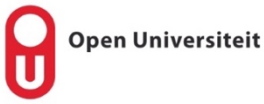
MASTER'S THESIS

Mining for workarounds in text fields with clustering algorithms

Spronk, J. (Janneke)

Award date:
2020

Mining for workarounds in text fields with clustering algorithms



Degree programme:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's programme
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	Janneke Spronk
Identification number:	
Date:	25-01-2020
Thesis supervisor	Dr. Lloyd Rutledge
Second reader	Dr. Guy Janssens
Version number:	1
Status:	Final

Abstract

This study describes how text mining and clustering algorithms can detect workarounds in a database. Finding workarounds can reduce risk and improve IT system resilience by guiding subsequent system improvement. Workarounds are methods used by employees that are not intended by the IT department. Analyzing workarounds provides important information about opportunities and threats in business processes and IT systems. Organizations base their choices of improvement on the intended use of IT, and do not include workarounds in their argumentation for change. With knowledge of workarounds, better choices on IT improvements can be made and organization resilience is improved. Therefore, a Resilience Mining Thesis Circle started to explore the possibilities of using algorithms to find workarounds in a database. This study is part of this Thesis Circle, and focusses on clustering algorithms to detect workarounds in text fields in IT systems. The research is conducted on data of the department of Collective Pensions of Achmea, a large insurance company in the Netherlands. We find that k-Means (kernel) and Agglomerative clustering are useful algorithms that can detect workarounds in text fields. Combined with the findings of the other members of the Resilience Mining Thesis Circle, we provide an overview of the possibilities for using algorithms to detect workarounds in IT systems.

Key terms

Workarounds, database, IT systems, algorithm, clustering, text mining, k-Means, DBSCAN, k-Medoids, Agglomerative clustering

Contents

1. Introduction

In many organizations, people adopt working methods that are not intended by the IT department, called workarounds (Alter, 2014). Workarounds exist in data and text, and they can be detected in data, text or logfiles. Insight in the use of workarounds provides information about opportunities and threats in IT systems (Silic & Back, 2014). This study aims to find a method to detect workarounds in text fields with clustering algorithms, to contribute to IT system resilience. In this chapter you will read an introduction to the theoretical background, the formulation of the research question and a brief introduction on the research methods.

1.1. Exploration of the topic

Workarounds are goal-driven changes to the intended use of an IT-system to overcome constraints that prevent the user from achieving the desired goal (Alter, 2014). There are two main types of workarounds concerning text fields. The first type is the use of text fields for other purposes than they are meant for. For example, text fields can be used to communicate with another party, when this field is not meant for this purpose (Patterson, 2018). The second type of workaround is the use of copy paste. This type is found when large amounts of text need to be filled that are similar to other instances, or when the same text is used very often (Huuskonen & Vakkari, 2012).

The use of workarounds in text fields can pose a threat to an organization. Compliance issues, risk of data loss and wasted investments are just a few of the identified risks (Silic & Back, 2014). When text fields are used for other purposes, there is a risk that not all parties are aware of the information the field contains. Besides these threats, the use of workarounds imposes a problem in the continuous development of the IT-systems of an organization. When the use of workarounds is unknown to the IT-department, the cycle of working, learning and innovating is based on incomplete information (Brown & Duguid, 1991). IT developments are based on the intended use of the IT-system, they do not consider the possibility of deviations from the intended use (Silic & Back, 2014) (Vos, 2018).

There are not only risks involved in the use of workarounds (Vos, 2018). Employees have a reason to resort to the use of workarounds. Workarounds can provide essential information to improve IT-systems. Finding workarounds uncovers this information and makes it possible to improve IT-system resilience (Vos, 2018).

1.2. Problem Statement

It has to be known when and where workarounds are used to reduce risk and to make use of the advantages of workarounds (Outmazgin & Soffer, 2013) (Vos, 2018). Therefore, it is important for an organisation to be able to detect workarounds. There are however many types of workarounds, for which different methods are needed to detect them.

1.3. Research objective and questions

In order to address this problem, the Resilience Mining Thesis Circle sets out to create a complete set of algorithms to find workarounds in text and in data. Four types of algorithms are divided over seven studies. Table 1 provides information this Thesis Circle. The Circle uses a sample database that is provided by the Open University. By focusing on text mining and using data from Achmea, this paper covers finding the best method for detecting workarounds in text fields with clustering algorithms. Using data from Achmea ensures that the results of this study are applicable on databases of comparable organizations.

	Structured data	Free text fields
Classification	**José Huisman** (2020) *Resilience Mining: Detecting Shadow IT in IT Systems with Data Classification*	**Ruben ten Cate** (2020) *Detecting shadow IT in free text fields using text mining and classification*
Association rule mining	**Thomas Sandfort** (2020) *Resilience mining: identifying workarounds in IT systems using association rule data mining*	**Jan van Rouwendal** (2020) *Tekst en association rule mining voor detecteren van workarounds in vrije tekst die gestructureerde invoer omzeilen*
Clustering	**Patrick van der Spoel** (2020) *Detecting workarounds with data clustering algorithms to improve Information Systems*	**Janneke Spronk** *Mining for workarounds in text fields with clustering algorithms*
Outlier detection	**Maarten Koskamp** (2020) *Mining for workarounds in information systems using outlier detection*	

Table 1. Studies of the resilience mining Thesis Circle

The results of these combined studies lead to a complete picture of the possibilities to detect workarounds in data and text with the use of algorithms, and this study focusses on clustering algorithms to detect workarounds in text fields. The main research question is:

How and to what extend can clustering algorithms detect workarounds in text fields in an IT-system?

To answer this question, we answer the sub questions:

What are indicators of workarounds in text fields?

What are the optimal settings for clustering algorithms to find workarounds in text fields?

What clustering algorithms are most suitable to find workarounds in text fields?

1.4. Relevance

Although workarounds have a bad image and are associated with incompliance, there is proof that there is value in the knowledge of the use of workarounds, and there is risk in not knowing what workarounds are used (Alter, 2014). Although several papers mention the use of workarounds in text fields, no effort has been made so far to find a method to automatically detect them.

The value of the knowledge of workarounds in text fields depends on the type of workaround, and the reason is it used. The creativity of the user that created the workaround can be used to improve IT system resilience (Alter, 2014). Patterson found that free text fields are used for communication to other parties. Since the field is not designed for this communication, there is great risk that the communication is missed (Patterson, 2018). In this case, it would be an improvement of the IT-system to add a field for communication, that cannot be missed by the receiving party. In the case of the use of copy paste, a selection of predefined texts can be offered.

1.5. Main lines of approach

Recent research is focussing on how to detect workarounds to use them to the advantage of an organization (Vos, 2018). This study is an addition to this tendency. So far, research on finding workarounds with datamining has focussed on the data in event logs, but not all workarounds leave traces in event logs. In order to be able to find all workarounds, the Resilience Mining Thesis Circle explores the possibility to detect workarounds in the database itself.

This study is conducted on a dataset of the department of Collective Pensions of Achmea, a large insurance company in The Netherlands. The known use of workarounds is explored with four interviews with process specialists. A selection of workarounds is made, to test whether the clustering algorithms can find these workarounds.

