the interested reader to [37]. In the following, we present *local* and *global* measures, i.e., measures that refer to single entities (nodes) within the network (local) or to the entire social network (global).

Local Measures

One measure for the interactions of an entity, reflected by a node n, is the degree of n, i.e., the number of adjacent edges to n in an undirected graph and the number of edges starting at n (out-degree) or ending in n (in-degree) in a directed graph (see Sect. 2.3 for the definition of degree, in-degree, and out-degree). The degree could reflect, for example, how important or popular the associated entity is. The intuition behind is that a central entity is "well-connected" within the sociogram. The degree of node `person003-lecturer` is 2 based on the sociogram shown in Fig. 8.2.

The question is whether this absolute measure is meaningful, i.e., relative measures such as degree over number of all nodes or degree compared to degree of the other nodes might be more helpful. Think, for example, of a node having a degree of 2 in a sociogram of 4 nodes (as in Fig. 8.2) compared to a node having a degree of 2 in a sociogram of 10,000 nodes. Hence, for SNA, the measure of *degree centrality* can be calculated in an absolute and relative manner. Relative means to divide the number of adjacent nodes to a node of interest by the overall number of

nodes. The degree centrality of node *person003-lecturer* in Fig. 8.2 is 2 (absolute) and $\frac{1}{2}$ (relative).

Degree centrality can be extended to counting paths[2] of a given length that originates from the node of interest (*k-path centrality*). In the example shown in Fig. 8.2, nodes `person001-lecturer` and `person002-lecturer` are connected by a path of length 2. Imagine an additional edge between `person003-lecturer` and `person002-lecturer`. For `person001-lecturer`, the degree centrality would still be 1, but the 2-path centrality would be 2 as `person001-lecturer` is connected to `person002-lecturer` and `person004-lecturer` via paths of length 2.

Another measure is the *closeness* of a node to the other nodes in the sociogram [33]. The closeness of a node of interest can be determined by counting in how many shortest paths[3] occur between two nodes. The length of the shortest path between two nodes is also referred to as the *distance* between them.

Global Measures

Density is a property of the entire sociogram. It measures how many edges are present in the sociogram compared to the complete graph on the same set of nodes. Intuitively, a complete graph contains all possible edges between the existing nodes. The formula for calculating the density dens of an undirected graph $G = (V; E)$ is

$$\text{dens}(G) := \frac{2 * \mid E \mid}{\mid V \mid * (\mid V \mid -1)}. \tag{8.1}$$

Consider the sociogram depicted in Fig. 8.2. The density of this graph turns out as $\frac{1}{3}$.

As a variant to density on the complete sociogram, *egocentric* measures of density can yield different insights. They are particularly interesting for analyzing the relations between the direct contacts of a given entity also called *ego* [33]. An interesting study about the implications of density is [38]. In summary, this study relates the density of personal networks to the family relations between members of the networks. The result is that the higher the density of the network, the higher is the percentage of people that are kin.

Evaluation and Reporting Task

Visualizations of the sociogram and the calculated measures support the interpretation of the SNA results. They empower the user to analyze the SNA results in an explorative manner, possibly supported by tools [4]. There are several techniques for visualizing sociograms, for example, circular layout, Kamada–Kawai layout

[2] A path between two nodes v, $w \in V$ consists of a set of (directed) edges connecting v and w. The length of a path is the number of edges it consists of, cf. Sect. 2.3.

[3] The shortest path between two nodes is the one with minimal length. Several algorithms exist to calculate the shortest paths between two or all nodes in a (directed) graph, for example, Dijkstra or Floyd-Warshall algorithm respectively, [16].

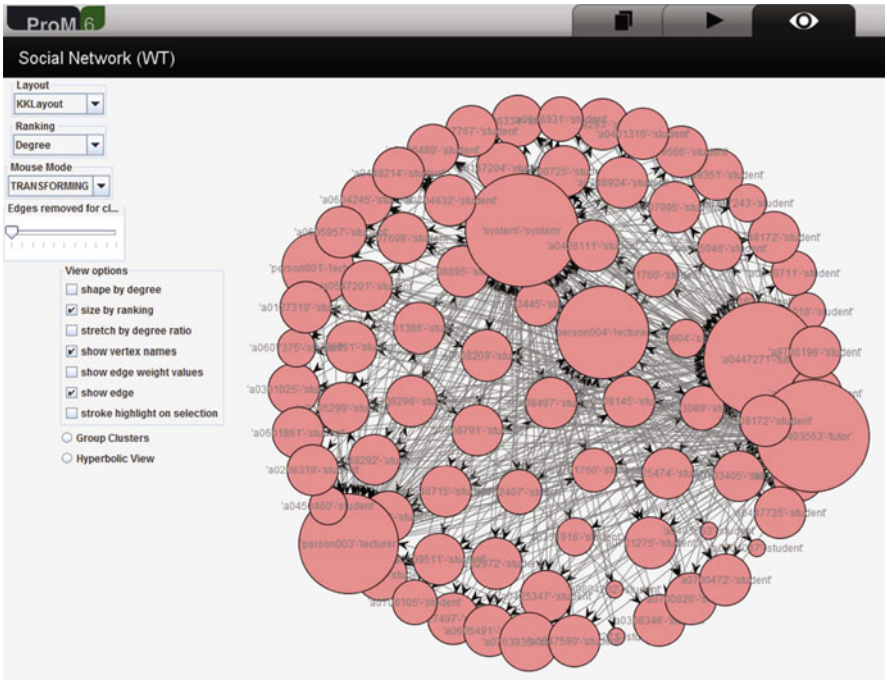

Fig. 8.3 HEP: sociogram using KKLayout, measure degree, and vertex size by ranking (using ProM 6.2)

(KKLayout) [17], or Fruchterman-Reingold layout [13]. Tools, such as Pajek[4] and ProM offer different options for layouting the sociograms. In this section, we do not dig deeply into visualization techniques, but will present different visualizations of the HEP data set in the following. Figure 8.3 shows the sociogram for the full HEP data set. It is produced by using ProM 6.2. We chose the KKLayout for the sociogram depicted in Fig. 8.3. It uses the degree of vertices as measures to determine their size.

8.2.2 Organizational Aspect in Business Processes

After introducing some SNA fundamentals, we explain in this section basics on organizational modeling regarding business processes.

In general, business processes do not only consist of process activities and their order relations but also incorporate aspects such as the data flow between

[4] http://pajek.imfm.si/doku.php?id=download.

process activities, the assignment of actors based on organizational structures, and invoked services (application components). All these aspects are of particular importance for the automation and execution of business processes within aProcess-Aware Information Systems (PAIS). At runtime, process activities are executed by human actors or invoke internal or external services that might exchange data. Typically, PAIS log which user has worked on which task. Consider, for example, the following MXML fragment for which the `Originator` tag shows that actor `Luke` performed the activity `registration`.

In the following, we present a set of techniques that refer to mining organizational structures from event logs (given that the `Originator` field is provided).

```
<AuditTrailEntry>
   <WorkflowModelElement>registration</WorkflowModelElement>
   <EventType>start</EventType>
   <Timestamp>2011-01-01T01:00:00.000+01:00</Timestamp>
   <Originator>Luke</Originator>
</AuditTrailEntry>
```

In general, process activities are not directly assigned to persons. Instead, organizational entities, such as *role* or *organizational unit*, are used as an abstraction layer. These organizational entities are captured and modeled by organizational models, often, based on role-based access control (RBAC) models [36]. Roles as well as organizational units can be related to actors. In addition, roles and organizational units can be also related in a hierarchical manner, i.e., expressing that one role is a subrole or junior role of another role. There exist several extensions to basic RBAC models, for example, capturing entities such as abilities or teams.

Figure 8.4 displays an example for an organizational model in a clinic. On top, organizational unit `Clinic` consists of organizational units `Ward` and `Admin`. Two roles are assigned to organizational unit `Ward`, i.e., `Nurse` and `Doctor`. Two actors have the role `Nurse`, i.e., `Sara` and `Bert`.

At the bottom of Fig. 8.4, process model PM refers to organizational model OM, i.e., the tasks in PM will be assigned to actors captured by OM. This is achieved by formulating a so-called actor assignment for a task. For task `C`, for example, we would like to express that only actors having the role `Nurse` can work on `C`. This is reflected by actor assignment `Role = Nurse`. At the process model level, this means that `Sara` and `Bert` can work on `C`.

Imagine now that process instances are created and executed based on process model PM. Then, during runtime, the actor assignments are resolved to the set of actors qualifying for the assignment. In the case of task `C`, `Sara` and `Bert` qualify. Hence, `C` is offered to both actors in their work list. Typically, the actor who first selects the task to work on is finally assigned to it. In turn, this task is then removed from the work list of all other actors.

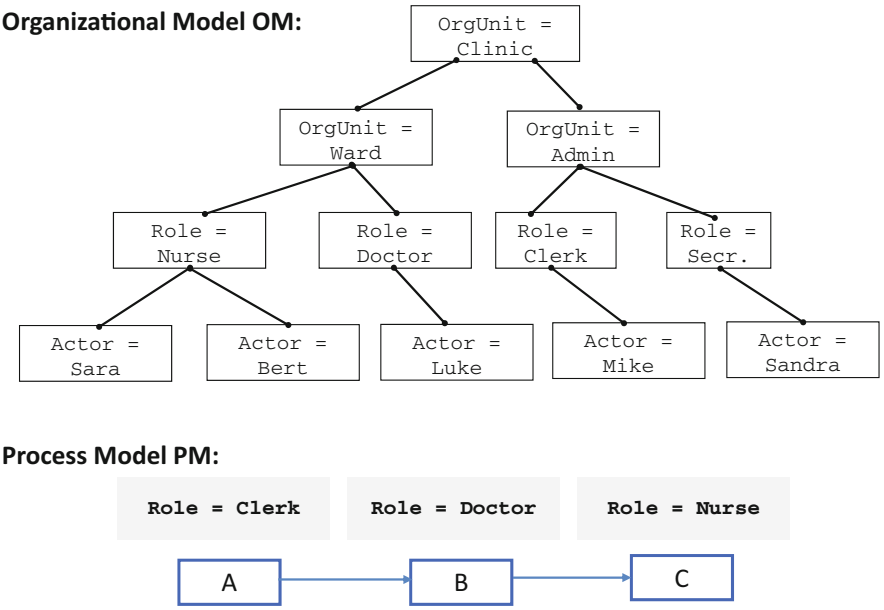

Fig. 8.4 Medical example: organizational model and process model with actor assignments

8.2.3 Organizational Mining Techniques for Business Processes

In this section, we describe how SNA techniques can be applied in combination with process mining techniques in order to address analysis questions at the interface between organization and production. The following paragraphs illustrate a selection of organizational mining techniques for business processes based on [34] and the associated implementation in ProM 5.2.[5] We opt for the version 5.2 of the ProM framework as it provides the implementations of the organizational and role hierarchy miner.

Organizational Mining

Organizational mining tries to derive the working behavior behind a process from event logs. The approach creates profiles for each actor in counting how many times an actor has performed a certain task. Based on the profiles, several distance metrics can be calculated addressing questions such as "who is doing similar tasks" or "who is working with whom" [34]. In turn, this information can be used to establish a suggestion for the organizational structure.

