VU University Deloitte

Master Thesis Business Analytics

Practitioner insights in the application of advanced text analytics

The best practices of clustering and topic modeling in e-discovery

by:

Sjanne NAP

Supervisors:

Dr. Ruben Izquierdo Bevia

Dr. Mark Hoogendoorn

Drs. Scott Mongeau

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

July 2015

This page intentionally left blank.

VU University Deloitte

Master Thesis Business Analytics

Practitioner insights in the application of advanced text analytics

The best practices of clustering and topic modeling in e-discovery

by:

Sjanne NAP

Supervisors:

Dr. Ruben Izquierdo Bevia

Dr. Mark Hoogendoorn

Drs. Scott Mongeau

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

July 2015

VU University, Amsterdam

Faculty of Sciences

De Boelelaan 1105

1081 HV Amsterdam

Deloitte, Amsterdam

Analytics professional services

Gustav Mahlerlaan 2970

1081 LA Amsterdam

This page intentionally left blank.

Declaration of Authorship

I, Sjanne Nap, declare that this thesis titled, 'Discovery workload reduction by clustering and topic modeling' and the work presented in it are my own. I confirm that:

- This work was done wholly during the master project Business Analytics.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Efficiency is doing things right. Effectiveness is doing the right things."

Peter Drucker

Abstract

The amount of unstructured data in business is immense. It characterizes 80 percent of the total data-consumed and is increasing at an exponential rate by approximately 60 percent per year. This huge amount has aroused great interest into the application of text analytics. Highly competitive organizations and companies apply these relatively new set of techniques to gain insights to improve business processes.

This study specifically investigated the use of text analytics for potential application in Deloitte's e-discovery projects. The use case of e-discovery has an important role within Data Analytics professional services. However it can be very time-consuming and a costly. An intelligent document review process is needed that reduces risks to a minimum and in which the effectiveness and efficiency of the process are maximized. This can be achieved by combining advanced text analytics and human expertise.

The main subject matters examined are the statistical methods *clustering* and *topic modeling*. The combination of these statistical methods can provide initial insights into the contents of the documents collected prior to fundamental reviews and scoping. It can be of great added value in addition to the traditional process of human review and keyword search.

In this thesis the following research question is answered:

"What are the clustering and topic modeling techniques that need to be considered as best-practices in e-discovery projects?"

Through literature and practical research best practices are developed in the form of guidelines. Both clustering and topic modeling, and the preceding text analytics process with methods associated have been studied step by step.

The final results show, amongst others, that the techniques investigated are powerful in focusing discovery efforts, but also computing intensive. Therefore, it is recommended to use this process as a second step in knowledge acquisition after initial refinement and reduction of the dataset.

Further, there is no single automated diagnostic algorithm that can be used to assess whether a particular unsupervised text analytics algorithm is better or worse in terms of results. The implication is that human review is always needed to validate the results as they are semantic in nature. Experts trained in the application of text mining and subject matter experts have to be involved as they understand the requiremetnts for investigation and are able to finally judge on the results.

This page intentionally left blank.

Acknowledgements

This research report written is part of the Master Business Analytics, a study at the VU University Amsterdam. It describes a problem of the company Deloitte Nederland, which is investigated, analysed, objectified and solved during an internship of six months.

Due to the growing interest in the market, of Deloitte Nederland and myself, research is performed on the topic 'advanced text analytics'. Of specific interest are the statistical techniques 'clustering'and 'topic modeling'. The central question to be answered in this report is:

"What are the clustering and topic modeling techniques that need to be considered as best-practices in discovery projects?"

I would like to thank everyone who contributed to this project. First of all, I would like to thank Deloitte Amsterdam for making it possible to conduct this research. I would like to thank my colleagues for the nice work atmosphere, their interest and enthousiasm.

Furthermore, I would like to thank my supervisors, Drs. Scott Mongeau and Dr. Ruben Izquierdo Bevia, for their time taken, the ideas discussed and the feedback they have provided.

Lastly, I would like to thank Dr. Mark Hoogendoorn, the second reader of this paper, for his involvement and support regarding the commitment to the study Business Analytics.

This page intentionally left blank.

Contents

List of Figures

List of Tables

This page intentionally left blank.

Abbreviations

ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
CAR	**C**omputer **A**ssisted **R**eview
CEO	**C**hief **E**xecutive **O**fficer
CFO	**C**hieve **F**inancial **O**fficer
CTM	**C**orrelated **T**opics **M**odeling
DF	**D**ocument **F**requency
DTM	**D**ocument **T**erm **M**atrix
DTTL	**D**eloitte **T**ouche **T**ohmatsu **L**imited
E-Discovery	**E**lectronic **D**iscovery
EDRM	**E**lectronic **D**iscovery **R**eference **M**odel
EN	**EN**tropy-based ranking
ESI	**E**lectronically **S**tored **I**nformation
FERC	**F**ederal **E**nergy **R**egulatory **C**ommission
ICA	**I**ndependent **C**omponent **A**nalysis
IR	**I**nformation **R**etrieval
LDA	**L**atent **D**irichlet **A**llocation
LSI	**L**atent **S**emantic **I**ndexing
MCMC	**M**arkov **C**hain **M**onte **C**arlo
NLP	**N**atural **L**anguage **P**rocessing
PAM	**P**artitioning **A**round **M**edoids
PCA	**P**rincipal **C**omponent **A**nalysis
PST	**P**ersonal **S**torage **T**able
SEC	**S**ecurites and **E**xchange **C**ommision
SMART	**S**ystem for the **M**echanical **A**nalysis **R**etrieval and **T**ext
SPE	**S**pecial **P**urpose **E**ntity

SVD	**S**ingular **V**alue **D**ecomposition
TAR	**T**echnology **A**ssisted **R**eview
TC	**T**erm **C**ontribution
TF	**T**erm **F**requency
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
TS	**T**erm **S**trength
UTF-8	8-bit **U**nicode **T**ransformation **F**ormat
VEM	**V**ariational **E**xpectation **M**aximization

Chapter 1

Introduction

Advanced text analytics offers a set of techniques of growing interest. The application of these relatively new approaches could be of great added value in improving the efficiency as well as effectiveness of analytics projects. The application investigated in this research is discovery, one of the core services provided by data analytics professional services. The human workflow in e-discovery processes is labor and cost intensive. Obtaining quick insights into large sets of documents under review by use of *clustering* and *topic modeling* could reduce this overhead. The aim of this research therefore is to investigate best practices in the use of these two statistical modeling techniques within e-discovery projects.

This introduction proceeds with a more extensive description of the research motivations, practical applications, e-discovery services and research objectives.

1.1 Research motivations: text analysis

The amount of unstructured data in business is immense. Often, we are not aware, but textual data is generated in each company daily. A well-known case is sending e-mails. Further examples are twitter messages, documents on the internet, blogs and customer reviews on products and services. Unstructured data characterizes 80 percent of the total data-consumed, which means that structured or numeric data only involves 20 percent. In addition, the amount of unstructured data is increasing at an exponential

rate by approximately 60 percent per year [Chakraborty and Pagolu, 2014, Gupta and Lehal, 2009].

This huge amount of unstructured data, has aroused great interest in the market. Text analytics is applied by highly competitive organizations and companies to gain insights to make better decisions. The text analysis market is growing, especially within marketing, sales and customer service domains [Halper et al., 2013].

However, advanced text analytics is not yet frequently applied in the Dutch market. Unstructured data can realize added value when the resources and tools are available to process large amounts of data. However, to date the solutions and software available on the market to analyze textual data in a methodical way are not broadly applied. In contrast to numerical data, machines have a difficulty in understanding natural language as it is rich in both its structure and form [Chakraborty and Pagolu, 2014]. A document with text might for example contain various languages, difficult grammatical sentences, dialects, idioms, jargon and acronyms. Beyond this, natural language can be quite ambiguous, which is often considered as the main problem. A sentence can be parsed in different ways and a word can have different meanings in the same sentence or statement. Vice versa, various words can have exactly the same meaning. Finally, a lack of understanding of the core techniques and methods hinders the proliferation of text analytics. Perhaps this is the most important and biggest problem, since understanding a topic is the foundation for successful practical application.

1.2 Backgound: Deloitte Nederland

'Deloitte' represents the cooperation of the independent Deloitte Touche Tohmatsu Limited ('DTTL') member firms that are located in more than 150 countries. More than 200,000 professionals work on accountancy, financial advisory, risk management, consulting, tax advisory and other related services.

The DTTL member firm in the Netherlands is Deloitte Nederland. Here, over 4,500 professionals work in the various areas. In this research, the focus will be on the department Risk Services, which assists clients in detecting, analyzing, reviewing and managing risks. Of specific interest is the Data Analytics professional services group within Risk Services.

1.2.1 Data Analytics professional services

Data Analytics is the practice of using data to drive business strategy and performance. It spans all of the Deloitte functional businesses to address a continuum of opportunities in Information Management, Performance Optimisation and Analytics Insights.

Data Analytics professional services carries out assignments in the fields of audit, compliance, financial and operational risk analysis, and financial crime analytics. The issues of projects performed within this team are mainly solved by use of advanced analytics including data conversion, machine learning, predictive modeling, visualization and text classification. To date, this largely is focused on structured data analytics, but there is an immense interest in the application of advanced text analytics. Text analytics could be of great added value to improve business processes and therefore this study investigates the use of text analytics for potential application in Deloitte's e-discovery projects.

1.3 E-discovery

E-discovery concerns the examination of information in compliance and fraud cases in order to find potentially relevant documents for evidence in litigation. Prior to this process, the requesting party asks for research to be conducted, which obligates the responding party to supply the relevant documents after a reasonable search has been conducted. Each party involved in this process, including the attorney and internal auditors, reviews immense amounts of data in a limited time. When these documents are electronic, also known as Electronically Stored Information (ESI), this process is called electronic discovery.

1.3.1 E-discovery process

The process that is followed during e-discovery projects contains a fixed number of steps, which is clearly described in the well-known Electronic Discovery Reference Model (EDRM) diagram [EDRM, 2015]. The diagram shown in figure 1.1 provides a theoretical view of an iterative e-discovery process. This model does not represent the only possible process. Each process model follows similar steps, though likely in a different order,

or with different dependencies on or connections to earlier stages. The EDRM model therefore serves as a representative example.

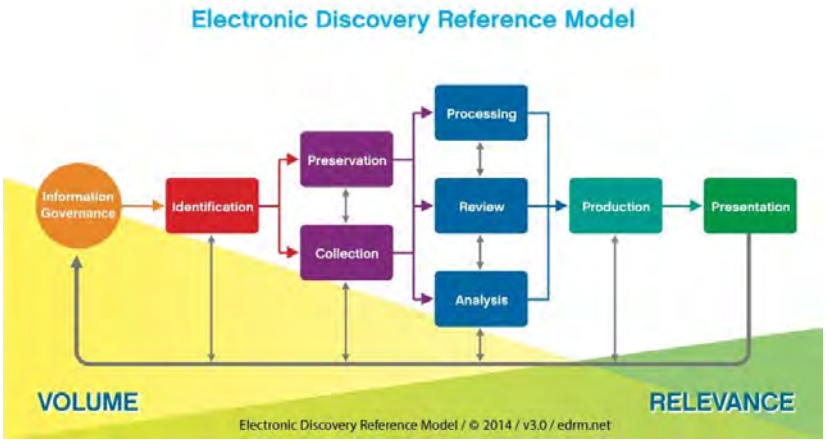


FIGURE 1.1: The EDRM model

Source: http://www.edrm.net/resources/edrm-stages-explained

The EDRM process starts on the left at Information Governance and proceeds through a number of steps to the conclusion with the Presentation step. However, Information Governance actually does not belong to the e-discovery process itself since it is a step which is taken prior to undertaking e-discovery. This step is a business process that is intended to lower the risks and expenses in case the company receives a discovery request. This is done by improving the process of storing and disposing Electronically Stored Information (ESI).

When a discovery request is received sources are investigated. This regards business entities, as well as IT systems, people and files. In this phase often interviews are conducted with the client or other individuals involved in order to identify the sources of potential relevance, how and where their data is stored, etcetera, and most important, the custodians[1] and their ESI. However, as the relevant sources, individuals and thus information changes over time, the process regularly returns to the investigation step.

The Preserving and Collecting boxes both concern the ESI identified. Within the preserving phase the information of potential relevance is protected against unintended and illegally events, such as manipulating or destroying the information. The collection box stands for the collection of ESI determined as potentially relevant in the identification

[1]Custodians: Persons of interest

phase in order to investigate these documents further in the rest of the process. This regards the content as well as the corresponding metadata[2].

The data collected is further processed by converting it to items that are suitable for the review and project phases. This includes, for instance, the extraction of e-mails from Outlook and converting them to plain text. The documents are then manually reviewed in order to acquire new information and, in the case of relevance, to use it when presenting before audiences. This is often accompanied by context and content analysis of the documents collected in order to improve the effectiveness and efficiency of the document review process. Nowadays, keyword search is the most commonly used method. In this approach, after conducting initial research in the case, keywords are determined by lawyers. These keywords are the basis for searching, identifying, ordering and thereby determining the importance of relevant documents before they are manually reviewed.

Lastly, the documents determined as being relevant as evidence are prepared and produced in their original form, such that they can be presented before audiences at, for example, legal trials in order to validate the facts, a position, or to persuade others[3].

1.4 Aim of research

In litigation, discovery plays a major role and therefore it can take up to 25% of litigation costs. This is largely due to the highly manual nature of the document review process. Until recently, the traditional approach has been typically followed, which means each document or mail is manually reviewed by large teams of reviewers in order to determine the responsiveness, or level of relevance. As the amount of data typically involved is immense and is growing steadily, it is a very time-consuming and costly process. It frequently consumes more than 70% of the total costs in the discovery effort [Pace and Zakaras, 2012].

An intelligent document review process is needed that reduces risks to a minimum and in which the effectiveness and efficiency of the process are maximized. This can be achieved

[2]Metadata: data about data. It contains the properties of data, such as the author, receiver(s), date and subject in case of an e-mail

[3]For further information on discovery, I would like to recommend the following documentary: The decade of discovery *by Joe Looby*

by combining efficient workflows and human expertise, but most importantly through the skillful use of the right analysis techniques and technologies within the workflow.

The majority of the companies already use keywords search in their discovery processes, but this technique is increasingly inadequate. Since the keywords are determined a priori, they do not capture all information needed and therefore do not provide a representative set of all potentially responsive documents. Therefore, more advanced text analysis techniques have been introduced, including clustering and topic modeling. The combination of these statistical methods can provide rapid initial insights into the contents of the documents collected prior to fundamental reviews and scoping. It can be of great added value in addition to the implementation of keyword search and therefore is the subject of this study.

The research objective in this masters thesis is to make best practice recommendations to the FCA professional services team at Deloitte Risk Services on the possible use of clustering and topic modeling. The best practices will provide guidelines for potential application in the future.

The question to be answered is:

"What are the clustering and topic modeling techniques that need to be considered as best-practices in e-discovery projects?"

1.5 Structure of report

Clustering and topic modeling can not simply be applied to raw textual data. A text analytics process precedes. Therefore, this report continuous with a review into the background of the text analysis process steps, methods and techniques investigated in this study. The next chapter covers the Enron case and dataset, which is applied for practical experiments. Chapter 4 to 7 describe these experiments per step taken. Chapter 4 concerns preprocessing of the data, chapter 5 discusses dimensionality reduction, chapter 6 treats clustering, and the last chapter contains topic modeling. Each of these four chapters have the same structure. First, the state-of-art is described, then the methods and techniques used in the experiments are covered, and finally the results are

discussed. The overall results are contained in chapter 8, which includes the conclusion and discussion with recommendations for further research.

This page intentionally left blank.

Chapter 2

Literature review: background

At high-level this research concerns text analytics. In order to obtain a better view of this topic, this chapter gives an introduction based on literature review. The subsections treat the generally accepted text analysis process, domains, uses and users of text analytics. Subsequently, the application of text analytics in e-discovery is discussed. Hereby, the specific methods of interest, clustering and topic modeling, are treated separately.

2.1 Introduction to text analytics

Text analytics is defined and applied in a number of different ways. It has no set definition, which is probably due to the fact that it is a relatively new topic of growing interest. However, the company Hurwitz and Associates defines a clear and comprehensive interpretation[1]:

Text analytics is the process of analyzing unstructured text, extracting relevant information, and transforming it into structured information that can be leveraged in various ways.

Hurwitz & Associates

Traditionally, this description is coupled to the term *text mining*. However, text mining is considered to be the application of machine learning techniques and statistical methods in order to examine the content of documents and to assess them individually. The

[1] [Halper et al., 2013]

term *text analytics* is a broader concept which originated over time. Text analytics is an interdisciplinary field that uses techniques from the fields of statistics and machine learning, as well as computational linguistics, Natural Language Processing (NLP), information retrieval, and data mining. [Grimes, 2015, Gupta and Lehal, 2009, Halper et al., 2013] Although this leads to a variety of use cases, one general process is followed.

2.1.1 Text analytics process

A text analytics process generally consists of five different steps as shown in figure 2.1 [Chakraborty et al., 2013].

The first step involves the collection of unstructured data relevant to the specific study. Information is gathered from multiple sources in order to be able to undertake the research completely. Several examples include the collection of documents, emails, user comments and unstructured data from web pages.

After collection, the textual, unstructured data is preprocessed. The aim of the preprocessing step is to manipulate the text into data that can be understood and 'learned' by the system, such that the best results are achieved.

First, the format and character set of the dataset is determined. Then, the words are manipulated via different methods, including stemming. Stemming involves reducing words to their root form (e.g. removing suffixes or changing to a simpler form if a verb), applying parts of speech tags (e.g. categorizing words as a verb or noun) and changing words into a more commonly recognized synonyms where applicable (e.g. changing car or automobile to the root word auto).

Subsequently text is filtered by for example removing punctuation, numbers and stop words (i.e. common articles and conjunctions such as the, a, an, and, etc.). This process step is meant to remove the terms that do not contain meaning, such that only relevant information is retained when transforming the textual data into numeric data. The structured data obtained is generally shown in a term-by-document matrix, of which the size is determined by the number of documents and number of terms. As such a matrix can become exponentially large, dimensionality reduction can be applied to control the size.

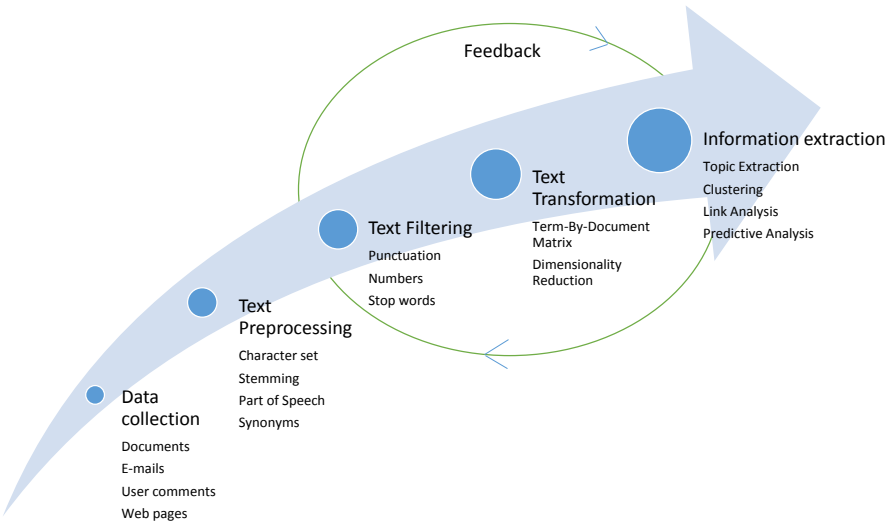


FIGURE 2.1: Text analytics process

Finally, the text analytics process is completed by extracting the information necessary to solve the 'problem'. This concerns several methods including topic extraction, clustering, link analysis or predictive analysis such as classification[2].However, sometimes it is necessary to cycle back to one of the previous phases until the desired information is extracted.

This text analysis process is followed in this research to answer the main question. The steps are discussed in the next chapters together with the corresponding methods that belong to various domains.

[2]For a more detailed description of the possible methods and techniques used in text analysis, I refer to the literature used.

2.1.2 Text analytics domains

Text analytics methods and techniques can on high-level be classified into a several sub-domains. Figure 2.2 below shows the domains defined by SAS, one of the leading providers of text analytics software [Chakraborty et al., 2013].

- **Advanced search:** Advanced search uses search queries to find information needed. Examples are keyword search in e-discovery projects or the search in a catalog for documents of interest.
- **Text mining:** Text mining is a combination of statistics and machine learning, often involving deriving patterns and pattern learning from unstructured data.
- **Taxonomies and ontologies:** Within this domain information is organized. Entities and its attributes are classified, grouped and related within a hierarchical structure. An example is the framework of departments of a company (taxonomy) and their internal relationships (ontology).
- **Natural language processing:** Concerns the interaction between human and machine, and the ability of the machine to understand what the human has written. This includes various tasks, such as summarization, topic recognition, parsing and part-of-speech tagging.
- **Content categorization:** The categorization and extraction of documents in order to classify them individually to one of the known labels or categories.
- **Sentiment analysis:** Analysis with the goal to determine the opinion and attitude of a writer towards a specific topic or overall emotional polarity of a text. It is also known as opinion mining

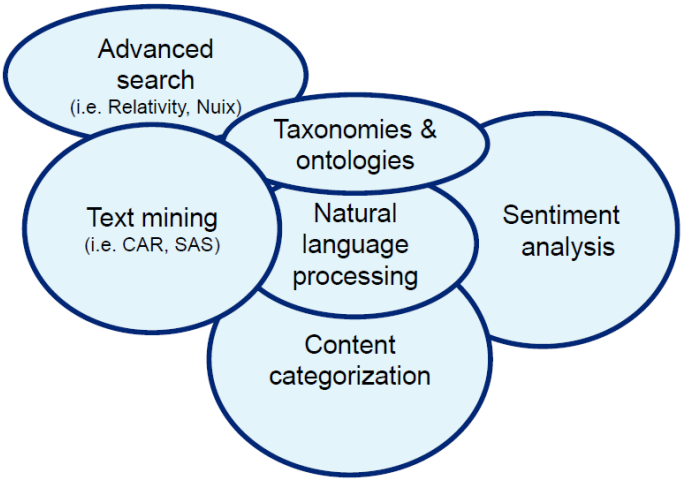


FIGURE 2.2: Text analytics domains

Source: [Chakraborty et al., 2013]

2.1.3 Uses and users of text analytics

As per a 2014 market study by Alta Plana[3] on text analytics solutions, text analytics is considered to be mainly applied in five industries: life sciences and clinical medicine, legal profession, consumer-facing businesses, public administration and government, and scientific and technical research [Grimes, 2014]. The primary applications of text analytics include financial services, life sciences / clinical medicine, competitive intelligence, customer experience, and product management, but also the search-driven e-discovery and fraud.

This varieties of application results in a a broad array of information types analyzed. Alta Plana reported a list of the fourteen most analyzed types of textual information, including social media, online review, customer and market surveys, news articles, blogs, and e-mails. Although the largest source of text analytics is the social field, the sorts of information extracted differ per use case. Of growing interest are the extraction of topics, events, entities, and personal characteristics including sentiment and other subjective information.