Personal information in the dataset is traced and deleted with the use of 'Named Entity Recognition'. The dataset is exported from the database and loaded in RapidMiner. The dataset is prepared for text mining, and five clustering algorithms are optimized for finding workarounds. The effectiveness of the algorithms is determined by comparing the efficiency of the algorithms in placing the workarounds in separate clusters.

2. Theoretical background and exploration of the topic

2.1. Research approach

The goal of the literature review is to gain a complete picture of the scientific knowledge on the subject of workarounds, and more specifically, workarounds in text fields and possibilities to detect them. The questions that we aim to answer in this literary review are:

- What are workarounds and why are they used
- What workarounds exist in free text fields
- What methods are known to find workarounds
- How can text mining and clustering be used to find workarounds

We started with search terms 'workarounds', 'shadow IT', 'feral practices'. The next step is to scan the articles for relevance and scan the cited works in the relevant articles. Then a new search is done with more specific queries like 'detecting workarounds', 'detecting shadow IT'. This process iterated until new queries do not deliver new relevant articles. The search was conducted in the Open University Library, ResearchGate.net, ScienceDirect and Google Scholar.

2.2. Implementation

The first search on workarounds gives results on definitions and classification of the subject. Many articles are found and scanned. The articles that are used for this study are: 'Theory of Workarounds' (2014) by S. Alter, 'Shadow IT - A view from behind the curtain' (2014) by M.B. Silic and 'Towards a Taxonomy for Shadow IT' (2016) by A.W. Kopper.

Next, a more specific search was conducted on workarounds in free text fields. Two articles are found: '"I Did It My Way": Social workers as secondary designers of a client information system' (2012) by S.V.P. Huuskonen and 'Workarounds to Intended Use of Health Information Technology: A Narrative Review of the Human Factors Engineering Literature' (2018) by E.S. Patterson.

The search for methods to detect workarounds delivered the following articles we used for this study: 'A process mining-based analysis of business process workarounds' (2016) by N.S. Outmazgin, 'Analyzing and Understanding workarounds in an IS to improve Business Processes in a Multinational Corporation' (2018) by L. Vos and finally 'Workarounds in retail work systems: prevent, redesign, adopt or ignore?' (2019) by I. van de Weerd.

2.3. Results and conclusions

2.3.1. What are workarounds

Many different terms are used in the literature on workarounds. Kopper and Westner (2016) created a taxonomy for all related terms, which is used in this paper. A definition of workarounds is:

> *"A goal-driven adaptation, improvisation, or other change to one or more aspects of an existing work system in order to overcome, bypass, or minimize the impact of obstacles, exceptions, anomalies, mishaps, established practices, management expectations, or structural constraints that are perceived as preventing that work system or its participants from achieving a desired level of efficiency, effectiveness, or other organizational or personal goals (Alter, 2014)".*

The overarching term is Feral Practices. Feral practices imply the use of Information Technology differently than the intended use, without the knowledge of the IT department (Kopper & Westner, 2016). Shadow IT and workarounds are a part of these feral practices. The difference between Shadow IT and workarounds is that Shadow IT is an addition to an IT-system, and workarounds concern a deviation of the official use of the IT-system (Kopper & Westner, 2016).

2.3.2. The use of workarounds

There are two main causes for using workarounds that can occur individually or together: there can be obstacles to performing work in the preferred manner, or there can be misalignment of goals of different stakeholders (Alter, 2014). An example of the first reason is when a process requires information that is not available, or when the IT-system malfunctions (Alter, 2014). The second reason can occur when systems are found to be too slow, when personal goals conflict with organizational goals, or when there is a personal motivation to bypass decisions of the organization.

The consequences of the use of workarounds are manifold. There are workarounds that have no effect (Vos, 2018). Negative effects can be compliance issues, risk of data loss and wasted investments (Silic & Back, 2014). There can be positive effects when workarounds are used to bypass bugs or anomalies, and they ensure continuation of work (Alter, 2014).

2.3.3. Workarounds in text

There are two main categories of workarounds in text fields. The first type is the use of text fields for another purpose than intended. Patterson (2018) has found that unstructured text fields were used for communication, where these fields were not designed for this purpose. They were used for communication either because this was the only way to communicate with the other party, or because this was the easiest way. The text fields had to be opened to reveal their information, which made the risk of overseeing the information substantial (Patterson, 2018).

The second type of workaround is the use of copy paste (Huuskonen & Vakkari, 2012). Entering large amounts of text can be a laborious task. This type of workaround can be used when large amounts of text are the same as other instances, or when the same text is used very often. This type of workarounds is used for example by social workers. In order to save time, they copy and paste parts of reports of clients that are similar (Huuskonen & Vakkari, 2012).

2.3.4. Finding workarounds

So far, research focusses on what workarounds are, why they are used and why they should be detected. In most research workarounds are identified using interviews or surveys (van de Weerd, Vollers, Beerepoot, & and Fantinato, 2019) (Outmazgin & Soffer, 2013). Research on finding workarounds automatically focusses on mining event logs (Outmazgin & Soffer, 2013). Finding workarounds in text fields has not been addressed yet. Using interviews may not reveal all workarounds. Interviewed employees may be aware of the risk involved, and they don't want to be associated with incompliance. Process mining only finds workarounds that leave traces in event logs.

Text mining

Text in free text fields is unstructured data. There are basic methods to transform text to make it possible to use data mining techniques (Provost & Fawcett, 2013). For text mining, text is seen as a bag of words. Each document is seen as a collection of words and each word is seen as a token. Each document is represented by either 1 or 0, depending if the document contains that word (Provost & Fawcett, 2013). This process is called tokenization. The bag of words method makes it possible to apply term frequency, inverse document frequency and the combination of these methods on text (Provost & Fawcett, 2013).

Clustering

Clustering is a data mining method that groups items in a population based on their similarities. This is an unsupervised method, which means that it has no specific purpose (Provost & Fawcett, 2013). There are four methods on which clustering algorithms are based.

With hierarchical clustering, clusters are based on distance and ordered in different levels. This method is based on how similarities between individual items link them together. On the highest level is the cluster that contains all other clusters, and on the lowest level are the smallest clusters that do not overlap each other (Provost & Fawcett, 2013). The results can be visualized by a dendrogram, see figure 1. Agglomerative clustering is a hierarchical clustering algorithm.

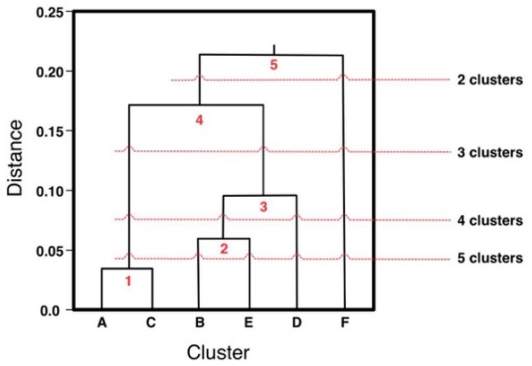


Figure 1.Hierarchical clustering (Provost, 2013)

Clustering around centroids starts with a specified number of clusters. The itemset is divided in the number of clusters that is chosen, and each of these clusters is assigned a centroid (Provost & Fawcett, 2013). The centroid is the average of the values of the features of the items in the cluster. The algorithm runs the clustering iteratively to optimize the clusters and centroids. An example with three clusters is given in figure 2. Two examples of clustering around centroids are the k-Means and the k-Medoids algorithm (Provost & Fawcett, 2013).