[5] http://promtools.org/prom5/.

Example 8.1 (Organizational Mining in Medical Example) Consider, for example, that activities A, B, and C of the process depicted in Fig. 8.4 were executed for ten instances by the three actors Mike, Luke, and Sara. The corresponding log for one instance would look like the following:

```
...
<AuditTrailEntry>
   <WorkflowModelElement>A</WorkflowModelElement>
   <EventType>start</EventType>
   <Timestamp>...</Timestamp>
   <Originator>Mike</Originator>
</AuditTrailEntry> <AuditTrailEntry>
   <WorkflowModelElement>B</WorkflowModelElement>
   <EventType>start</EventType>
   <Timestamp>...</Timestamp>
   <Originator>Luke</Originator>
</AuditTrailEntry> <AuditTrailEntry>
   <WorkflowModelElement>C</WorkflowModelElement>
   <EventType>start</EventType>
   <Timestamp>...</Timestamp>
   <Originator>Sara</Originator>
</AuditTrailEntry>
```

For all ten executed instances, the actor profiles turn out as:

Mike(10,0,0), Luke(0,10,0), and Sara(0,0,10),

meaning that Mike has worked on A, Luke on B, and Sara on C for all ten instances. Note that this means that for some reason, Bert has never executed task C even though he qualifies for it.

For subsequent analysis, these profiles can be transformed into the following representation:

Mike: $10 \times (1,0,0)$, Luke: $10 \times (0,1,0)$, and Sara: $10 \times (0,0,1)$,

The second representation builds the basis for cluster analysis (cf. Sect. 5.4), i.e., we interpret each execution as a case and want to find clusters of persons having similar work profiles. Usually, one uses hierarchical clustering or k-means clustering based on distances between the profiles, e.g., Hamming distance, Euclidean distance, or distances based on Pearson's correlation coefficient. These cluster solution defines profiles, and the distance between the clusters can be interpreted as the distance between the working profiles.

Obviously, a hierarchical cluster analysis of the working profiles in the above example would lead to a solution with three clusters and the profiles have maximal distance, i.e., no one is *working together*. These clusters are afterwards identified as three organizational units by the organizational miner and visualized by a bipartite graph (cf. Sect. 2.3.2). The result is depicted in Fig. 8.5a.

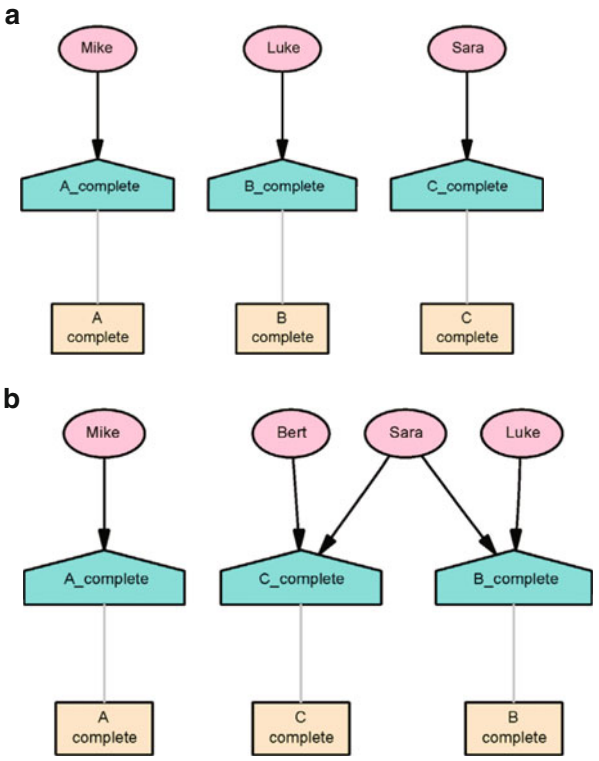

Fig. 8.5 Results of organizational mining on the medical example (using the *organizational miner* in ProM 5.2). (**a**) Organizational Miner, Profiles Mike(10,0,0), Luke(0,10,0), Sara(0,0,10) and (**b**) Organizational Miner, Profiles Mike(10,0,0), Bert(0,0,5), Luke(0,5,0), Sara(0,5,5)

Example 8.2 (Organizational Mining in Medical Example with Modifications) Assume now a modification of Example 8.1 with profiles:

`Mike(10,0,0)`, `Luke(0,5,0)`, `Bert(0,0,5)`, and `Sara(0,5,5)`.

Applying *organizational miner* results again in three clusters depicted in Fig. 8.5b. As `Luke` and `Sara` worked on task `B` and `Bert` and `Sara` worked on task `C`, `Sara` has now been assigned to two clusters. One assignment is in agreement with the role model and the other one in disagreement. This result can be interpreted in three ways. At first sight, this could mean that `Sara` has both roles `Nurse` and `Doctor` that is not reflecting the original model. As a second interpretation, task `B` could have been assigned to both roles `Nurse` and `Doctor` by a corresponding actor assignment:

```
B ⟵ Role='Doctor' AND Role='Nurse
```

This is not reflected by the original actor assignment either. As a third interpretation, during process execution, actors deviated from the original assignments due to, for example, substitution or emergency cases [30]. Such workarounds are neither

reflected by the organizational nor the process model (including actor assignments), but can be detected by techniques such as organizational mining.

Role Hierarchy Mining

In practical settings, the role structure within the organizational model is not "flat," i.e., roles might be ordered in some hierarchical relation to each other. More precisely, a "more powerful" role can be defined as a *senior* role to a "less powerful" *junior* role [32]. More precisely, the senior role inherits all permissions of its junior roles and might have additional permissions.

Role hierarchy mining aims at finding these senior-junior role relations based on logs and the contained originator information. As for organizational mining, the user profiles build the basis for determining the role hierarchy.

Example 8.3 (Role Hierarchy Mining in Medical Example) Consider the user profiles provided in Example 8.2. Using the *role hierarchy mining*, plug-in of ProM 5.2 results in the role hierarchy depicted in Fig. 8.6. As `Sara` has worked on the same tasks as `Luke` and `Bert`, it can be assumed that `Sara` has a senior role related to the roles of `Luke` and `Bert`.

Another organizational mining technique is *staff assignment mining* [23], i.e., deriving the actor assignments connecting process tasks to organizational units. Note that this technique does not only require a log as input but also an existing

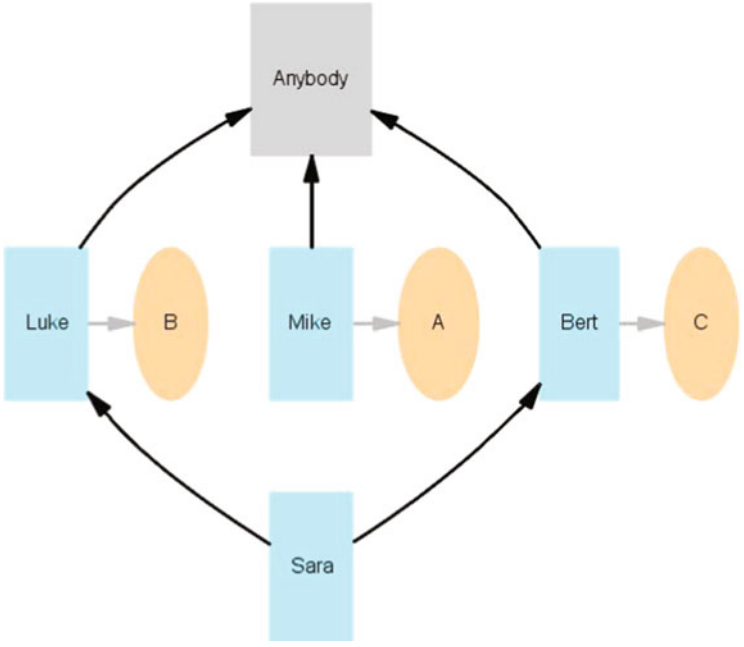

Fig. 8.6 Results of role hierarchy mining on medical example (using ProM 5.2)

organizational model. The algorithm itself uses decision trees to separate the set of actors (from the log) along the organizational entities (from the organizational model). Hence, the approach constitutes an example for combined application of analysis techniques, i.e., process mining and data mining. In other words, staff assignment mining applies the classification of actors along the organizational entities.

Social Network Mining

Beyond deriving organizational structures, it can also be of interest how the actors participating in the process execution interact with each other. Interaction can mean that they directly work together, hand over work to each other, or work on similar tasks. Such analyses can be performed by using techniques from social network mining (cf. Sect. 8.2.1) on the process execution logs. These logs yield two kinds of input information, i.e., the user profiles and the underlying process structure. The process structure is important for determining the relations between the actors. If, for example, two actors are only working on tasks that are ordered in parallel, they do not work together.

Example 8.4 (Social Network Mining on the Medical Example) Consider the process displayed in Fig. 8.7. In this process, the tasks `B` and `C` are ordered in parallel. `Sara` works on both `B` and `C` for five times each, `Luke` works on `B` for five times, and `Bert` works on `C` for five times. `A` and `D` are performed by `Mike` and `Sara` ten times each. Using the Social Network Miner of ProM 5.2. the result for analysis *working together* is also shown in Fig. 8.7. We can conclude that based on the profiles and the process structure, `Mike` works together with, for example, `Luke` in 50 % of the cases, whereas `Mike` works with `Sara` in 100 % of the cases. It can be also seen that `Bert` and `Luke` do not work together directly as there is no edge connecting the nodes reflecting `Bert` and `Luke`.

The resulting visualization of the social network based on the relation *working together* is shown on the bottom of Fig. 8.7. As expected, the relationship between `Mike` and `Sara` states that `Sara` takes over 100 % of the work from `Mike` and vice versa. There is no relationship between `Bert` and `Luke`. `Bert` takes over 100 % of his work from `Mike`, and `Mike` hands over 50 % of his work to `Bert`. The other 50 % is handed over to `Luke`.

So far, we have illustrated existing techniques for organizational mining in business processes as an example for combined analysis. The examples were rather simple and abstract. Hence, we provide some results of using the social network miner and the role hierarchy miner to the higher education process data in the following.