[3]Alta Plana is an IT strategy consulting corporation with a focus on the use of advanced analytics and the positioning of the analytics products and market to the providers and user organizations.

Due to the diversity, it can be concluded that a single application, technology, solution or user does not exist. Users and uses differ per industry, business operation, source of information and goal [Halper et al., 2013].

2.2 Text analytics in e-discovery projects

As described above, e-discovery is one of the primary applications of text analytics, which is heavily used in legal profession. It concerns a whole process that the experts Grossman and Cormack define as follows[4]:

E-discovery is the process of identifying, preserving, collecting, processing, searching, reviewing and producing Electronically Stored Information that may be relevant to a civil, criminal, or regulatory matter.

M. R. Grossman and G. V. Cormack

Although a limited amount of research has been conducted on e-discovery compared to for example the clinical areas, it is an important area for future investigation [Eves et al., 2015]. The most studied advanced analytics technique is Computer Assisted Review (CAR), also known as Technology Assisted Review (TAR) or predictive coding. Within CAR, a set of documents are reviewed and coded manually as responsive or not responsive, after which they are provided to the system as input. Based on the manually coded documents, the system makes decisions and codes the rest of the documents itself. This is an iterative workflow process which is continued until 'enough' documents are reviewed and labeled [Cormack and Grossman, 2014].

CAR is able to save much work and costs in litigation, as only reasonable effort is needed to find relevant documents. However, there are more possibilities in which text analytics can help in making document review more effective and efficient. The e-discovery company kCura[5] proposes eight principles for e-discovery projects [kCura, 2015]:

1. Tie the conversation together with e-mail threading

[4] [Grossman and Cormack, 2015]

[5] kCura is an e-discovery software company and developer of Relativity, software offering solutions for corporations and law firms as well as government with the aim to support e-discovery processes.

2. Review near-duplicate emails at the same time
3. Quickly batch foreign language documents to the right reviewers
4. Expand your awareness of critical case language
5. Uncover conceptually similar documents
6. Prioritize your review with document clusters
7. Categorize case data with sample documents
8. Tackle complex cases with Computer-Assisted Review

As can be seen from above, CAR is the last step in kCuras process for document review. Before CAR can be applied, it is important to divide and group documents based on first insights in language, concept and duplicates *(Step 1 to 4)* [Barnett et al., 2009]. Then, clustering is used to prioritize documents that can be manually reviewed in Computer-Assisted Review in order to achieve a good performance from the machine. Clustering is a much explored advanced analytics technique which can be used as a preprocess for CAR *(see the next section)*.

From legal view, it is not necessary for discovery processes to be perfect and to obtain perfect results. According to Rule 26(g) of the U.S. Federal Rules of Civil Procedure, the effort should be 'reasonable', which of course is a legal question [Government, 2015]. Therefore, the aim of the information retrieval systems using text analytics is to obtain reasonable results that approximate reasonable effort in human review.

2.2.1 Human vs. System performance

Human reviewers do not always agree on responsiveness and do not always give documents the same label. Multiple studies have discussed the performance of human review, including the TREC Studies of 2006 to 2008. They all came to the same conclusion that the level of agreement between reviewers is between 70 and 76%. The reason for this is that during reviews various judgements have to be made. It includes judgements about the document, the content and the situation. Judgements might be made inclusively, but sometimes also less inclusively. This depends on the aim of the reviewer when labeling the documents. Everyone has their own goals, which may change over time. This

strategic judgement affects the bias of the reviewer and affects the agreement between individuals. However, bias differs per case and per individual [Roitblat et al., 2010]

Research into the performance of text analytics resulted in the growing idea that automated analytics tools are able to substitute the current practice of keyword search and full manual review. Most research compared the traditional approach with Computer Assisted Review and came to the conclusion that the performance of machine learning is approximately at the same level as human performance. Sometimes the system performance was slightly worse (5% less relevant documents were found) and in some cases the performance was even better. Besides, as multiple papers suggest a workload reduction between 30 and 70%, CAR and other advanced text analytics techniques such as clustering are considered promising from an efficiency standpoint [Cormack and Grossman, 2014, Eves et al., 2015].

However, up to today the most commonly used approach is still the traditional approach, containing human review and keyword search. This might be due to the fact that change and innovation often come with resistance. In this case, judges resisted for a long time and still do. Yet, the acceptance of text analytics methods in e-discovery projects has become inevitable as the amount of data and costs are expanding. The desire to solve legal issues and thereby to decrease the expenses, is gaining acceptance by legal entities and practitioners.

However, full automation of discovery is not possible. Lawyers and other individuals involved in the case need to know and understand the content of the data. They have to know their evidence, to be able to respond to the arguments of the counterpart and they have to devise and carry out a strategy in order to win. Human knowledge, experience and judgement is needed [Barnett et al., 2009, Reissner and Hochman, 2012, Roitblat et al., 2010]. In general, the following conclusion can be drawn:

- The use of text analytics with the aim to prioritize the order in which the documents are reviewed, can be treated as safe. It can be used for live document review.
- The use of text analytics as a second reviewer, may also be used cautiously. However, the assumption has to be made that the human reviewer has not consistently missed relevant documents.

- The use of text analytics to automatically exclude non-relevant documents is promising, but not yet fully proven. In highly technical and clinical areas a lot of research has been done, so it can be used with a high degree of confidence, but not yet in other areas. More research is needed.

As mentioned before, the traditional approach is not sufficient but the use of advanced text analytics should be considered as promising. The combination of clustering and topic modeling, as defined below, can be used to organize documents by priority based on similarity and content, and so they are good, safe methods to use in document review processes. It is a good approach to prioritize by use of clustering and topic modeling next to or instead of keyword search. Besides, in a later stage the results could be used as the basis for the promising application of CAR [Eves et al., 2015].

2.2.2 Clustering and topic modeling

Document clustering is a statistical method that is often used as a foundation for different purposes, such as summarization, navigation, information retrieval, and, the main purpose, efficient document categorization. This variety of use cases has resulted in extensive studies in multiple disciplines, including e-discovery [Aggarwal and Zhai, 2012, Beil et al., 2002, Liu et al., 2005].

Clustering is mostly defined as a category segmentation method since it groups related documents. However, in contrast to categorization methods, clustering is a method of unsupervised learning. The documents are not coded, which means it is not known in advance to which cluster a document belongs. A general clustering algorithm works as follows: It defines a vector for all documents separately, with topics describing the content. Based on these topics similarity measures are calculated in order to determine the fit of each document in each proposed cluster [Gupta and Lehal, 2009]. Finally, documents containing similar content are grouped together, such that within one cluster all documents are similar and such that they are different from the documents in the other clusters. Thus all clusters are unique and contain unique documents. However, finding the relevant topics and thus content from the set of documents is difficult. Effective and efficient knowledge retrieval is challenging due to the “curse of dimensionality” that is

related to natural language [Milios et al., 2006]. The three largest difficulties within clustering are [Beil et al., 2002]:

- The high-dimensional data. The number of terms over the various documents quickly increase to an amount of at least 10.000. Since most terms only occur in a couple of documents, this results in sparse data. Clustering methods need to be able to deal with sparsity or dimensionality reduction should be applied otherwise.
- An immense amount of documents. Clustering methods have to be efficient in their run time and extensible if the amount of documents increases.
- The comprehensibility and interpretability of the clusters. Results need to be understood by users in order to be able to search the clustering.

As previous research has shown that high dimensionality and sparseness make the clustering performance decline very quickly, this is probably the largest challenge of clustering [Liu et al., 2003]. Therefore, usually a general approach is followed. First documents are preprocessed in order to clean the data. Here, already some reduction in the high-dimensionality is addressed by removing information that is of no added value. Next, a numerical representation method is chosen in which the content of the documents is converted such that the system 'understands' the data. The resulting dataset is used to apply efficient clustering methods. Then, the clusters can be used for various purposes. Associated best practices are [Milios et al., 2006]

- Get first insights in the content of the unknown documents quickly
- Get to know the part of custodians in a certain case or topic
- Review the immense datasets in relatively little time
- Group documents based on content and topics such that review is finished faster
- Separate documents of no relevance for legal proceedings.

The first best practice is the purpose of this study. However, first insights in the content of the documents are not immediately obtained by the clustering results. As the results only indicate the documents that are similar to each other, the clusters are not easily interpreted. Therefore, clustering is often studied in combination with topic modeling.

Topic models, also known as probabilistic topic models, are statistic algorithms that have the aim to find underlying thematic infrastructure of a set of documents by use of unsupervised learning. This means documents do not need to be labeled. Topic models search for and discover patterns in the term frequencies within documents, after which the documents which similar patterns are connected. However, documents do not necessarily contain and belong to one topic. They are mixed models. Documents might contain multiple topics and the topic distributions might differ amongst and across document.

Previous research has shown that topic modeling works very well and is able to find unexpected thematic patterns in a set of documents that would not have been found by the human otherwise [Blei and Lafferty, 2009, Grossman and Cormack, 2015, Sethi and Upadrasta, 2012]. Still, as described in previous research , the performance of text categorization methods and techniques might differ per dataset. A perfect methodology does not exist. This also holds for the combination of clustering and topic modeling techniques [Liu et al., 2003]. Still, good working methods can be determined based on stability and the content of topics obtained.

This page intentionally left blank.

Chapter 3

Enron case

The case used within the practical experiment is the well-known Enron case, also known as the "Enron scandal". It involves one of the biggest company scandals in history and it is also one of the biggest collapses in an audit case.

3.1 Case description

The company 'Enron'was founded by Kenneth Lay in 1985 through merging the utility company 'Houston Natural Gas'and the gas pipeline company Internorth of Omaha. Enron was located in Houston, Texas, and possessed 37,000 miles of pipelines in- and outside the United States with the aim of transporting natural gas between utilities and producers. Within a few years regulatory changes arose which resulted in an increased flexibility and supply of natural gas. Enron, as largest owner of pipelines within the United States, profited from this. In the 1990s Enron was growing. The pipeline business even expended abroad. Projects were established and managed in Central and South America, Eastern Europe, the Middle East, Africa, China, and India [Diesner et al., 2005, Healy and Palepu, 2003].

Besides the business in pipelines, the company started to construct power plants and became known as an energy broker. However, they were most distinctive by creating new, innovative markets such as weather forecasts, bandwidth of electronic communications, and ad time. This made them one of the most innovative companies in the United States, at the end of the 1990s. In addition, Enron belonged to the top ten business organizations

with the most revue in the United States. In 2001, Enron had approximately 21,000 employees and was located in 40 countries [Li, 2010].

This rapid growth and innovation arose in part through the ideas and efforts of Jeffrey Skilling, who was hired via McKinsey in 1988. In that year, he proposed a trading model for Enron. Jeffrey Skilling perceived that the pipeline business could not be used for competitive advantage and therefore new markets were created. However, it was questionnable whether this success and advantage could be perpetuated, considering the competitors that could enter the market. In addition, the creation and expansion to new markets entailed risks. For example, the expansion to countries outside the United States was accompanied by political risks due to different laws and regulations.

In addition to risks taken, Enron's financial reporting and accounting was questionnable. At the start of 2000 financial losses arose, which where kept hidden. Jeffrey Skilling, then CEO of Enron Finance, invented a way to make this possible: mark-to-market accounting. This approach makes use of the current accounting principles in which the actual value of an asset at a certain moment is determined by a prediction on the future profits. In the Enron case, when the revenue was less then expected, the asset loss was not reported, but was removed from the ledger and transmitted to a so-called "off-the-books corporation", such that the loss was hidden. This approach gives the impression that any loss can simply be written off, a company does not need profits and financial fundamentals are flexible to interpretation.

As the losses continued, the company looked more profitable than it actually was. Andrew Fastow, CFO of Enron since 1998, contributed to this image by creating special purpose entities (SPE). These entities were used to hide the assets that had failed. People or organizations investing in these SPEs, payed for these losses, but obtained shares of Enron in return in order to compensate [Seabury, 2015].

It went increasingly downhill and in October 2001 Enron announced a loss for the first time. Since the company closed an SPE at the same time, such that the issue of shares was not needed, the U.S. Securites and Exchange Commision (SEC) started an investigation into the company as it was suspected. The SEC discovered a debt of approximately $628 million and an amount of losses of $590 million. This was made public and resulted in a massive decline in Enrons share prices. Shareholders lost $11 billion and finally, on December2, 2001, Enron went bankrupt.

Eventually, sixteen employees pleaded guilty for their involvement in fraudulent actions within the company. In addition, four individuals were judged guilty.

This page intentionally left blank.

Chapter 4

Preprocessing

As mentioned before, a text analytics process starts with the collection of data and the preprocessing of text. The data collected for this experiment is a sample of the Enron dataset, and is preprocessed in four steps: document triage, text segmentation, post-tokenization, and text representation. All steps are described below in the 'state of the art'-section and in the second section regarding the experiment performed.

4.1 Enron dataset

The Enron dataset was made publicly available by the agency FERC (Federal Energy Regulatory Commission) in May 2001. This dataset included 0.5 million emails and attachments from exactly 158 employees of Enron during a period of 3.5 years. Today, this dataset is widely used for (academic) research into the processes of the large organization and for the improvement of e-discovery projects in business. However, it needs to be noted that some emails are removed from the dataset due to legal issues and privacy. Besides, the corpus also contains private conversations between employees who were not part of the investigation into Enron [Diesner et al., 2005].

The dataset used for this experiment is a selected subset consisting of approximately 70,000 documents from six custodians, Darron Giron, Kenneth Lay, Don Baughman, David Delainey, Diana Scholten, and Louis Kitchen, who were principle figures in the Enron scandal. The reason for a sample instead of the whole dataset concerns computing power. The experiment is run locally on a laptop, using the statistical software package

R[1]. R is a well-known package often used by colleagues at Deloitte, and therefore is chosen for this experiment *(Appendix A contains the main characteristics of the system and software used)*. However, on a single machine R can not handle too large amounts of data. Therefore, a sample is chosen with an amount of data that is large enough to have differences in content and thus to define and perform proper research. The reason for a selected subset of data is that the scope of this research is to focus on the best practices in methods and techniques. The purpose is not to find persons involved in the Enron scandal. However, the sample taken is a specific subset as it contains e-mails sent by Kenneth Lay and other key players in the fall of Enron.

4.2 State of the art

Text preprocessing is an important step in text analysis, and thus an important component of NLP systems. Within the text preprocessing stage, it is crucial to determine the characters, words, and sentences in a text clearly, since they are the foundation for further processes. These units are passed to other stages in order to perform further analysis.

4.2.1 Document triage

Document triage converts the digital documents that need to be reviewed into the correct form of plain text [Indurkhya and Damerau, 2010]. Initially, the files consist of bits. These bits are linked to a character encoding, that represents a sequence of characters, whereby one or multiple bytes signify one character. Determining the character encoding, ensures the machine is able to read and convert the documents to plain text properly. Often, an automatic encoding algorithm is used that knows the various encoding systems. First, it determines the ranges of bytes used in the documents, in order to make a selection of options. Then, the patterns recognized in the bytes are compared to the possible character encodings. Based on this information, the best fitting encoding is chosen.

[1]http://www.r-project.org/ - [Jockers, 2014, Sanchez, 2013, Stewart, 2010, Team, 2015, Williams, 2014]

The most adopted encoding types are the 8-bit character sets which represent a character in one or more bytes of 8 bits. This holds that $2^8 = 256$ characters can be encoded, of which the first 128 are mostly used for the characters of the ASCII-encoding. Until 2007, ASCII was the most used encoding on websites. However, as ASCII now is a subset of the 8-bit encodings, the 8-bit encodings are used more often. The diagram with ranked website encodings produced by W3Techs[2] in July 2015 even shows that UTF-8 is by far mostly applied by 84.4%.

Next to character encoding identification, the identification of the language is necessary as it determines the natural language and NLP system used in the further process. The language can largely be determined by the character encoding, but not totally. Namely, besides the various symbols, each language indicates its boundaries between linguistic terms differently, such as sentences or words. It therefore is important to identify one or multiple languages used in a document. First, this can be done by determining the set of characters in the text. This decreases the possibilities. Second, the correct language(s) can be identified by use of models that have trained on the distribution of the different characters for each language.

Finally, the unimportant elements of the documents are removed. This includes the elimination of links, headers, images, HTML formatting etc. Now, only the desired content is retained in a text corpus. This text corpus is further investigated and transformed by text segmentation.

4.2.2 Text segmentation

The aim of text segmentation is to convert the text corpus into sentences and words such that the data can be further processed [Indurkhya and Damerau, 2010]. It can be divided into sentence and word segmentation. **Sentence segmentation** determines a sentence, i.e. a sequence of words, by the identification of the sentence boundaries between sentences. **Word segmentation** identifies the words and separates them by use of the word boundaries in the text. Within computational linguistics, this process is often known as tokenization, and the words are referred to as tokens.

[2] www.w3techs.com

Segmenting text has multiple challenges, which depend of the form of language used. For example, unsegmented languages such as Chinese do not have well-defined word boundaries. In that case, additional linguistic and lexical information is needed in order to perform word segmentation. In addition, languages with clear boundaries, such as the English language, often have problems on the level of text segmentation. A sentence generally ends with a dot, but these dots can not be simply removed in order to partition the sentences as these punctuation marks also occur in acronyms and abbreviations. So, the acronyms as abbrevations in a text need to be detected first.

These are just two of the many more examples of difficulties in text segmentation, but generally the following holds: word and text segmentation are related. The two segmentation tasks have to be performed together in order to obtain successful results that are further processed in the post-tokenization step.

4.2.3 Post-tokenization

Post-tokenization concerns the preprocessing methods that transform the corpus after text segmentation is applied. Several methods exist that can be carried out in this process, but it depends on the task which of them are used. As for this research the main purpose is to understand the content of documents, the first three methods discussed in this sub-section are the methods mainly necessary to understand the structure and meaning of text [Milios et al., 2006, Sebastiani, 2002].

- Punctuation removal
- Numbers and symbols removal
- Stopwords removal, i.e. the removal words that are topic-neutral, such as prepositions and articles

Especially the last method, removal of stopwords, is of big importance. Namely, it is shown by van [van Rijsbergen, 1979] that a large frequency over one or all documents does not necessarily mean a term has a significant contribution to the semantics *(see figure 4.1)*. The least and most common words are of no significant importance.

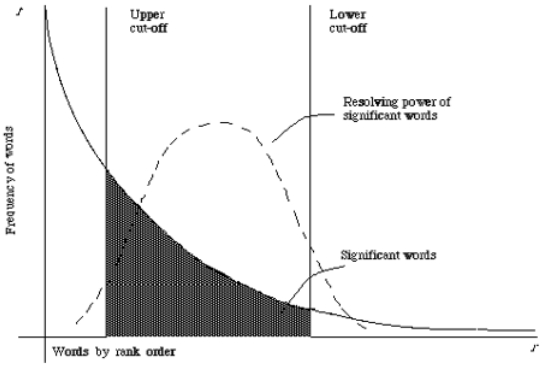


FIGURE 4.1: Word significance in text

Source: [van Rijsbergen, 1979]

Further, two other commonly applied methods are the following:

- Convertion of text to lower case
- Stemming, i.e. the grouping of words that have the same linguistic stem

By converting the words to lower case or to its linguistic stem, similar words having the same content are recognized as one. This reduces the corpus, but retains the content. However, according to Baker and McCalum [1998] stemming might hurt the performance. Namely, a lot of terms contain the same stem, also known as etymology, but have different meaning. An example is the following sequence of words:

Genre, generic, generate, genus and *generic*

They all are derived from the Latin word *genus*, which means kind or class.

This kind of stemming is called derivational stemming, whereby the words and corresponding stem do not necessarily have the same part of speech[3] As derivational stemming is not totally reliable, partly due to the ambiguities in natural language, various people are restricting to inflectional stemming. This concerns returning the basis word form as a stem, such that the part of speech is retained. A possible sequence of words is:

Walk, walks, walked, and *walking*

The stem returned from inflectional stemming is *walk*.

[3] A part of speech is a category of lexical units that have equal linguistic properties. Examples are nouns, prepositions and verbs.

Inflectional stemming works well, but restricting to this type of stemming is often not well enough as it misses relevant information that might be obtained from derivational stemming. As such, stemming does not provide comprehensive results, but it is often still performed. Namely, it reduces the number of terms and therefore the dimensionality. In addition, it decreases the existing dependency between the terms.

4.2.4 Text representation

The text corpus, which is refined in the post-tokenization step, can not yet be read and interpreted by text analytics algorithms such as clustering. The text corpus is still a form of unstructured data. Therefore, a text representation method is chosen to abstract the results [Milios et al., 2006, Sebastiani, 2002]. This process is called document indexing. Here, a text d_j is converted into a structured and compact form of representation, whereby the choice of terms depends on the units in the text that are considered as most meaningful.

An extensively used text representation method is the vector space model, which was established at the Cornell University in New York in the 1960s. The vector space model became part of the SMART[4] Information Retrieval System and represents each document d_j individually as a vector of weights:

$d_j = (w_{1,j}, w_{2,j}, , w_{t,j})$

Here, $w_{i,j} \geq 0$ is the weight value corresponding to the term i and document j, with t different terms over all documents. This value determines the contribution of the term to the meaning, also known as the semantics, of the document. Here, terms are assumed to be independent.

In case of n documents, these vectors are normally shown in an n x t document-term matrix, in which each row represents one document, and each column represents one term as shown in figure 4.2. However, for each use case the definition of a 'term' and the way in which term weights are calculated, differ. There is no single solution, but there are certain approaches that are commonly used.

Three different techniques applied are word representation, N-grams, and multi-word terms:

[4]SMART: System for the Mechanical Analysis and Retrieval of Text

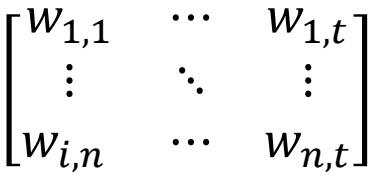


FIGURE 4.2: Document-term matrix

- The most traditional is the **word representation**, also known as 'bag of words', which means each term is a word occurring at least once in one of the documents. This results in a high dimensional matrix, whereby the dimensionality easily rises to thousands.

- This high dimensionality also holds for the **N-grams**, in which a term corresponds to an adjacent substring from a sequence of text. This substring contains N symbols, often characters, consisting of letters from the alphabet A or a space, resulting into a maximum dimensionality of $(A+1)^N$. The N-grams are determined by taking the first N characters of the text and then moving one character at a time. Here, is a character is encountered that is not a letter, it is substituted by a space. Besides, two spaces are joined to one. This approach makes the terms less language dependent, and more robust against errors with in the natural language. No linguistic preprocessing is needed. However, for a small number of N, the dimensionality already is extremely high. So, the use of N-grams does not work well in reducing the dimensionality of the document-term matrix.

- **Multi-word terms**, on the other hand, are able to reduce the dimensionality of the terms significantly. In this case, multiple words are extracted as one term, by use of an automatic algorithm. The idea behind the usage of multi-word terms, is that the semantic information contained in the word representation is not enough. More semantic information is needed to obtain good results from the information extraction methods.