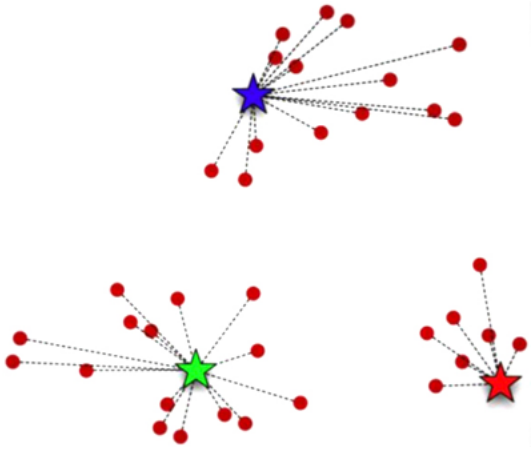


Figure 2. Clustering around centroids (Provost, 2013)

Another method for clustering is DBSCAN (Density Based Spatial Clustering for Applications with Noise). This model uses minimum density level estimation, with a threshold for the number of neighbors, to find core points. All neighbors within a specified range of the core points are considered to be a part of the same cluster. If there are core points among the neighbors, these are included in the cluster (Schubert, Sander, Ester, & Xu, 2017). A visualization of the DBSCAN model is given in figure 3.

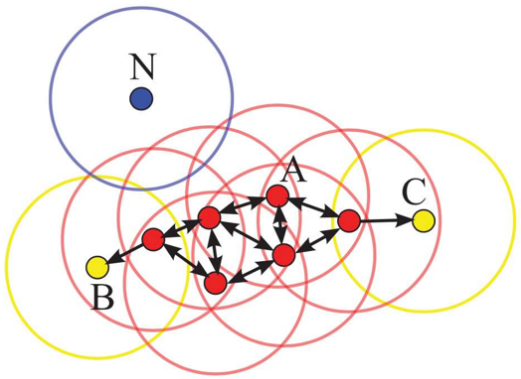


Figure 3. DBSCAN (Schubert, Sander, Ester, & Xu, 2017)

2.4. Objective of the follow-up research

Much is known about the reasons to use workarounds and the consequences of the use of workarounds. It is found that finding workarounds can help to improve IT-system resilience. Two types of workarounds in text fields have been identified so far: the use of text fields in another way than intended, and de use of copy and paste.

There is more need for information and methods for finding workarounds. As part of the Resilience Mining Thesis Circle, we will contribute to the field by testing the use of clustering algorithms to find workarounds in free text fields, using data from Achmea Pensions.

3. Methodology

3.1. Conceptual design

To explore the possibilities of finding workarounds in text fields with clustering algorithms, we use a dataset in which the text fields that contain a workaround are labeled. We apply different algorithms and optimize those algorithms for best results. In this study the subject cannot be randomly allocated to an experimental or control group, which makes this a quasi-experiment (Saunders & Thornhill, 1997). The control group is a group of text fields in which we know a workaround is used. The methods are highly objective, and this is a quantitative research.

3.2. Data gathering

We use two methods of data gathering. First, we need to know what workarounds are used in text fields in the database. To gain this knowledge we use unstructured, informal interviews with four process specialists. Each of the interviewees has knowledge of different specialisms, and all specialisms are covered by these four specialists.

The interviews are based on the question whether the specialist knows any examples of workarounds that present themselves in text fields. If workarounds are known, the interview continues to find out how often the workarounds are used, in which text fields they are present, and how they can be found. The interviews are recorded and transcribed. The results are interpreted the three most suitable workarounds are selected. The introduction and questions for the interview are available in the document 'Interviews'.

The second method of data gathering is the collection of text fields from the database. Only workarounds that are still in use are of value for a company, so we select text fields that are added in the previous two years. In order to be able to use data from Achmea Pensions, we need to remove personal information from the dataset. This is done with the use of 'Named Entity Recognition' (NER). All personal information that could be present in the dataset is gathered and when present in the dataset it is replaced with 'xxxxx'.

3.3. Modeling

3.3.1. Technical design: CRISP-DM

The process of data mining is a craft that involves science as well as technology. In order to guide this process, a standard process codification is developed: The Cross Industry Standard Process for Data Mining (CRISP-DM) (Provost & Fawcett, 2013). This method is used for this study. The basic process is visualized in figure 4. The steps are repeated until no improvement in the results is achieved.

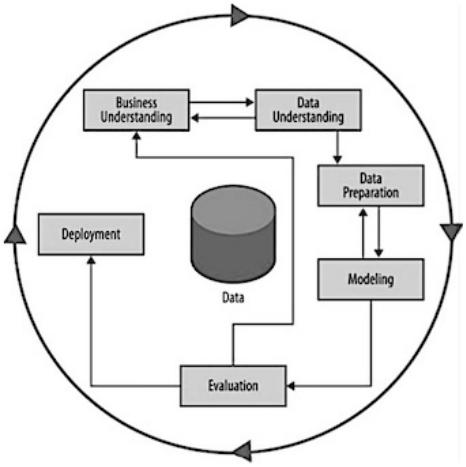


Figure 4. The Crisp-DM data mining process (Provost, 2013)

3.3.2. Data preparation

When the dataset is imported in RapidMiner, the true preprocessing of the data starts. Based on the information gained from the interviews, the text fields in the datasets are assigned a label. The label indicates if and what workaround is present.

3.3.3. Modeling

RapidMiner contains the following algorithms that are suitable for non-numerical data:

1. k-Means (Kernel)
2. DBSCAN
3. k-Medoids
4. Agglomerative clustering

We conduct three experiments with each algorithm. One with four clusters and the group without workarounds sampled to the average size of the groups that do have workarounds (experiment 1), one with four clusters and the group without workarounds sampled to 1000 items (experiment 2) and one with 6 clusters and the group without workarounds sampled to 1000 items (experiment 3). This way we see the effect of the percentage of workarounds in the total dataset, and the difference in performance with different numbers of clusters.

3.3.4. Evaluation

First, the model runs with all operators in default settings. RapidMiner calculates the accuracy of the model. The accuracy is used to optimize the algorithms. Next, step by step the operators are optimized to reach the best accuracy. This is an iterative process, repeated until the best performance is found.

3.4. Data analysis

3.4.1. Confusion matrices

After having optimized the algorithms, we compare the different algorithms with each other with the use the confusion matrices. The confusion matrix gives the precision for each label. The precision is the percentage of true workarounds within the predicted cluster (Provost & Fawcett, 2013).

The advantage of the use of clustering is that you only have to look at a number of items in a cluster to see if it contains a workaround, instead of looking at all items in a dataset. Therefore, it is important that the precision is high in the clusters that contain a workaround.

To compare the models, we look at the average precision of the clusters that identify a workaround. The result of this comparison is a ranking of the four algorithms based on their suitability to detect workarounds in text fields in a database.

3.4.2. False positives

A false positive means a wrong prediction of a workaround. Clustering, being an unsupervised method, is not predictive. In this experiment however, we want to see if the algorithms find certain clusters containing a workaround. In this respect a false positive is a text fields without workaround x, in a cluster that predicts workaround x. This occurs when there are similarities in the text fields that have nothing to do with the workaround.