`HEP Use Case: Visualization of a Sociogram`

Figure 8.8 depicts the result of using the *social network miner*, *working together* (ProM 6.3) to the higher education data. Statistics show that the process was executed 74 times, producing 4,018 events and involving 80 originators. The application of social network mining to

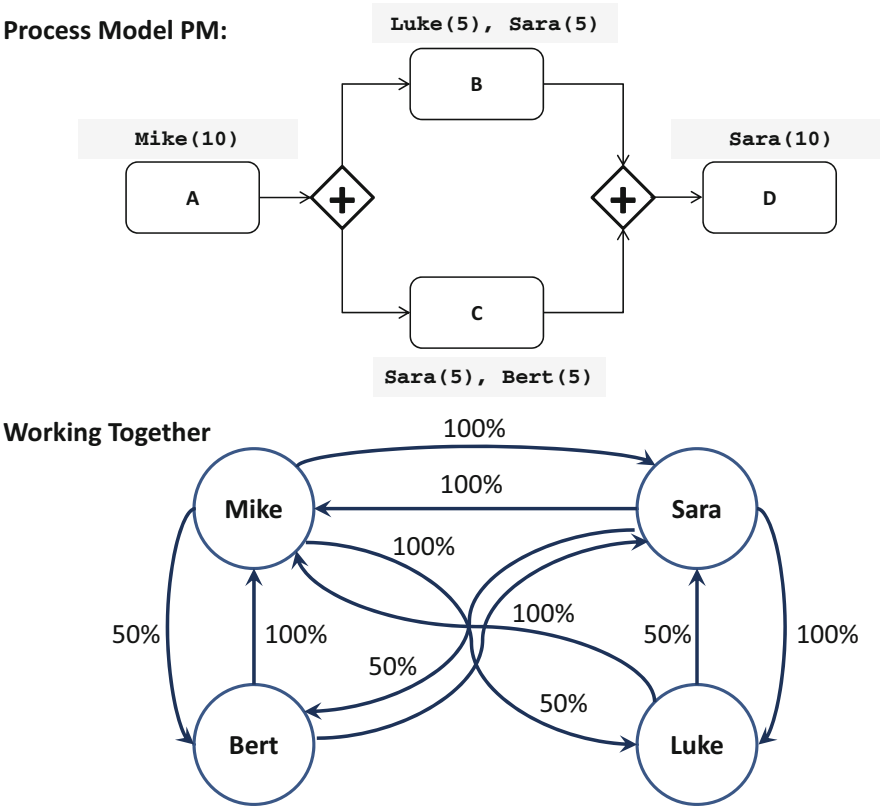

Fig. 8.7 Process model with actors and the number of task executions; result of the application of social network mining using the relationship *working together*

the log produces the result displayed in Fig. 8.8. Obviously, teaching staff including actors `'a0403563'-'tutor'`, `'a0447271'-'tutor'`, `'person003'-'lecturer'`, and `'person004'-'lecturer'` are *working together*, i.e., communicates often with the `system`, i.e., the online learning platform. Note that some of the edges have been removed as their weight does not exceed a certain threshold. Details on the HEP project can be found on the homepage of the book:

www.businessintelligence-fundamentals.com

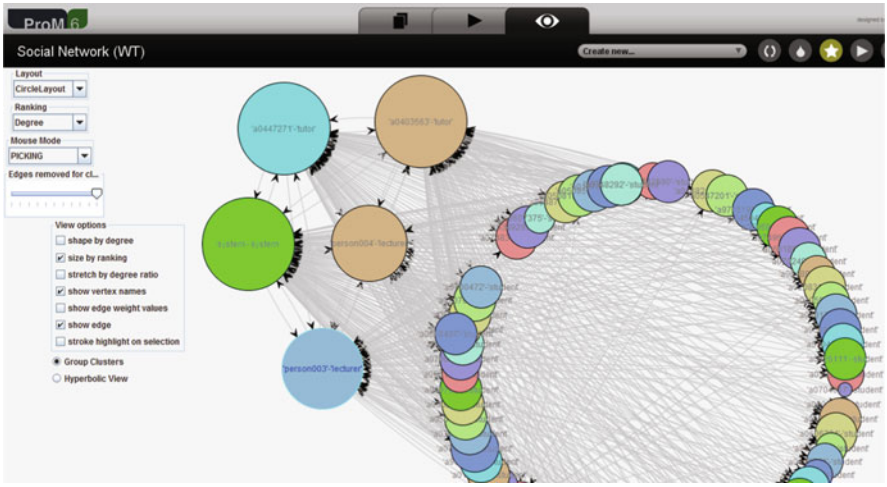

Fig. 8.8 HEP: Social network mining (using ProM 6.3)

8.2.4 Summary: Social Network Analysis and Organizational Mining

Social network analysis provides means to analyze BI perspective organization. The analysis model is a sociogram that consists of entities reflected as nodes and relations between them reflected by edges. These relations can be analyzed based on different metrics, either for the entire sociogram (e.g., density) or for single nodes (e.g., centrality).

Organizational mining aims at addressing analysis questions at the interface between BI perspective organization and production by an application of social network analysis techniques on process logs. An analysis of the organizational perspective can be very valuable for enterprises in order to, for example, detect anomalies in user behavior. Recently, the mining of role-based access control (RBAC) models over event logs for anomaly detection, policy maintenance, and policy specification errors has gained interest, not only in the business process community but also in, for example, security and RBAC applications [18, 26].

8.3 Decision Point Analysis

Business processes unfold aspects beyond the control flow that can be subject to analysis and mining such as organizational structures or process data. How to analyze and mine organizational structures from process logs have been explained in Sect. 8.2.2, aiming at analysis questions at the interface between organization and

production. In this section, we show how analysis questions at the interface between BI perspectives customer and production can be tackled by the combined analysis of process instance data (cross-sectional analysis) and process mining.

The related technique is called *decision mining* or *decision point analysis* (DPA) introduced by Rozinat and van der Aalst [31]. DPA aims at deriving the decision rules that are connected with decision points in a business process. The essence is to find rules based on the data values that have triggered the different decisions at process runtime along the paths chosen and stored in the event log.

DPA is illustrated by means of the logistics use case:

Logistics Use Case: Mining a Process Model

Consider the container transportation process as illustrated by Fig. 8.9. Obviously, this process contains a decision point, i.e., after execution process activity `Move to D`, it is decided whether the vehicle unloads at D or moves to P. We can also see that the decision depends on the value of data element `ContainerTemperature`. Based on the process model, we created 25 synthetic logs simulating different values for `ContainerTemperature`. The following log fragment displays how the values for data element `ContainerTemperature` are stored within the log. In this case, container temperature was logged with a value of 55.

```
...
<AuditTrailEntry>
   <WorkflowModelElement>Move to D</WorkflowModelElement>
   <EventType>complete</EventType>
   <Timestamp>2013-02-27T10:29:06.373+01:00</Timestamp>
   <Originator>unknown</Originator>
   <Data>
      <Attribute name="ContainerTemperature">55</Attribute>
   </Data>
</AuditTrailEntry>
```

We applied the α-algorithm (see Sect. 7.4) using ProM 5.2 to the logs. One can see that the resulting Petri net (cf. Fig. 8.10) reflects the decision point of the original model by the place between the transitions reflecting process activities `Move to D` and `Move to P` and `Unload at D`.

The question for DPA is to find the rule behind this decision, for example, "for a container temperature above 30 °C, return to P; otherwise, continue and unload at D." Details on the logistics use case can be found on the homepage of the book: www.businessintelligence-fundamentals.com

DPA works as follows: first of all, the structure of the underlying process is determined by applying process mining. If the process structure contains a decision point, this means that the set of all process execution paths can be discriminated along this decision. Decisions in business processes are mainly based on values of

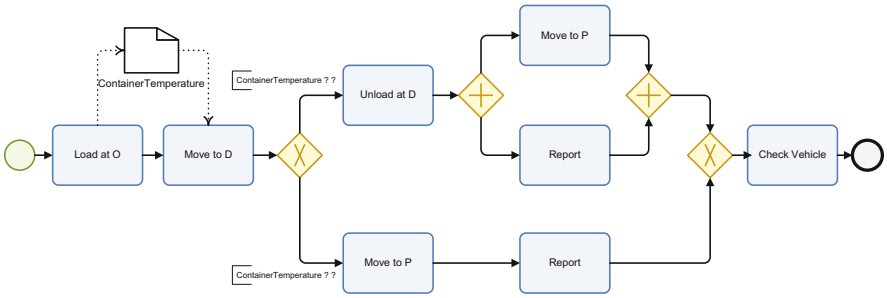

Fig. 8.9 Container transportation use case (based on [27])

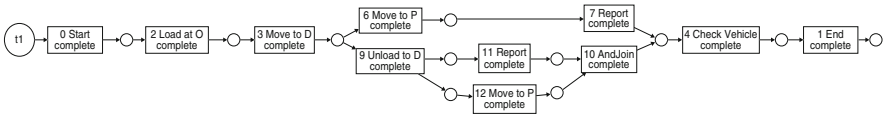

Fig. 8.10 Result of applying the α-algorithm to the container transportation example (using ProM 5.2)

process data. The analysis question is to determine how the paths can be classified based on the data.

For doing so, DPA employs decision trees (introduced in Sect. 5.3.3). More precisely, it is determined for which data values the distinction into the different execution paths becomes significant.

Logistics Use Case: Decision Point Analysis

As the first step of DPA, we applied the α-algorithm to the 25 container transportation logs resulting in the Petri net model shown in Fig. 8.10. This model contains a decision point which we want to explain. Hence, in a second step, we employ the DPA plug-in (ProM 5.2). In Fig. 8.11, we can see that the DPA plug-in determines the decision point (shaded) within the Petri net model. DPA invokes the Weka

Data mining software in order to run decision tree analysis on the process data connected with the decision point. Figure 8.12 shows the corresponding decision tree. It reflects a significant distinction of execution traces at a container temperature value below or greater or equal to 40. Finally, in Fig. 8.13, we see the integration of the decision rule into the underlying process model. In detail, for a container temperature below 40, the vehicle moves to P. Otherwise (container temperature $\geq$ 40), the vehicle unloads at D. Details on the logistics use case can be found on the homepage of the book:

www.businessintelligence-fundamentals.com

DPA offers a promising approach for combining different analysis methods, i.e., cross-sectional analysis and process mining, in a staged approach in order to address additional analysis questions. As a repetition, in the container transportation

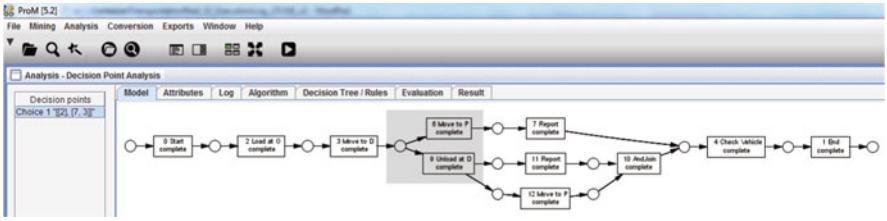

Fig. 8.11 Finding a decision point with DPA (using ProM 5.2)

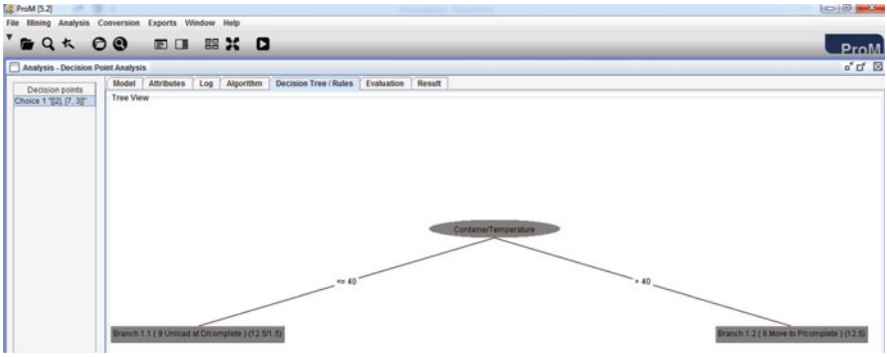

Fig. 8.12 Determining the decision tree (using ProM 5.2)

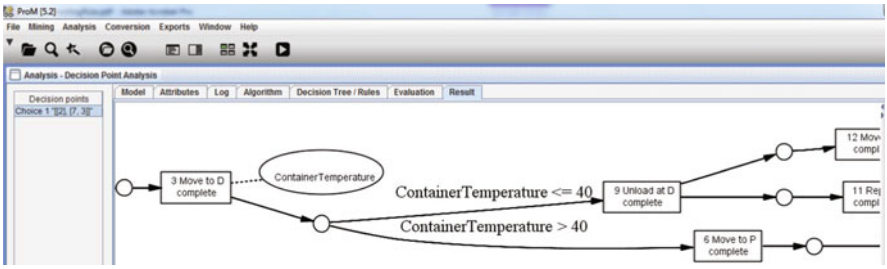

Fig. 8.13 Determining the decision rule (using ProM 5.2)

example, the container can be perceived as the customer of the process. By applying process mining, it is revealed how the container transportation works (production perspective). Applying additional cross-sectional analysis shows how the customer, i.e., the container, determines the choice of the process execution paths.