The most typical choice for representation is word representation. Namely, experiments have shown that the more sophisticated representations give comparable or even worse

results in Information Retrieval (IR) [Salton and Buckley, 1988]. Besides, their effectiveness is not significantly better. In particular, research has been conducted into the use of multi-word representation. It is believed that multi-word representation should obtain better results compared to the word representation ad N-gram representation. However, the results from experiments that have been conducted thus far, are not always promising. According to Lewis [1992], this is due to the inferior quality in the statistical information. Multi-word representation is associated with lower dimensionality, but also with a lower frequency within each document. Besides, it results in more synonyms compared to the other representation techniques. As these synonyms do not appear in the same documents, the assignment of terms is less consistent.

Regarding the assignment of weights to the terms, a distinction is made between binary and non-binary weights:

- Binary weights: weight $w_{(i,j)} \in (0, 1)$. A 0 is assigned in case a term is not present in a document, and 1 otherwise.
- Non-binary weights can be determined by multiple methods. The two most commonly used are:
  - Term frequency (tf)[5]: weight $w_{(i,j)} \in \mathbb{N}$. It is the number of times term i appears in document j.
  - Term frequency-inverse document frequency (tf-idf): weight $w_{(i,j)} \geq 0$. The tf-idf method is the most used commonly used method. It does not only take into account the number of times a term occurs (term frequency), but also the importance of the term by calculating the number of documents in which the term occurs (document frequency). The formula of the standard tf-idf method is as follows:

    $\text{tf-idf}(t_q, d_j) = tf(t_q, d_j) * idf(t_q, D)$

    Here,

    $tf(t_q, d_j) = \#(t_q, d_j)$, the number of times term t_q occurs in document d_j

    $idf(t_q, D) = log\frac{|D|}{\#(t_q, D)}$, the importance of term t_q over all documents D

[5] $\mathbb{N}$ is the mathematical symbol for numeric numbers

It follows intuition, as the tf-idf and thus weight of a term t_q increases as it occurrence is higher in a document d_j, but decreases in case it occurs in a larger number of all documents D. Tf-idf, can thus be seen as a normalization technique.

Many studies have shown that tf-idf weights obtain better results compared to the other weighting methods [Sebastiani, 2002]. First of all, the information retrieved from binary weights is too limited. It only returns whether a term is present in a document. This holds that if two terms are both present in a document, they are of equal value. This is not necessarily the case. Besides, the term frequency method gives more weight to terms occurring more often in a document. This is not always the right method, since it is also important that the term appears infrequently in the rest of the documents. Therefore, overall the tf-idf method gives the best results. However, since the preprocessing step of stopword removal already removes most terms that occur frequently in all documents, the term frequency method does not perform much worse.

4.3 Methods and techniques used

The methods and techniques used for preprocessing of the Enron dataset are based on the state of the art provided in the previous section.

Figure 4.3 shows the document triage process followed to obtain readable documents in plain text.

FIGURE 4.3: Document triage process

The Enron dataset obtained from the internet consists of 151 PST (Personal Storage Table) files{footnote: PST files are files of open personal format in Microsoft, such as messages, mails and calendar appointments in Microsoft Outlook) of which a sample is loaded into Nuix[6]. The PST files are containers that can be considered as zip files containing multiple sub files. By reading these PST files into Nuix, a tree structure

[6]Nuix is a document search and analysis suite, http://www.nuix.com

is created, that splits on the PST files, and the different subfiles inside. From this hierarchical structure the non-readable files, such as outlook folders, are removed, after which the remaining files are converted into TXT files.

The original sample received obtained 71,429 documents. As approximately 17,000 documents were not readable, Nuix converted and returned 69,527 documents. The encoding used to convert these PST files was UTF-8, as expected from the previous section. This character encoding was used again in the text segmentation and transformation process followed *(see figure 4.4)* when loading and reading the files in R.

FIGURE 4.4: Text segmentation and transformation process

One of these files is shown below in figure 4.5. As most files in the Enron dataset, it is an e-mail. However, since e-mails are often forwarded, the dataset contains a lot of duplicates. Besides some files are empty. So before looking into the files itself, the empty and duplicate files are removed. This results in a unique dataset of 15,783 documents, of which the content is separated from the metadata as we are only interested in performing analyses in the content of the data. For the e-mail shown above, the metadata includes the subject, date, sender, and receivers *(see figure 4.5)*.

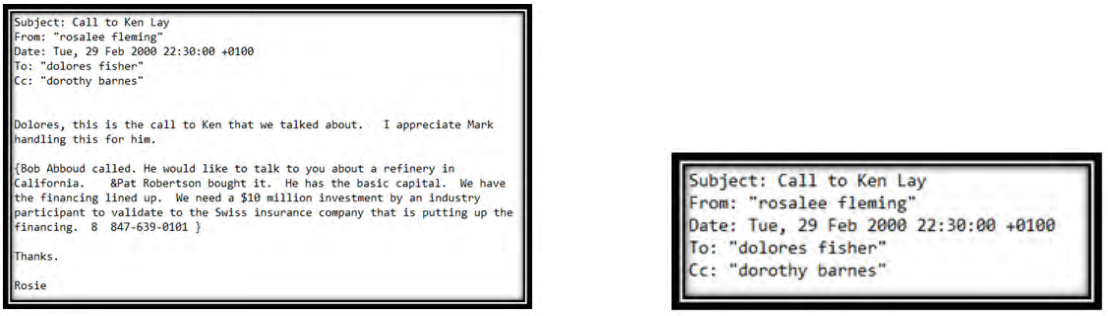


FIGURE 4.5: Example of an e-mail and the corresponding metadata

The resulting corpus was then transformed by the transformation methods indicated in the previous 'state of the art'-section. All text is converted to lower case, the punctuation is removed, as well as the numbers and symbols. English stopwords are deleted and stemming is performed on the words which are retained.

Since there is no best choice for the determination of the weights in the document-term matrix, two matrices with different weighting methods are created, on which further analyses are performed. As shown in figure 4.6, these are the term frequency and tf-idf methods.

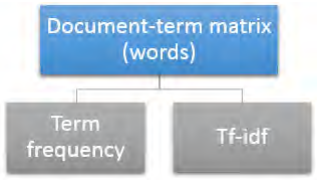


FIGURE 4.6: Practical experiment: Text and term representation

4.4 Statistics on the dataset

The document-term matrices created are high dimensional matrices with a total of 15,783 rows (documents) and 46,177 columns (terms) from which the first insights are obtained. The wordclouds below show the most important words that have the highest weightings in the document-term matrix. Here it holds, the larger the words, the bigger their importance.

As expected, the wordclouds consider different words as important. For example, *jan*, *vlookup* and *colmmatch* do occur in the wordcloud based on term frequency (tf), but not or only in small size in the wordcloud based on tf-idf weighting. This probably means that these words occur frequently and in many documents. On the other hand, *internet-shortcut* and *recipi* are shown in the second plot, while they are not visible in the first one. This probably means they occur frequently, but not in a large amount of documents.

Although the word clouds give a good first insight into the contents of the data, it is important to keep in mind that even the most frequent words are relatively rare. As is shown by figure 4.8, over all documents, the most frequent word only occurs in 0.02% of the cases. This agrees with intuition: since natural language is complex, there is a broad vocabulary for each language.

FIGURE 4.7: Wordclouds based on tf (left) and tf-idf (right)

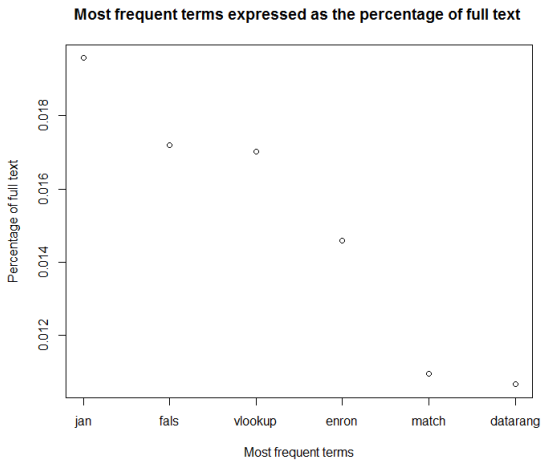


FIGURE 4.8: Most frequent terms expressed as percentage of full text

Finally, the document-term matrix provides insight into the distribution of the weights over all terms. As can be seen from figure 4.9, for both methods around 70% of the weights are approximately 0. This means that besides the high dimensionality, both document-term matrices are very sparse.

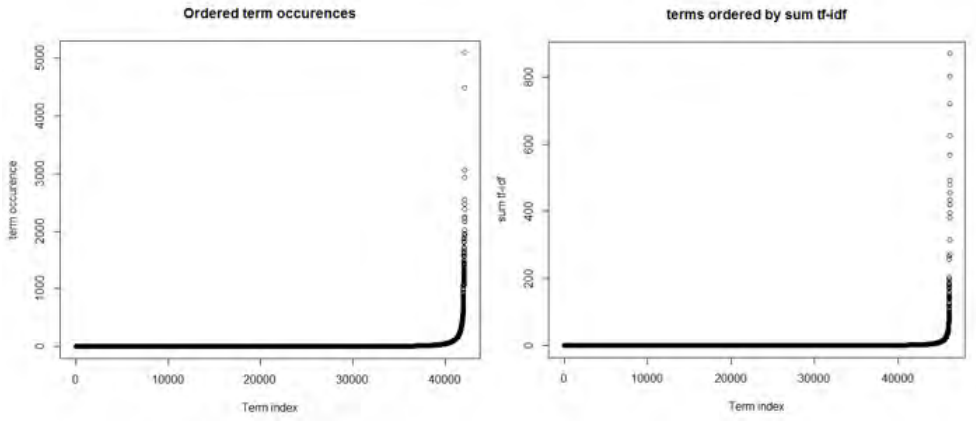


FIGURE 4.9: Terms ordered by tf (left) and tf-idf (right)

This page intentionally left blank.

Chapter 5

Dimensionality reduction

It is well known that document-term matrices created from a set of documents are high-dimensional and sparse. The total number of different words occuring in the document set is high, but the overall the word occurences are low. This means that a large component of the matrix is empty, which results in a major decline in the performance of text analytics methods. It especially holds for the results of clustering algorithms [Aggarwal and Yu, 2001]. Therefore, dimensionality reduction is necessary and part of the text analytics process as described in chapter 2.

As in the previous chapter, the state of the art will be discussed as well as the methods and techniques used in the experiment, and finally the results. Further, Appendix B contains the formulas of the methods discussed in this chapter.

5.1 State of the art

As not all text analytics algorithms are able to handle high-dimensional data a large number of terms, $|T|$, might present difficulties in further analyses. Dimensionality reduction reduces the number of terms of a document-term matrix, i.e. it reduces the number of columns. In mathematical terms: it reduces $|T|$ to $|T'| \ll |T|$. Here, T' is the set of reduced terms [Sebastiani, 2002].

In addition to reducing the term space and thus reducing computation time required for algorithms, dimensionality reduction has the advantage of decreasing the chance of

overfitting. It has shown that in case of a classification problem a certain ratio between the number of items to train a model and the amount of terms is needed to avoid overfitting. According to Fuhr and Buckley [1991] this ratio has to be at least 50 to 1, which means a lot of terms need to be removed. Research by Yang and Pedersen [1997], has shown that in case of text categorization, a removal of even 98% of the terms is possible, such that it increases the efficiency of the algorithm significantly and even improves the performance to a certain degree [Liu et al., 2003].

However, dimensionality reduction should be performed carefully, as the reduction of terms could be accompanied by a loss of information pertaining to the meaning and content of the set of documents. As such, dimensionality reduction is an important compontent of the text analytics processes, and therefore, much research is currently conducted regarding this topic [Fodor, 2002]. Mostly, (comparative) research is conducted into two types of methods: feature selection and feature transformation. Both are discussed in the two subsections below.

5.1.1 Feature selection

Within feature selection a subset of the features are chosen, based on criterion that is pre-determined. Feature selection can be applied to classes with or without labels, so it can be used for both supervised and unsupervised learning. Since this study is focused on unsupervised learning, the methods discussed here are unsupervised methods. As such, it is not possible to test the quality of the features selected by comparing resulting classes with pre-determined labels.

A well-known traditional and often used method is based on document frequency. Document frequency is a measure of the amount of documents in which a certain term appears. In mathematical terms, feature selection based on document frequency can be described as follows: given a matrix of m x n, where m the number of terms and n the number of documents, DF_t is the amount of documents in which term t occurs once or more. The dimensionality of the matrix is reduced by taking k from the m terms, such that $k \ll m$ and such that these terms have the k highest values of DF. This process requires a computation time of $O(m * n)$ [Milios et al., 2006].

This method is based on the idea that terms occurring rarely, do not contain much information and thereby have little influence on the overall performance. Yang and Pedersen [1997] extensively studied this method and they believe that this method might give results that are as good as advanced methods for feature selection. However, not everyone agrees. As mentioned in the previous chapter, tf-idf takes more information into account and is often considered as a more promising method within both text representation and dimensionality reduction. Therefore, a method based on the mean tf-idf has been introduced to reduce dimensionality. This method selects the k terms that have the highest average tf-idf values *(see chapter 4 for an explanation on tf-idf)*. Yet, the mean tf-idf is rarely used and compared to other dimensionality reduction methods. The first reason is that this method is relatively new. Secondly, it is sometimes still considered as complicated compared to for example the method based on document frequency. Document frequency is more often used since it is a fairly simple and very intuitive method.

Further, in the course of time, other methods have been developed and investigated. An example is term strength (TS) [Yang, 1995], which is the conditional probability that a term appears in the second part of a document set, given that it did in the first part. In addition, two newer feature selection methods are entropy-based ranking and term contribution (TC). Entropy based ranking (EN) was introduced by Dash and Liu [2000], ordering the terms based on the reduction in entropy when they were removed from the feature set. Term contribution, introduced by Liu et al. [2003], can be seen as an extension to the document frequency (DF). It takes into account the number of documents in which a term occurs, but also the similarity between the documents. The idea behind this method is that results from clustering text are dependent upon document similarity.

Although more feature selection methods exist than discussed in this section, the four methods often compared are DF, TS, TC and EN. Most research conducted resulted in the same conclusion. It was found that term contribution and term strength perform somewhat better than Document Frequency and Entropy-based ranking. The exact order is the following: $TS > TC > DF > EN$. However, as can be seen from table 5.1, term strength has a really high computation cost that is quadratic to the amount of documents. This also holds for entropy-based ranking. Therefore, term contribution

and document frequency are often preferred [Liu et al., 2003]. Further, these methods are very intuitive compared to the feature transformation methods.

Feature selection method	Time complexity
Document frequency	O(mn)
Term strength	$O(n^2)$
Entropy-based ranking	$O(mn^2)$
Term contribution	$O(m\bar{n}^2)$

m = #terms, n = #documents,
$\bar{n}$ = average #documents in which terms occur

TABLE 5.1: Time complexity feature selection methods

5.1.2 Feature transformation

Feature transformation is the process of reducing dimensionality by combining features, both linearly and non-linearly. This is also called functional mapping. Feature transformation methods are very successful as they are able to discover the latent structure[1] in the data. However, the features created from these methods do not have a visible meaning, which makes the results from clusters created difficult to understand and interpret [Dash and Liu, 2000].

The transformation of features is generally executed by one of the three methods below [Milios et al., 2006]:

- **Principal Component Analysis:** The PCA is a second-order transformation technique that creates new terms by use of the covariance matrix of the original feature set. According to Fodor [2002], PCA is the best linear technique taking the mean-square error as a validation technique. The new terms created are the so called principal components that are linear combinations of the terms from the original dataset and orthogonal to each other. These principal components are ordered on the basis of their variance. The first principal component has the largest variance, the second has a variance that is second-largest etc. The reduced dimensionality k then depends on the number of principal components chosen. However, there is no best answer for that. For each case, is must be re-examined, on the basis of the variance explained by the principal components and the permissible variance by the user.

[1] Latent structure is the structure in the variables that can not be examined directly. These variables are the latent variables

- **Independent Component Analysis:** The ICA is quite similar to the PCA, however it does not require the new created terms to be orthogonal to each other. Here, the terms created are called independent components as the requirement of ICA is that the components are statistically independent. This requirement is heavier compared to the requirement of PCA. Further it is a function of higher-order, which means the terms are not necessarily combined linearly. Overall, the ICA is considered to be an expansion to the PCA.

- **Latent Semantic Indexing:** LSI applies singular value decomposition (SVD) to a document-term matrix whereby the terms are represented by words. Within SVD, the original matrix M is transformed to a new matrix M' such that their distance is minimized. The distance measure used is the 2-norm. Again, the choice of k, the number of dimensions in M' , is a point of discussion. Reduction might reduce the noise. However, information might get lost as well if the reduction becomes too large. Still, it works well for a relatively small amount of k.

In previous research, ICA and LSI are most compared [Liu et al., 2005]. Overall this resulted in the same conclusion, namely that ICA delivers better results than LSI. Both methods are able to reduce a high-dimensional dataset of thousands of terms to a dimensionality of up to hundred. However, ICA is much more stable in its results than LSI.

In addition, comparing only ICA and LSI, these two methods are also often compared to the Document Frequency Method. This results in the following ranking: $ICA > LSI > DF$. The feature transformation methods perform better than the method based Document Frequency. However, Document Frequency obtains its best results at an amount of dimensions somewhere in the middle part. ICA and LSI obtain their best results at a much lower number of dimensions k. This best performance of DF can become almost equal to the best results of ICA and LSI. Still, this only holds for a larger number of dimensions. For a smaller number of k, ICA and LSI perform much better.

It is well known that PCA provides very good results as well, which get close to the results of ICA. However, little research had been done on comparing this method to LSI and DF. Therefore, it is not possible to include PCA in the ranking above. However, something can be said about the time complexity. Although, there is no complete

agreement on the time complexity of PCA, ICA and LSI, it is well known that the time complexity of PCA is smaller than that of ICA, since it is often used as a preprocess step of ICA. Besides, computing LSI takes a lot of time. It is seen as one of the major disadvantages of this method.

Overall, both types of methods have their advantages and disadvantages. The feature selection methods work very intuitively and their results are are easily interpretable. The feature transformation methods other hand, generally provide better results as they have the possibility to discover latent structure in text. Therefore, both types of methods are examined in the practical research.

5.2 Methods and techniques used

For both types of dimensionality reduction, several methods are investigated.

The two feature selection methods investigated are: document frequency and mean tf-idf. The reason for investigating document frequency is that a lot of research is performed into this method and it has been proven that dimensionality reduction based on document frequency provides good results. In addition, since it has been shown that the tf-idf weighting scheme in document-term matrices is promising, dimensionality reduction based on this scheme is expected to be promising as well. The reason for not using the term strength and entropy-based ranking is their high time complexity. Further, term contribution could not be tested due to a lack of a TC function in R.

The feature selection method investigated in this experiment is Principal Component Analysis. The reason for choosing this method is that the principal components created by PCA have shown to produce good results in clustering documents. Further, the computation time required was reasonable. As was expected, running ICA and LSI took too much time given computational resources available.

Figure 5.1, shows the tree structure again, which is applied for this experiment. As described in the previous chapter, two document-term matrices are created, one with the term frequency and the other with the tf-idf weightings. In this process step, the dimensionality of both matrices is reduced. Since the dimensionality reduction method based on document frequency is often applied to matrices with term-frequency weights,

this choice is also made in this experiment. The same applies to PCA. The principal components will be created based on the term frequency weights. However, the reduction on the basis of the average tf-idf will be performed on the second matrix with the tf-idf weights.

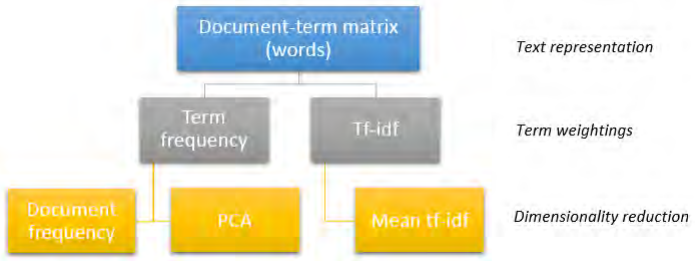


FIGURE 5.1: Process practical experiment: Dimensionality reduction

As discussed in the 'start of the art'-section the number of reduced dimensionality k has to be determined per method and per use case individually. This decision can be made on the basis of several methods, depending on the user's desire In this research the decisions are made based on plots, as shown in figure 5.2. The first two plots show the terms ordered by the document frequency and average tf-idf. Based on the idea that terms with a value 0, do not add information and meaning to the content of the text, k is in the first two cases set equal to 12,000, since around 35,000 terms are equal to 0. For PCA, the number of principal components is chosen based on the variance reduction if the number of components increases. As the variance is approximately 0 in case of 4,000 principal components, k is set equal to 4,000 in this case.

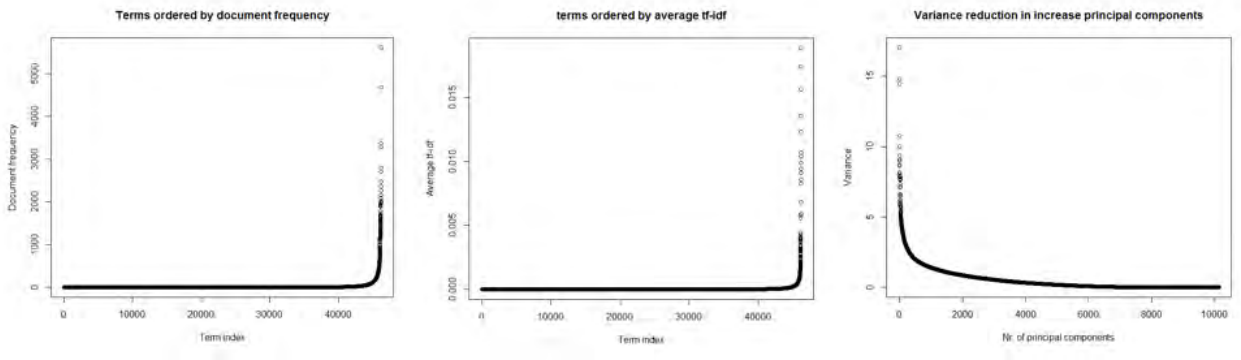


FIGURE 5.2: Plots created to reduce dimensionality

5.3 Results

The dimensionality reduction step followed, with the methods and techniques used as described above, resulted into three document-term matrices (DTMs). Their characteristics are shown in table 5.2.

Text repres.	**Term repres.**	**Term weighting**	**Dim. reduction**	**#columns**	**#rows**
DTM	Word	Term frequency	Document frequency	12,000	15,783
DTM	Word	Term frequency	PCA	4,000	15,783
DTM	Word	Tf-idf	Average tf-idf	12,000	15,783

TABLE 5.2: Characteristics DTMs after dimensionality reduction

These three matrices are used in the further research on best practices for the use of clustering and topic modeling methods.

Chapter 6

Clustering

As mentioned in the introduction and literature review, text clustering is applied to a group of documents based on their similarities. Once clustered, documents can be organized, classified and summarized, which facilitates the retrieval of information [Aggarwal and Zhai, 2012, Beil et al., 2002, Berkhin, 2006]. Since text clustering can be applied in multiple contexts, it is extensively studied. In the first section the state of the art is described, after which the second section describes the application of the most frequently used and 'best'methods. This results and best practices in the field of text clustering are discussed in the conclusion.