3.5. Reflections

3.5.1. Reliability

Reliability is the extent to which research findings are consistent. The results should be the same when repeated on a different moment or by another researcher (Saunders & Thornhill, 1997). The use of standard data mining techniques and following the CRISP-DM process ensures the reliability of the technical aspects of this study. The selection of workarounds we use for the experiment is based on the knowledge of the interviewees. It is possible that other specialists would name different workarounds, or that a specialist has reasons not to mention workarounds he uses. This risk is reduced by interviewing three specialists instead of only one.

3.5.2. Validity

Validity is the extent to which we measure what we want to measure (Saunders & Thornhill, 1997). The validity of a study can be split in internal validity and external validity. The quasi-experiment contains an internal check on validity. If the workarounds that we look for are not found, the algorithms are not suitable for this purpose. The scope of this study is however limited, and three selected workarounds may not be representative of all workarounds.

The fact that realistic data from Achmea was used contributes to the external validity. The data was not designed for the use for a research. Also, the data is comparable to data in other insurance companies. This increases the likelihood that the algorithms that work on this dataset, also work on datasets of other companies.

3.5.3. Ethics

For this study, data from Achmea is used that contained personal information of clients. We proceed with caution to guarantee their privacy. We clear the dataset of personal information thoroughly, but some personal information of clients or personnel may be overseen. The dataset will for this reason not be made public and will not be shared with anyone other than those who need to see it to check for authenticity.

4. Results

This chapter contains the results of this study. First, the selection of three workarounds from the interviews is described. Then the gathering and preparing of data for the experiments is explained. What follows is a description of the experiments and the results for each algorithm. Finally, the results from the different algorithms are compared.

4.1. Preparation

To be able to conduct this experiment, we need to know what workarounds are used in the database. Four specialists who are responsible for the main specialisms are selected for an interview. The interviews are transcribed and the text is added in the document 'Interviews'. Three workarounds in the client's text fields and three workarounds in the mutation job's text fields were identified.

4.1.1. Selecting workarounds

The three workarounds in the mutation jobs text fields turned out to be unsuitable for this experiment. They were either no longer in use, used too little or there was no consequent use of the text field that could be used for labeling the text field.

The three workarounds in the client's text are special administration, email addresses and correspondence addresses. Estimated numbers are between 50 and 200 each year. They are all easy to find to be labeled. They are all workarounds where the text field is used for another purpose than it is intended for.

4.1.2. Gathering the data

Considering the speed of development of automation and improvement of processes on the department of pensions, we choose to use data from the past two years. A request is submitted to get all client's text fields that were created in the period from 1-9-2017 to 1-9-2019.

In the query the client's number, text number and date of creation of the text field are used for selecting unique records. The name, date of birth, address and social security number of the client and (ex-)partners are selected for applying Named Entity Recognition (NER).

After having received the queries, an employee from the Achmea Data Expertise Centre applied NER and removed all personal information in the query from the text fields. All other fields but the text fields are then removed. A manual check is done to remove any personal information that remained, and to restore some text that was accidentally removed by applying NER.

4.1.3. Exploration of the dataset

Having gathered the data, the workarounds are identified, and the text fields are labeled with numbers from 1 to 3. Label 1 is for Special administration, label 2 is for Email addresses, label 3 is for Correspondence addresses. Text fields with label 0 contain no workaround.

Label 1. Special administration

Special administration means the client is not able to take care of his or her financial administration. A special administrator is appointed to take over the administration and all correspondence about finances. If a client is subject to special administration, this information is registered in the client's text field. The Dutch word for special administration is "bewind" and for special administrator is "bewindvoerder". There are 248 of the 9225 text fields that contain information about special administration or a special administrator. This is 2.69% of the dataset.

Label 2. Email addresses

There is a special field available for the client's email address. This field can only be filled by the client by logging in to the web portal. This way the identity of the client is confirmed before using the email address. Employees sometimes receive an email address by email or phone, and then fill the email address in the client's text field. For privacy reasons, all email addresses in the dataset have been replaced by xxx@xxx.xxx. There are 224 of the 9225 text fields that contain clients email address. This is 2.43% of the dataset. There are more text fields with email addresses, but they are not of the client. They are not counted as a workaround.

Label 3. Correspondence addresses

In some cases, a client wants to receive mail on a different address then their residential address. This second address is called a correspondence address. In Dutch this is "correspondentie adres". The administration system supports the use of a correspondence address, but there is no option to add additional information about the reason for the use of a correspondence address. This additional information is filled in the client's text field. When the reason for a correspondence address is special administration, the text field is labeled for special administration. There are 102 of the 9225 text fields that contain information about a correspondence address. This is 1.11% of the dataset.

4.1.4. Preprocessing and modeling

The dataset has three attributes, the text field, the label that indicates the presence of a workaround and an ID. The database is not balanced. There are 8.650 items with label 0, 249 items with label 1, 224 items with label 2 and 102 items with label 3. The 9225 items contain an average of 17.97 words. We work in RapidMiner Studio with the Educational label 09.0.003. The models for the four algorithms are very similar. An example is shown in figure 5.

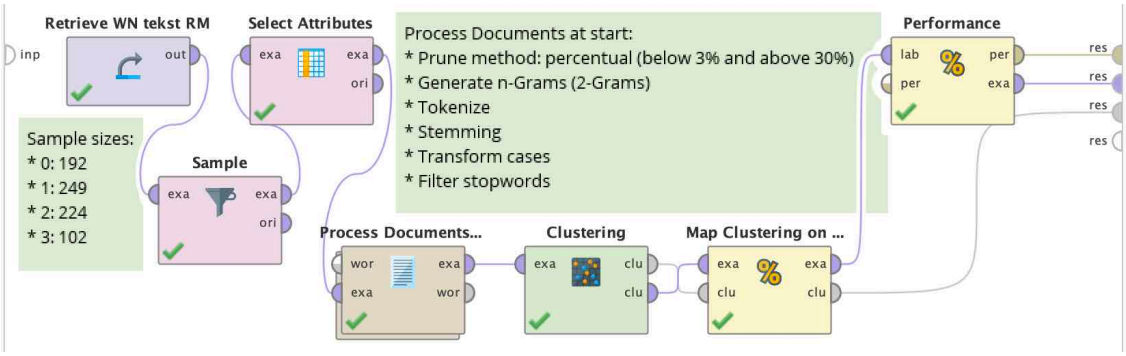


Figure 5. The model in RapidMiner

Select Attributes

The items are labeled to be able to evaluate the performance of the clustering algorithms. The label should not be a part of the clustering. With the operator 'Select Attributes' we exclude the label from the example set.

Process Documents from Data

In the 'Process Documents from Data' operator, we add operators for text preprocessing. The available operators are 'Tokenize', 'Stemming', 'Transform cases', 'Filter Stopwords' and 'Generate n-Grams'. The operator 'Process Documents from Data' offers the vector creation options and the possibility for pruning.

Map Clustering on Labels and Performance

To test the performance of the algorithms we add the operator 'Map Clustering on Labels'. This operator adds an attribute that gives the clusters a predicted label. To use this operator, the number of clusters needs to be equal to the number of labels. For the experiment with 6 clusters, we export

the dataset with the clusters as an attribute and place them in a confusion matrix in Excel. This data is available in the file 'Confusion matrices'. The average precision in the clusters with a workaround is calculated with the highest percentage of each workaround in a cluster. The operator 'Performance' uses the predicted labels from the 'Map Clustering on Labels' operator to create a confusion matrix and calculate the accuracy of the model in RapidMiner.