DPA can be extended in different ways. For example, de Leoni et al. [9] suggest different techniques to derive more complex decision rules containing more than one variable and possibly connected by arithmetic operators.

In [11], DPA has been also extended by considering time series process data, i.e., data that is not written once a process activity completes, but during process execution in a continuous way. An example is the continuous measurement of

container temperature during the time a vehicle is moving from its origin to its destination. The basic idea is to apply process mining and cross-sectional analysis in an iterative way. The results of the cross-sectional analysis within one iteration are captured within a complex variable. The complex variable can then be used as input for DPA. If DPA results in a significant discrimination, the complex decision rule can be unfolded from the complex variable again.

8.4 Text Mining

The Internet offers a plethora of data and applying text mining for extracting useful information from such data for business decisions has become more and more important in the last 15 years. Text mining combines ideas of different scientific areas which have worked with textual information for a long time. The most important ones are databases and information retrieval, computational linguistics, artificial intelligence, machine learning, statistics, and data mining. We will focus in this section on a text mining approach using mainly analytical techniques from data mining and statistics. Section 8.4.1 introduces a data structure frequently used in this approach towards text mining and identifies the analytical goals treated in detail in this section. Data preparation and modeling for text data is discussed in Sect. 8.4.2. Section 8.4.3 considers analysis techniques for descriptive goals and Sect. 8.4.4 analysis techniques for clustering, classification, and understanding of text data. In Sect. 8.4.5, we briefly discuss other approaches and applications, in particular opinion mining.

8.4.1 Introduction and Terminology

The basic entity for our considerations are text documents. These text documents may be of different origins and can occur in various formats. Typical sources for documents are reports, abstracts, blogs, tweets, emails, journal articles, or notes stored in a database. Corresponding to the variety of sources, the documents are stored in miscellaneous formats, for example, PDF, MS-Word, HTML, or XML. Text mining software offers readers for the different formats and converts the document into a simple text format together with some metadata. These document-specific metadata cover entries which are useful for search and retrieval of text documents and getting basic information, e.g., the author and title of the document, the content of the document, the creation date, the access rights, and how to access the documents. A standard representation of such information is a structure proposed by the Dublin Core Metadata Initiative[6] (DCMI). This standard has its

[6] http://dublincore.org/about-us/.

origin in the library science and is based on 15 tags which allow a unified description of Web resources, books, CDs, and objects like artwork. Such metadata describe documents by well-structured data and allow application of the analysis techniques presented in the previous chapters. For example, one can use basic descriptive analysis for the metadata, one can retrieve documents in the corpus according to specific properties of metadata, or one can use methods of process analysis for exploring threads in a discussion.

In the following, we will not pursue metadata approaches in detail but focus on analytical goals which look at the entire text in the documents. Moreover, in most cases, we are not interested in one single document but in analyzing a collection of documents, called a *corpus*. The following overview box lists the goals treated in this section in some detail. Besides these goals, there are a number of other goals in text mining which will be briefly reviewed in Sect. 8.4.5.

Analytical Goals in Text Mining

- *Descriptive goals:* Corpus description of the contents of the documents based on word frequencies.
- *Segmentation and understanding goals:* Find clusters of documents which are similar with respect to content and identify the topics in these clusters.
- *Classification goals:* Learning of a classification rule from already classified documents which allows the classification of a new document in one of the classes.

The model we use for the analysis is the document term matrix, which will be introduced in Sect. 8.4.2. The main emphasis in this section is on the necessary data preparation activities, taking into account the peculiarities of text data. Section 8.4.3 considers techniques for the descriptive analysis and Sect. 8.4.4 considers methods for document understanding. The demonstration of these methods is done by using a corpus defined by earlier versions of the 33 subsections of Chaps. 1 and 2 of this book.

In order to achieve the analytical goals, we follow the analysis steps described in Chap. 1, which are summarized in the text mining template.

`Template: Text Mining for a Corpus`

- **Relevant Business and Data:** A text corpus defined by a collection of text documents
- **Analytical Goals:**
  - Description of the documents in the corpus
  - Clustering the documents in the corpus
  - Finding topics of the corpus
  - Classification of documents based on rules derived from a training corpus
- **Modeling Task:** Definition of a document term matrix

- **Analysis Task:**
  - *Description of Corpus:* Determination of type-token relation and association measures; visualization of the content in the corpus using word clouds and correlation plots
  - *Clustering documents:* Use cluster analysis methods for clustering the documents
  - *Topic Models:* Define a number of topics and find the probability of assignment of the documents to the topics
  - *Classification:* Learn classification rules for assignment of new documents
- **Evaluation and Reporting Task:** Represent the results of the analysis by word clouds, by correlation plots and by characterization of the topics with terms.

8.4.2 Data Preparation and Modeling

In text mining, the basic unit for describing documents and corpora are terms. Basically, we can understand a term as a string derived from the words in a document. However, in most cases, we do not use the words as they appear in a text document but apply some transformations. These transformations are to some extent language specific, and we assume that the documents are written in English. The following transformations are offered by text mining software packages and are usually applied for the standardization of the documents.

Transformations for Standardization of Documents

- *Removal operations:* In general, numbers and punctuation do not contribute to the content of the text and are removed. Additionally, special characters like "/" or "@" are removed. In case of email documents, email address and signatures are usually removed.
- *White space and lowercase letters:* After the removal operations, it may be useful to remove extra white space produced by the removal operations. Furthermore, all words are transformed to lowercase letters.
- *Stop words:* Words that do not contribute to the intended meaning of the text are removed. Typical examples are articles, preposition, or auxiliary verbs.
- *Stemming:* words in a text are composed from a stem and affixes which mainly have syntactic meaning. Typical examples are endings like "ed" or "s." The standard procedure is to remove these affixes.

Example 8.5 (Example for Transformations) For demonstration, let us consider Sect. 1.3.5 of this book. For the description of the document in the corpus, we use the following local metadata:

```
Metadata:
  author       : Rinderle-Ma, Grossmann
  datetimestamp: 2014-09-28 08:09:19
  description  : brief task description
  heading      : 1.3.5 Evaluation and Reporting Task
  id           : 11
  language     : en
  origin       : Fundamentals of Business Intelligence
                 V1.0
```

The text described by the metadata reads as follows:

```
1.3.5 Evaluation and Reporting Task

The evaluation and reporting task looks at the analysis
results from a global business perspective and positions
the results of the analysis in the context of the
business. Its main goals are the interpretation of the
results in reference to domain knowledge and coming to
a decision of how to proceed further. Usually, the
evaluation task employs reporting techniques which are
similar to data description and visualization
techniques.Depending on the intended audience of the
report, different types of reporting can be
distinguished. We will sketch some ideas in Chapter 4.
```

Using the transformations described above, the text of the document is transformed into the subsequent:

```
[1] " evaluation reporting task"
 [2] ""
 [3] " evaluation reporting task looks analysis results
       global busi"
 [4] "perspective positions results analysis context
      busi"
 [5] " main goals interpretation results reference
       domain knowledg"
 [6] " coming decision proceed usually evaluation
       task"
 [7] "employs reporting techniques similar data
      description visu"
 [8] "techniques depending intended audience report
      different types "
 [9] "reporting can distinguished will sketch ideas
      chapter "
```

The stemmer used for stemming is a snowball version of the Porter stemmer[7] which is available in the text mining environment of R.

For further analysis, a number of decisions have to be made which may require further transformations. Let us mention the following ones:

1. *Consideration of synonyms:* Sometimes it may be useful to replace synonyms, i.e., words with the same or a similar meaning, by one term. For example, we could replace in context of BI the word "client" by the word "customer."
2. *Consideration of homonyms:* By homonyms, we understand words with the same spelling and pronunciation but with different meaning. An example in BI applications may be the term "article" which may refer to something a customer orders or to a paragraph in a business contract.
3. *Definition of specific stop words:* Depending on the intended analysis, some words may not contribute to the content. Typical candidates may be adjectives, adverbs, or written numerals.
4. *Usage of abbreviations:* In some cases, it may be useful to replace two or more words occurring together by a commonly used abbreviation. In the context of the book, an example is the abbreviation "BPMN" for "business process model and notation." On the contrary, it may be useful to expand abbreviations, for example, BI to "business intelligence."

After preprocessing, the next step is *tokenization*, i.e., breaking the documents into a list of tokens. Generally speaking, tokens are defined by *n-grams*, i.e., a sequence of n contiguous words in a text. The most simple and frequently used cases are 1-*grams*, i.e., the tokens correspond to words in the text, or *bigrams* defined by two contiguous words. Bigrams are of interest in the case of composed terms like "business intelligence." For special purposes it may be of interest to define tokens by mixtures of different n-grams, for example, words and bigrams. The different types of tokens are called *terms*.

Each term has a frequency of occurrence in a document which defines the *type-token relation*. In the case of tokens and terms defined by words, a model for the type-token relation is *Zipf's law* which states that the frequency of a term in a document is inverse proportional to its rank in the frequency table. If a document contains altogether T words of V different types and the rank of a type w in the list of types ordered by a decreasing frequency is $\text{rank}(w) = k$, then the probability of the rank is defined by a power relation, i.e.,

$$f(k) = \frac{1/k^{\alpha}}{\sum_{n=1}^{V}(1/n^{\alpha})}, \quad k = 1, 2, \ldots V. \tag{8.2}$$

Here, α is a document-specific parameter. A more theoretical exposition of this law can be found in [15]. In close relation to Zipf's law is Heaps' law, sometimes called

[7] http://snowball.tartarus.org/algorithms/porter/stemmer.html.

Herdans' law. Heap's law defines a relation between the number of types V and the number of words T in a document by a power relation. A visual check for both laws can be done by double logarithmic plots. In the case of Zipf's law, the logarithms of the frequencies of the types are plotted against the logarithms of the ranks. In the case of Heaps' law, the logarithms of the number of types V is plotted against the logarithm of the text length T.