6.1 State of the art

Although each cluster method and technique generates clusters differently, they all have the same goal. Namely, they group documents in clusters, such that the clusters C together contain all documents D, and each document is assigned once to a cluster c_i, i.e. there is no overlap. In mathematical terms [Rokach and Maimon, 2005]:

Set of documents $D \in C = \{c_1, c_2, ..., c_p\}$ such that $\sum_{i=1}^{p} c_i = D$ and $c_i \wedge c_j = \emptyset \ \forall i, j$

Most clustering methods create clusters based on the relative aggregate term frequency distances between documents. As a term frequency counting mechanism, this approach can be considered a latent indicator of semantic content in the document. These methods

are called 'distance based′ clustering methods. The best known and most frequently applied algorithmic measure applied for text clustering is the Euclidean distance:

$$dist(d_i, d_j) = \sqrt{\sum_{k=1}^{m}(d_{j,k} - d_{i,k})^2}$$

This measure calculates the distance between document d_i and d_j in Euclidean space. It is determined by squaring the term values k for both documents, summing the resulting values and finally, taking the resulting root. The smaller the distance value, the more similar documents d_i and d_j are. The distance value reaches its minimum if the documents are identical.

However, in some cases similarity measures are used instead of distance measures to cluster similar documents. The most often used similarity measure is the cosine similarity:

$$s(d_i, d_j) = \frac{d_i^T * d_j}{\|d_i\| * \|d_j\|}$$

This is the inner product of two vectors representing document d_i and d_j normalized by their length. The larger the value of s, the more similar the documents are. The value s is largest for documents that are exactly the same. Further, for both similarity and distance measures, it holds that they are symmetric [Beil et al., 2002, Steinbach et al., 2000]. In other words:

$dist(d_i, d_j) = dist(d_j, d_i)$ and $s(d_i, d_j) = d(d_j, d_i)$:

Distance based methods (sometimes based on similarity) are generally divided into two categories: hierarchical clustering algorithms and partitional clustering algorithms. Both are explained below, together with the related methods most commonly used.

6.1.1 Hierarchical clustering

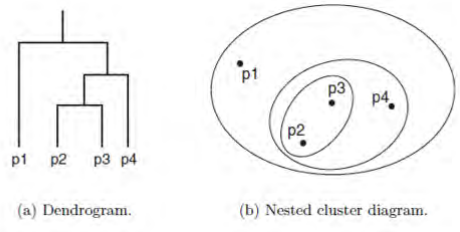
Hierarchical clustering methods produce hierarchically created clusters step by step. These can be created bottom-up, which is called *agglomerative clustering*, or top-down, *divisive clustering* [Müllner, 2011, Rokach and Maimon, 2005, Steinbach et al., 2000]:

- **Agglomerative clustering:** This bottom-up approach works in reverse, starting with one cluster per document and aggregating clusters upwards. In other words,

its initial status is n clusters, all containing 1 document. For each subsequent level sets of two clusters are aggregated into single clusters based on shared affinity. The clusters merge based on the distance or similarity measures chosen.

- **Divisive clustering:** A top-down approach starts with a single cluster containing all documents, and bifurcates these clusters until n clusters remain having in each cluster exactly one document.

For both types of methods, this results in a nested cluster diagram and dendogram, as shown in figure 6.1. Both structures show the groupings of the documents and merges or splits at each level of similarity. It can be seen as a hierarchical taxonomy. In addition, the dendrogram shows that for hierarchical clustering methods the desired number of clusters is not specified upfront. This is one of the main advantages of hierarchical clustering. However, the level of specification can be obtained by selecting the desired similarity value. At the specified level, the dendogram is 'cut off'. The level of specification desirable is not a standard measure, being unique to the semantic context of the corpus under investigation. This is best understood through the lens of domain expertise and highlights the importance of scoping the corpus with subject matter experts prior to applying such approaches.



(a) Dendrogram. (b) Nested cluster diagram.

FIGURE 6.1: Graphical representations of hierarchical cluster

Source: [Bramer, 2013]

Although, is seems that the methods of agglomerative and divisive clustering are quite similar, the agglomerative methods are applied much more frequently. This is due to the computational overhead of divisive clustering *(see table 6.1)*, which takes time proportional to the computational resources available. Although, the computational overhead of agglomerative clustering is already large, that of divisive clustering is even larger. In addition, research has established that divisive clustering performs approximately equal to and sometimes worse than agglomerative methods. As a result, little research has

been conducted on divisive clustering algorithms, and they are considered out-of-scope for this research.

Hierarchical clustering method	Time complexity
Agglomerative clustering	$O(n^2)$ - $O(n^3)$
Divisive clustering	$O(2^n)$

n = #documents,

TABLE 6.1: Time complexity hierarchical clustering methods

Agglomerative clustering always follows the same, general bottom-up approach *(see Appendix C.1 for the exact algorithm)* as described above. However, various techniques can be used to calculate the distances and similarities. The three most well-known techniques are single linkage, average linkage and complete linkage clustering *(see table 6.2)*:

- **Single linkage clustering:** This method defines the distance from cluster C_i to C_j as the shortest distance between one of the documents in C_i and one of the documents in C_j. In case a similarity measure is used, it is the largest similarity found between one of the documents in C_i and one of the documents in C_j.
- **Average linkage clustering:** This method defines the distance as well as the similarity between of documents C_i and C_j as the average value between a pair of documents, one of cluster C_i and one of C_j.
- **Complete linkage clustering:** This method defines the distance from cluster C_i to C_j as the longest distance between one of the documents in C_i and one of the documents in C_j. In case a similarity measure is used, it is the smallest similarity found between one of the documents in C_i and one of the documents in C_j

Agglomerative cluster method	Distance measure	Similarity measure
Single linkage	$min_{d_i \in C_i, d_j \in C_j}\ d(d_i, d_j)$	$max_{d_i \in C_i, d_j \in C_j}\ s(d_i, d_j)$
Average linkage	$\frac{1}{\lvert A \rvert \lvert B \rvert} \sum_{d_i \in C_i} \sum_{d_j \in C_j} d(d_i, d_j)$	$\frac{1}{\lvert A \rvert \lvert B \rvert} \sum_{d_i \in C_i} \sum_{d_j \in C_j} s(d_i, d_j)$
Complete linkage	$max_{d_i \in C_i, d_j \in C_j}\ d(d_i, d_j)$	$min_{d_i \in C_i, d_j \in C_j}\ s(d_i, d_j)$

TABLE 6.2: Measures agglomerative cluster methods

Each of these methods has their advantages and disadvantages. Single linkage is very easy to implement and to execute, as similarities or distances are first calculated for

each possible set of documents and then only need be ranked in order to determine which clusters to merge. However, single linkage clustering has the disadvantage of the chaining phenomenon. This holds that two clusters are merged quickly if they have some similarities, even when they both have a large number of elements that are dissimilar. This may result in clusters which are not homogeneous, but heterogeneous.

The average linkage method does not suffer from the chaining problem. However, its run time is longer, as the average distance or similarity has to be calculated after computing the paired values.

This also holds for complete linkage. As complete linkage merges clusters based on the worst values, it also does not suffer from the chaining problem. It is more robust compared to single linkage, but again takes more time to compute.

6.1.2 Partitional clustering

Unlike hierarchical clustering only one partition is created by partitional clustering algorithms. A graphical representation of a possible result is shown below.

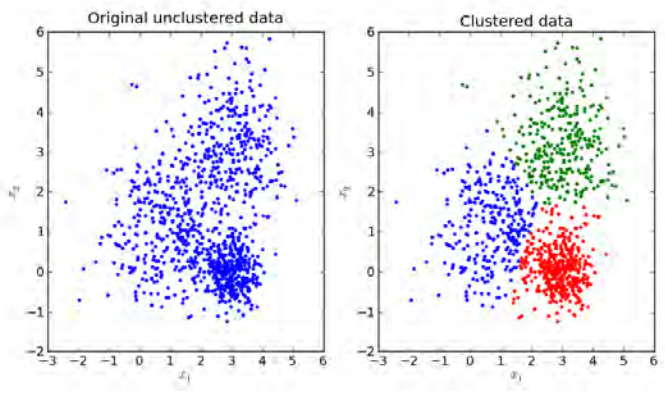


FIGURE 6.2: Graphical representation of clusters created by a partitional algorithm

Source: http://trendsofcode.net/kmeans/

For partitional algorithms, the number of clusters K is given as input upfront, together with an initial positioning of K documents within these clusters. From this initial positioning, documents are then moved and assigned to clusters iteratively until 'optimality'is reached. This optimality is obtained when the measurement of error is minimized, i.e. when the largest distance between clusters and the smallest distance within clusters is achieved. The most commonly used error measure is the Error Sum of Squares

(SSE), which is the sum of squared distances between the point of representation r_i and document d_i for all documents in each cluster:

$$SSE = \sum_{i=1}^{K} \sum_{d_i \in C_i} dist^2(r_i, d_i)$$

In order to obtain the best result, it would be necessary for a domain expert to examine all possible cluster combinations, and thus all possible allocations. However, this approach is infeasible due to the time and effort involved. This problem is NP Hard. In this context, partitional clustering algorithms are greedy methods to a certain degree, improving the results iteratively. Upon each iteration the placement of the documents in the k clusters are changed until the allocations of all documents are returned at once. These methods therefore are more efficient than manual assessment, and are thus often applied in the task of clustering documents [Berkhin, 2006, Müllner, 2011, Steinbach et al., 2000].

The two most applied clustering methods for document classification are k-medoid clustering and k-means clustering. Their algorithms are quite similar, but they differ in their cluster representation. K-medoids represents its clusters by one of the documents in it (medoids), while k-means clusters are defined by their centroids. The exact algorithms are shown below. In addition, appendix C.2 contains an example of k-means for a better understanding:

K-means algorithm

1. Randomly select K documents $d_1, ..., d_K$, all as centroids[1] $ce_i, ..., ce_K$ for one of the K clusters.

2. Assign all documents d_i for all i=1,...,n, to one of the centroids. The centroid, and thus cluster, ce_j chosen is the one that is closest based on a distance or similarity measure.

3. Calculate the new centroids $ce_1, ..., ce_K$ for each of the clusters by use of the measure chosen. $ce_j = \bar{D}_j$ for j=1,...,K, the average location of the documents in cluster j

4. Repeat steps 2 an 3 until the K centroids are fixed.

[1] centroid: the center or average location of all documents in a cluster, not necessarily a document

K-medoids algorithm

1. Randomly select K documents $d_1, ..., d_K$, all as medoids[2] $me_i, ..., me_K$ for one of the K clusters.

2. Assign all documents d_i for all i=1,...,n, to one of the medoids. The medoid, and thus cluster, me_j chosen is the one that is closest based on a distance or similarity measure.

3. Calculate the new medoids $me_1, ..., me_K$ for each of the clusters by use of the measure chosen. $me_j = D_j^*$ for j=1,...,K, the document with the largest similarity in cluster j

4. Repeat steps 2 an 3 until the K medoids are fixed.

The three biggest differences between these two methods when clustering documents are robustness, ability to handle sparse data, and time complexity *(see table 6.3)*:

- **Robustness:** The results of k-medoids are more robust compared to the results of k-means. This is due to the fact that a medoid is less sensitive to outliers and noise than a mean is. Besides, the k-means algorithm is very sensitive to the documents initially positioned to represent the K clusters.

- **Sparse data:** K-medoids isnt able to handle sparse data, in this case text, very well. K-medoids represents a cluster by its medoid, one of the documents in it. As this document only contains a small part of the terms in the overall cluster, and thus the similarity with the other documents is relatively low, it doesnt have all information needed. This results in clusters that are not built effectively.

- **Time complexity:** Of note, the time complexity of the k-medoid algorithm is much larger than that of the k-means, which is linear. This is partly due to the calculation of the medoid for each cluster, which takes much more time than that of the computation of means. In addition, the k-medoids algorithm requires many iterations to converge. For k-means relatively few iterations are required, since it converges quickly. As multiple research has shown, the amount of iterations needed

[2]medoid: the representative document in a cluster that has the largest similarity with the other document in that cluster

for a large amount of documents is only five, or ten at maximum. This results in a higher computational overhead, and thus time to compute, for K-medoids.

Paritional clustering method	**Time complexity**
K-means	$O(K * n * i)$
K-medoids	$O(K * i * (n - K)^2)$

K = #clusters, n = #documents, i = #iterations

TABLE 6.3: Time complexity partitional clustering methods

As mentioned in the introduction, the three largest difficulties of clustering are high-dimensionality and sparse data, increasing computational overhead accompanying large sets of documents, and the interpretability of the algorithms. Therefore, a clustering algorithm should be able to accommodate these three factors. Although k-means and k-medoids are both quite easily interpretable, k-means is much more frequently used. Beyond ease of interpretation, the popularity of k-means can be understood in terms of its ability to handle sparse data and large sets of documents, a fast convergence rate, linear computational overhead and rapid implementation of the algorithm.

Compared to hierarchical clustering, k-means is more often used in document clustering. Although hierarchical clustering generally performs better due to its robustness, it is still less popular. Hierarchical clustering methods are not able to handle the immense amount of data that the k-means algorithm can handle. This is partly due to computational overhead as shown in figures 6.1 and 6.3, but also due to the space complexity that is needed for storage of the similarity matrix *(see table 6.4)*. The space complexity of k-means is linear, while hierarchical clustering utilizes the higher computational complexity of quadratic space.

Clustering method	**Space complexity**
K-means	$O(K + n)$
Hierarchical	$O(n^2)$

K = #clusters, n = #documents

TABLE 6.4: Space complexity clustering methods

However, the main disadvantage of k-means clustering is, as mentioned above, sensitivity to the documents initially chosen as cluster representatives. Therefore, multiple techniques have been investigated that might improve the performance of k-means clustering. First, k-means can be run numerous times with various initial cluster representations. However, for a large amount of documents and a large number of clusters K the chance of selecting a specific document is very small. Assuming that each cluster contains the same amount of documents N_c

$$\text{P(document selection)} = \frac{\#\text{possibilities to select a center in each cluster}}{\#\text{possibilities to select K centers}} = \frac{K! * N_c^K}{(K * N_c)^K} = \frac{K!}{K^K}$$

Even for small number of clusters K, the probability is already quite small, and declines rapidly. The probability remains high that a local rather than a global optimum is obtained for each of the runs.

A second possibility is to use a scatter-gather method. This method initializes the documents chosen as centroids by use of agglomerative hierarchical clustering, after which clusters are created through the use of k-means. Often, a sample is used to obtain the initial values. Lastly, bisecting k-means, a variant on k-means, can be used. This algorithm uses the characteristics from k-means in addition to hierarchical clustering. Starting with all documents in a single cluster, it creates a tree structure, each time splitting a chosen cluster into two clusters until the desired numbers of clusters K is reached *(see Appendix C.3 for the exact algorithm)*.

Both bisecting k-means and scatter-gather methods have been the subject of numerous investigations. Bisecting k-means is most often compared to standard k-means clustering, but sometimes to hierarchical clustering as well. For this method the same essential conclusion is drawn, namely that bisecting k-means gives better results than standard k-means and results that are quite similar to hierarchical clustering. Testing scatter-gather methods also give promising results, however this method is rarely compared to paritional and hierarchical clustering algorithms. In addition, the number of studies performed on these two methods is relatively limited. Therefore, no strong conclusions can be drawn [Beil et al., 2002, Steinbach et al., 2000].

A corpus-specific judgement of fit applies both to the effectiveness of the clustering method as well as to the determination of the ideal number of clusters K. Best fit must be determined per corpus, as relevant to the objectives of the search, by running different

algorithms with different values K. However, it should be noted that a 'perfect', all-purpose clustering algorithm, by nature, does not exist. There is no standard algorithmic procedure or measure for validating the 'best' results, as results are specific to each unique document set, or corpus. The final arbitrator of the quality of results are the stakeholders and domain experts who motivated the research and participated in framing the semantic context surrounding the corpus.

The final validation of clustering results must ultimately be performed by people, ideally those with great expertise in the subject matter. This is due to the fact that clustering is an unsupervised learning method, which means no labels are present to automatically determine accuracy. Labeling 'fit' is highly specific to domain-specific factors related to the corpus selected and the surrounding body of knowledge under investigation. It is crucial to understand that clustering in the end is a rough term frequency and correlation measure which at best gives latent indications of semantic content, whereas expert human comprehension is a much more highly refined specification which incorporates implicit semantic context (i.e. language comprehension, cultural and social factors, legal context, domain knowledge and expertise, etc.) [Rokach and Maimon, 2005].

This said, it is possible to cross-validate which methods (approximately) yield the same results, and thus, which methods distinguish themselves when all other factors are held equal. Two often used similarity measures are jaccard similarity and the rank similarity. These measures compare the similarity in which two methods have assigned the set of documents $D = \{d_1, ..., d_n\}$ to a set of clusters S and T. The number of clusters for S and T are not necessarily equal.

Jaccard Similarity:

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|} \qquad J(S,T) \in (0,1)$$

The Jaccard similarity is determined by calculating the number of documents that are assigned to the same cluster (intersection), and dividing this value by the total number of documents n assigned to the various clusters (union). Figure 6.3 shows this method visually.

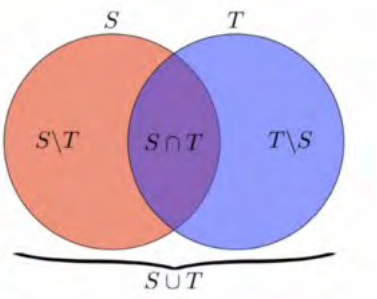


FIGURE 6.3: Visual display of the Jaccard similarity

Source: http://www.insight-journal.org/browse/publication/707

Rand Similarity:

$$R(S,T) = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{\begin{pmatrix} n \\ 2 \end{pmatrix}} \qquad R(S,T) \in (0,1)$$

Here,

TP = $d_i, d_j \in S_k$; $d_i, d_j \in T_o$

TN = $d_i \in S_k$ and $d_j \in S_l$; $d_i \in T_o$ and $d_j \in T_p$

FP = $d_i, d_j \in S_k$; $d_i \in T_o$ and $d_j \in T_p$

FN = $d_i \in S_k$ and $d_j \in S_l$; $d_i, d_j \in T_o$

The rand similarity measure first defines all possible pairs of documents in the document set D. Then for each pair of documents (d_i, d_j) it determines separately whether they are assigned to the same clusters by the first method and then by the second method. If both methods assign the documents to a single cluster, the result is a True Positive (TP). If they are assigned to different clusters in both cases, it is a True Negative (TN). Further, the number of False Positives (FP) and False Negatives (FN) are calculated, as described above. Finally, the number of True Positives and False Negatives are then summed, and divided by the total number of TP's, TN's, FP's and FN's. This is equal to the number of different pairs that can be formed from the document set: $\begin{pmatrix} n \\ 2 \end{pmatrix}$

Both similarity measures are used in the practical experiment in order to cross-validate the similarity in clusters generated by different methods.

6.2 Methods and techniques used

As discussed in the previous chapters, document clusters provide first insights into the content of unknown documents. The previous steps taken to prepare the data in a proper way are of great importance in answering the main question, but with document clustering the actual research objective is reached.

However, as within each process step, several methods are developed. This step includes multiple possibilities regarding distance measure, clustering method, and similarity measure. Based on knowledge acquired from previous research, the following choices are made for conducting practical experiments.

Firstly, the cluster methods and techniques investigated in this experiment are all tested by use of the Euclidean distance measure. This choice was made based on its proper functioning and popularity. No other methods were used, such that the comparison of cluster results is statistically supported.

Secondly, as both the hierarchical and partitional clustering have (dis-)advantages, both types of methods are studied and compared. First, the three hierarchical agglomerative methods, single, average, and complete linkage, are examined in order to create a clear view of the actual differences. The following process was carried out:

Clustering:

For *each of the three DTM's created:*

> ***For*** *an amount of clusters ranging from 2 to 10:*
>
> > ***Assign*** *each document to a cluster by use of the chosen cluster method in R* [3]
>
> ***End***

End

[3] Use is made of the function 'hclust', which contains multiple methods for hierarchical clustering

Validation:

For *each of the three DTM's created:*

> ***For*** *an amount of clusters ranging from 2 to 10:*
>
> > ***Compute*** *the Jaccard and Rank similarities between all cluster methods*
> >
> > ***Compare*** *the computation timed needed for all cluster methods*
>
> ***End***

End

The number of clusters K that are examined ranges from two to ten. This amount is chosen since a relatively small number of clusters makes it easier to interpret and validate the results of this experiment. In addition, a limited amount of clusters is also desirable in practice. The aim of this study is to investigate the best practices in the use of clustering in e-discovery projects, such that without knowledge quick insights concerning the content can be obtained. This is only possible, when the overview is retained.

Therefore, the amount of clusters should not be too large. Further, the results are validated by the two similarity measures previously described, in order to investigate whether the results of the cluster methods differ. As per the human semantic context, a single 'best'method cannot be chosen which applies to all cases, but something can be said about the efficiency based on the computation run-time of the algorithms.

Given the results, the hierarchical agglomerative algorithms that are most distinctive are used for further investigation. Here, agglomerative and partitional cluster methods are compared with each other. The partitional methods that are explored in this study are k-means and Partitioning Around Medoids (PAM), which is the most used algorithm of k-medoids. These are chosen due to their intuitive operation, which will provide better understanding when applied in practice, and due to the relatively low computational overhead compared to the hierarchical methods. The variances of k-means mentioned

earlier are not tested. The reason for this relates to their use of hierarchical clustering, which increases the calculation time heavily. Beyond this, little research has been conducted concerning the functioning of these methods.

6.3 Results

The comparison of the single, average, and complete linkage methods gave expected results. Apart from the calculation of the distance matrix, calculating clusters by use of the single linkage method indeed takes the least time *(see table 6.5)*. This holds for all three document matrices. At first sight, these differences seem very small, but for a large number of documents the difference becomes bigger quickly. Further, table 6.5 shows a clear difference in computation time for the various document matrices, especially in calculating the distance matrix. Given the computing resources applied, the calculation time for the document-term matrix was at least 1.5 hours shorter compared to document clustering based on the other two matrices. Further, it takes the longest to cluster the documents if their terms are represented by principal component analysis (PCA).

	Computation time			
Document matrix	**Distance matrix**	**Single link.**	**Average link.**	**Complete link.**
Document term	3h; 14m; 24s	12s	13s	13s
Tf-idf	4h; 56m; 5s	26s	31s	29s
PCA	5h; 31m; 28s	12s	13s	14s

h = #hours, m = #minutes, s = #seconds

TABLE 6.5: Computation time hierarchical cluster algorithms

However, in spite of the difference in the calculation time, the documents were largely assigned to the same clusters by the different methods. Figures 6.4-6.6 show the jaccard similarities between the three methods, for each document matrix and the amount of clusters varying from two to ten. The rand similarities gave almost exactly the same outcome. These results are not presented here, but are available in Appendix C.4.

For each document matrix, each hierarchical method and any number of clusters the similarity is between 0.997 and 1. This is a very large number. However, there are some small differences. Namely, for a small number of clusters the similarity for all methods compared is equal to 1. For documents represented by principal components, this holds for two and three clusters, for documents tf-idf weighted it applies to two, three, and

four clusters and the similarity for documents weighted by their document frequency is always one for even five clusters. For a larger number of clusters differences arise. However, they are minimal.