4.2. k-Means (Kernel)

The process for k-Means (Kernel) is optimized with the smallest sample and four clusters. The settings with the best performance are given in table 2.

Process Documents from Data	
Pruning	Below 2% and above 20%
Vector creation	Term Frequency
Tokenize	On
Transform Cases	On
Filter Stopwords	Off
Stemming	Off
Generate n-Grams	Off
Clustering k-Means (Kernel)	
Kernel type	Radial
Use local random seeds	On
Kernel gamma	1.0
Use weights	Off

Table 2. Settings with best performance for k-Means (kernel)

k-Means (kernel), experiment 1: four clusters, small sample

This model gives an accuracy of 76.66%, the confusion matrix is shown in table 3.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	69	3	19	2	93
Cluster 2	1	208	7	21	237
Cluster 3	32	13	223	81	349
Cluster 4	0	0	0	88	88

Table 3. Confusion matrix of experiment 1 with k-Means (kernel)

The green cells are the largest group of each workaround, on which the precision is based. In this case they are divided over three clusters. The average precision for the workarounds is 75.28%. The best precision of each workaround is:

- Correspondence addresses: 74.19% of cluster 1
- Email addresses: 87.76% of cluster 2
- Special administration: 63.90% of cluster 3

All three workarounds are the majority of their own cluster, so if you examine the clusters without any knowledge, you find all three workarounds. There are two relatively large groups of items in the clusters with special administration that are either correspondence addresses or no workaround. There are similarities in these fields with the text fields with special administration that are stronger than the similarities with the text fields with the same label. With this sample size and number of clusters, this algorithm is able to find 3 workarounds.

k-Means (kernel), experiment 2: four clusters, large sample

The group with label 0 is sampled to 1000 items. Now the accuracy is better with TF-IDF than with Term Frequency, so we use TF-IDF. This model gives an accuracy of 59.97%, the confusion matrix is shown in table 4.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	92	4	86	508	690
Cluster 2	10	220	12	1	243
Cluster 3	0	0	133	23	156
Cluster 4	0	0	18	468	486

Table 4. Confusion matrix of experiment 2 with k-Means (kernel)

The green cells are the largest group of each workaround, on which the precision is based. They are divided over three clusters. The average precision for the workarounds is 63.04%. The best precision for each workaround is:

- Correspondence addresses: 13.33% of cluster 1
- Email addresses: 90.53% of cluster 2
- Special administration: 85.26% of cluster 3

Email addresses and special administration are the majority in their cluster. If you look at the clusters without any knowledge about workarounds, you will find these two. Correspondence addresses are only 13.33% of a cluster, which means they will not be found. There is a large number of items in the cluster with correspondence addresses, that are no workaround. This is a group of text fields in the large sample without workarounds that are more similar to each other than the group of text fields with correspondence addresses is. With this sample size and number of clusters, this algorithm is able to find 2 workarounds.

k-Means (kernel), experiment 3: six clusters, large sample

The number of clusters is set to 6 clusters. The operators 'map clustering on labels' and 'performance' are disabled. The resulting example set is exported to Excel to create a confusion matrix. This confusion matrix is shown in table 5.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	0	0	0	75	75
Cluster 2	0	0	18	467	485
Cluster 3	13	127	28	1	169
Cluster 4	0	0	1	54	55
Cluster 5	89	2	202	402	695
Cluster 6	0	95	0	0	95

Table 5. Confusion matrix of experiment 3 with k-Means (kernel)

The green cells are the largest group of each workaround, on which the precision is based. The majority of the special administration and correspondence addresses are in the same cluster. Email addresses are spread over 2 clusters. The average precision in the clusters that contain a workaround is 43.15%. The best precision for each workaround is:

- Correspondence addresses: 12.81% of cluster 5
- Email addresses: 87.57% of cluster 3 and 6
- Special administration: 29.06% of cluster 5

If you examine the clusters without knowledge of existing workarounds, in cluster 1, 2, 4 and 5 you find no workarounds. In cluster 3 and 6, you find email addresses (75.15% and 100%). Special administration and correspondence addresses will not be found, for they are only 29.06% and 12.81% of a cluster. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

4.3. DBSCAN

With this algorithm it is not possible to choose the number of clusters. First, we run the algorithm in default settings with the operators 'map clustering on labels' and 'performance' disabled and it creates 5 clusters. The clustering parameters are set to create 4 clusters. In the 'Process documents from data' operator only 'Tokenize' and 'Generate n-Grams (2-grams) are enabled. In the clustering operator Epsilon is set to 0.5 and minimum points is set to 50 to create 4 clusters. The settings with the best performance are given in table 6.

Process Documents from Data	
Pruning	Below 1% and above 20%
Vector creation	TF-IDF
Tokenize	On
Transform Cases	Off
Filter Stopwords	Off
Stemming	Off
Generate n-Grams	Off
Clustering DBSCAN	
Epsilon	0.9

Table 6. Settings with best performance for DBSCAN

DBSCAN, experiment 1: four clusters, small sample

This model gives an accuracy of 56.32%, the confusion matrix is shown in table 7.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	0	71	2	5	78
Cluster 2	0	97	0	2	99
Cluster 3	102	56	247	97	502
Cluster 4	0	0	0	88	88

Table 7. Confusion matrix of experiment 1 with DBSCAN

The green cells are the largest group of each workaround, on which the precision is based. The majority of special administration and correspondence addresses are in the same cluster. Email addresses are spread over two clusters. The average precision in the clusters that contain a workaround is 49.06%. The best precision for each workaround is:

- Correspondence addresses: 20.32% of cluster 3
- Email addresses: 94.50% of cluster 1 and 2
- Special administration: 49.20% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 1 and 2 you find email addresses (97.98% and 91.03%). In cluster 4 you find only items without workaround. In cluster 3 special administration is 49.20%. This is not high enough to be certain the workaround will be found. Correspondence addresses are not found, it is at best 20.32% of a cluster. Just like in experiment 1 with k-Means (kernel), there are many items in the cluster with special administration that are correspondence addresses or no workaround. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

DBSCAN, experiment 2: four clusters, large sample

The group without workarounds is sampled to 1000 items. This model gives an accuracy 50.41%, the confusion matrix is shown in table 8.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	0	0	0	70	70
Cluster 2	6	203	3	1	213
Cluster 3	96	21	242	580	939
Cluster 4	0	0	4	349	353

Table 8. Confusion matrix of experiment 2 with DBSCAN

The green cells are the largest group of each workaround, on which the precision is based. Special administration and correspondence addresses are mostly in the same cluster. The average precision in the clusters that contain a workaround is 43.77%. The best precision for each workaround is:

- Correspondence addresses: 10.22% of cluster 3
- Email addresses: 95.31% of cluster 2
- Special administration: 25.77% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 1 and 4 you find no workarounds. In cluster 2 you find email addresses. Special administration and correspondence addresses will not be found, for they are at best 25.77% and 10.22% of a cluster. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

DBSCAN, experiment 3: six clusters, large sample

The number of clusters is set to 6. The operators 'map clustering on labels' and 'performance' are disabled. The resulting example set is exported to Excel to create a confusion matrix. The result is shown in table 9.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	101	27	245	581	954
Cluster 2	1	30	0	1	32
Cluster 3	0	94	0	0	94
Cluster 4	0	73	0	0	73
Cluster 5	0	0	4	348	352
Cluster 6	0	0	0	69	69

Table 9. Confusion matrix of experiment 3 with DBSCAN

The green cells are the largest group of each workaround, on which the precision is based. The majority of special administration and correspondence addresses are in the same cluster. Email addresses are spread over 3 clusters. The average precision in the clusters that contain a workaround is 43.77%. The best precision for each workaround is:

- Correspondence addresses: 10.59% of cluster 1
- Email addresses: 97.92% of cluster 2,3 and 4
- Special administration: 25.68% of cluster 1

If you examine the clusters without knowledge of existing workarounds, in cluster 5 and 6 you find no workarounds. In cluster 2, 3 and 4, you find Email addresses (94%, 100% and 100%). Special administration and correspondence addresses will not be found, as they are at best 25.68% and 10.59% of a cluster. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

4.4. Agglomerative clustering

The process for agglomerative clustering has one extra operator. The clustering operator creates a dendrogram with as many levels as are needed to create one cluster for each item. The operator 'Flatten Clustering' is added after the clustering operator to stop the tree at the point where 4 clusters are created. The settings which give the best performance are given in table 10.

Process Documents from Data	
Pruning	Below 3% and above 20%
Vector creation	Term Frequency
Tokenize	On
Transform Cases	On
Filter Stopwords	Off
Stemming	Off
Generate n-Grams	2-Grams
Agglomerative Clustering	
Mode	Average link

Table 10. Settings with best performance for Agglomerative Clustering

Agglomerative clustering, experiment 1: four clusters, small sample

This model gives an accuracy of 61.02%, the confusion matrix is shown in table 11.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	0	0	0	1	1
Cluster 2	3	134	3	6	146
Cluster 3	99	90	246	97	532
Cluster 4	0	0	0	88	88

Table 11. Confusion matrix of experiment 1 with Agglomerative Clustering

The green cells are the largest group of each workaround, on which the precision is based. The majority of the special administration and correspondence addresses are in the same cluster. The average precision in the clusters that contain a workaround is 52.21%. The best precision for each workaround is:

- Correspondence addresses: 18.61% of cluster 3
- Email addresses: 91.78% of cluster 2
- Special administration: 46.24% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 1 and 4 you find no workarounds. In cluster 2 you find email addresses. In cluster 3 you find special administration, but only for 46.24% of the cluster. Again, there are many items in the cluster with special administration, containing email addresses, correspondence addresses or no workaround. Correspondence addresses will not be found, for it is at best 18.61% of a cluster. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

Agglomerative clustering, experiment 2: four clusters, large sample

The group without workarounds is sampled to 1000 items. With this sample size, accuracy is higher when vector creation is set to TF-IDF. This model gives an accuracy of 53.65%, the confusion matrix is shown in table 12.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	0	0	0	66	66
Cluster 2	18	222	31	37	308
Cluster 3	82	2	216	490	790
Cluster 4	2	0	2	407	411

Table 12. Confusion matrix of experiment 2 with Agglomerative Clustering

The green cells are the largest group of each workaround, on which the precision is based. The majority of the special administration and correspondence addresses are in the same cluster. The average precision in the clusters that contain a workaround is 36.60%. The best precision for each workaround is:

- Correspondence addresses: 10.83% of cluster 3
- Email addresses: 72.08% of cluster 2
- Special administration: 27.34% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 1 and 4 you find no workarounds. In cluster 2 you find Email addresses. Special administration and correspondence addresses will not be found, as they are at best 27.34% and 10.83% of a cluster. With this sample size and number of clusters, this algorithm is able to find 1 workaround.

Agglomerative clustering, experiment 3: six clusters, large sample

The number of clusters is set to 6 clusters. The operators 'map clustering on labels' and 'performance' are disabled. The resulting example set is exported to create a confusion matrix. The result is shown in table 13.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	82	2	86	489	659
Cluster 2	2	0	2	407	411
Cluster 3	0	0	130	1	131
Cluster 4	0	0	0	66	66
Cluster 5	17	222	30	5	274
Cluster 6	1	0	1	32	34

Table 13. Confusion matrix of experiment 3 with Agglomerative Clustering

The green cells are the largest group of each workaround, on which the precision is based. The workarounds are all in different clusters. The average precision in the clusters that contain a workaround is 64.23%. The best precision for each workaround is:

- Correspondence addresses: 12.44% of cluster 1
- Email addresses: 81.02% of cluster 5
- Special administration: 99.24% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 2, 4 and 6 you find no workarounds. In cluster 3 you find special administration. In cluster 5 you find email addresses. Correspondence addresses will not be found, as it is at best 12.44% of a cluster. With this sample size and number of clusters, this algorithm is able to find two workarounds.

4.5. k-Medoids

The k-Medoids process is very similar to the k-Means process. The settings that give the best performance are given in table 14.

Process Documents from Data	
Pruning	Below 3% and above 40%
Vector creation	TF-IDF
Tokenize	On
Transform Cases	Off
Filter Stopwords	On
Stemming	Off
Generate n-Grams	Off
Clustering k-Medoids	
Numerical measure	EuclidianDistance

Table 14. Settings with best performance for k-Medoids

k-Medoids, experiment 1: four clusters, small sample

This model gives an accuracy of 70.93%, the confusion matrix is shown in table 15.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	2	1	0	20	23
Cluster 2	46	223	20	2	291
Cluster 3	54	0	229	80	363
Cluster 4	0	0	0	90	90

Table 15. Confusion matrix of experiment 1 with k-Medoids

The green cells are the largest group of each workaround, on which the precision is based. Correspondence addresses are spread over two clusters. One part together with email addresses, and the other part with special administration. The average precision in the clusters that contain a workaround is 51.84%. The best precision for each workaround is:

- Correspondence addresses: 15.81% of cluster 2
- Email addresses: 76.63% of cluster 2
- Special administration: 63.09% of cluster 3

If you examine the clusters without knowledge of existing workarounds, in cluster 1 and 4 you find no workarounds. In cluster 2 you find email addresses. In cluster 3 you find special administration. There are many items in the cluster with special administration, being either correspondence addresses or no workaround. With this sample size and number of clusters, this algorithm is able to find 2 workarounds.

k-Medoids, experiment 2: four clusters, large sample

The group without workaround is now sampled to 1000 items and performance is measured again. This model gives an accuracy of 45.52%, the confusion matrix is shown in table 16.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	1	0	0	105	106
Cluster 2	0	0	0	0	0
Cluster 3	101	224	249	428	1002
Cluster 4	0	0	0	467	467

Table 16. Confusion matrix of experiment 2 with k-Medoids

The green cells are the largest group of each workaround, on which the precision is based. All workarounds are in the same cluster. The average precision in the clusters that contain a workaround is 19.10%. The best precision for each workaround is:

- Correspondence addresses: 10.08% of cluster 3
- Email addresses: 22.36% of cluster 3
- Special administration: 24.85% of cluster 3

If you examine the clusters without knowledge of existing workarounds you find no workarounds. All workarounds are placed in the same cluster, with a large number of text fields without workarounds. With this sample size, the algorithm is not able to find workarounds.

k-Medoids, experiment 3: six clusters, large sample

The number of clusters is now set to 6 clusters. The operators 'map clustering on labels' and 'performance' are disabled. The resulting example set is exported to create a confusion matrix. The result is shown in table 17.