Given M documents $D = \{d_1, d_2, \ldots, d_M\}$ containing all together V different terms (words) $W = \{w_1, w_2, \ldots, w_V\}$, we denote the term frequency of a term w_j in document d_i by t_{ij} and the $M \times V$ *document term matrix* by

$$\text{DTM} = [t_{ij}], \quad i = 1, \ldots, M, \quad j = 1, \ldots, V \tag{8.3}$$

The rows of the DTM define the type-token relation for each document and the column sums of the DTM define the type-token relation for the entire corpus. Besides the DTM, also the transposed matrix called *term document matrix* (TDM) is used. The TDM is also known as *bag of words*.

Example 8.6 (Document Term Matrix) Let us demonstrate the concepts developed up to now using the corpus containing the 33 documents defined by the sections of Chaps. 1 and 2. After the removal of punctuation, numbers, and stemming of the words, we obtained by tokenization a DTM with 2,012 terms. The most frequent term occurring 2,381 times was "the," followed by "and" and "for" with frequencies 730 and 598. After removal of stop words, there were only 1,962 terms. The most frequent terms after removal were "model" and "busi" (which is the stem for "business") and "data" with frequencies 490, 475, and 439. The visual inspection of Zipf's law is shown in Fig. 8.14 for the corpus with and without removal of stop words. This figure shows that the removal of the stop words reduces the fit to Zipf's law.

The application of a bigram tokenizer resulted in a DTM with 11,377 terms, but 87 % of the bigrams occurred only once in the corpus. Not surprisingly, the most frequent bigrams were "business intelligence," "business process," and "key performance" (indicator).

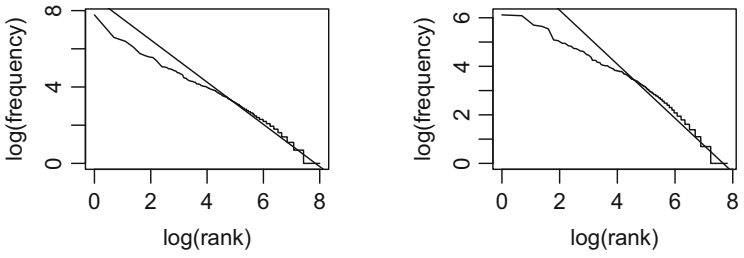

Fig. 8.14 Zipf's laws without removal of stop words (*left*) and with removal of stop words (*right*) for the example corpus (R package `tm`)

As the example shows, the DTM usually encompasses a large number of terms, and an important issue is the reduction to those terms that give important information about the documents. Such a reduction has to take into account two different reasons for frequent occurrence of terms. On the one hand, a term w_i occurs frequently in a document because it is important, i.e., gives more information about the content than a term occurring only once or twice. On the other hand, frequent terms occur only due to language usage and are not helpful for describing the contents of the documents. Typical examples are verbs or adverbs.

Depending on the analysis goal, different strategies can be applied for reducing the number of terms in the DTM. One basic strategy is defining a lower and an upper threshold for the term frequencies and remove all words which are not inside these thresholds. Another method is defining only a lower threshold and remove those frequent terms afterwards which are not important for the contents by enlarging the stop word list.

A more theoretically motivated method is using the *term frequency-inverse document frequency* (TF-IDF), in particular, if one is interested in finding terms which separate documents. Besides the term frequency t_{ij} of a term w_j in document d_i, i.e., the entries in the DTM, TF-IDF uses the document frequency DF_{ij} for the term defined by the number of documents which contain the term. Using these two quantities, TF-IDF is defined as

$$\text{TF-IDF}_{ij} = t_{ij} * \log(\text{IDF}_{ij}) = t_{ij} * \log(|D|/\text{DF}_{ij}). \tag{8.4}$$

Here, $|D|$ denotes the number of documents. This basic definition is often modified. First of all, the definition of the term frequency may be modified. For example, one can standardize the term frequencies by the length of the documents in order to reduce the importance of large documents. Another frequently used modification is to take $\log(|D|/(1 + \text{DF}_{ij}))$ which allows the application to terms which occur in none of the documents in the corpus.

TF-IDF can be understood as a weight for frequencies in the DTM which increases the importance of terms occurring in only few documents. A more theoretical analysis of TF-IDF and comparison with other weighting schemes may be found in [2]. For further aspects, we refer to Chapter 4 in [1].

The following example shows how one can apply TF-IDF for selection of terms using a summary statistic for the values of the TF-IDF for a corpus.

Example 8.7 (TF-IDF for Document Term Matrices) In the DTM for the 33 documents of Chaps. 1 and 2, there were altogether 1,962 terms after removal of stop words and stemming. An inspection of the term frequencies showed that in the list of terms with term frequencies above 20, there were altogether 186 terms. The list of words showed many words like "also," "like," "often," or "will," have no direct connection to the contents defined by BI. Hence, we computed the TF-IDF with term frequencies standardized by the size of the documents measured by the number of words in the documents. The mean was used as summary measure for all documents. Looking at the summary statistics of the average TF-IDF over the documents, we

selected terms with a value larger than the median. The reduced DTM contained only 1,007 terms. An inspection of the terms with a frequency above 10 showed that according to our understanding, all the 132 terms had a connection to the contents of the chapters and differentiated between the documents. On the other hand, the most frequent terms like "business," "model," or "data" were not in the list due to the occurrence in many documents.

Besides the data preparation techniques treated in this section, there exist also a number of other techniques which modify the DTM. A frequently proposed method is dimensionality reduction similar to the idea of principal components introduced in Sect. 4.4.3. The matrix $(\text{DTM}) \cdot (\text{DTM})^T$ is factorized according to its eigenvalues and only those dimensions of the matrix are retained which correspond to the largest eigenvalues. We will not pursue this approach and refer the interested reader to monographs on text mining, for example, [5].

8.4.3 Descriptive Analysis for the Document Term Matrix

Based on a document term matrix, different kinds of descriptive analysis can be done. The most popular is probably the visualization of the terms using a word cloud. A word cloud represents terms in the DTM with size according to the frequency in the corpus or in a document. The word cloud allows quick perception of the most frequent terms in the corpus. Besides size, colors can also be used for representing the terms. The order of the terms can be done according to the frequency with most frequent words in the center. Another option is to use a random position for the words. A further parameter allows the representation of a proportion of terms rotated by 90°. With respect to the overall layout, rectangular or circled shapes can be used. Usually one includes only terms with frequency above a certain threshold which is an additional parameter in the design of the word cloud. In the case of DTMs produced with stem terms, expansion of the stem terms to words improves readability.

Example 8.8 (Word Cloud for a Document Term Matrix) Taking the basic DTM with 1,962 terms from the 33 documents as described in Example 8.6, we decided to reduce the DTM to words above a frequency threshold of 20. The stem terms were expanded to the words with the highest frequency. However, some manual correction was necessary after applying the expansion function of R. Furthermore, a number of words had a frequency above the threshold 20 and were removed as additional stop words. Overall, 138 terms remained for the cloud. The cloud in Fig. 8.15 shows all the terms in a circled layout and with fixed order. The most frequent terms are shown in the center of the cloud.

For a comparison of the frequencies of the terms in the DTM over the documents a *comparison cloud* can be produced by the R package *wordcloud* ([28]). If the DTM contains M documents, the deviation of the relative frequencies of a term w_i

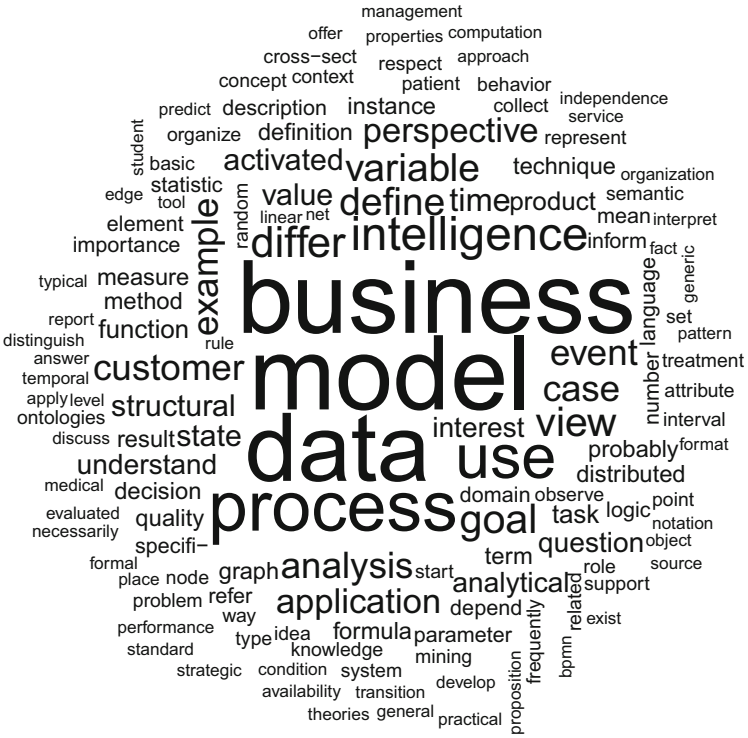

Fig. 8.15 Word cloud for frequent terms in Chaps. 1 and 2 (R package `wordcloud`)

in a document d_j from the mean relative frequency of the document is measured by

$$d_{ij} = (p_{ij} - p_j), \quad p_{ij} = t_{ij} / \sum_j t_{ij}, \quad p_j = \sum_i p_{ij} / M. \tag{8.5}$$

The comparison cloud shows the different documents in an outer circle. The angular position of the terms is defined by the document in the outer circle for which d_{ij} attains its maximum. The size of the term is chosen according to the value of the maximum deviation. Additionally, one can use colors for identification of the documents.

Example 8.9 (Comparison Cloud for a Document Term Matrix) Figure 8.16 shows the comparison cloud for the 33 documents defined by the sections of Chaps. 1 and 2. As one can see, there is a good coincidence with the contents of the sections and the position of the terms.

Besides word clouds, another descriptive technique for a DTM is the calculation of the association between the terms in the different documents. Association can be made precise in different ways, for example, as a correlation between the term

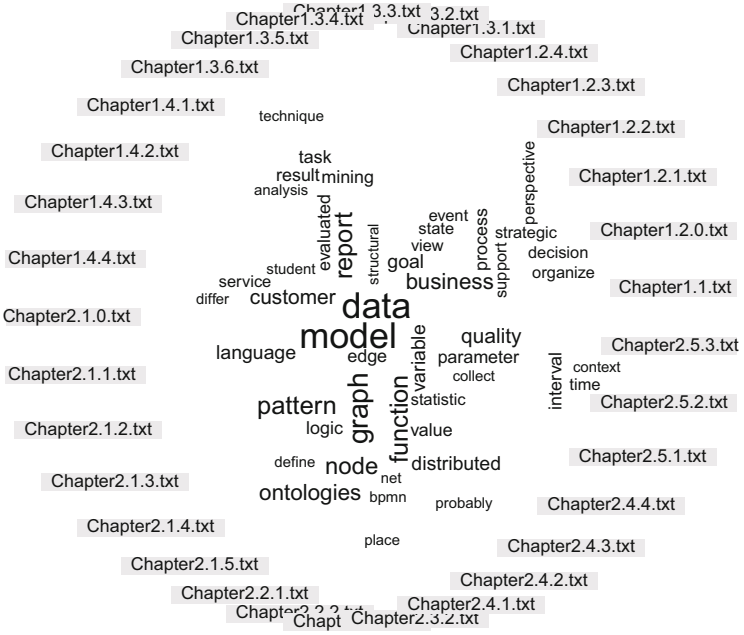

Fig. 8.16 Comparison cloud for frequent terms in Chaps. 1 and 2 (R package `wordcloud`)

frequency vectors of the different documents or as an association in the sense of association analysis in Chap. 6. In the following example, we will show an application of correlation analysis.