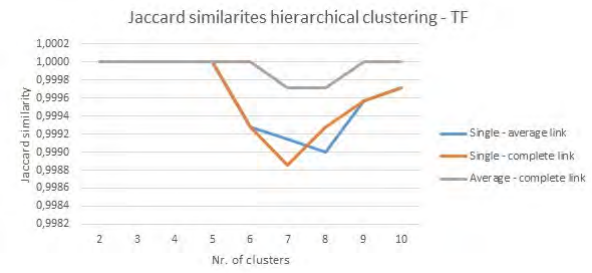


FIGURE 6.4: Jaccard similarities on hierarchically clustered tf weighted documents

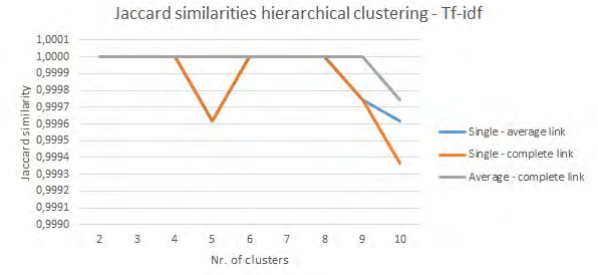


FIGURE 6.5: Jaccard similarities on hierarchically clustered tf-idf weighted documents

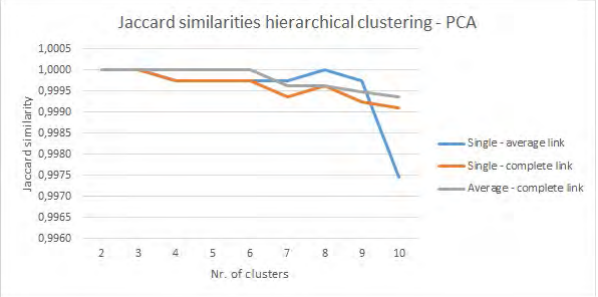


FIGURE 6.6: Jaccard similarities on hierarchically clustered PCA weighted documents

There are several possible reasons for these large values of similarity. The first one might be that the sample of documents selected for this experiment are quite similar, i.e. their content is quite the same. This will result in linkage methods clustering the documents the same way. A second reason might be the usage of stemming and stopword removal. Both methods reduce the dimensionality of the dataset heavily, but they also have an impact on the content, which becomes a bit more generalized. Lastly, it is possible

there is generally little difference within the results generated by the different linkage algorithms.

Further, the fact of decreasing similarity as the number of clusters K increases, follows human intuition. Consider a dendogram. As K is larger, then we are concerned with the bottom of the hierarchical tree. As the linkage methods are bottom-up, the results will differ the most on the bottom, whereas moving upwards more documents belong to an aggregated cluster. On top, all documents belong to one cluster, so in that case the results are always equal.

In general, the results of document assignment are practically the same for all three different methods. Therefore, further research was conducted applying a single hierarchical cluster method. The method chosen is the single linkage method, due to its lower relative computational overhead.

This single linkage method is compared to the partitional methods, k-means and PAM, again on the basis of the computation time and similarity measures used. Table 6.6 provides the computation time that was needed for every method to cluster the documents for different document matrices. However, it needs to be noted that the computation time for one run of the single linkage method cannot directly be compared with the computation time for one run of a partitional cluster method. Namely, a hierarchical cluster method assigns the documents to a number of clusters K ranging from 1 to n at once. Partitional methods assign the documents to clusters only for one predetermined value of K per run.

	Computation time		
Document matrix	**Single link.**	**K-means *(1 run)***	**PAM *(1 run)***
Document term	3h; 14m; 36s	1 - 4m	4.5 - 5h
Tf-idf	4h; 56m; 5s	1 - 4.5m	4.5 - 5.5h
PCA	5h; 31m; 40s	1 - 4m	5 - 6.5h

h = #hours, m = #minutes, s = #seconds

TABLE 6.6: computation time single linkage and partitional cluster algorithms

Yet, it can be seen immediately that PAM is slower than the single linkage approach. Within the bounds of the computational resources applied in the test, PAM needs at least 4.5 hours in order to allocate the document assignments to a single cluster K. Single linkage also requires about 4 to 5.5 hours of run time, depending on the document-term

matrix, but this method calculates the document assignments for all possible cluster quantities. Based on the results measured in computational time, it therefore can be concluded that the PAM method incorporated in R is not a suitable method for this experiment given the computational overhead involved.

Further, k-means takes only 1 to approximately 5 minutes per run. Since we are interested in an amount of clusters ranging from 2 to 10, this means that running k-means for all cluster quantities takes a maximum $5 * 9 = 45$ minutes. In this case, based on the run time, k-means is preferred. If the examined cluster quantity increases, single linkage becomes increasingly advantageous.

However, since clustering results cannot simply be judged based on run time, again the jaccard similarities are calculated. Figures 6.7-6.9 show the similarities for all three algorithms *(see Appendix C.4 for the rand similarities)*. Comparing the similarities of the hierarchical cluster methods mentioned above, these figures show less resemblance. The similarities are generally between 0.7 and 1. Here, the similarities also contain more fluctuations over the various amounts of clusters. It appears that the similarities between PAM and single linkage are most steady (gray line). One reason could be that both methods are quite robust. K-means is more dependent on the initially selected centroids, which is likely the cause of the fluctuating orange and blue lines. For each number of clusters k-means has to be run again, which means that each time another set of documents is selected randomly as initialization values. This has a big impact on the results of k-means as well as on measures of similarity with the other two methods. However, although the similarity of the k-means with the other algorithms has larger fluctuations, it also has a higher average similarity compared to the similarity of PAM and single linkage. In particular, the similarity between k-means and PAM is higher. One possible cause is that both are partitional cluster methods, such that the base algorithms are much alike.

From these results, no 'best' cluster method can be determined. However, it is expected that the application of topic modeling will yield better results in combination with single linkage and PAM, then with k-means. This is expected due to the unstable results.

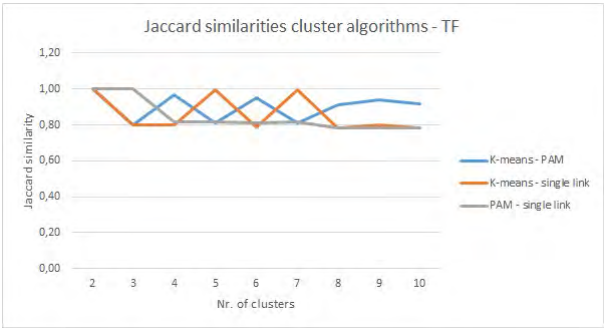


FIGURE 6.7: Jaccard similarities on clustered tf weighted documents

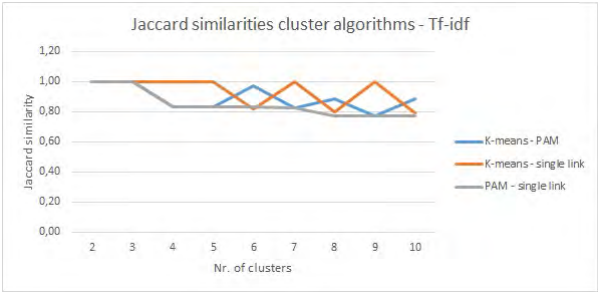


FIGURE 6.8: Jaccard similarities on clustered tf-idf weighted documents

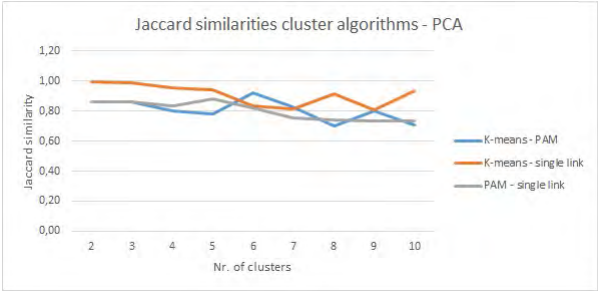


FIGURE 6.9: Jaccard similarities on clustered PCA weighted documents

Chapter 7

Topic modeling

The previous chapter discussed document clustering, a widely studied statistical method which groups similar documents. The main purpose of document clustering, just as in this research, is gaining insight into the contents of data. However, self-contained clusters are not easily interpretable and thus do not provide sufficient understanding.

Topic modeling is therefore of great value to clustering and clusters generated. This method discovers the topics, i.e. latent semantics, in a given set of documents. The topics do provide insight into the content as well as the quality of the clusters generated and can be used for various purposes [Xie and Xing, 2013]. For instance, topic modeling on-top-of clusters can help to validate that the correct cluster cut-off level was chosen. A wide variety of non-related topics within clusters could mean the number of clusters is too small. Vice versa, too specific topics within a cluster could mean that the number of clusters is again too large. Furthermore, for the specific area of e-discovery, topic modeling on-top-of-clusters can help project stakeholders to assess the semantic content in each cluster, beyond the cluster results. This will allow them to make decisions regarding which clusters to review.

Although topic modeling is a relatively new subject, a good deal of research there has been conducted, which is reviewed and summarized in the first section. The application of topic modeling, subsequent to the cluster results, are discussed in the second section, together with the final results on the experiment performed.

7.1 State of the art

As mentioned in the second chapter, topic modeling is a statistical technique. It has the aim to obtain a probabilistic model that extracts latent indications of thematic structures in a set of documents and categorizes documents, i.e. assigns documents to topics, based on the structures identified. This method enables the summarization and organization of documents at a speed that cannot be matched by humans[Sethi and Upadrasta, 2012].

Multiple approaches exist which all make the same basic assumptions [Aggarwal and Zhai, 2012]:

- Given the fact that a document set consists of a certain number of topics, each document has probabilities of being assigned to each of these topics. In total, the probability for each document is 1.

  In mathematical terms, this assumption can be described as follows:

  The set $D = \{d_1, ..., d_n\}$ consisting of n documents, has for each document d_i a probability of being assigned to one of the topics $T = t_1, ..., t_K$. As the number of topics T is analogous to the number of clusters C, i.e. each cluster consists of one topic, the following holds:

  $P(t_j|d_i) = P(c_j|d_i)\ \forall i, j$ where $P(t_j|d_i) \in (0, 1)$

  Further:

  $\sum_j P(t_j|d_i) = 1\ \forall d_i$

- Each topic is described by a number of terms which appear in the set of documents. Thus each term in the vocabulary has probabilities of belonging to the lexicon of the different topics. The total probability for a term to be assigned to a topic is 1.

  In mathematical terms, this assumption can be described as follows:

  Each topic t_j has a probability vector, indicating the probabilities that a term of the set $W = \{w_1, ..., w_N\}$, containing N terms, occur in the lexicon of topic t_j.

So, topics can be determined as distribution of a vocabulary with N terms. The probabilities $P(w_l|t_j)$ meet the following requirement:

$$\sum_l P(w_l|t_j) = 1 \ \forall t_j \text{ where } P(w_l|t_j) \in (0,1)$$

These two assumptions define a document as being a mixed model. Each document is a mixed model, with probabilities of belonging to a certain topic, whereby each topic is specified as a sequence of extrapolated words.

In order to find these mixed models for a document set, each topic modeling method basically conducts two different steps. Firstly, the probabilities are estimated of assigning a document to a certain topic. Secondly, the probabilities of terms describing the topics are determined.

That is, in mathematical terms: $P(t_j|d_i)$ and $P(w_l|t_j)$.

The aim of each topic modeling method is to find the correct probability values for both components, such that the best possible fit on the contents of the data is obtained. This means that the document set is assigned to topics which actually describe the contents. However, every algorithm has its own unique approach.

Two popular methods are the Latent Dirichlet Allocation (LDA) and Correlated Topic Modeling (CTM). Both are generative models, which means that an initial seed set of random topics with random terms are created. These are tested and subsequently refined to iteratively improve assignment accuracy. In other words, the algorithms first extrapolate hidden thematic structure through the generation of a random set and improve through iterative testing and refinement[Steyvers and Griffiths, 2006]. However, it needs to be noted that the specific topics are not known directly. Therefore, the aim of these methods is to infer the topics from the generative models based on the observed data.

The generative algorithms are shown below. They look very similar, though it needs to be mentioned that the methods contain one big difference in their assumptions with regards to the correlation of words. Latent Dirichlet Allocation (LDA) makes an assumption regarding the correlation of words which is different from the assumption made by Correlated topic Modeling (CTM). LDA assumes that words are uncorrelated, that the order in which words occur in the text are not significant. As such, this method uses the

'bag of words' approach. CTM however, assumes words are correlated, and thus treats as significant the order of words in documents [Blei and Lafferty, 2009].

Latent Dirichlet Allocation

The LDA algorithm contains four steps in the process of generating the observed variables from hidden features [Blei, 2012, Blei and Lafferty, 2009]. Figures 7.1 and 7.2 show the algorithm via an image and graph representation.

Firstly, the number of topics K is chosen. Hereafter, the mixes models are determined by determining the term distributions for each topic, and determining the topic distributions for each document. Both term and topic distributions are Dirichlet distributions. Lastly, words are generated, equal to the number of words in the document. This is done by choosing a topic, and then from that topic chosen to generate a word.

The idea behind this algorithm is that in case of the correct distribution, the words created approximately reproduce the actual set of documents. As such, assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

Below, the algorithm is shown in mathematical terms:

1. Choose a number of topics K
2. For each topic t_k determine the term distribution $\beta_k \sim Dirichlet(\eta)$
3. For each document D_d determine the topic distribution $\theta_d \sim Dirichlet(\alpha)$
4. For all words W_n in every document d, choose:
   (a) a topic $z_{d,n} \sim Multinomial(\theta_d)$
   (b) a word $w_{d,n} \sim Multinomial(\beta_{z_{d,n}})$

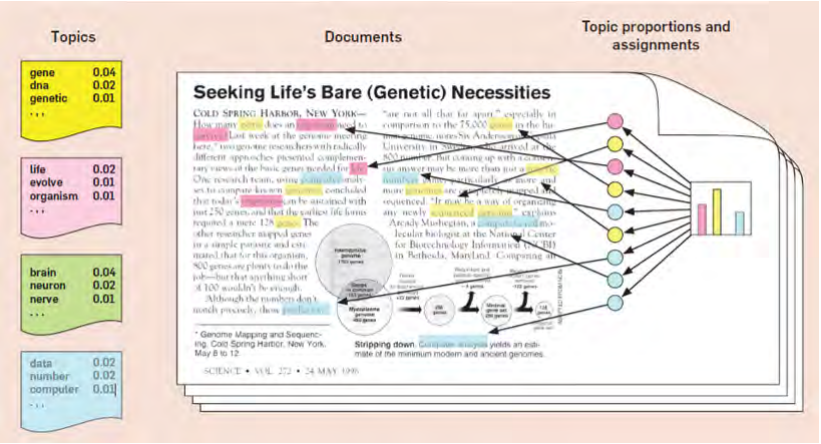


FIGURE 7.1: Image representation of LDA process

Source: [Blei, 2012]

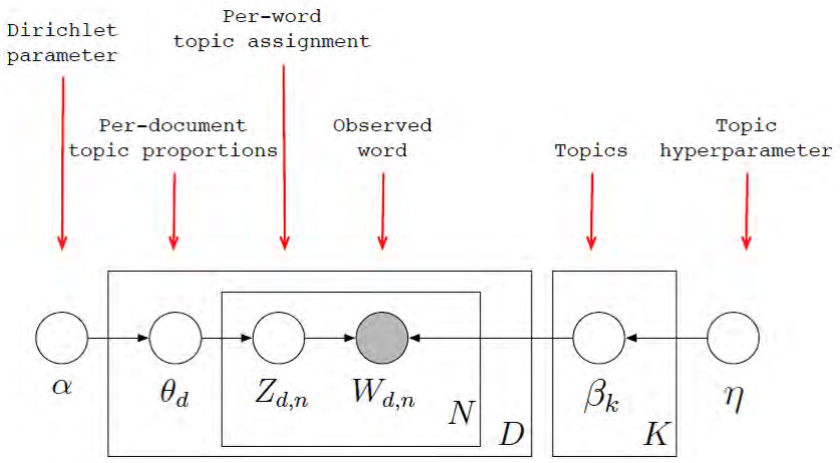


FIGURE 7.2: Graph representation of LDA process

Source: [Sethi and Upadrasta, 2012]

Correlated Topic Modeling

The correlated topic modeling approach is quite similar to LDA except that the topic distribution is drawn from a logistic normal distribution instead of the Dirichlet distribution. *(see figure 7.3 for the graphical representation)*. In addition, it is important to notice that for this method the notation used has an alternate meaning. As CTM assumes words are correlated $w_{d,n}$ is not just a word with index n in document d, but it is the n^{th} word occurring in the d^{th} document. The same holds for topic $z_{d,n}$ assigned to the n^{th} word in the d^{th} document [Blei and Lafferty, 2009, Grün and Hornik, 2012].

In mathematical terms, the algorithm is as follows:

1. Choose a number of topics K
2. For each topic t_k determine the term distribution $\beta_k \sim Dirichlet(\eta)$
3. For each document D_d determine θ, the rates of the topic distribution:
   (a) Set $\eta_d \sim N(\mu, \Sigma)$ with η a vector of K values, Σ a K*K covariance matrix
   (b) Compute $\theta_d = \frac{e^{\eta}}{\sum_{i=1}^{k} e^{\eta_i}}$
4. For all words W_n in every document d, choose:
   (a) a topic $z_{d,n} \sim Multinomial(\theta_d)$
   (b) a word/term $w_{d,n} \sim Multinomial(\beta_{z_{d,n}})$

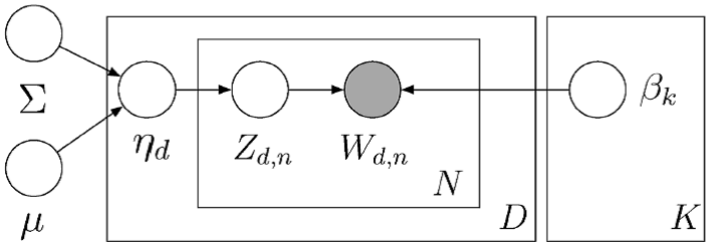


FIGURE 7.3: Graph representation of CTM process

Source: [Sethi and Upadrasta, 2012]

For more detailed descriptions of the methods I refer to the papers used.

Both algorithms result in a joint probability distribution, i.e. a framework of all possible combinations of the unobserved and observed variables. For the Latent Dirichlet Allocation it is given as follows:

$$p(\beta, \theta, z, w) = p(\beta|\eta) * p(\theta|\alpha) * p(z|\theta) * p(w|\beta_{z_{d,n}})$$

Here, the topics β, the topic distributions θ, and the per-word topic assignments z are unobserved variables, of which the values can be determined by use of the posterior

distribution. This is the distribution of the unobserved variiabes, given the obeserved variables, in this case the words w. In mathematical terms this is formulated as the joint probability distribution divided by the marginal probability distribution of the words.

$$p(\beta, \theta, z|w) = \frac{p(\beta,\theta,z,w}{p(w)}$$

However, calculating the posterior by this formula is a big problem. Theoretically, it is possible, but the amount of different topic frameworks possible is very large, even exponential. So in practice it is almost impossible. Therefore, a couple of methods have been discovered which estimate the posterior closely. Two well-known and often used methods are Gibbs sampling and the Variational Expectation Maximization [Sethi and Upadrasta, 2012].

- **Gibbs sampling:** Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method. It constructs a Markov chain, which is a process of random events. In each state a transition is made, depending on the current state, such that it finally converges to the distribution preferred. In case of topic modeling, the goal is to find the posterior distribution. First, every word in the vocabulary is semi-randomly assigned to the K topics. Then in each state of the Markov chain, an assignment of word $w_{d,n}$ to topic $z_{d,n}$ is updated by calculating the probability of that word $w_{d,n}$ occurs in topic t_j and the probability that topic t_j belongs to document d_i. This process is run iteratively until the distribution converges.
- **Variation Expectation Maximization (VEM):** VEM does not make use of sampling. It is a deterministic approach trying to optimize the fit with the posterior by trying different methods from the same family of distributions. First, a family of distributions is chosen over the hidden features, and then each member is compared to the posterior. The member that has the best fit is adopted as the estimate of the posterior.

For more detailed descriptions of the methods I refer to the papers used.

The estimated posterior distribution generated is the main result of both LDA and CTM. However, differences have been detected in the accuracy of the different topic modeling methods. It needs to be noted that compared to traditional cluster methods,

relatively little research has been conducted on topic modeling. This is because the subject is fairly new. (LDA was developed in 2003, and CTM was only developed in 2007). However, some clear conclusions have been drawn in the few studies performed to date. The main conclusion concerns the assumptions made by LDA and CTM regarding the correlation of text. Namely, LDA does not only assume that words and topics are not correlated, it is also not able to detect correlations. This is due to the Dirichlet allocation used to compute the topic distribution. CTM uses the logistic normal distribution for its topic distribution, which detects the correlations automatically. This generally results in higher topic coverage compared to the amount of topics supported by LDA. In addition, CTM generally gives a better fit to the semantic content of the data, as it is well known that words indeed are correlated based on proximity [Blei and Lafferty, 2009].For example, the words *economics* and *politics* will be mentioned more often together in texts, then *economics* and *sunglasses*. However, since LDA is a simpler method compared to CTM, its computational overhead is generally lower.

Further, it is important to note that there is also a difference in the posterior approximation methods applied: Gibbs sampling and Variation Expectation Maximization (VEM). Since Gibbs sampling requires more steps in order to reach a good approximation, it is much slower compared to VEM. For datasets of moderate size operating on a standard workstation, the computation time can vary from a couple of hours for VEM to multiple days for Gibbs sampling. However, Gibbs sampling is more accurate: it fits to the posterior distribution more closely.

Lastly, topic modeling has a couple of initialization problems that affect the results. Therefore the initialization has to be handled carefully. The first problem is the parameter values chosen for α and η. The correct values depend per case. However, in their paper written, Griffiths and Steyvers [2004] identified a set of best practices that generally work well. Today, these values are often used in topic modeling initialization:

$\alpha = \frac{50}{K}$ with K the number of topics

$\eta = 0.1$

Secondly, the number of topics K has to be provided as input as well. This is the biggest obstacle in topic modeling, since as with clustering it is not possible to determine the number of topics a-priori. In addition, it is important to find the correct value of K,

since there is a probability of under- or overfitting otherwise [Sethi and Upadrasta, 2012]. Therefore, topic modeling methods are run several times with different values K in order to determine the best fit for the specific corpus. Often, use is made of cross-validation. Subsets are created on which a topic modeling method is run with a specific value K given as input. Then, the resulting topics are checked on their robustness, which is calculated by likelihood or similarity measures. the greater the agreement of the results amongst the subsets, the robuster the topics. The most robust topics correspond to the right amount of K.

The value for the number of topics K necessary to fully describe the content of the Enron dataset is examined in the practical experiment. K depends upon and is determined for various combinations of the methods investigated.

7.2 Methods and techniques used

Since the goal of the investigation of the Enron dataset is to develop an approach to prioritize documents for review by grouping and profiling them, the application of topic modeling on-top-of clusters could be a useful addition to understand the topics covered in each document and cluster. Therefore, in the practical experiment the two methods clustering and topic modeling are integrated.

This integration concerns a process followed, as shown in figure 7.4. After preprocessing the data and reducing the dimensionalities, the documents were clustered as discussed in the previous chapters via the traditional clustering method. For each resulting cluster of similar documents, topic modeling was then applied. Then, for the set of clusters created the optimal number of topics is chosen by use of cross-validation. Finally, the process is repeated for another amount of clusters.