	Correspondence addresses	Email addresses	Special administration	No workaround	Total
Cluster 1	1	0	0	105	106
Cluster 2	0	0	0	6	6
Cluster 3	0	0	0	462	462
Cluster 4	88	0	219	426	733
Cluster 5	13	126	30	1	170
Cluster 6	0	98	0	0	98

Table 17. Confusion matrix of experiment 3 with k-Medoids

The green cells are the largest group of each workaround, on which the precision is based. The majority of the special administration and correspondence addresses are in the same cluster. The average precision in the clusters that contain a workaround is 42.98%. The best precision for each workaround is:

- Correspondence addresses: 12.01% of cluster 4
- Email addresses: 87.06% of cluster 5 and 6
- Special administration: 29.88% of cluster 4

If you examine the clusters without knowledge of existing workarounds, in cluster 1, 2, and 3 you find no workarounds. In cluster 5 and 6, you find email addresses (74.56% and 100%). Special administration and correspondence addresses will not be found, as they are at best 29.88% and 12.01% of a cluster. With this sample size and number of clusters, this algorithm is only able to find 1 workaround.

4.6. Comparison of the algorithms

In comparing the algorithms, the accuracy of each model is not as important as the ability to create clusters in a dataset that help find workarounds. The results of the three experiments with the four algorithms are shown in table 18. The best results of each experiment are shown in green.

		Special administration	Email addresses	Correspondence addresses	Average Precision
k-Means (kernel)	Experiment 1	63.90%	87.76%	74.19%	75.28%
	Experiment 2	85.26%	90.53%	13.33%	63.04%
	Experiment 3	29.06%	87.57%	12.81%	43.15%
DBSCAN	Experiment 1	49.20%	94.50%	20.32%	49,06%
	Experiment 2	25.77%	95.31%	10.22%	40.36%
	Experiment 3	25.68%	97.92%	10.59%	41.20%
Agglomerative clustering	Experiment 1	46.24 %	91.78%	18.61%	52.21%
	Experiment 2	27.34%	72.08%	10.38%	36.60%
	Experiment 3	99.24%	81.02%	12.44%	64.23%
k-Medoids	Experiment 1	63.09%	76.63%	15.81%	51.84%
	Experiment 2	24.85%	22.36%	10.08%	19.10%
	Experiment 3	29.88%	87.06%	12.01%	42.98%

Table 18. Comparison of the results of the different algorithms

Email addresses are visible a cluster in 11 of 12 experiments. Special administration is visible in 4 of 12 experiments. Correspondence addresses are only visible in a cluster in 1 of 12 experiments.

In experiment 1 with 4 clusters and a small sample, k-Means (kernel) gives the best results with an average precision of 75.28% and all three workarounds are found. DBSCAN, Agglomerative clustering and k-Medoids all find 2 workarounds, with an average precision around 50%.

In experiment 2 with 4 clusters and a large sample, k-Means (kernel) also gives the best result with an average precision of 63.04% and finding 2 workarounds. DBSCAN and Agglomerative clustering have an average precision of 40.36% and 36.60% and find only email addresses. The k-Medoids algorithm is not able to find any workaround in this experiment.

In experiment 3, with 6 clusters and a large sample, Agglomerative clustering gives the best result with an average precision of 64.23% and finding 2 workarounds. DBSCAN, k-Means (kernel) and k-Medoids have an average precision just around 40% and find 1 workaround.

Conclusion

The performance of the algorithms is very much dependent of the percentage of the workarounds in the dataset. If this percentage is very high, k-Means (kernel) gives the best results. If the percentage of workarounds in the dataset is low, Agglomerative clustering gives the best results.

5. Conclusions

5.1. Discussion

Literature on workarounds mostly describes reasons for the use of workarounds and classifications of workarounds. Most research on finding workarounds focusses on interviews and process mining of event logs. With these methods, not all workarounds can be found (Outmazgin & Soffer, 2013). The Resilience Mining Thesis Circle contributes to the field by searching for methods to find workarounds with algorithms in a database. We contribute specifically by exploring the possibilities of detecting workarounds in text fields with clustering algorithms.

We found that workarounds in text have characteristics that make it possible to detect them with clustering algorithms. This is a novel contribution to the existing knowledge on methods to detect workarounds. Our findings are rudimentary, and more research needs to be done to confirm our findings and gain more detailed information on the use of clustering algorithms for this purpose.

As has been the experience in literature so far, one method is not able to find all workarounds (Vos, 2018) (Outmazgin & Soffer, 2013). Only in the experiment with four clusters and a small sample, one algorithm finds all three workarounds in the dataset. If the percentage of workarounds is lower in the dataset, the best result is two out of three workarounds. The percentage of workarounds in the dataset has to be relatively high to be able to use clustering as a method for detection of workarounds.

The dataset that was used is very specific. It contains one type of text field from the department of Collective Pensions. The data and the workarounds are very specific for this department within Achmea. Although it is a dataset directly from practice, data from other companies may contain different characteristics that are specific for that product and IT-system, that influence the clustering algorithms.

5.2. Reflection

Only workarounds in one type of text field were used for this experiment. The text field that is used is an open note field for information about the clients. The text fields contain many words, with an average of 18 words in one text field. Other text fields may contain different types of workarounds, that may be easier or more difficult to find with clustering algorithms. In smaller text fields, the characteristics of workarounds may be more strikingly present.

The dataset is anonymized for protection of the privacy of the clients of Achmea. The x's that replace personal information have an effect on the dataset and the clustering. Also, In the process of anonymizing two problems occurred. The first problem was that certain names in the dataset were equal to words in the text. For example, the Dutch word 'regeling' in 'pensioen regeling' (pension plan), is also the name of a client. This caused the word to be replaced by 'xxxxx' in all text fields. These faults were manually repaired as much as possible, but it may have influenced the dataset.

The second problem in anonymizing is that all email addresses have to be anonymized. One of the workarounds is the use of the text field for the email address of the client. Since all email addresses were replaced by "xxx@xxx.xxx", results on this workaround are different in reality.

Finally, a selection of four specialists was interviewed to identify the workarounds that are used within the department. These specialists may not know all workarounds that are used by their colleagues. Also, they may have reasons not to tell about all workarounds, when a workaround creates a compliancy or security risk. This means, we cannot be sure that we labeled all workarounds in the dataset.

This causes the part of the dataset that was labeled ‘no workaround’ to be a black hole. In this part of the dataset, there is one very large group of text fields that are very similar. This group contains 3947 of the 8650 items that are labeled ‘no workaround’. Next to this very large group, other groups that are very similar are present in this part of the dataset. Further inspection of this data falls out of scope for this study due to the limited time that is available. We see that in many experiments labeled workarounds are placed in the same cluster. This means that they are more similar to each other than to the items that are labeled ‘no workaround’, but in reality, it may mean that other workaround clusters are larger and more similar than the workarounds we labeled.

5.3. Conclusions

Indicators of workarounds in text fields

The assumption on which this study is based is that workarounds in text have similarities that make it possible to detect them with clustering and text mining. These similarities are calculated based on either Term Frequency, Inverse Document Frequency or a combination of these (Provost & Fawcett, 2013). This means that the number of times a token is used within a text field and the number of text fields that contain that word is used to add a weight to each token, and the clusters are formed based on similarities in these weights. The Agglomerative clustering algorithm gives better results with forming 2-grams. This means that a combination of two tokens in a row is an indication of a workaround in the text field.

The results of the experiment with 4 clusters and a small sample show that the assumption is true. The occurrence of specific tokens in a text field that contains a workaround, can be used to apply clustering algorithms to detect workarounds. The k-Means (kernel) algorithm clearly divides the dataset in 4 clusters, three with the workarounds and 1 without workarounds. This means that this algorithm is able to recognize shared characteristics of the workarounds and place them in the same cluster.

Optimal settings for clustering algorithms to find workarounds in text fields

There are many different possibilities for preparing text for text mining. For all algorithms, tokenizing is the most important step in gaining results with the algorithms. The operator ‘Transform cases’ often gives a better performance. If not, it does not have a negative influence on the performance. The other settings are dependent of the algorithms. K-Means (Kernel) and Agglomerative clustering have a better performance in experiment 1 with Term Frequency than with TF-IDF. For both, results with TF-IDF is better when the dataset is larger.

What clustering algorithm is most suitable to find workarounds in text fields

The k-Means algorithm is most suitable to find workarounds with a high occurrence in a dataset. If the dataset is large and the percentage of workarounds is low, Agglomerative clustering is more effective.

How and to what extend can clustering algorithms find workarounds in text fields in an IT-system?

Clustering algorithms can be used to detect workarounds in text fields. It is not possible to find all workarounds, since a certain percentage of the workaround needs to be present in the dataset and characteristics of the data without workarounds influences the clusters.

5.4. Recommendations for practice

Clustering algorithms can be used to find workarounds in text fields. With clustering you only have to scan a few items of each cluster to see if there is a workaround in the cluster. This means that instead of scanning thousands of items, you only scan 40 to 60 items, depending on the number of clusters. This method should give an indication of the presence of workarounds. It should be kept in mind that not all workarounds will be found when only one method is used.

In this study we only looked at the precision, we did not include recall in the assessment of the workarounds. The k-Means (kernel) and Agglomerative clustering algorithms will help to identify the workarounds that are used in a database. In the case of a workaround that forms a threat to security or compliance, the work of fellow member of the Resilience Mining Thesis Circle Ruben ten Cate provides a method for detecting all instances of a workaround with classification algorithms and text mining (Cate ten, 2020).

5.5. Recommendations for further research

From the literature review we learned that knowledge of workarounds in text fields is very limited. Text being unstructured data it is a little more laborious to study, but on the other hand it contains much information and the free format of a text field makes it easy to use them for workarounds. In general, it is important that more effort is taken to get a full idea of what workarounds are used in text fields.

The dataset used in this study comes directly from practice which contributes greatly to the external validity. The dataset is however very specific. It contains one type of text field from a database for administration of one financial product. It is important to see if the same result will be found in text fields with other functions and from other companies.

Clustering is an unsupervised method. In this study, we knew what we were looking for, through the interviews we conducted. Now we know we were able to find workarounds, but we do not know what other information can be gained by examining the clusters we created. There may be workarounds in the dataset we do not know of, that are clearly visible in the clusters. Building on the knowledge that workarounds have characteristics that can be used to form clusters, in future research the algorithms could be applied truly unsupervised to form the strongest clusters, and then the clusters could be analyzed to see if workarounds present themselves in these clusters.

Finally, this study confirms that workarounds are very diverse and not one method for detection will find all workarounds. We recommend that the results of the Resilience Mining Thesis Circle should be used as input for researching the possibilities of using a combination of algorithms to find different types of workarounds in data and text.

Acknowledgments

This master thesis is the result of my graduation that completes the master Business Process Management and IT at the Open University. I hereby want to thank my supervisor Lloyd Rutledge for his guidance and advice during the process of conducting this master thesis, and second reader Guy Janssens for his time and constructive feedback. Also, thanks to my fellow students for their support and the inspiring discussions.

From Achmea I would like to thank Michel de Boer, head of the department of Pensions of Achmea, for giving me the opportunity to use data from Achmea Pensions for this study. I want to thank my Achmea supervisors Reinder de Haas and Hennie van der Veer for their support, Jens Vogel for obtaining the dataset and Cyril Cleven for anonymizing the dataset.

Finally, I want to thank my partner for his moral support and patience.

Janneke Spronk

References

Alter, S. (2014). Theory of Workarounds. *Communications of the Association for Information Systems, 34 (55)*.

Brown, J., & Duguid, P. (1991). Organisational learning and Communities-of-Practice: Toward a unified View of Working, Learning and Innovation. *Organization Science, 2.1*.

Cate ten, R. (2020). Detecting shadow IT in free text fields using text mining and classification.

Huisman, J. (2020). Resilience Mining: Detecting Shadow IT in IT Systems with Data Classification.

Huuskonen, S., & Vakkari, P. (2012). "I Did It My Way": Social workers as secondary designers of a client information system. *Information Processing and Management*.

Kopper, A., & Westner, M. (2016). Towards a Taxonomy for Shadow IT. *The Americas Conference on Information Systems.*

Koskamp, M. (2020). Mining for workarounds in information systems using outlier detection.

Outmazgin, N., & Soffer, P. (2013). A process mining-based analysis of business process work-arounds. *Software Systems Model, 15*, 15, 309-323.

Patterson, E. (2018). Workarounds to Intended Use of Health Information Technology: A Narrative Review of the Human Factors Engineering Literature. *Human Factors*, 60 (3) 281-292.

Petrides, A., McClelland, S., & Nodine, T. (2004). Costs and benefits of the workaround: inventive solution or costly alternative. *The International Journal of Educational Management*, 100-108.

Provost, F., & Fawcett, T. (2013). Data Science for Businesses. United States of America: O' Reilly Media.

Rouwendal van, J. (2020). Tekst en association rule mining voor detecteren van workarounds in vrije tekst die gestructureerde invoer omzeilen.

Sandfort, T. (2020). Resilience mining: identifying workarounds in IT systems using association rule data mining.

Saunders, M., & Thornhill, A. (1997). Research Methods for Business Students. Edinburth Gate: Pearson Education Limited.

Schubert, E., Sander, J., Ester, M., & Xu, X. (2017). DBSCAN Revisited, Revisited. Why and How You Should (still) Use DBSCAN. *ACM Transactions on Database Systems*.

Silic, M., & Back, A. (2014). Shado IT- A view from behind the curtain. *Computers & Security*, 45, 274-283.

Spoel van der, P. (2020). Detecting workarounds with data clustering algorithms to improve Information Systems.

van de Weerd, I., Vollers, P., Beerepoot, I., & and Fantinato, M. (2019). Workarounds in retail work systems: prevent, redesign, adopt or ignore? Proceedings of the 27th European Conference on Information Systems .

Vos, L. (2018). *Analyzing and Understanding workarounds in an IS to improve Business Processes in a Multinational Corporation.* Amsterdam: Vrije Universiteit.