Example 8.10 (Association Analysis for a Document Term Matrix) Starting with the DTM of Example 8.6 we looked for terms for which the correlation with the term "business" is larger than 0.6. For example, the correlation between "business" and "intelligence" is 0.76 and between "business" and "understanding" 0.72, respectively.

Next, we restricted the DTM to the 32 terms with a frequency above the threshold 70. For these terms, the correlation matrix was calculated and a threshold for a meaningful correlation between two terms was defined by 0.6. Figure 8.17 shows the relationships between the terms with correlations above the threshold. Here, we used only the stems of the words.

8.4.4 Analysis Techniques for a Corpus

Based on the example for a corpus defined by the sections of Chaps. 1 and 2 of the book, we show in this section how the methods for clustering introduced in Chap. 5 can be applied for a corpus. For understanding the structure behind the documents

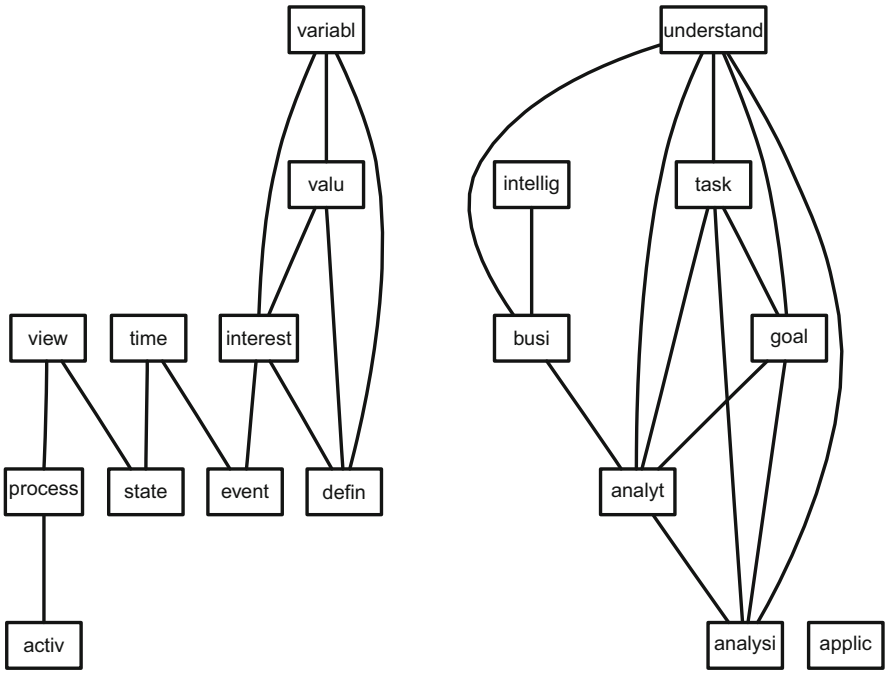

Fig. 8.17 Associations between frequent terms in Chaps. 1 and 2 (R package `tm`)

in a corpus, we introduce topic models. Finally, we discuss some peculiarities in connection with the classification of documents.

Cluster Analysis for Text Data

Cluster analysis for text documents is one of the most frequently used analysis techniques in text mining. Frequently, clustering techniques are used based on the distances between documents. An important decision for the analyst is what kind of preprocessing should be done for the DTM representing the corpus. As discussed in Sect. 8.4.2, it may be useful to use a selection based on TF-IDF instead of the entire DTM.

For finding the similarity or distance between the documents in a corpus, the cosine function is the first choice. If d_i and d_j are two documents in the corpus with term frequency vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ defined by the rows of the DTM, the similarity is defined by

$$\text{sim}(d_i, d_j) = \frac{\mathbf{v_i} \cdot \mathbf{v_j}'}{||\mathbf{v_i}|| \cdot ||\mathbf{v_j}||}. \tag{8.6}$$

As usual "$\cdot$" defines the inner product of vectors and $||.||$ denotes the norm of the vector.

After the definition of the distance, one can use hierarchical cluster methods or k-means clustering. Sometimes, partitioning around medoids (PAM, Sect. 5.4.3) is proposed. A nice property of PAM is that the center of the clusters are elements of the data. There are two main disadvantages of PAM. The first one is that the computation is rather slow in the case of large data, and the second one is that the representation of the clusters by the medoid may miss important terms in the case of a sparse DTM.

Example 8.11 (Clustering a DTM) Using the same DTM as in the case of demonstration of the word cloud in Example 8.8, we used hierarchical clustering with the Ward method. The solutions at different levels can be explained quite well. In the case of two clusters, which show a high level of separation, one cluster with 14 documents can be interpreted as a cluster with documents treating general aspects of modeling, logical and algebraic structures, and analytical structures. The second cluster with 19 elements mainly contains the documents from Chap. 1 dealing with data and general aspects of BI. A grouping into three clusters splits the analytical and modeling cluster into one cluster containing documents about general aspects of modeling and one cluster with documents about analysis techniques. Further splits from the cluster containing the 19 documents about data and general aspects of modeling separate two small clusters at almost the same level. The two new clusters can be labeled by the terms "data cluster" (the three documents of Sects. 1.3.1, 2.5.1, and 2.5.3) and "general considerations cluster" (Sects. 1.1 and 1.2.2 and the introduction of Chap. 2).

Besides these basic approaches, many other techniques have been proposed. For example, instead of the cosine function, one can use string kernels for definition of the distances (cf. [12] for an application using R). Another technique for clustering documents is *co-clustering*, which aims at simultaneous clustering of the documents and the terms. Clustering of the documents and the terms is done in such a way that there is an optimal fit between the two cluster solutions. One method for the definition of this fit is the interpretation of the DTM as a bipartite graph, where the documents and the terms define the two sets of nodes and the entries in the DTM define the edges. The algorithm introduced in [10] defines a partition in the two node sets which minimizes the edges between different partitions in the two clusters.

Topic Models

The idea behind topic maps is similar to model-based clustering in Sect. 5.4.4. We assume that a document d_i in the corpus $D = \{d_1, d_2, \ldots, d_M\}$ is composed from different topics $t_1, t_2, \ldots, t_K$ with probabilities for topic t_k given by $p(t_k|d_i)$. For each topic t_j, the occurrence of a word w_ℓ from a set of words $W = \{w_1, w_2, \ldots, w_N\}$ follows a topic-specific distribution defining the probability $p(w_\ell|t_k)$. Using Bayes' theorem, we can represent the probability of the word in the document by

$$p(w_\ell|d_i) = \sum_{k=1}^{K} p(w_\ell|t_k)\, p(t_k|d_i). \tag{8.7}$$

Based on this model, one can understand the occurrence of words in a document as the result of the following process (cf. [6]):

Data Generation in Topic Models

1. Choose randomly for a document a distribution over the topics.
2. For each word in the document.
   a. Choose randomly a topic according to the distribution of the topics in the document.
   b. Choose randomly a word according to the distribution of the words in the topic.

The estimation task is now to find for a given DTM the number of topics, the distribution of the topics for each document, and the distribution of words within the topics.

Given the number of topics, two solutions are frequently used. The first one is known as *probabilistic semantic index (PLSI)*. This method uses the model specification defined in Eq. (8.7) and estimates the topic probabilities and the word probabilities by maximum likelihood estimation in an iterative way. An outline of the algorithm can be found in [1, Chapter 4].

The second method nowadays most frequently used is *latent Dirichlet allocation (LDA)* which reformulates the problem in a Bayesian framework. The priors for the topics and the words are defined by Dirichlet distributions (cf. Sect. 6.4.1 for the definition of the Dirichlet distribution). The estimation of the parameters is done by a special version of the EM algorithm (cf., Sect. 5.4.4). The results of this algorithm are posterior distributions for the topics in each document and for the terms within the topics. A detailed description of the algorithmic solution can be found, for example, in [14]. For the estimation of the number of topics, one can use the method of cross-validation. An alternative is proposed in [35] and implemented in the R package `maptpx`.

Example 8.12 (Topic Models for the Sample Corpus) Using the DTM of the corpus of the sections of Chaps. 1 and 2, we first reduced the DTM according to TF-IDF as described in Example 8.7 and fitted topic models from two to five. The model with two topics corresponds, not surprisingly, to Chaps. 1 and 2. A model with five topics showed that for most of the sections, there is one dominant topic. For example, topic 4 dominates in Sect. 1.2.4 (Goals of BI), 1.3.2 (Business and Data Understanding Task), 2.3.1 and 2.3.2 (Graph Structures), and 2.4.4 (Modeling Methods Using Analytical Structures). Correspondingly the five most important terms for this topic are "graph," "goal," "node," "variable," and "transition." An example for a document which is a mixture of two topics is Sect. 2.1.3 about model building, which is mainly a mixture of topic 2 (0.634) and topic 3(0.364). A detailed description of the results can be found on the homepage of the book.

Classification of Documents

A straightforward solution for the classification of documents is the application of the algorithms of Chap. 5 to the DTM. Practically, all different methods for classification have been used in document classification. For an overview, see [1], for example.

However, many times the DTM of the training data is a sparse matrix, and it is advisable to reduce the number of terms. Besides the general methods for term reduction discussed in Sect. 8.4.2, techniques that use the information about the class membership in the training data have been proposed. The basic idea is to find terms with high discriminatory power and apply the classification algorithms to the modified DTM. A well-known example is spam filtering for emails. In this case, it is useful to focus on terms which occur frequently in spam mails, for example, emails promising high winnings. Identification of such terms can be based on different methods. For example, one can use measures of the information gain for the selection of terms which have high discriminatory power in the training data. This method is similar to the selection criteria for variables in decision trees (cf. Sect. 5.3.3) and is described in detail in [1, Chapter 6]. Other methods for term selection are frequently specific to the problem. Methods for the selection of terms in connection with opinion mining will be discussed in Sect. 8.4.5.

As soon as the terms of interest have been identified, one can use either a DTM with this terms or the incidence matrix corresponding to the DTM, i.e., we consider only the occurrence of the term.

8.4.5 Further Aspects of Text Mining

Up to now, we have considered analytical goals in text mining at the level of a corpus based on the DTM. Besides these goals, a number of other goals can be formulated which are not confined to a corpus of documents but refer to different types of text data like words, sentences, single documents, or linked documents (cf. [25] for a systematic overview of text mining goals). Achieving these goals requires additional analytical techniques; however, for evaluation purposes, the reference to a corpus is often necessary. In the following, we will briefly discuss some frequently used techniques and then show how different techniques can be applied in opinion mining.