Here, the assumption is made that clusters containing similar documents also have quite similar topic distributions and approximately discuss the same topics. The hypothesis is that the resulting topics within a cluster are related, and that these topics are distinct conceptually from topics assigned to other clusters.

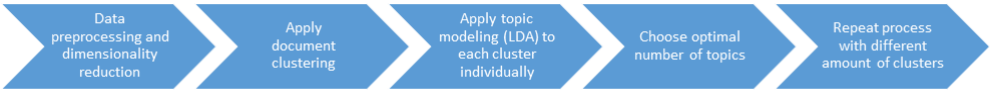


FIGURE 7.4: Integrated clustering and topic modeling process

An visual example of a result from the process followed in shown in figure 7.5. It represents two clusters created, each with a set of documents from which three topics are extracted.

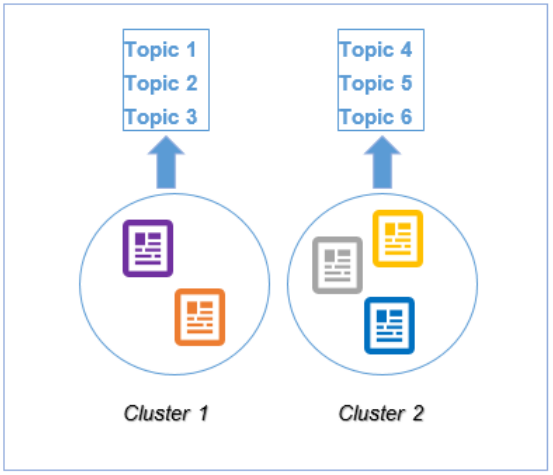


FIGURE 7.5: Graphical representation of desired clustering and topic modeling results

The topic modeling method used to realize the topics is Latent Dirichlet Allocation (LDA). Although Correlated Topic Modeling (CTM) seems to achieve slightly better results, it also requires higher computational overhead. Unfortunately the program R used to conduct the test developed instabilities when applying CTM, so topics were extracted using LDA.

The LDA was calculated in R by using the function LDA(), with multiple arguments given as input:

- **a document-term matrix of documents belonging to cluster c**: In the preprocessing step, a document-term matrix was extracted with a specific term weighting and reduced dimensionality based on one of the three methods discussed. The document-term matrix was provided as input to the LDA method is a part of the total document-term matrix, containing the documents belonging to cluster c.

- **the number of topics *K* for cluster c:** The number of topics desired is between 2 and 10. The practical reason for this range is that the results should be distinguishable without being reductive or overwhelming in number. A larger number of topics could result in losing the overview. However, this method is of course able to handle larger numbers of K.

- **the posterior estimation method:** The default estimation method in the LDA() function in R is Variation Expectation Maximization (VEM). Gibbs sampling is possible as well, but it is not recommended for use due to the computational overhead involved. Therefore, VEM is chosen as the posterior estimation method in this experiment.

- **the initial value of α and η:** The default values are $\frac{50}{K}$ and 0.1, as discussed in the 'state of the art'-section. Because of the successes mentioned in several papers these values are held.

Figure 7.6 shows the final results generated in this experiment in the form of a tree structure.

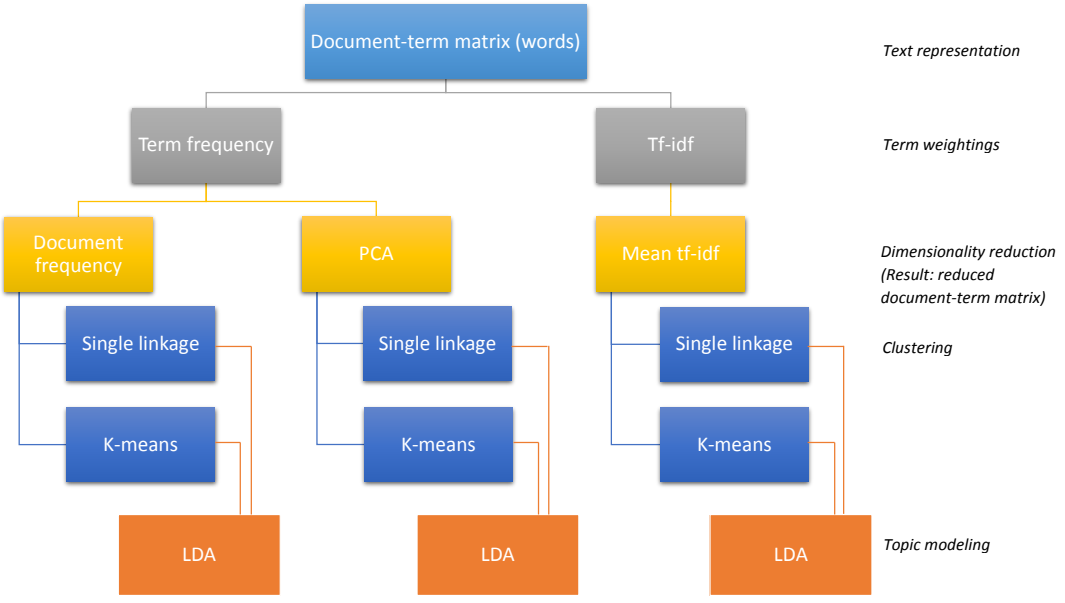


FIGURE 7.6: Process followed in practical experiment

It shows that Latent Dirichlet Allocation (LDA) is applied to six different combinations of clustering methods, single linkage and k-means, and document-term matrices, created by different term weightings and dimensionality reduction methods. As can be seen, the clusters created by Partitioning Around Medoids (PAM) are not included in this process, since it has been shown that R has a very long run time for this clustering method. In combination with LDA, the run time was no longer reasonable for this practical experiment given the single workstation computing resources applied and therefore the choice was made to leave this method out.

Six different combinations of clusters were created with the amount of clusters ranging from 2 to 10, as discussed in the previous chapter. In this last step the corresponding topics are obtained for each cluster within each set of clusters. The optimum number of topics for each quantity of clusters together is calculated on the basis of cross-validation as discussed in the previous section. The resulting amount K is given as input to the LDA()-method.

7.3 Results

Figure 7.7 *(left)* shows the optimal number of topics generated for each quantity of clusters from 2 to 10. Each line represents the results of LDA applied on top of clusters that were created by a certain method and on a certain document-term matrix. It was observed that for almost all methods, the optimal number of topics decreases as the number of clusters becomes larger. This is a logical consequence as the general semantic complexity or breadth of a cluster, the amount of topics covered, should decrease as the cluster becomes more specific. Likewise, as a cluster becomes larger (contains more aggregate documents), it becomes conceptually broad and thus requires a larger set of topics to generalize across documents. If all documents are grouped in one and the same cluster, all possible topics characterize the single cluster. However, if the documents are grouped by similarity in a number of clusters, not all topics will occur in all clusters anymore (they will be segmented across the range of clusters). As the number of clusters increases, groupings require a higher similarity between documents, which results in more specific topics in each cluster.

As mentioned and demonstrated, most but not all methods discussed in this paper observe this concept approximately. It does not apply to topics found in clusters produced by the single linkage method using the data reduced through Principal Component Analysis (PCA). It is not clear why this is the case. The reason has not been found, in part due to the fact that PCA is a feature transformation method. This makes the performance of the method more difficult to interpret.

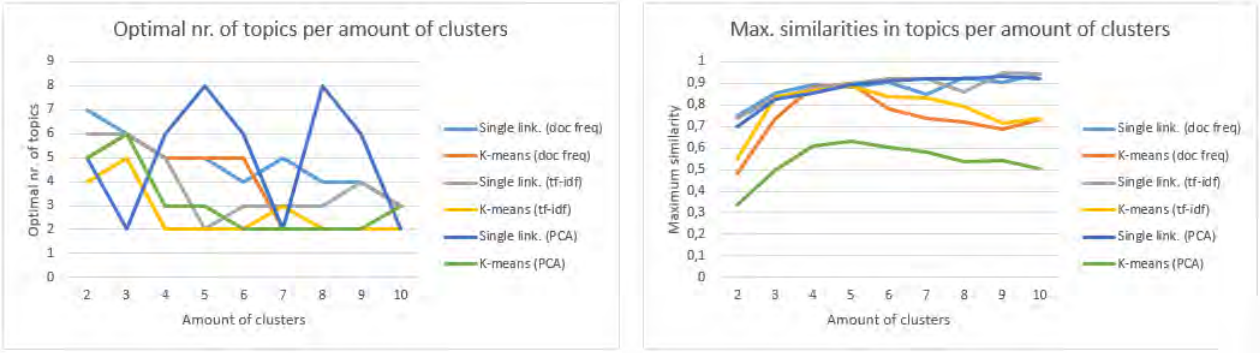


FIGURE 7.7: Topic modeling results regarding optimal nr. of topics

However, in spite of the fluctuation in the optimal amount of topics per cluster, this method, LDA integrated with single linkage and applied on data reduced by use of PCA, generated the highest similarities in this experiment. The results are shown in figure 7.7 as well, on the right. This plot represents the average Jaccard similarities corresponding to the number of topics chosen per cluster. As mentioned in the previous section, 10 fold cross validation was applied to each cluster created. This results in ten sub-sets, for each of which topics were determined. The average Jaccard similarity was calculated based on the similarities in the topics per fold. The highest similarities shown in the plot are the average similarities corresponding to the optimal number of topics found.

The combination of LDA and single linkage generally seems to give the highest similarities. The three lines on top of the plot are the grey and blue lines, each representing the results of LDA on clusters created by the single linkage method for the three different document-term matrices. Furthermore, it appears that the documents clustered by k-means on the features reduced by principal components, give poorer results.

Based on similarity measures, the optimal number of topics per amount of clusters can be determined, but no strong conclusions can be drawn regarding to the 'best' integrated

clustering and topic modeling method which provides all of the following wishes:

- related topics within clusters and not related topics across clusters
- best fit of topics on actual contents of documents
- reasonable computational overhead
- an ideal number of clusters
- an optimal number of topics

As already mentioned in 6, an all-purpose algorithm, by nature, does not exist. Results are specific per corpus, case, and user objectives. Further, as topic modeling and clustering are both unsupervised learning methods, the results need to be validated by people based on knowledge and their expertise.

However, something more can be said about the computational overhead of the method used to calculate the optimal number of clusters for a particular number of documents. Figure 7.8 (left) shows that between 12 and about 15 hours were needed on the single workstation utilized.

The outliers are the clusters created by k-means based on the dataset reduced by the tf-idf, and the single linkage on the dataset reduced by means of the document frequency. It appears that the differences are reasonable, but for a large number of documents this difference can become fairly high. However, these results are only valid for this particular experiment. The results could be different for other cases.

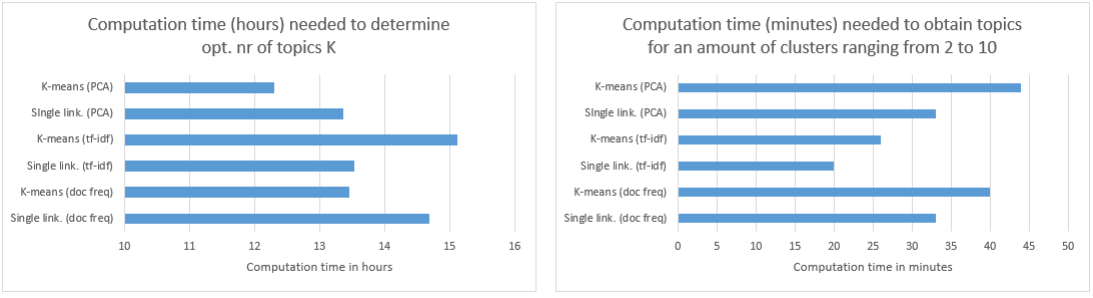


FIGURE 7.8: Computation time required for topic modeling

The computation time required for the determination of the topics after the optima were known (figure 7.8 (right)), however, was much shorter. This was between 20 and

45 minutes. In this case, the calculation was via K-means. The calculation of k-means for data sets reduced by means of PCA and document frequency took the longest time.

In summary, figure 7.8 shows that each method has its advantages and disadvantages in terms of run time. Therefore, given the computation time a method will be chosen depending on the wishes of the user. However, some methods, such as Correlated Topic Modeling in this process step, were not feasible at all.

As such, the only topic modeling method tested in this practical experiment was LDA. By use of this method topics were determined for a total of six different combinations of datasets and cluster methods. Since the number of clusters tested for each combination ranged from 2 to 10, this resulted in $6 * 9 = 54$ topic modeling results generated. An example of results generated is shown in figure 7.9. This shows topics for five clusters generated by k-means on a tf-idf weighted dataset. A more comprehensive set of results is added in Appendix D. The decision was made not to add all results in this paper, given the large number of pages and relatively few content-based insights obtained.

As just mentioned, it is difficult to validate the resulting topics via an algorithm as this is an unsupervised approach. Therefore, validation was performed by a group of different and independent people on the basis of their human intuition without having knowledge of the exact contents.

As expected, although not major, the results differ per combination of dataset and methods used. Still, three general conclusions could be drawn:

- The topics within clusters are conceptually related,
- The terms describing topics are unique to each cluster,
- Multiple topics describe one cluster but sometimes repeat for a cluster.

The first two conclusions are desired. The goal is that first insights can be obtained in the content of documents by dividing them and the topics related into clusters. The topics provide a clean distinction of the content. The latter conclusion is less desirable. It shows that the optimal number of topics defined, in the example three per cluster, is not always correct. As can be seen from figure 7.9, the first cluster contains terms occurring only once. The second cluster, however, includes terms that occur several

times. This might hold that for the first cluster, indeed 3 topics are optimal, but for the second cluster a smaller amount of topics would be enough.

Cluster1			Cluster2			Cluster3		
Topic1	**Topic2**	**Topic3**	**Topic1**	**Topic2**	**Topic3**	**Topic1**	**Topic2**	**Topic3**
Enron	Com	Jan	Jan	Mar	Ffffff	Darron	Darron	Darron
Mapi	Will	Gas	Mar	Feb	Jan	Kristi	Kristi	Round
Ect	Eomonth	Max	Jun	Apr	Jun	Round	Round	Jan
Employe	http	Min	Apr	Ffffff	Aug	Jan	Jan	Kristi
Compani	Market	Sum	Jul	Sep	Jul	Growth	Growth	Total

Cluster4			Cluster5			Cluster6		
Topic1	**Topic2**	**Topic3**	**Topic1**	**Topic2**	**Topic3**	**Topic1**	**Topic2**	**Topic3**
Vlookup	Vlookup	Fals	Slrtb	Slrtb	Slrtb	Assum	Assum	Assum
Fals	Isna	Match	Electr	Elect	Region	Sum	Edat	Cashflow
Match	Data	Colmmatch	Region	Rate	Current	Cashflow	Sum	Basi
Datarang	Fals	Datarang	Rate	Heat	Elect	Edat	Basi	Sum
Jan	Colmmatch	Transact	Year	Region	Heat	Incom	Draw	Draw

FIGURE 7.9: Example of topics generated for 5 clusters

Chapter 8

Conclusion and discussion

This research conducted led to several results, both on high-level and low level. The main findings on a quite high-level are discussed and interpreted in this chapter. In addition, the added value of this study is defined in terms of strengths and limitations, leading to recommendations for further research and application in practice.

8.1 Main findings

As raised in the introduction, the techniques investigated hold great promise for speeding-up document reviews by segmenting documents automatically based on subject matter. Such approaches are increasingly essential as data overload is flooding most organizations and is becoming a stumbling-block in discovery projects.

The main findings on these techniques are considered to be the so-called best practices, from which guidelines for the application in practice are obtained:

- The research examined the computational performance and outputs of various text analytics algorithms. As was pointed out several times in the research, there is no single automated diagnostic algorithm that can be used to assess whether a particular unsupervised text analytics algorithm is better or worse in terms of results. The implication is that human review is always needed to validate the results as they are semantic in nature, they are tied to complex human factors: language, business and legal context, culture, and societal.

- The overhead associated with applying these techniques is non-trivial. The computation time required as well as the sources needed to achieve the projects objectives vary per case, corpus and methods used.
- The methods investigated are computing intensive. In the practical experiment conducted, methods were tested on a relatively small set of documents. Howewer, the computation time required is quite high, and increases for a larger number of documents.
- While the techniques investigated are powerful in focusing discovery efforts, they do not appear to be appropriate as an initial step in the discovery process. This is due to the intensive computational overhead required, as described above. The corpus needs to be reduced before the text analytics process is applied (see section 8.4.

8.2 Interpretation of results

As discussed in the previous chapters, the specific results generated in the practical experiment are difficult to interpret. Firstly, due to the unsupervised learning methods investigated. Human intuition is needed in order to evaluate the topics generated, however, this is a subjective and time consuming task.

Further, it is difficult to compare the results with previous research due to the very specific area investigated in this study regarding the use case and combination of methods. Within the use case of e-discovery relatively little research on text analysis is done so far. Since the results of the methods investigated may vary per use case, comparative research therefore can not be done properly. This also holds for the integration of clustering and topic modeling in which limited research has been done. Since generally clustering and topic modeling are carried oud separately, little comparative research can be done into the topics resulting from the integration of these statistical methods. The interpretation of the results by means of comparison is thus mainly left to future research.

8.3 Strengths and limitations

The results of this study are best practices in the process that can be followed in e-discovery projects with the aim of grouping documents and the corresponding topics cleanly. The implementation of this process before starting document review, can give easy and fairly quick insight into the content of the data. Running the process in a software package might take up to a couple of hours, but this is still much faster than using human review. In addition, as documents and related topics are separated cleanly, it is possible to determine based on the first insights which document clusters will be reviewed.

However, it is also important to value the quality of the results, i.e. topics, returned. Due to the fact that this research performed is a form of unsupervised learning, quality assessment is almost not possible. In addition, this study had some other limitations with respect to the practical experiment. The methods discusses in the state of the art sections are not all explored in practice due to the limited capacities of R. Finally, the process and methods discussed are tested on a dataset publicly available, but the added value has not been studied in a real case. This is one of the recommendations for further research, which are further discussed in the next section.

8.4 Recommendations for further research and application in practice

Below, the recommendations are given for both further research and the application in future projects. These recommendations are based on the limits of current research and opportunities discovered.

- In this research, a single corpus was investigated. In order to draw comprehensive conclusions, cross-sectional research is needed concerning how different document sets perform. Then, the it can be validated how unique subject matter and conceptual breadth accompanied affects performance and results.

- Even before pre-processing text, it is preferred to focus and refine the document corpus by interviewing key stakeholders and experts, in particular to identify keywords, subjects, and people of interest in the investigation. By refining and reducing the corpus before the application of text analytics, the semantic breadth of the content is reduced, which allows the algorithms to divide into more meaningful sub-categories. As well, computing and human overhead is reduced by reducing the corpus up-front prior to application of text analytics.

As the overhead associated with the application of techniques investigated is non-trivial the following recommendations are made regarding the use in future e-discovery projects:

- Time is needed to pre-process documents, to apply the algorithms, to run diagnostics, and to validate and refine the application of the algorithms. Clients may tend to have overly high expectations concerning automation and rapidity due to misunderstandings related to computing power and automated computer decision making. Expectations concerning time and overhead should be set up-front at the beginning of the project.

- As discussed above, the methods investigated are computing intensive: a single workstation is not enough in most cases. A number of the algorithms could not be run due to insufficient computing resources. In order to speed computation, a multi-processer server with a large amount or RAM is preferable, or a cloud-computing approach.

- While R was applied in this study, R is not the only option for conducting text analytics. A number of other open source and commercial options exist all having their (dis)advantages. Cost, computing power available, security, privacy, and software licensing restrictions, and cost are need to be taken into account when selecting a software approach.

- An expert trained in the application of text mining is necessary. An expert understanding of the text processing, text analytics algorithms, algorithmic configuration, software operation, and interpretation of results is necessary in order to obtain the best results.

- Finally, next to a text analytics expert key stakeholders and subject matter experts are needed to assist in determining the objectives for the investigation, specifying

criteria for selection of the document set, and for validating results from the algorithms. They particularly assist the text analytics expert in specifying the correct cut-off level for clustering and validating topics extracted (when applying topic modeling).

The final judge on results of unsupervised text analytics is thus a subject matter expert familiar with the domain and documents investigated. However, human review is expensive with regard to time and resources required. Typically experts do not have time to properly review results. As such, a potential future research program would be to:

1. identify ways to make human review more efficient, for instance to develop an algorithm to determine how many documents in a corpus need to be sampled and reviewed for a particular algorithm or set of algorithms to ensure the results are representative

2. identify and propose standard ways to utilize human reviews of representative samples to validate multiple unsupervised algorithm results. If such research validated many cases, it would be possible to assess if particular algorithms performed better according to human reviews across a broad range of document sets and cases. For instance, a confusion matrix could be applied on text analytics results utilizing representative results from human review validation.

This page intentionally left blank.

Appendix A

System and software characteristics

This appendix contains most relevant information of the system and software package used with regard to the practical assignment.

System

1. Brand:	HP
2. Model:	W8-13-11-0001
3. Processor:	Intel Core i5-4300U CPU @ 1.90GHz 2.50
4. Installed memory (RAM):	16,0 GB (15,0 GB usable)
5. System type processor:	64-bit Operating System, x64-based

Software

1. Package:	R
2. Version:	3.1.3 (2015-03-09)
3. Platform:	x86_64-w64-mingw32/x64 (64-bit)

This page intentionally left blank.

Appendix B

Dimension reduction methods

During the experiment conducted, multiple methods were investigated that reduce dimensionality. The methods discussed in chapter 5, are extensively described here on the basis of their formulas.

B.1 Feature selection

Document Frequency: This method ranks and selects terms based on their document frequency, the number of documents d_i in which term k occurs at least once:

$$DF(t_k) = \sum_{i=1}^{n} f_{t_k}(d_i)$$

with

$$f_{t_k}(d_i) = \begin{cases} 1 & \text{if } \#(t_k, d_i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Term Strength: This method ranks and selects terms based on the term strength, a measure indicating the conditional probablity of a term t_k appearing in the second part of a set of documents with similarity S, given the knowledge that it appeared in the first part:

$$TS(t_k) = p(t_k \in d_i | t_k \in d_j) \qquad d_i, d_j \in D \text{ and } S(d_i, d_j) > \alpha$$

Entropy-based Ranking: This method ranks ans selects the terms based on the decrease of the entropy when deleting term t_k from the feature set. The Entropy is in mathematical terms defined as follows:

$$E(t_k) = \sum_{i=1}^{n}\sum_{j=1}^{n}(S(d_i,d_j) * log(S(d_i,d_j)) + (1 - S(d_i,d_j)) * log(1 - S(d_i,d_j))$$

with similarity between documents d_i and d_j defines as:

$$S(d_i,d_j) = e^{-\beta * dis(d_i,d_j)} \qquad \beta = -\frac{ln(0.5)}{dis}$$

Here, $dis(d_i,d_j)$ is the distance between both documents.

Term Contribution: This method ranks and selects terms based on their contribution to the similarity of the documents, determined by the term frequency-inverse document frequency (tf-idf) weighting of term t_k in document d_i. The similarity between documents is in mathematical terms defines as follows:

$$S(d_i,d_j) = \sum_{t_k} f(t_k,d_i) * f(t_k,d_j)$$

This formula is used to compute the term contribution of term t_k

$$\sum_{i;i\neq j}^{n}\sum_{j;i\neq j}^{n} f(t_k,d_i) * f(t_k,d_j) \qquad \text{with} f(t_k,d_j) = \text{tf-idf}(t_k,d_j) = \#(t_k,d_j) * log\frac{|D|}{\#(t_k,D)}$$

B.2 Feature transformation

Principal Component Analysis: This method transforms terms to the so-called principal components and selects them based on the extent to which they explain the variance of the dataset.The approach consists of four steps:

1. **Scale the original terms:** The terms are transformed to (approximately) the same scale, by for example substracting the mean such that each feature has a mean of 0, or by changing the number formatting to percentages.

2. **Compute the covariance matrix:** For every two terms t_k and t_p their covariance is calculated by multiplying their correlation with the standard deviations:

   $$Cov(t_k,t_p) = Corr(t_k,t_p) * (\sigma_{t_k} * \sigma_{t_p})$$

The covariance indicates the extent to which the two terms move together. Calculating the covariance for each combination of terms, results in a square m*m - covariance matrix Σ, with m the number of terms:

$$\mathrm{Var}[X] = \begin{bmatrix} \mathrm{Var}[X_1] & \mathrm{Cov}[X_1,X_2] & \dots & \mathrm{Cov}[X_1,X_K] \\ \mathrm{Cov}[X_1,X_2] & \mathrm{Var}[X_2] & \dots & \mathrm{Cov}[X_2,X_K] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_1,X_K] & \mathrm{Cov}[X_2,X_K] & \dots & \mathrm{Var}[X_K] \end{bmatrix}$$

FIGURE B.1: Covariance matrix

Source: http://www.statlect.com/mcdnrm1.htm

3. **Calculate eigenvalues and eigenvectors:** The terms are transformed by use of eigenvalues and eigenvectors. The eigenvalues provide the direction in which the dataset is transformed, and the corresponding eigenvectors indicate the degree of elongation. They are found by the following equation:

   $A * \mathbf{v} = \lambda * \mathbf{v}$

   For each eigenvalue λ corresponding to the covariance matrix A, a vector $\mathbf{v}$ exists, such that the equation above is true. Firstly, λ is found by calculating the determinant:

   $determ(A - \lambda * I) = |(A - \lambda * I)| = 0$ with I an m*m Identity matrix

   Then, a possible solution for $\mathbf{v}$ is found by solving the rest of the equation.

   This results in m eigenvalues of length 1 and m eigenvectors of length m*1. These eigenvectors are the principal components, and the eigenvalues indicate the variability explained by the corresponding eigenvector.

4. **Transform original to new dataset:** Finally, the new dataset X is calculated by multiplying the transposed principal components with the transposed original dataset of scaled terms:

$$X = \begin{vmatrix} PC_1 \\ PC_2 \\ \vdots \\ PC_n \end{vmatrix} * \begin{vmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{vmatrix} \qquad PC_1 = |PC_{1,1} \quad PC_{1,2} \quad \dots \quad PC_{1,m}| ; \qquad t_1 = |t_{1,1} \quad t_{1,2} \quad \dots \quad t_{1,n}|$$

FIGURE B.2: Dataset transformation by use of PCA

The content of the original data is (mostly) maintained in dataset X. However, the data is displayed differently. It is displayed in the patterns of relationships that have been detected in the data.

Independent Component Analysis: This method transforms and selects terms by revealing hidden signals. Three steps are followed:

1. **Apply whitening transformation:** This approach removes the correlations among terms in the data, by transforming the original terms to new terms such that the covariance matrix is the Identity matrix. This means that the covariance matrix only consists of variances with value 1 and no correlations.

2. **Create the model of latent variables:** As with PCA, the dataset is then transformed and rotated. In this case the goal is to minimize the Gaussianity on all different axis. However, the axis created don't have to be orthogonal.

   The model is defined as follows:

   $X = A * S$

   Here,

   X is the m*n matrix of observed variables, i.e. the original dataset A is a unknown m*s matrix, called the mixing matrix S is the s*n source signal matrix, i.e. the latent variables

   with n, the number of documents m, the number of terms s, the number of sources

   The unknown variables in this equation are A and S. They are calculated based on the assumption of statistical independence between the sources, or latent variables, s_i.

3. **Calculate independent components:** The independent components W can be calculated from the mixing matrix A, determined in the previous step:

   $W = A^{-1}$

   So, W is an s*m matrix, with s independent components and m the number of terms.

Latent Semantic Indexing: This method transforms the document-word matrix M into a new reduced matrix such that the distance between the two matrices is minimal. The consists of the following steps:

1. **Apply Singular Value Decomposition (SVD):** This least-squares method transforms matrix M as a combination of three matrices:

   $M_{mxn} = T_{mxp} S_{pxp} (D_{nxp})^T$

   Here,

   m = the number of terms

   n = the number of documents

   p = min(m,n)

   Further, the matrices T and D represent the terms and documents in the new space, and S is a diagonal matrix with singular values. Each singular value, gives the variation taken by that component.

2. **Reduce the matrix of interest S to S':** The singular values with the highest variation are chosen. This amount of singular values $k \ll p$ is usually between 100 and 150, and reduces S_{pxp} to S_{kxk}, ehich is the diagonal matrix S':

3. **Approximate matrix M in a lower dimensional space:** Now k singular values are chosen, M can be reproduced, by taking the first k rows of matrix T and D. However, as $k \ll p$, this results in a lower dimensional matrix:

   $M'_{mxn} = T_{mxk} S_{kxk} (D_{nxk})^T$

   M' is the least squares approximation of the original matrix M, i.e. the distance between the two matrices is minimal.

This page intentionally left blank.

Appendix C

Clustering methods

During the experiment conducted, multiple methods were investigated that cluster documents. The methods discussed in chapter 6, are extensively described here on the basis of their algorithms

C.1 Agglomerative hierarchical cluster algorithm

The figure below shows the general algorithm followed by the agglomerative hierarchical cluster methods in mathematical terms.

1: **procedure** PRIMITIVE_CLUSTERING(S, d) ▷ S: node labels, d: pairwise dissimilarities
2: $N \leftarrow |S|$ ▷ Number of input nodes
3: $L \leftarrow [\,]$ ▷ Output list
4: $size[x] \leftarrow 1$ for all $x \in S$
5: **for** $i \leftarrow 0, \ldots, N-2$ **do**
6: $(a, b) \leftarrow \operatorname{argmin}_{(S\times S)\setminus\Delta} d$
7: Append $(a, b, d[a, b])$ to L.
8: $S \leftarrow S \setminus \{a, b\}$
9: Create a new node label $n \notin S$.
10: Update d with the information

$$d[n, x] = d[x, n] = \text{FORMULA}(d[a, x], d[b, x], d[a, b], size[a], size[b], size[x])$$

for all $x \in S$.
11: $size[n] \leftarrow size[a] + size[b]$
12: $S \leftarrow S \cup \{n\}$
13: **end for**
14: **return** L ▷ the stepwise dendrogram, an $((N-1)\times 3)$-matrix
15: **end procedure**
(As usual, Δ denotes the diagonal in the Cartesian product $S \times S$.)

FIGURE C.1: Algorithm agglomerative hierarchical cluster methods

Source: [Müllner, 2011]

C.2 K-means example

Below, a small example is shown of the k-means algorithm discussed in chapter 6. It contains four steps in which 22 documents are assigned to two clusters. The example starts with the random initialization and ends with the attainment of fixed centers.

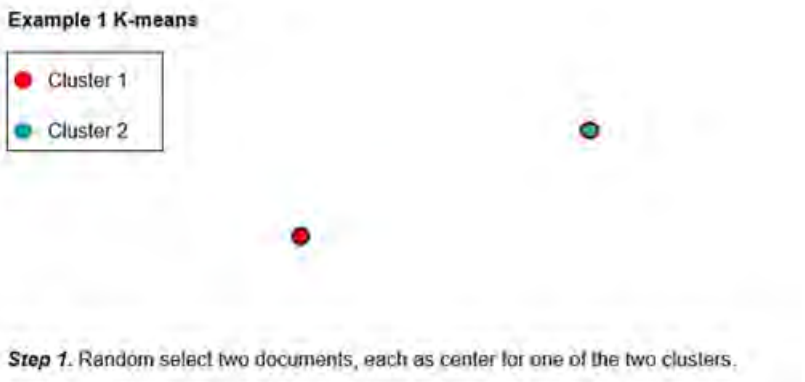


FIGURE C.2: Example k-means step 1: Random initialization of two documents

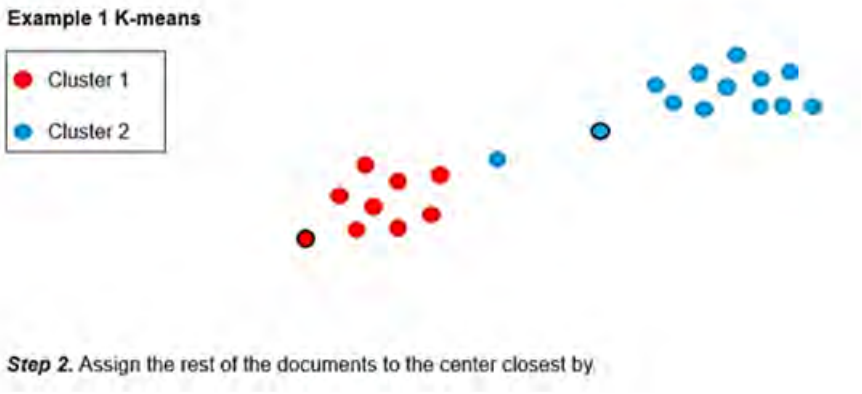


FIGURE C.3: Example k-means step 2: Assign documents to closest center

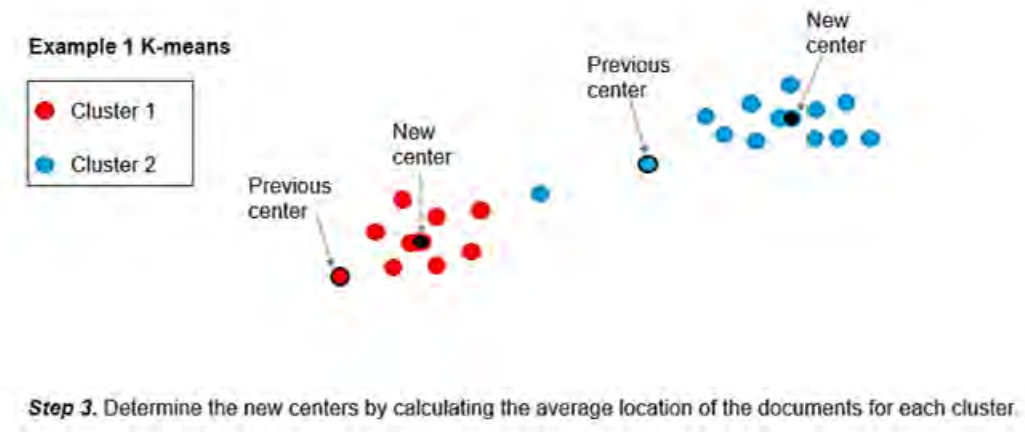


FIGURE C.4: Example k-means step 3: Determine new centers

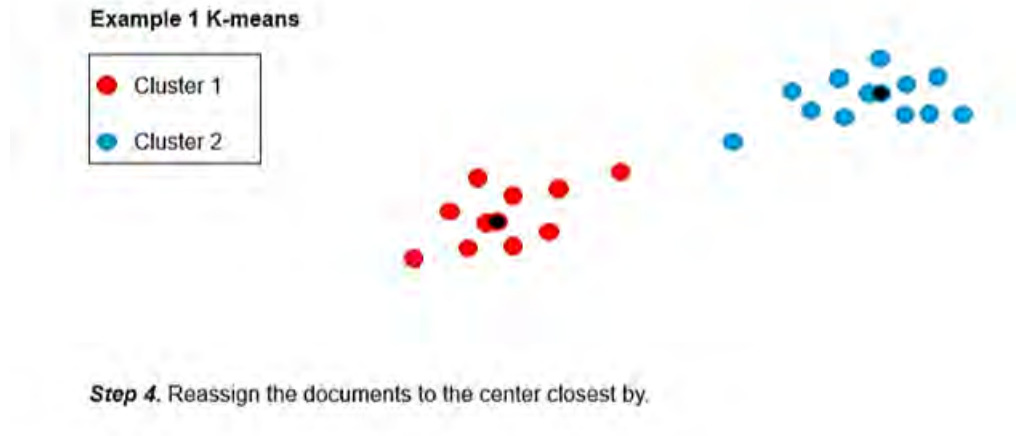


FIGURE C.5: Example k-means step 4: Reassign documents to closest center

C.3 Bisecting k-means algorithm

Below, the bisecting k-means algorithm is shown which is followed step by step until the desired number of clusters K is found.

1. Randomly choose a cluster which will be split.
2. Apply the basic k-means method in order to separate the documents into two sub-clusters
3. Repeat the second step i times, i.e. apply k-means multiple times and select the split with the highest document similarity.
4. Iteratively repeat the first, second and third step until K clusters are created.

C.4 Rand similarities

The plots below show similarities in which various methods have assigned the set of documents to a set of clusters. The similarity measure used is the rank similarity.

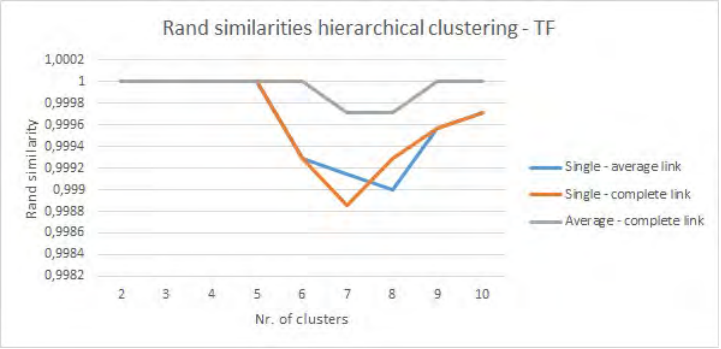


FIGURE C.6: Rand similarities on hierarchically clustered tf weighted documents

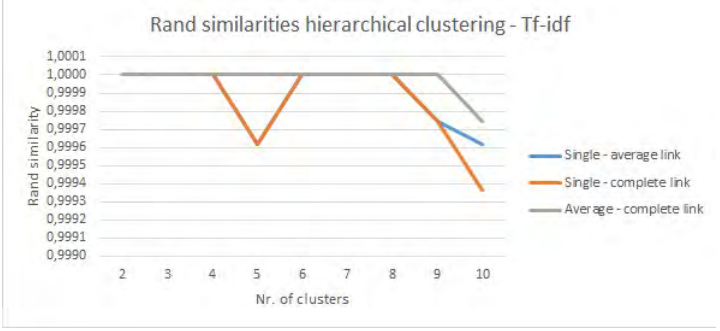


FIGURE C.7: Rand similarities on hierarchically clustered tf-idf weighted documents

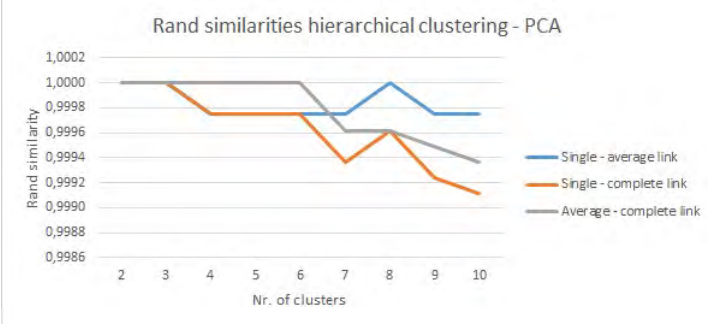


FIGURE C.8: Rand similarities on hierarchically clustered PCA weighted documents

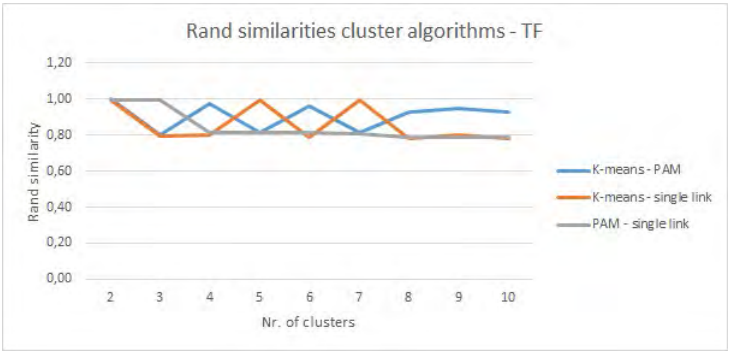


FIGURE C.9: Rand similarities on clustered tf weighted documents

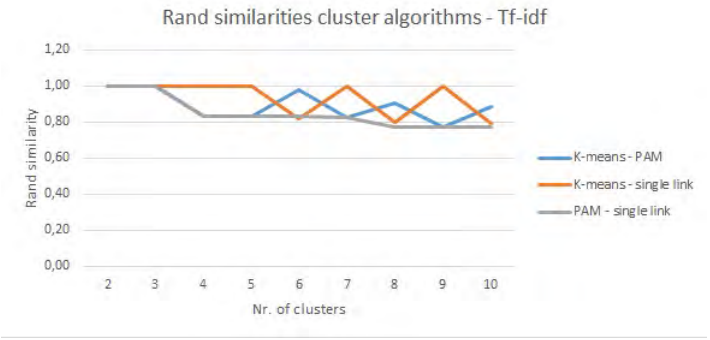


FIGURE C.10: Rand similarities on clustered tf-idf weighted documents

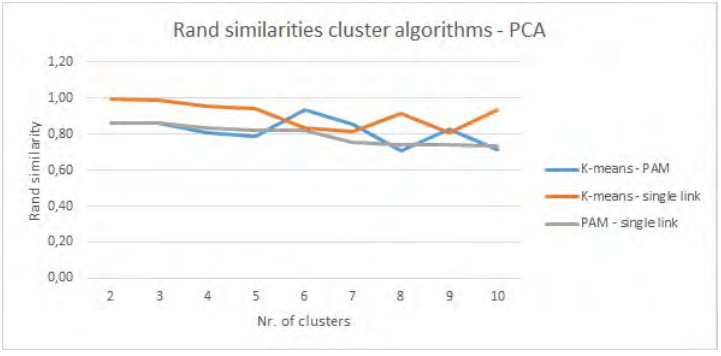


FIGURE C.11: Rand similarities on clustered PCA weighted documents

Appendix D

Topics generated

The figures below show the topics generated by use of Latent Dirichlet Allocation for cluster amounts ranging from 2 to 10. The clusters are created by use of single linkage and k-means, both on a term frequency weighted document-term matrix.

Single linkage clustering (TF)

Cluster1							Cluster2						
Topic1	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic6**	**Topic7**	**Topic1**	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic6**	**Topic7**
enron	Com	Darron	Fals	Mapi	Max	Will	Fffffff	Jan	Jan	Jan	Jan	Jan	Mar
Ect	http	Sum	Jan	Eomonth	Min	Market	Mar	Mar	Ffffff	Oct	Jul	Ffffff	Jan
Employ	www	Assum	Match	Enron	Energi	Compani	Feb	Fffff	Feb	Aug	Sep	Feb	Sep
Compani	url	Round	Day	Will	Total	New	Sep	Oct	Jul	Sep	Apr	May	Jul
million	internetshortcut	Jan	Yes	Message	Privat	Price	Jun	Sep	Mar	Feb	May	aug	Apr

Cluster1						Cluster2						Cluster3					
Topic1	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic6**	**Topic1**	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic6**	**Topic1**	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic6**
Max	Enron	Sum	Enron	Will	Fals	May	Ffffff	Ffffff	Jan	Ffffff	ffffff	Darron	Darron	Darron	Jan	Kristi	Darron
Min	Ect	Mapi	Employe	Market	Jan	Apr	Jul	May	Feb	Feb	Jan	Round	Round	Kristi	Kristi	Jan	Jan
Eomonth	Peopl	Com	Compani	New	Match	Jul	May	Feb	Mar	Mar	Feb	Kristi	Kristi	Jan	Round	Darron	Kristi
Power	Hou	Assum	Million	Compani	Day	Mar	Apr	Mar	Jun	Jan	May	Jan	Jan	Round	Darron	Round	Round
Energi	Gas	Http	california	time	Yes	Jun	Mar	Sep	Aug	Aug	Jul	Growth	total	growth	total	Growth	Total

Cluster1					Cluster2					Cluster3				
Topic1	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic1**	**Topic2**	**Topic3**	**Topic4**	**Topic5**	**Topic1**	**Topic2**	**Topic3**	**Topic4**	**Topic5**
Enron	Will	Sum	Enron	Jan	Ffffff	Aug	Jan	Feb	Feb	Round	Round	Darron	Darron	Round
Com	Market	Mapi	Employe	Gas	Feb	Ffffff	Mar	Ffffff	Ffffff	Darron	Darron	Round	Round	Darron
Ect	Compani	Max	Compani	Eomonth	Mar	Jul	Fffffff	Aug	May	Jan	Kristi	Kristi	Kristi	Jan
Hou	New	Min	Million	People	Oct	Apr	Jul	Sep	Jan	Kristi	Jan	Jan	Jan	Kristi
Will	Year	Assum	energy	power	Jan	May	Sep	Mar	Oct	Growth	Match	Total	Total	Total

FIGURE D.1: Topics generated on single linkage clusters, K = 2 - 4

Cluster4				
Topic1	Topic2	Topic3	Topic4	Topic5
Fals	Fals	Fals	Match	Fals
Day	Match	Jan	Fals	Match
Jan	Day	Yes	Summari	Jan
Weekday	Transact	Weekday	Day	Day
Transact	Weekday	Match	Weekday	Summari

Cluster1					Cluster2					Cluster3				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Enron	Max	Will	Com	Enron	Jan	Jun	Ffffff	Jan	Jan	Darron	Kristi	Round	Round	Darron
Employe	Min	Market	Jan	Mapi	May	Ffffff	Feb	Aug	May	Round	Round	Darron	Jan	Jan
Compani	Sum	Compani	Eomonth	Ect	Jun	Jan	Mar	May	Aug	Kristi	Jan	Kristi	Darron	Kristi
Million	Power	New	http	Hou	Jul	May	Jan	Jul	Jul	Jan	Darron	Jan	Kristi	Round
Energi	Gas	Busi	Peopl	Ena	Aug	Feb	Sep	Oct	Apr	Growth	Total	Total	Total	Total

Cluster4					Cluster5				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Dals	Fals	Day	Fals	Jan	Sum	Assum	Assum	Sum	Sum
Match	Match	Match	Jan	Fals	Incom	Sum	Cashflow	Assum	Cashflow
Jan	Day	Eol	Day	Match	Max	Cashflow	Jan	Jan	Jan
Yes	Jan	Data	Ltd	Day	Assum	Intern	Intern	Cashflow	Incom
Day	Monday	Transact	Weekday	Ltd	Year	Incom	Draw	Draw	Assum