Analysis at the Word Level

Natural languages have a high expressive power and allow the denotation of similar concepts with different words, called synonyms. Moreover, nouns representing a concept are usually embedded into a hierarchy of other concepts. Terms representing a broader concept are called *hypernyms*, and terms representing a narrower concept are called *hyponyms*. Other possible relations between concepts is the "part of" relation. In the case of verbs, it is also possible to express a special way of doing an activity (*troponyms*). In the case of adjectives, one can define *antonyms*, which are

adjectives with opposite meaning. Additionally, one can define relations between different types of words (e.g., nouns and verbs).

For usage of such information in text mining, a database is needed which keeps lexical information and semantic information. For English words, WordNet[8] offers such a database which is freely available and can be downloaded from the website. R offers access to this database in the library `wordnet`. Similar projects exist for other languages but are in general not free of charge.

Example 8.13 (Relation Between Terms Using WordNet) Looking for synonyms in WordNet for the word "business" provides altogether 16 words which are grouped into 9 senses. The most frequent sense is given by the synonyms "concern," "business concern," "business organization," and "business organisation." Not so frequently used are senses of business with synonyms "occupation," "job," "line of work," or "line." As senses without synonyms, the interpretation of business as a volume of commercial activity ("business is good today") or concern ("mind your own business") is listed. The different senses together with the synonyms are called the *synset* of the word.

A hierarchy of hypernyms for business in the first sense is defined by the terms "enterprise," "organization," "social group," "group," "abstraction," "abstract entity," and "entity." Hyponyms are according to WordNet terms like "agency," "brokerage," "carrier," "chain," "firm," "franchise," and others. For the sense of the term "business" as a "commercial enterprise," a part of relation is defined between the terms "business" and "market place."

If one looks for the synset of the word "model," one can find sysnsets for the noun "model," as well as the verb "model,". In both cases, different senses are possible. In this case, the relation between the noun and the verb is also indicated.

Looking for the term "busy," the word is identified either as verb or as adjective. In case of the adjective, the most important sense is related to active. Correspondingly, the antonym is given by the word "idle."

Analysis at the Sentence Level

Text mining at the sentence level is mainly concerned with understanding the structure of a sentence, i.e., identification of the position and the role of the words in a sentence. This identification is known as *part-of-speech tagging (POS tagging)*. Part-of-speech tagging identifies the role of the words in a sentence using tags as standard descriptors. A frequently used set of tags are the Penn Treebank tags.[9] Finding tags in a sentence has to resolve disambiguation, and different machine learning algorithms have been proposed. There are many open-source taggers for different languages available; for example, the text mining package of R offers a tagger of the Apache OpenNLP library.[10]

[8] http://wordnet.princeton.edu/.

[9] http://www.cis.upenn.edu/~treebank/.

[10] https://opennlp.apache.org/.

Example 8.14 (Part-of-Speech Tagging) Applying part-of-speech tagging to the sentence:

The evaluation and reporting task looks at the analysis results from a global business perspective.

leads to the following structure

```
{(TOP
  (S
    (NP (DT The) (NN evaluation) (CC and)
                 (NN reporting) (NN task))
    (VP (VBZ looks)
    (PP (IN at) (NP (DT the) (NN analysis)
                 (NNS results)))
    (PP (IN from) (NP (DT a) (JJ global)
                 (NN business) (NN perspective))))}
```

The sentence starts with the tag `S` and contains a noun phrase `NP`, a verb phrase `VP`, and two prepositional phrases `PP`. Within the phrases, the words are identified by tags. For example, `NN` stands for noun singular and `NNS` for nouns in plural, `VBZ` is a verb, 3rd person singular present, `IN` indicates prepositions, `DT` determiners, `JJ` adjectives, and `CC` coordinating conjunction.

Keyword Extraction

The extraction of keywords is of utmost importance in indexing Web documents, and many different techniques have been proposed. Most analytical techniques for the automatic extraction of keywords from documents combine text mining methods described in this section. The precise formulation of the analytical task depends on the available information. We will outline here an approach which is used, for example, in the open-source tools *keyword extraction algorithm* (KEA)[11] and the further development *multipurpose automatic topic indexing* (MAUI).[12] Another tool using his approach is RAKE which is described in detail in [5].

Usually, the first step in keyword extraction is the selection of candidate terms based on statistical measures for the terms in the document. These candidates are found by tokenization using n-grams, in most cases up to three words, and the most frequent terms are used. If there is a thesaurus of controlled keywords available, these candidates can be matched against the vocabulary in the thesaurus. Based on these candidates, a number of features for the document can be calculated. If one is interested in finding keywords for documents in a corpus, a basic feature is the TF-IDF (cf. Sect. 8.4.2). Other features of interest can be based on part-of-speech tagging (POS) described above, on the occurrence of the word in the text (usually a word at the beginning of the text is more likely to be a keyword than a word at the end of a text), or on the semantic relatedness of a term to words in a thesaurus. Using

[11] http://www.nzdl.org/Kea/.

[12] https://code.google.com/p/maui-indexer/.

these features, a scoring of the candidates for the keywords is done. In the case of training data these scores can be calculated using a supervised learning approach. If no training data are available, heuristic methods can be applied. Afterwards, the keywords are defined by the terms with the highest scores.

Opinion Mining and Sentiment Analysis

Understanding how customers perceive products or services is a core topic in marketing. The traditional methods used for learning about the opinion of customers is market and opinion research. However, nowadays, one can find customer opinions about products and services at e-business portals, opinions about politicians and political decisions in blogs, tweets and fora, or customer reviews of movies or theater performances at specific portals. Hence, it is not surprising that there is great interest in understanding and interpreting these customer opinions on the Internet using text mining methods.

One can find a lot of research activities and applications in this area under the headings *opinion mining*, *sentiment analysis*, or *subjectivity analysis*. According to [29], subjectivity analysis is the oldest and broadest term subsuming the other two. Subjectivity is defined in this paper as *linguistic expressions of a persons' private states like opinions, evaluations, or emotions which cannot be completely verified by objective criteria*. This lack of objectivity discerns opinionated information from factual information. In combination with the linguistic diversity in the expression of sentiments and opinions, the analysis of subjectivity in text documents creates a number of challenging problems for text mining.

Consequently, many different approaches have been proposed and a rich set of techniques is available based on data mining methods described in the previous sections, methods from artificial intelligence, computational linguistics, or psychology. For an excellent summary of the problems, together with an extensive bibliography we refer to [29]. In the following we will describe some basic tasks shown in the overview box. These tasks are elaborated in more detail in [21] and start with a problem formulation for sentiment analysis which resembles some concepts of market and opinion research and points to specificities of opinion mining.

Tasks in Opinion Mining

- Finding in a document all opinionated sentences
- Identification of the objects and the features about which an opinion is expressed
- Classification of the expressed opinion

The basic unit for opinion mining is a document in which an opinion holder expresses her/his opinion about certain features of an object of interest. This opinion is given at a certain time and has an orientation, in the simplest case positive, negative, or neutral. Note the similarity of the set-up with a traditional survey: The document plays a role similar to a questionnaire, the opinion holder corresponds to a surveyed person, the features correlate to the questions about the product, and the

orientation of the opinion replaces the answers to the questions. However, making this similarity operational for analysis by machines causes a number of challenging problems.

Finding Opinionated Sentences

Opinionated sentences occur in two guises: either a sentence expresses an opinion explicitly, for example, a sentence like "The performance of Sandra Bullock in this movie was exceptional" or in comparative form, for example, a sentence of the form "In this movie the performance of Sandra Bullock was much better than in her previous film."

As the first example shows, explicit opinions are usually expressed by adjectives, e.g., "good," "beautiful," "poor," or "awful." Another way for expressing opinions is using certain verbs as "like" or "hate," adverbs and nouns, for example, "remarkable" and "junk," or opinion phrases like "deserves attention." Opinions expressed in a comparative way many a time use adjectives in comparative or superlative form. Using part-of-speech tagging allows the identification of such phrases.

Identification of Objects and Features

Frequently, identification of the objects and the features about which the opinion is expressed in a document starts with the assumption that there is only one object of interest in the entire document. Such an assumption is justified in the case of review documents about products or documents expressing the opinion about persons. The identification of the object is often done according to the context of the creation of a document, e.g., a product review on a business portal. Other methods for the identification of the objects are using keyword matching of documents using the name of a person or the name of a brand.

The identification of the features of interest is more complicated because one has to take into account the linguistic diversity for naming the features. For example, for the evaluation of a movie, one can use features such as script, director, music, and the performance of the actors. For each feature, different terms can be used. Take as an example the performance of an actress which can be described using phrases like "she acts," "she plays," or "she represents." For the resolution of this diversity, one can use semantically oriented databases such as WordNet.

Opinion Classification

The opinion expressed in a sentence is called the *polarity* of the sentence. Finding the polarity of a short sentence with only one object of interest can be rather simple by using the polarity of words in the sentence. This polarity of words can be accessed by comparison with a dictionary with polarity tags. Such dictionaries exists, for example, SentiWordNet which is an extension of WordNet including the polarity of words (cf. [3]).

Using a list of positive and negative terms for the English language,[13] one can assign polarity in the case of direct expression of the polarity simply by matching the words with such a list. Such an application has got rather popular in the last years for analyzing tweets. A tutorial for explaining this approach using R for the evaluation of airlines can be found in [7].

The evaluation of complex sentences needs a more complex representation of words and sentences. On the one hand, the syntax of the sentences has to be analyzed, for example, negation has to be identified. Another aspect is finding the overall opinion in the case of differing opinions about different features of the object of interest. The opinions about features can be summarized in scores or can be represented individually using techniques like radar plots introduced in Sect. 4.5.1. Another aspect which hampers opinion classification is the identification of ironic statements.

An interesting project in this area is *SenticNet* described in detail in [8]. The approach was developed in the spirit of artificial intelligence and aims at modeling common sense in sentences by looking more explicitly at psychological aspects in expressions and the recognition of emotions. To achieve this goal, a *bag of concepts* model is used instead of the bag of words model described in Sect. 8.4.2. Using this bag of concepts, a number of aspects are distilled which correspond to the features of the object which is opinionated by the sentence. These aspects are evaluated with respect to the polarity in the sentence and mapped into an affective space, and an overall opinion score is calculated. A prerequisite for application of SenticNet is the assumption that the sentence is an opinionated sentence and not a factual statement.

Example 8.15 (SenticNet Example) Suppose in a movie review occurs the following summary:

The script is stupid but the performance of the actors is excellent

Using SenticNet,[14] the following concepts are derived:

"script," "stupid_script," "performance," "performance_of_actor," "actor," and "excellent_performance."

The aspects extracted are:

"performance," "actors," and "script."

The overall sentence polarity is evaluated with *0.5/null*.

If one replaces the word "stupid" by "illogical," the polarity of the sentence

The script is illogical but the performance of the actors is excellent

changes to *positive*.

As in other applications of text mining, the development of knowledge which allows the classification of opinions is based on the analysis of existing text corpora. Similar to the case of keyword extraction, the first step in the analysis is the definition of features of the document which are used afterwards in techniques for

[13] A list of positive and negative words can be found in http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon.

[14] http://sentic.net/demo/.

supervised and unsupervised learning for text documents. The tuning of the methods often depends on the application domain. Details can be found in [22].

In the evaluation of results of opinion mining, two effects have to be taken into account. The first one is spam opinion, i.e., persons post a faked opinion about a product. For example, on tourist portals, an accommodation may post faked positive evaluations or on a platform about politicians, negative statements may be posted about a candidate. Identifying such deceptive opinions has become a special kind of text mining in the last years. In [20, 27] one can find applications of supervised learning for the identification of a deceptive mail. A second problem is a phenomenon similar to *publication bias* in *meta-analysis*. Meta-analysis aims at an improvement of research results by combining the findings of different publications, particularly in medical research. Publication bias occurs due to the fact that many a time insignificant research results are not published. A similar effect can occur in published opinions because we do not know what motivates a person to post an opinion on the Internet. For example, persons satisfied with a service will not post their opinion, whereas people with negative experience may be interested to post their opinion.

8.4.6 Summary: Text Mining

Text mining combines ideas from different scientific disciplines and has got more and more importance in the last 15 years. This section focused on a text mining approach based on the document term matrix for a corpus. The definition of the document term matrix needs a number of data preparation steps like removal of special characters, removal of stop words, or stemming. Next, one has to decide which method is used for tokenization and how the different terms are weighted. An important method for weighting is the calculation of the so-called *term frequency-inverse document frequency* matrix.

Descriptive analysis of the document term matrix allows the representation of the documents as word clouds and the comparison of documents can be done by a comparison cloud. For understanding the structure of a corpus, one can apply various cluster algorithms. A more theoretically oriented method for analyzing the corpus structure are topic models.

Finally, the basics of text mining at the word level and at the sentence level were defined, and the application of mixed methods for keyword extraction and for opinion mining and sentiment analysis was demonstrated.

8.5 Conclusion and Lessons Learned

In the following, we comment on existing methods to evaluate and assess the analysis methods presented in this chapter.

For social network analysis, the basic question is what should the results be compared to in order to evaluate them. If the task is to classify in a social network the type of the relation between actors using a number of attributes, the quality of the classification can be measured using metrics such as precision and recall (an introduction to these metrics can be found in Sect. 5.3.1). This was applied in a study for extracting the relations between researchers from Web data [24]. This paper also investigates questions of finding networks with similar affiliation of researchers and compares the results with networks obtained from questionnaires.

Mining organizational models from an event log is often done in the presence of a so-called *prescriptive* organizational model. Prescriptive organizational models set out how the organizational structure of the company looks like and how the access of actors to process activities is managed. However, as mentioned in several studies and works, e.g., [18, 19, 23, 26], actors might deviate from the prescriptive model due to various reasons such as delegation or unavailability. Thus, the results from organizational mining will deviate from the prescriptive models. Of course, one could apply metrics, such as precision and recall, to compare the results of organizational mining and prescriptive models. However, the conclusion would not be very meaningful. In this case, it is more interesting to find differences in order to pinpoint and analyze deviations. This can be achieved by applying *delta analysis* between the prescriptive and actual organizational model [18].

To the best of our knowledge, there is no evaluation or assessment method that has been proposed for decision point analysis (DPA). DPA combines two analysis methods, i.e., process mining and cross-sectional analysis (more precisely, decision trees). Evaluation techniques exist for both methods. Discovered process models can be evaluated based on conformance checking (see Sect. 7.6) and related quality metrics fitness, precision, generalization, and structure.

In the case of text mining, we have shown how one can apply the methods of supervised an unsupervised learning to text data. The main challenge in text mining is data preparation and the definition of a model which transforms unstructured text into a structured model. The focus in this section was on models based on the document term matrix DTM. The DTM is basic for analyses of a corpus of text documents. Depending on the analytical goal, a number of additional features of texts have to be considered. These features are often derived by methods of computational linguistics. An important resource for many text mining applications are databases, such as WordNet, and algorithms for natural language processing such as part-of-speech (POS) tagging. These tools are of utmost importance in the case of opinion mining and sentiment analysis.

With respect to the evaluation of text mining results, the evaluation criteria are basically those defined in Chap. 5 for supervised and unsupervised learning. In the case of opinion mining, an additional benchmark is defined by the human perception of expressed opinions. This perception is usually not as unique as for factual information (cf. [29]). An additional problem in opinion mining is the relevance and the validity of the results. Even if the problems of analysis of opinion by machines are solved successfully, the interpretation of the results may be rather tricky. Contrary to traditional market research, secondary data are frequently used

and it is by no means evident that the data correctly represent the target group. Finally, the effects of spam opinion and the bias of published opinions have to be taken into account.

8.6 Recommended Reading

The reference work for organizational mining is Song (2008). For text mining, we recommend from a more practical point of view Berry (2010). A more theoretical exposition of text mining can be found in Aggarwal (2012). The topic of opinion mining is treated from a more practical point of view in Pang (2008) and Liu (2010).

- Aggarwal CC, Zhai C (2012) Mining text data, Springer
- Berry MW, Kogan J (2010) Text mining: applications and theory. Wiley Online Library
- Liu B, Zhang L (2012) A survey of opinion mining and sentiment analysis. In: Aggarwal CC, Zhai C (eds) Mining Text Data, Springer, pp. 415–463
- Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2(1–2):1–135
- Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. Decision Support Systems 46(1):300–317

References

1. Aggarwal CC, Zhai C (2012) Mining text data. Springer, New York
2. Aizawa A (2003) An information-theoretic perspective of tfidf measures. Inf Process Manag 39:45–65
3. Baccianella S, Esuli A, Sebastiani F (2010) SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. LREC 10:2200–2204
4. Batagelj V, Mrvar A (2004) Pajek—analysis and visualization of large networks. Springer, New York
5. Berry MW, Kogan J (2010) Text mining: applications and theory. Wiley Online Library, Chichester
6. Blei DM (2012) Introduction to probabilistic topic models. Commun ACM 55(4):77–84
7. Breen J (2011) R by example: mining Twitter for consumer attitudes towards airlines. https://jeffreybreen.wordpress.com/2011/07/04/twitter-text-mining-r-slides/
8. Cambria E, Hussein A (2012) Sentic computing. Springer, Dordrecht
9. de Leoni M, Dumas M, García-Bañuelos L (2013) Discovering branching conditions from business process execution logs. In: Cortellessa V, Varró, D (eds) FASE'13: 16th int'l conference on fundamental approaches to software engineering. Lecture notes in computer science, vol 7793. Springer, Heidelberg, pp 114–129
10. Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Lee D, Schkolnick M, Provost FJ, Srikant R (eds) ACM SIGKDD'01: International conference on knowledge discovery and data mining. ACM, New York, pp 269–274

11. Dunkl R, Rinderle-Ma S, Grossmann W, Fröschl KA (2014) Decision point analysis of time series data in process-aware information systems. In Nurcan S, Pimenidis E, Pastor O, Vassiliou Y (eds) CaISE forum: joint proceedings of the CAiSE 2014 forum and CAiSE 2014 doctoral consortium, CEUR workshop proceedings 1164, CEUR-WS.org, pp 33–40
12. Feinerer I, Hornik K, Meyer D (2008) Text mining infrastructure in R. J Stat Softw 25(5):1–54
13. Fruchterman TMJ, Reingold EM (1991) Graph drawing by force-directed placement. Software 21(11):1129–1164
14. Grün B, Hornik K (2011) topicmodels: an R package for fitting topic models. J Stat Softw 40(13):1–30
15. Johnson NL, Kotz S, Kemp AW (1992) Univariate discrete distributions, 2nd edn. Wiley, New York
16. Jungnickel D (1994) Graphen, Netzwerke und Algorithmen, 3rd edn. BI-Wissenschaftsverlag (in German)
17. Kamada T, Kawai S (1989) An algorithm for drawing general undirected graphs. Inf Process Lett 31(1):7–15
18. Leitner M, Rinderle-Ma S (2014) Anomaly detection and visualization in generative RBAC models. In: Osborn SL, Tripunitara MV, Molloy I (eds) SACMAT'14: ACM symposium on access control models and technologies. ACM, New York, pp 41–52
19. Leitner M, Baumgrass A, Schefer-Wenzl S, Rinderle-Ma S, Strembeck M (2013) A case study on the suitability of process mining to produce current-state RBAC models. In: La Rosa M, Soffer P (eds) Business process management workshops. Lecture notes in business information processing, vol 132. Springer, Heidelberg, pp 719–724
20. Li J, Ott M, Cardie C, Hovy E (2014) Towards a general rule for identifying deceptive opinion spam. In: ACL'14: annual meeting of the association for computational linguistics. The Association for Computer Linguistics, Stroudsberg, Pa, pp 1566–1576
21. Liu B (2010) Sentiment analysis and subjectivity. In: Indurkhya N, Damerau FJ (eds) Handbook of natural language processing. Chapman&Hall/CRC, Boca Raton, pp 627–666
22. Liu B (2012) Sentiment analysis and opinion mining (synthesis lectures on human language technologies). Morgan & Claypool Publishers, San Rafael
23. Ly LT, Rinderle S, Dadam P, Reichert M (2005) Mining staff assignment rules from event-based data. In: Bussler C, Haller A (eds) Business process management workshops. Lecture notes in computer science, vol 3812. Springer, Heidelberg, pp 177–190
24. Matsuo Y, Mori J, Hamasaki M, Nishimura T, Takeda H, Hasida K, Ishizuka M (2007) POLYPHONET: an advanced social network extraction system from the web. Web Semantics 5(4):262–278
25. Miner G, Delen D, Elder J, Fast A, Hill T, and Nisbet R (2012) Practical text mining and statistical analysis for non-structured text data applications. Elsevier, Waltham, Ma
26. Molloy I, Park Y, Chari S (2012) Generative models for access control policies: applications to role mining over logs with attribution. In: Atluri V, Vaidya J, Kern A, Kantarcioglu M (eds) SACMAT'12: ACM symposium on access control models and technologies. ACM, New York, pp 45–56
27. Ott M, Cardie C, Hancock JT (2013) Negative deceptive oponion spam proceedings of NAACL-HLT 2013, pp 497–501
28. Package 'wordcloud' http://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf. Accessed 12 December 2014
29. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
30. Rinderle S, Reichert M (2007) A formal framework for adaptive access control models. J Data Semant IX:82–112
31. Rozinat A, van der Aalst WMP (2006) Decision mining in ProM. In: Dustdar S, Fiadeiro JL, Sheth AP (eds) BPM'06: International conference on business process management. Lecture notes in computer science, vol 4102. Springer, Heidelberg, pp 420–425
32. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47

33. Scott J (2013) Social network analysis. SAGE, London
34. Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. Decis Support Syst 46(1):300–317
35. Taddy MA (2012) On estimation and selection of topic models (2012). In: Proceedings of 15th international conference on artificial intelligence and statistics
36. Wainer J, Barthelmess P, Kumar A (2003) W-RBAC—a workflow security model incorporating controlled overriding of constraints. Int J Coop Inf Syst 12(4):455–485
37. Wasserman S (1004) Social network analysis: methods and applications, vol 8. Cambridge University Press, Cambridge
38. Wellman B (1004) Are personal communities local? A Dumptarian reconsideration. Soc Netw 18(4):347–354