Cluster1				Cluster2				Cluster3			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Will	Enron	Enron	Com	Jan	Jun	Ffffff	Aug	Darron	Jan	Darron	Darron
Market	Mapi	Ect	Jan	Feb	Oct	Mar	Ffffff	Jan	Round	Round	Round
Sum	Employe	Max	Eomonth	Ffffff	Jan	Feb	Jan	Round	Kristi	Kristi	Jan
Energi	Peopl	Min	http	May	May	Sep	Jul	Kristi	Darron	Jan	Kristi
Compani	Million	Hou	Www	Apr	Ffffff	Jul	Oct	Growth	Growth	Growth	Total

FIGURE D.2: Topics generated on single linkage clusters, K = 4 - 6

Cluster4				Cluster5				Cluster6			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Match	Fals	Fals	Fals	Aep	Aep	Aep	Aep	Assum	Sum	Sum	Cashflow
Fals	Match	Day	Day	Hour	Sheet	Power	Hour	Sum	Cashflow	Cashflow	Total
Day	Jan	Monday	Match	Power	Src	Hour	Schedul	Cashflow	Year	Interest	Interest
Jan	Yes	Summari	Jan	Sheet	Hour	Schedul	Power	Incom	Assum	Basi	Incom
Transact	Summari	Data	Summari	Src	Power	Sheet	Total	Jan	Interest	Draw	Outsid

Cluster1					Cluster2					Cluster3				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Com	Enron	Eomonth	Max	Mapi	Mar	Jan	May	Ffffff	May	Round	Darron	Darron	Darron	Round
Enron	Ect	Market	Min	Jan	Jan	Mar	Mar	Feb	Ffffff	Darron	Round	Round	Round	Kristi
Will	Employe	Will	Price	Peopl	Jun	May	Feb	Oct	Feb	Jan	Kristi	Kristi	Jan	Darron
Pleas	Compani	Compani	Com	Gas	Ffffff	Jul	Jun	Mar	Mar	Kristi	Jan	Jan	Total	Jan
Http	Million	Power	Http	Ena	Sep	Jun	Ffffff	Aug	Apr	Ltd	Total	Estim	Kristi	Estim

Cluster4					Cluster5					Cluster6				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Jan	Jan	Summari	Fals	Data	Aep	Aep	Aep	Aep	Aep	Cashflow	Sum	Sum	Assum	Assum
Fals	Data	Match	Match	Transact	Sheet	Hour	Sheet	Shet	Src	Sum	Cashflow	Assum	Sum	Sum
Summari	Day	Jan	Day	Ytd	Src	Schedul	Src	Src	Power	Jan	Jan	Cashflow	Intern	Cashflow
Match	Match	Transact	Jan	Day	Schedul	Daili	Power	Hour	Schedul	Incom	Assum	Incom	Incom	Basi
Day	Yes	Yes	Yes	Fals	Power	Sourc	Daili	Schedul	Sheet	Assum	Intern	Max	Jan	Draw

Cluster7				
Topic1	Topic2	Topic3	Topic4	Topic5
Walton	Sum	Sum	Sum	Sum
Sum	Walton	Capit	Capit	Doyl
Turbin	Emc	Ect	Nepco	Project
Doyl	Doyl	Actual	Total	Text
Project	Week	Summari	Actual	Ena

FIGURE D.3: Topics generated on single linkage clusters, K = 6 - 7

Cluster1				Cluster2				Cluster3			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Enron	Enron	Jan	Will	Jan	Mar	Feb	Ffffff	Darron	Darron	Darron	Darron
Ect	Mapi	Gas	Com	Mar	Jan	Ffffff	Mar	Round	Round	Round	Round
Com	Max	Eomonth	http	Feb	Jun	Jan	Aug	Kristi	Kristi	Kristi	Kristi
Hou	Min	People	Busi	Ffffff	Sep	Jun	May	Jan	Jan	Jan	Jan
Privat	Employe	Market	New	Oct	Feb	Sep	Apr	Total	Total	Total	Total

Cluster4				Cluster5				Cluster6			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Jan	Jan	Fals	Day	Aep	Aep	Aep	Aep	Sum	Assum	Assum	Assum
Fals	Match	Match	Ltd	Sheet	Src	Schedul	Power	Assum	Sum	Sum	Sum
Match	Ltd	Jan	Summari	Src	Sheet	Hour	Sheet	Cashflow	Cashflow	Cashflow	Cashflow
Summari	Summari	Day	Data	Power	Schedul	Sheet	Demand	Jan	Incom	Jan	Incom
Weekday	Day	Yes	Match	Sourc	Hour	Power	Daili	Incom	Jan	Incom	Draw

Cluster7				Cluster8			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Sum	Sum	Sum	Sum	Sum	Sum	Sum	Sum
Capit	Capit	Ect	Capit	Doyl	Max	Max	Doyl
Actual	Ect	Capit	Draw	Walton	Project	Ena	Week
Nepco	Expens	Total	Actual	Project	Walton	Walton	System
Cost	Eecc	Expens	Nepco	Text	Ena	Doyl	Emc

Cluster1				Cluster2				Cluster3			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Jan	Will	Com	Enron	Mar	Ffffff	Ffffff	Aug	Darron	Kristi	Round	Darron
Max	Market	Mapi	Ect	Jan	Jan	Feb	Mar	Round	Jan	Kristi	Round
Min	Compani	Http	Employe	Feb	Feb	Mar	Feb	Kristi	Darron	Darron	Kristi
Gas	Busi	Enron	Eomonth	Jul	Sep	Jul	Jan	Jan	Round	Jan	Jan
Peopl	New	Www	Compani	Aug	Oct	Jan	Jul	Total	Growth	Annual	Estim

FIGURE D.4: Topics generated on single linkage clusters, K = 8 - 9

Cluster4				Cluster5				Cluster6			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Fals	Match	Match	Fals	Aep	Aep	Aep	Aep	Assum	Assum	Assum	Sum
Jan	Jan	Fals	Match	Sheet	Sheet	Sheet	Sheet	Sum	Cashflow	Basi	Assum
Weekday	Summari	Yes	Jan	Src	Src	Src	Src	Cashflow	Sum	Sum	Cashflow
Data	Day	Weekday	Day	Power	Daili	Daili	Hour	Jan	Total	Outsid	Incom
Yes	Data	Jan	Transact	Daili	Demand	Power	Power	Incom	Jan	Year	Intern

Cluster7				Cluster8				Cluster9			
Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4	Topic1	Topic2	Topic3	Topic4
Wish	Wish	List	List	Sum	Sum	Ect	Sum	Project	Walton	Project	Sum
Public	Total	Wish	Wish	Capit	Capit	Sum	Capit	Sum	Sum	Turbin	Doyl
Privat	List	Common	Privat	Cost	Total	Part	Ect	Ena	Turbin	Date	Walton
Special	Round	Special	Asset	Project	Turbin	Turbin	Actual	Walton	Power	Sum	Max
Convert	Common	Equiti	Common	Nepco	Part	Includ	Expens	Text	Doyl	Week	Text

Cluster1			Cluster2			Cluster3			Cluster4			Cluster5			Cluster6		
Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
Enron	Enron	Mapi	Ena	Ena	Enov	Mar	Jan	Ffffff	Darron	Darron	Round	Fals	Fals	Fals	Aep	Aep	Aep
Jan	Energi	Will	Gas	Peopl	Gas	May	Ffffff	Jul	Kristi	Round	Darron	Transact	Jan	Match	Sheet	Schedule	Sheet
Ect	Compani	Com	Peopl	Gas	Peopl	Feb	Feb	Mar	Jan	Jan	Kristi	Jan	Match	Day	Power	Src	Src
Com	Employe	Max	Enov	Enov	Ena	Ffffff	Jul	Feb	Round	Kristi	Jan	Yes	Day	Yes	Demand	Total	Hour
Peopl	Milllion	Min	Fetch	Counterparti	Fetch	Aug	Apr	Oct	Total	Estim	Growth	Weekday	Summari	Transact	Src	Sheet	Power

Cluster7			Cluster8			Cluster9			Cluster10		
Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
Assum	Assum	Sum	List	Wish	Wish	Sum	Sum	Sum	Doyl	Doyl	Sum
Incom	Jan	Assum	Privat	List	Privat	Construct	Expens	Capit	Project	Walton	Walton
Sum	Sum	Cashflow	Asset	Round	Round	Actual	Estim	Ect	Sum	Project	Project
Cashflow	Cashflow	Draw	Energi	Total	Common	Expens	Total	Actual	Walton	Sum	Doyl
Intern	Intern	Incom	warrant	common	sum	Capit	Summari	Nepco	Ena	Ena	Max

FIGURE D.5: Topics generated on single linkage clusters, K = 9 - 10

K-means clustering (TF)

Cluster1					Cluster2				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Oct	Darron	Ffffff	Day	Fals	Enron	Enron	Com	Sum	Market
Mar	Round	Feb	Match	Match	Employe	Ect	Eomonth	Jan	Will
Apr	Kristi	Jan	Monday	Jan	Peopl	Hou	Will	Mapi	Compani
Jun	Jan	Mar	Jan	Yes	Compani	Privat	http	Max	Power
May	Total	Sep	Yes	Transact	Million	Aep	Enron	Min	Price

Cluster1						Cluster2						Cluster3					
Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
Fals	Fals	Fals	Fals	Darron	Darron	Jan	Jan	Ffffff	Jan	Mar	Jan	Will	Enron	Enron	Com	Market	Sum
Data	Match	Match	Weekday	Match	Round	Mar	Jun	Feb	Sep	Jan	Mar	Enron	Employe	Mapi	Max	Power	Jan
Yes	Jan	Day	Day	Jan	Kristi	Jul	Aug	Jun	Apr	Sep	Sep	Busi	Eomonth	Ect	Min	Energi	Assum
Jan	Yes	Summari	Summari	Round	Jan	May	Mar	May	Ffffff	Ffffff	Oct	Pleas	Million	Hou	http	Price	Peopl
Weekday	Day	Yes	Yes	Total	Total	Feb	Jul	Oct	Mar	Feb	Apr	Time	Compani	Privat	Www	Gas	Gas

Cluster1					Cluster2					Cluster3				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Apr	Apr	Ffffff	Apr	Feb	Enron	Enron	Will	Com	Sum	Fals	Fals	Fals	Yes	Fals
Jan	Ffffff	Feb	Ffffff	Ffffff	Jan	Mapi	Min	Peopl	Assum	Match	Match	Jan	Match	Jan
Feb	Jan	Mar	Jan	Apr	Employe	Ect	Max	Http	Energi	Day	Jan	Yes	Jan	Match
Mar	Oct	Jan	May	Aug	Compani	Eomonth	Market	Www	Gas	Summari	Day	Day	Summari	Yes
Jun	Sep	Jun	Mar	Jul	Million	Hou	Compani	Gas	Total	Good	Summari	Transact	Transact	Transact

FIGURE D.6: Topics generated on k-means linkage clusters, K = 2 - 4

Cluster4				
Topic1	Topic2	Topic3	Topic4	Topic5
Kristi	Darron	Darron	Darron	Darron
Darron	Kristi	Round	Kristi	Kristi
Round	Jan	Kristi	Round	Round
Jan	Round	Jan	Jan	Jan
Total	Growth	Total	Total	Growth

Cluster1					Cluster2					Cluster3				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Assum	Assum	Sum	Sum	Sum	Jan	Com	Will	Mapi	Enron	Jan	Ffffff	Feb	Mar	Ffffff
Cashflow	Cashflow	Assum	Assum	Assum	Enron	Peopl	Market	Max	Ect	Jun	Jun	Jul	Aug	Jan
Incom	Jan	Cashflow	Incom	Cashflow	Employe	Sum	Compani	Min	Ou	Feb	Oct	Mar	Jul	Feb
Sum	Sum	Jan	Year	Max	Compani	http	Price	Eomonth	Energi	Apr	Sep	Jun	Jun	May
Intern	Draw	Draw	Cashflow	Jan	Million	Gas	New	Aep	Corp	Oct	Mar	Aug	Feb	Mar

Cluster4					Cluster5				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Darron	Kristi	Round	Darron	Darron	Jan	Fals	Jan	Fals	Jan
Round	Round	Kristi	Round	Round	Match	Yes	New	Match	Fals
Kristi	Darron	Darron	Kristi	Jan	Fals	Jan	Data	Day	Match
Jan	Jan	Jan	Jan	Kristi	Day	Weekday	Weekday	Yes	Summari
Total	Estim	Total	Growth	Total	Weekday	Data	Monday	Summari	Weekday

Cluster1					Cluster2					Cluster3				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Jan	Enron	Market	Privat	Enron	Assum	Cashflow	Assum	Assum	Sum	Will	Com	Market	Mapi	Enron
Sum	Ect	Will	Equiti	Max	Sum	Assum	Cashflow	Incom	Cashflow	Houston	http	Energi	Eomonth	Will
Peopl	Aep	Price	Enron	Min	Cashflow	Intern	Sum	Sum	Assum	New	Enron	Power	Messag	Ect
Gas	Hou	Compani	Common	Employe	Incom	Incom	Incom	Max	Jan	Busi	Www	Manag	Sent	Pleas
Ena	Corp	Year	Energi	Compani	Intern	Draw	Jan	Basi	Basi	Time	Recipi	Price	Repres	Thank

FIGURE D.7: Topics generated on k-means linkage clusters, K = 4 - 6

Cluster4					Cluster5					Cluster6				
Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5	Topic1	Topic2	Topic3	Topic4	Topic5
Jun	Feb	Sep	Oct	Ffffff	Darron	Darron	Darron	Round	Kristi	Fals	Fals	Yes	Match	Yes
Feb	Aug	Mar	Feb	Feb	Kristi	Round	Kristi	Darron	Round	Match	Jan	Match	Yes	Fals
Jan	Oct	Nov	Mar	Jan	Jan	Jan	Round	Kristi	Jan	Jan	Day	Day	Fals	Jan
Oct	Jan	Jan	Jun	Mar	Round	Kristi	Jan	Jan	Darron	Day	Transact	Jan	Transact	Day
Mar	Apr	Feb	May	Jul	Growth	Total	Growth	Growth	Growth	Summari	Yes	Fals	Day	Match

Cluster1		Cluster2		Cluster3		Cluster4		Cluster5		Cluster6		Cluster7	
Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2
Jan	Darron	Jan	Sum	Mapi	Enron	Cashflow	Assum	Enron	Enron	Yes	Fals	Ffffff	Ffffff
Round	Round	Peopl	Doyl	Max	Will	Sum	Sum	Employe	Ect	Data	Match	Feb	Jan
Kristi	Kristi	Gas	Walton	Min	Com	Jan	Incom	Million	Market	Transact	Jan	Mar	Feb
Darron	Jan	Ena	Project	Eomonth	Pleas	Draw	Cashflow	Compani	Will	Match	Day	Jan	Sep
Total	Total	Intra	Capit	Com	New	Max	Basi	California	Energi	Jan	Summari	May	Aug

Cluster1		Cluster2		Cluster3		Cluster4		Cluster5		Cluster6		Cluster7		Cluster8	
Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2
Peopl	Jan	Sum	Assum	Sum	Sum	Fals	Fals	Com	Enron	Kristi	Darron	Ffffff	Aug	Max	Privat
Gas	Page	Assum	Sum	Doyl	Capit	Match	Jan	Will	Mapi	Jan	Round	Jan	Mar	Min	Equity
Intra	Total	Equiti	Cashflow	Walton	Ect	Day	Match	Enron	Employe	Darron	Kristi	Feb	Feb	Aep	Common
Border	Gdp	Cashflow	Incom	Project	Actual	Jan	Data	Ect	Eomonth	Round	Jan	Mar	Sep	Tot	Enron
Harper	Christoph	Incom	Jan	Text	Nepco	Summari	Weekday	Http	Compani	Growth	Total	Jul	Apr	Price	Total

Cluster1		Cluster2		Cluster3		Cluster4		Cluster5		Cluster6		Cluster7		Cluster8		Cluster9	
Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2
Aep	Eomonth	Com	Mapi	Sum	Jan	Ena	Privat	Sum	Assum	Fals	Match	Darron	Round	Feb	Ffffff	Will	Enron
Sheet	Jan	Max	Messag	Doyl	Peopl	Gas	Equiti	Cashflow	Cashflow	Jan	Fals	Kristi	Darron	Jan	Mar	Market	Employe
Src	Cury	Min	Attach	Walton	Gas	Peopl	Common	Assum	Sum	Transact	Day	Jan	Jan	Aug	May	Com	Ect
Hour	Match	Enron	Ect	Project	Intra	List	Enron	Jan	Incom	Yes	Jan	Round	Total	Ffffff	Jul	Price	Compani
Power	Weekday	Will	Sent	Capit	Border	Wish	Total	Basi	Intern	Summari	Data	Total	Match	Mar	Jun	Compani	Million

FIGURE D.8: Topics generated on k-means linkage clusters, K = 6 - 9

Cluster1		Cluster2		Cluster3		Cluster4		Cluster5	
Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2
Fals	Fals	Assum	Assum	Round	Darron	Jan	Eomonth	Jan	Ffffff
Jan	Match	Sum	Cashflow	Jan	Round	Match	Round	May	Feb
Match	Day	Incom	Sum	Darron	Kristi	Cury	Sum	Oct	Mar
Yes	Weekday	Jan	Draw	Kristi	Jan	Aquila	Cury	Jul	Sep
Transact	Jan	Cashflow	Max	Growth	Total	Socal	Weekday	Mar	Aug

Cluster6		Cluster7		Cluster8		Cluster9		Cluster10	
Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2	Topic1	Topic2
Com	Mapi	Ena	Peopl	Sum	Privat	Aep	Aep	Will	Enron
Will	Max	Gas	Gas	Doyl	Equity	Hour	Sheet	Market	Employe
Enron	Min	Peopl	Jan	Walton	Common	Sheet	Src	Compani	Ect
http	Enron	Enov	Intra	Project	Enron	Demand	Hour	Price	Compani
Pleas	Messag	Counterparti	Border	Capit	Total	Src	Power	Power	Million

FIGURE D.9: Topics generated on k-means linkage clusters, K = 10

This page intentionally left blank.

Bibliography

Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. 2001 ACM SIGMOD international conference on Management of data.

Aggarwal, C. C. and Zhai, C. (2012). A survey of text clustering algorithms. In *Mining Text Data*, chapter 4. Springer.

Baker, L. D. and McCalum, A. K. (1998). SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval.

Barnett, T., Godjevac, S., Renders, J., Privault, C., Schneider, J., and Wickstrom, R. (2009). Machine learning classification for document review. Technical report.

Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping Multidimensional Data: Recent Advances in Clustering*. Springer.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4).

Blei, D. M. and Lafferty, J. D. (2009). Topic models. In *Text Mining: Classification, Clustering, and Applications*. Chapman and Hall/CRC.

Bramer, M. (2013). Introduction to data mining. In *Principles of Data Mining*, chapter 1. Springer.

Chakraborty, G., Pagolu, M., and Garla, S. (2013). *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS*. SAS Institute Inc.

Chakraborty, G. and Pagolu, M. K. (2014). Analysis of unstructured data: Applications of text analytics and sentiment mining. Technical report, SAS Institute Inc.

Cormack, G. V. and Grossman, M. R. (2014). Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *SIGIR '14 - Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval*.

Dash, M. and Liu, H. (2000). Feature selection for clustering. Springer-Verlag.

Diesner, J., Frantz, T. L., and Carley, K. M. (2005). Communication networks from the enron email corpus - "it's always about the people. enron is no different". In *Computational and Mathematical Organization Theory*. Springer.

EDRM (2015). *EDRM Stages*. EDRM.

Eves, A. O., Thomas1, J., McNaught, J., Miwa, M., and Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. Technical report.

Fodor, I. K. (2002). A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory.

Fuhr, N. and Buckley, C. (1991). A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems (TOIS) - Special issue on research and development in information retrieval*.

Government, U. (2015). Federal rules of civil procedure.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*.

Grimes, S. (2014). Text analytics 2014: User perspectives on solutions and providers. Technical report, Alta Plana Corporation.

Grimes, S. (2015). Text analytics 2015 technology and market overview.

Grossman, M. R. and Cormack, G. V. (2015). The grossman-cormack glossary of technology assisted review.

Grün, B. and Hornik, K. (2012). Topicmodels: An R package for fitting topic models.

Gupta, V. and Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, 1(1).

Halper, F., Kaufman, M., and Kirsh, D. (2013). *Text Analytics: The Hurwitz Victory Index Report*. Hurwitz and Associates.

Healy, P. M. and Palepu, K. G. (2003). The fall of enron. *Journal of Economic Perspectives*, 17(2).

Indurkhya, N. and Damerau, F. J., editors (2010). *Handbook of Natural Language Processing - Second Edition*. Chapman and Hall/CRC.

Jockers, M. L. (2014). *Text Analysis with R for Students of Literature*. Springer.

kCura (2015). *Accelerate Your e-Discovery Efforts - 8 Ways to Speed Up Review with Text Analytics*. kCura.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*.

Li, Y. (2010). The case analysis of the scandal of enron. *International Journal of Business and Management*, 5(10).

Liu, H., Stine, R., and Auslender, L. (2005). Proceedings of the workshop on feature selection for data mining: Interfacing machine learning and statistics. SIAM International Conference on Data Mining 2005 (SDM05).

Liu, T., Liu, S., Chen, Z., and Ma, W. (2003). An evaluation on feature selection for text clustering. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. The AAAI Press.

Milios, E. E., Shafiei, M. M., Wang, S., Zhang, R., Tang, B., and Tougas, J. (2006). A systematic study on document representation and dimensionality reduction for text clustering. Technical report, Faculty of Computer Science, Dalhousie University.

Müllner, D. (2011). *Modern hierarchical, agglomerative clustering algorithms*. PhD thesis, Department of Mathematics, Stanford University.

Pace, N. M. and Zakaras, L. (2012). *Where the Money Goes - Understanding Litigant Expenditures for Producing Electronic Discovery*. RAND Corporation.

Reissner, H. J. and Hochman, I. K. (2012). Every good document review starts with human expertise. *New York Law Journal*.

Roitblat, H. L., Kershaw, A., and Oot, P. (2010). Document categorization in legal electronic discovery: Computer classification vs. manual review. *Journal of the American Society for Information Science and Technology*, 61(1).

Rokach, L. and Maimon, O. (2005). Clustering methods. In *The Data Mining and Knowledge Discovery Handbook*, chapter 15. Springer.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*.

Sanchez, G. (2013). *Handling and Processing Strings in R*. Trowchez Editions.

Seabury, C. (2015). *Enron: The Fall Of A Wall Street Darling*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1).

Sethi, A. and Upadrasta, B. (2012). Introduction to probabilistic topic modeling. Technical report, Innovation and Development Group, Mu Sigma Business Solutions.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. Technical Report 00-034, Department of Computer Science and Egineering, University of Minnesota.

Stewart, B. M. (2010). *Practical Skills for Document Clustering in R*. University of Washington.

Steyvers, M. and Griffiths, T. (2006). Probabilistic topic models. In *Latent Semantic Analysis: A Road to Meaning*.

Team, R. C. (2015). *R Data Import/Export*. R Core Team.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth-Heinemann.

Williams, G. (2014). *Hands-On Data Science with R - Text Mining*.

Xie, P. and Xing, E. P. (2013). Integrating document clustering and topic modeling. Technical report.

Yang, Y. (1995). Noise reduction in a statistical approach to text categorization. Technical report.

Yang, Y. and Pedersen, J. O. (1997). ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning.