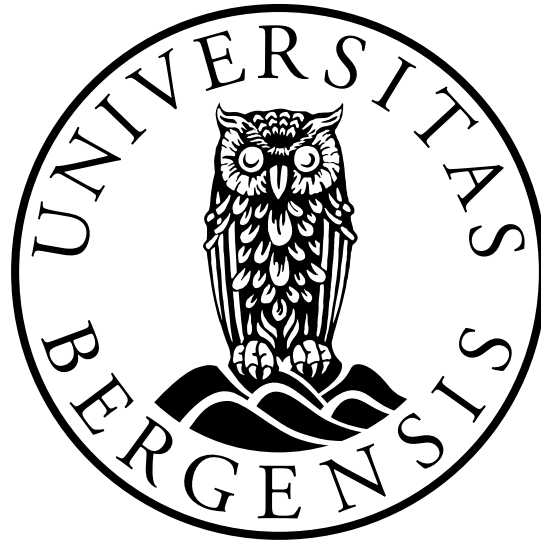


UNIVERSITY OF BERGEN



Department of Information Science and Media Studies

MASTERS THESIS

**Natural Language Processing of fiscal
yearly reports for use in risk assessment**

Author: Magnus Tenmann

Supervisor: Csaba Veres

December 9, 2019

Abstract

The stability and accuracy of products in the financial sector is maintained by various measures within each organisation in the field they operate in. After a meeting with DNB Livsforsikring, which offers insurance products, it was identified that the current processes of risk assessment applied in this context could benefit from the language processing technologies. Consequently, this could lead to profit optimization for the company and decreased costs of human labour, and potentially in reduction of error, depending on the accuracy of the implemented technology.

This research is conducted in cooperation with DNB with an aim of developing an application, which utilises the functionalities of existing libraries for Natural Language Processing (NLP) to perform the task of text extraction and topic modelling of the fiscal reports, provided by DNB.

Design science research has been used to create an artifact that use text extraction for analytics of fiscal yearly reports. Other textual visualisations are implemented, such as word clouds and Latent Dirichlet Allocation (LDA). The implementation utilizes a variety of technologies, including the NLTK library, as well as other common data science libraries, such as sci-kit learn. The main functionalities of the resulting artifact are text extraction and visualisation of topic modelling, TF-IDF, wordcloud generation and frequency distribution of which were fully functional as separate components. As part of the development process, a number of subject-specific methods have been used and implemented, such as agile development and minimum viable product. The evaluation of the prototype has shown perceived usefulness, relevance to the intended application, understandability, practicality and the ability to produce some relevant results.

Acknowledgment

Thanks to family and friends, and my lovely girlfriend for enduring my anti-social behaviour and at times shoddy mood during the writing of this thesis.

Thanks to DNB Livsforskring for contributing with feedback and data that made the basis of this thesis.

Thanks to my fellow students at the Department of Informaton Science and Media Studies for providing helpful conversations and motivation.

A big thanks to Csaba Veres for his guidance as my supervisor.

Bergen, December 9, 2019

Magnus Tenmann

Contents

Abstract	ii
Acknowledgment	v
1 Introduction	1
1.1 Goals	4
1.2 Research Questions	4
1.3 Overview of Thesis	5
2 Theory	7
2.1 Natural Language Processing (NLP)	7
2.2 Natural Language toolkit (NLTK)	9
2.3 Gensim	10
2.4 Fiscal Yearly Reports Design Standard	11
2.5 LDAvis	12
2.6 Word Cloud	13
2.7 Topic Modelling	14
2.8 Sentiment Analysis	15
2.9 Text Extraction	16
3 Research Methodology	19

3.1	Design Science Research	19
3.2	Design and Creation	22
3.3	Data Collection	24
3.4	Artifact	25
3.5	Design Prototyping	26
3.6	Semi-structured Interviews	27
3.7	Unstructured Interviews	27
3.8	Expert Interviews	28
4	Tools	29
4.1	Programming Language and Libraries	29
4.2	Applications	32
4.3	Summary	34
5	Development	35
5.1	First Iteration	36
5.1.1	Initial Requirements	36
	Semi-structured Interview	36
	Functional Requirements	37
	Non-Functional Requirements	37
5.1.2	Obtaining Data: Getting the Fiscal Yearly Reports	38
5.1.3	Summary	41
5.2	Second Iteration	42
5.2.1	Input from Bank	42
5.2.2	Text Extraction	45
5.2.3	Wireframes	48
	Expert Interview	51

- 5.2.4 Summary 51
- 5.3 Third Iteration 51
 - 5.3.1 Input from bank 52
 - 5.3.2 Analysis and Visualisation 52
 - 5.3.3 Wireframes 60
 - 5.3.4 Summary 60
- 5.4 Fourth Iteration 61
 - 5.4.1 Input from Bank 61
 - 5.4.2 Interactive Wireframes 61
 - 5.4.3 Summary 63
- 6 Prototype Evaluation 64**
 - 6.1 Observation 64
 - 6.2 Semi-structured Interviews 64
 - 6.3 IT Expert Interview 65
- 7 Discussion 67**
 - 7.1 Answering the Research Questions 67
 - 7.2 Semi-structured Interviews 68
 - 7.3 Prototype Development 69
 - 7.4 Evaluation 70
 - 7.5 Design Science Research 70
 - 7.6 Design Prototyping 71
 - 7.7 Limitations 72
- 8 Conclusion 74**
 - 8.1 Conclusion 74

8.2 Future Work	75
References	78
A Fiscal reports	89
A.1 Yearly report	89
A.2 Independent accountant report	89
B Wireframes	92
B.1 Low Fidelity	92
B.2 Medium Fidelity	94
C Semi-structured interview	102
C.1 Initial requirements	102
C.2 Prototype testing	102
C.3 Evaluation questions	103
D Visualization	104
D.1 Frequency distribution	104
D.2 TF-IDF	106
D.3 LDA	108
E Source code	109

List of Figures

2.1	LDA topic model visualization, using LDAvis [1], p. 49	12
3.1	Design process model (generic) [2], (p.3)	22
5.1	Kanban board on Trello	36
5.2	Brønnøysundregistrene (Official Registry) User Interface	39
5.3	Initial (manual, left) versus Subsequent (semi-automated, right) process	40
5.4	Excel Sheet of Companies and their Risk Factor	41
5.5	Sample Annotated Report	43
5.6	Positive and Negative Words/Phrases	44
5.7	Sample Extracted Text	48
5.8	Main screen, hand-drawn	49
5.9	Summary screen, hand-drawn	49
5.10	Main screen, low-fi wireframe	50
5.11	Summary screen, low-fi wireframe	50
5.12	LDA visualisation	56
5.13	Wordcloud Low-Risk	57
5.14	Wordcloud Medium-Risk	57
5.15	Wordcloud High-Risk	57
5.16	Term Frequency Distribution	58

5.17 TF-IDF 2-D vector space mapping of reports	59
5.18 Overview of medium fidelity wireframes	60
5.19 Landingpage wireframe	62
5.20 Dashboard wireframe	63
B.1 Main window	92
B.2 Summary of analyzes	93
B.3 Lo-fi Main Window made in Wiresketcher	93
B.4 Lo-fi Summary of Analyzes made in Wiresketcher	94
B.5 Dashboard Iteration 1	95
B.6 Summary Iteration 1	95
B.7 Frequency Distribution Iteration 1	96
B.8 Landing screen Iteration 2	97
B.9 Dashboard Iteration 2	98
B.10 Frequency Distribution Iteration 2	99
B.11 Risk Assessment Iteration 2	100
D.1 Frequency distribution, inital reports	105
D.2 Frequency distribution, all reports	106
D.3 TF-IDF, inital reports	107
D.4 TF-IDF, all reports	107
D.5 LDA, all reports	108

List of Tables

2.1	Language processing tasks and corresponding NLTK modules with examples of functionality [3], p.427.	10
3.1	Benefits of Agile Development [4]	24
5.1	Iteration cycles	35

List of source codes

5.1	Stopwords	45
5.2	Clean Function	45
5.3	Dictionary creation and risk factor sorting	46
5.4	Processing PDF	46
5.5	Multiprocessing using batches	47
5.6	CSV Creation	52
5.7	Loading of text into classification categories	53
5.8	Implementation of Gensim LDA	54

List of abbreviations

DNB Den Norske Bank

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

ML Machine Learning

MVP Minimum Viable Product

NLTK Natural Language Toolkit

NLP Natural Language Processing

Chapter 1

Introduction

The financial sector is broad as it encapsulates a variety of industries, such as the banking, insurance, accounting, stock markets, funds for investment etc. The financial system plays an important role in the development of countries worldwide, which make its stability and performance a priority for analysts [5]. The companies that operate within the sector have various means of reporting their financial status and strategic performance and plans; financial stability is measured through a variety of numeric indexes, such as quantile, leverage ratio or liquidity, among others, all of which have certain shortcomings. For example, it is considered that an approach of manual inspection of financial performance can be cumbersome for stakeholders due to information being located in parts of the report which are not traditionally associated with financial figures. It can also be one-sided, i.e only evaluating one aspect of financial performance [5]. These challenges have encouraged researchers to identify new ways of quantifying and visualising financial stability using fiscal yearly report data, which has led to the integration and utilisation of machine learning technologies, specifically natural language processing (NLP), for analysis of textual data in such documents.

Textual data is more available than numeric data and it can be argued that high-level human language contains a large amount of complexity and nuance. Machine interpretation and analysis of textual data is a field that is continuously moving forward, yet it still has a considerable number of shortcomings when compared to human interpretation. For example, a person would successfully interpret the following statement from a fiscal report: "the company had a year with many challenges, but in the end we powered through" as a sign of potential financial hardship, while a machine interpreter would struggle to extract meaning

from this sentence due to the inherent semantic ambiguity of spoken language [6]. Another challenge of inductive machine learning systems are that they often require huge collections of pre-labelled data so that they can be trained and tested on whether they correctly identify a given relationship [7]. Also when extracting text from fiscal reports, the structure of the documents could pose a challenge, since they have no formal requirement as to what they need to contain even though they are mandatory in all countries.

Due to constraints of machine learning software in performing text analytics, companies in the financial sector currently have considerable human resource-related spending for staff to analyze fiscal reports manually, with reports often being up to one hundred pages long as part of their auditing procedures for companies they wish to work with, insure or invest in. It can be argued that this presents an opportunity to create value by solving an existing business problem through the implementation of natural language processing of such documents. Thus, the following research is conducted in cooperation with DNB Liv, and is aimed at scoping the requirements of developing an application which utilises the functionalities of existing libraries for NLP to perform the task of topic modelling of fiscal reports, provided by the company.

The case study company involved is a bank which sells life and pension insurance to companies, municipalities, organizations and private citizens. Prior to selling insurance, a risk assessment is performed if the customer is not a private citizen. DNB have automated systems for handling individual private citizens and this customer category requires less oversight. As for companies the process is currently done semi-automatically: there is an automated process that processes over data they have available and later an actuary conducts a manual assessment. The motivation for the creation of the application is that even though the bank has an abundance of data available to them, the management has a strategic objective of using a variety of data points, in which regard textual data can be used as a means to improve the assessments' efficiency and accuracy. The current processes by which many companies in the financial sector retrieve information is through popular search engines, where relevant information is gathered from the web. That information could indicate future financial instability. For example, they find a news article that says that the company have to put employees on paid leave or fire them due to economic hardships in the company. Other sources of information such as social media websites (e.g Facebook, Twitter, Instagram) are also used in this regard. Up to date, yearly fiscal (otherwise known as financial or annual) reports are

most commonly used as a means of of corporate credit and financial health assessment [8]. Such reports can be submitted to Brønnøysundregistrene, as well as being published on public websites or on the company's own website [9], which enables ease of access. The bank sets the price of insurance based on components (1) the risk factor, with (2) an added main tariff.

$$\text{Price} = \text{Main tariff} + \text{Risk factor}$$

This research and the end artifact will attempt to assist with the evaluation of the risk factor, without taking into consideration the main tariffs that the bank might apply in different stages of their pricing process. Specifically, the reports will be scanned for information that hints at financial instability in the text, such as human resource or staffing issues, discussions of lawsuits or other public relations matters that could indicate that the company might face financial difficulty.

To summarize, the driver of this research from an organizational standpoint is the need to optimise the internal processes through implementing process automation in the aspect of financial report text extraction and analysis. If achieved, this could lead to a reduction of human resource cost in this area, which is considered a benefit for the organization as it will reduce cost and potentially lead to more efficient and streamlined processes [10]. Furthermore, the implementation of an automated solution could reduce the risk of human error in the interpretation of the data, which could result in a better financial performance for the bank, since they could achieve a risk reduction of their client portfolio. As well as the potential financial benefits for the organization, there is potential for freeing up time for the banks employees as the time they previously spent reading through these documents could be utilized elsewhere [11].

1.1 Goals

The goals of the research are to:

1. identify how DNB Liv do risk assessment
2. implement NLP techniques, such as data extraction, topic modelling and visualization to analyze the textual data in financial reports
3. present a solution which can be used to replace existing practices
4. develop the application in cooperation with the organization
5. implement feedback and present potential improvements of the developed system prototype MVP

The aim of the system is to present employees with relevant information that is extracted from text in the fiscal reports that concerns the way they asses risk. This information can be used by employees as a means to more efficiently and accurately analyze the potential of a new client for the banks insurance products.

1.2 Research Questions

The research questions that will be used to guide the research process are as follows:

- **RQ1:** *How can we use fiscal yearly reports to assess the risk of a company by analysing its natural language?*
- **RQ2:** *Does the fact that fiscal yearly reports do not have a standard structure affect data extraction and natural language analysis?*
- **RQ3:** *How can we visually present the result of a report analysis to the end user?*

1.3 Overview of Thesis

The following is an outline of how the thesis is structured:

Chapter 2: Theory will present some research in relevant areas in the field of NLP and relevant literature.

Chapter 3: Research Methodology presents the research methodology, where the development method of design science is discussed in-depth.

Chapter 4: Tools has an identification of the appropriate technologies, as well as a summary of how they are commonly applied in practice.

Chapter 5: Prototype Development shows the development process and goes into detail over the four iterations.

Chapter 6: Evaluation presents the findings of the testing done on the fourth iteration of the prototype.

Chapter 7: Discussion will have a discussion that concerns the research in design science, the system prototyping process, following which the discussion will refer back to the research questions, demonstrating the answers that stems from the research and the development.

Chapter 8: Conclusion and Future Work is a conclusive chapter, where future research work will be identified and presented and the insights of this research will be summarised.

Chapter 2

Theory

The Theory chapter provides an overview of the literature available on various topics that concern how the problem can be addressed in the context of the current research. Section 2.1 provides an overview of the field of NLP, discussing some challenges of the field and offering insight into some of the technologies that are currently trending in academic research. Section 2.2 discusses NLTK - a commonly-used toolkit for solving NLP issues in computer science and development. Section 2.3 explains the use of gensim and shows some uses of it in similar fields of research. Section 2.4 discusses the structure of fiscal reports, demonstrating the complexities in implementing a text analytics solution to such documents. Sections 2.5 and 2.6 discuss the industry application of the technologies for visualization of topics and words - LDAvis and word-clouds, respectively. Section 2.8 shows an overview of the field of topic modelling, illustrating relevant research and demonstrating the benefits and limitations of one of the approaches, which will be used as part of the development, whereas in Section 2.9 discusses sentiment analysis and classification. Section 2.10 features research in the field of text and knowledge extraction for textual and image documents.

2.1 Natural Language Processing (NLP)

NLP is a field of research that studies the ability to decode data from natural language using computational means. This field also examines how this decoded data can be incorporated into machine learning and statistical programming software. If this is achieved, computer programs will be able to perform data analysis using machine learning algorithms on tex-

tual data, which there is a lot of in a public-source format [12]. Programs can also be coded to extract meaning from data, otherwise referred to as text semantics [12]. The holistic aim of the NLP field is to bridge the gaps in communication between computer programs and humans, with the former being improved on a continuous basis to decode natural language and speech data into meaningful semantic insights through processing, analysis and synthesis [12].

One of the challenges that NLP faces is that the information that is useful and relevant to the specific business problem is often located in a large pool of data, which is often clustered chaotically [13]. Another challenge is that the data might not provide any insight or might be considered useless in relation to the business problem when approached for consideration by a human agent. A computer program at the current stage of research, however, has no understanding of the relationship between data and business strategy. Therefore, there is no understanding of the purpose or meaning of text. This creates an important stream of research and application development within NLP which is concerned with the extraction of relevant information for the machine learning algorithm or program to analyze and extract meaning from, commonly referred to as data (and more specifically) text mining. Text mining is defined by Berry and Linoff as 'the exploration and analysis of substantial quantities of data in order to discover meaningful patterns and rules' [14].

These problems have sparked academic interest, with many researchers training NLP algorithms in different areas, with the aim of many of those being to understand the constructs of language and expression through computer science and statistical models [15]. Such insights can then be used to reduce the need for human evaluation of text and can improve the efficiency of text analytics in many organizations. Research studies have also shown that the application of software for analysis of data in the public domain, such as social media data, can improve an organizations competitive advantage [16][17]. As the field of NLP develops, organizations have been reported to implement its insights into operational processes. For example, Naratani and Cuang's research demonstrates that companies might adapt their corporate communications and content to reflect analyzed public sentiment that they monitor through social media text analytics [18].

2.2 Natural Language toolkit (NLTK)

The NLTK is a Python library that has various packages, which support text pre-processing, including a stop-words pre-defined library [19]. The suite includes a range of open-source program modules, tutorials and problem sets, which offers researchers ready to use computational linguistics algorithms [20]. It was developed in conjunction with a Computational Linguistics course at the University of Pennsylvania back in 2011 [3]. NLTK covers both symbolic and statistical NLP and is already interfaced with a corpora of words [20]. Research shows that this toolbox can be used for both educational and scientific purposes. For example, it can be used not only as a training complex, but also as an analytical tool or a prototyping basis in the development processes of applied text analytics systems [3]. In recent times, some common applications of this toolkit include linguistics, machine learning and artificial intelligence software applications [3]. Some advantages of this software are that it is entirely self-contained, it provides raw and annotated versions of real data in the form of data corpora, grammar collections and some trained models, as well as functions that can be utilised as building blocks for NLP tasks, making it suitable for prototyping [3]. The corpora within the toolkit are sectioned, which is useful for programmers to exploit them. NLTK is also integrated with WordNet. WordNet is a database of semantic relationship for nouns, verbs, adverbs and adjectives in the English language [3]. A similar version exist for Norwegian words, which encompass around 50.000 synonym sets [21]. Finally, NLTK provides frequency, plotting and visualization tools and its instructions and wide-spread use in the academic field make it suitable for programmers with little experience and for students. Below is a summary of the NLTK modules and their functionality, created and featured in Lobur et.al work [3].

Table 2.1: Language processing tasks and corresponding NLTK modules with examples of functionality [3], p.427.

Language processing task	NLTK modules	Functionality
Accessing corpora	<code>nltk.corpus</code>	standardised interfaces to corpora and lexicons
String processing	<code>nltk.tokenize</code> , <code>nltk.stem</code>	tokenisers, sentence tokenisers, stemmers
Collocation discovery	<code>nltk.collocations</code>	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	<code>nltk.tag</code>	n-gram, backoff, Brill, HMM, TnT
Classification	<code>nltk.classify</code> , <code>nltk.cluster</code>	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	<code>nltk.chunk</code>	regular expression, n-gram, named-entity
Parsing	<code>nltk.parse</code>	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	<code>nltk.sem</code> , <code>nltk.inference</code>	lambda calculus, first-order logic, model checking
Evaluation metrics	<code>nltk.metrics</code>	precision, recall, agreement coefficients
Probability and estimation	<code>nltk.probability</code>	frequency distributions, smoothed probability distributions
Applications	<code>nltk.app</code> , <code>nltk.chat</code>	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	<code>nltk.toolbox</code>	manipulate data in SIL Toolbox format

Some disadvantages of using NLTK are also identified. For example, it uses strings as input and returns strings or lists of strings as an output, which is simple to deal with when working with a small data set, but becomes more complex when processing big data. There are other alternatives to this library, which take a comparatively more object-oriented approach, such as spaCy, which is considered more suited to modern Python programming [22]. Another disadvantage is that not all tools included in NLTK are compatible with languages of non/-alphabetic transcription [23]. Finally, when incorporating NLTK in software applications there can be integration and speed challenges, as all tools used (e.g. for text classification, tokenization, stemming, etc.) need to be launched separately [23].

2.3 Gensim

Somewhat similar to NLTK, gensim is a free open-source library that is used for natural language processing and topic modelling, and is made for Python [24]. This was a product of the PhD dissertation of Radim Rehurek in 2011, and Rehurek says that gensim can be used as a building block for more concrete algorithms for use in scalable unsupervised learning

[25]. Gensim is often used in work that involve topic modelling of large corpora of text and it contains a variety of useful tools that include sentiment analysis, Latent Dirichlet Allocation and tf-idf and tools to turn documents into words. Gensim has been used to in research that explore data and documents from social media, and after language processing used LDA to do a topic modelling of the data, similar to the goal of this thesis [26].

2.4 Fiscal Yearly Reports Design Standard

Although the use of fiscal yearly reports is mandatory in some countries, there are no clear indication of the information that should be included, and not many academic research papers that focus on the topic. There are researchers that have suggested the implementation of a world-wide policy for financial reporting [27]. An interesting study, written by Etteredge et al., has examined financial and annual reports of companies which have been published online, demonstrating that although companies publish reports online, these publications are specifically tailored for the web (i.e they are different to what is submitted to an official regulator). In such reports, companies commonly omit important financial information which can hurt their relationships with investors [28]. This allows companies to use such reports as a communication and investment attraction tool, even if the company is not performing well financially. Most commonly though, fiscal reports will include some form of accounting information or data, data for financial assets and performance, a letter to shareholders and investors from a board member and a balance sheet, where the company's assets can be researched from a holistic standpoint [27].

There is a conceptual framework for financial reporting proposed by the Financial Accounting Standards Board, in which it is proposed that the information presented in fiscal reports is moderated so that it is informative for the decision-making process of investors and offers factual information through non-misleading language [29]. It also says that in some cases: 'A decision not to disclose certain information or recognize an economic phenomenon may be made, for example, because the amounts involved are too small to make a difference to an investor or other decision maker' [28], yet this decision should be made without bias in the reporting process and with concern for the investor. What this demonstrates is that fiscal reports, although filled with quantitative and numeric data, also offer a great deal of ambiguity and challenge for extracting information. One reason for this is that the information

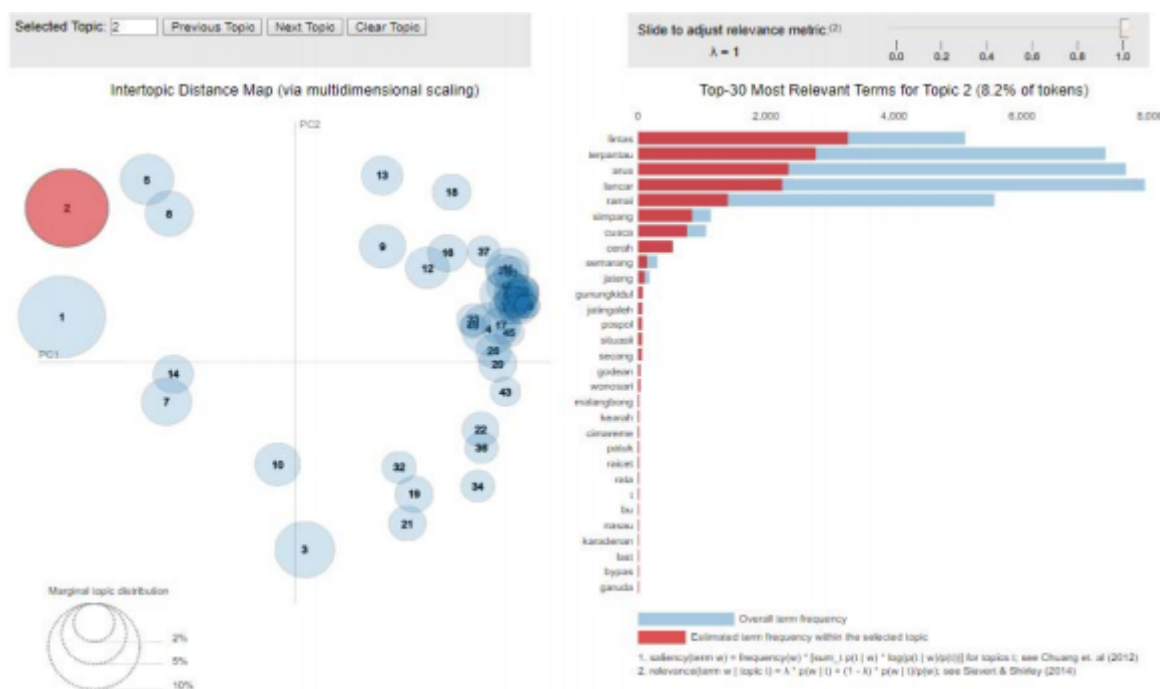


Figure 2.1: LDA topic model visualization, using LDAvis [1], p. 49

that is useful for an investor or risk assessor who is seeking to ensure that their future potential client has financial stability can be difficult to locate in a report that contains ambiguous wording. A further reason is the lack of apparent, commonly agreed upon structure, there is only an agreement on a handful of elements that should be present throughout the reports.

2.5 LDAvis

LDAvis is a Python library that allows visualization of topic models. It is created using the Latent Dirichlet Allocation algorithm (discussed in depth in Section 3.6, below) in a dynamic manner, as soon as results are processed by the machine learning algorithm [1]. It is accessible through a web browser, which has enabled it to become a commonly-used option for topic visualization in the field of NLP. The tool provides two visualization panels, as demonstrated by the figure 2.1, originally published as part of Hidayatullah et al.'s paper[1]. On the left is shown a holistic topic view through a map that reflects the interconnectedness of topics within the text. The right side shows term bar charts.

Although not being the only topic modelling visualization tool that has been developed in recent years, LDAvis is open-source and unique in that it is the only tool that measures the

relevance of the topic [30]. Topic relevance is used to rank terms within the topics, which helps the task of topic interpretation. The key aspect of the system is that it allows the user to select a topic and reveal the most relevant terms associated with it, and it offers users the ability to select a term (by hovering over it) to reveal its conditional distribution over topics. This allows the user to verify whether the multidimensional scaling of topics has clustered similar topics in the two-dimensional space. Finally, LDAvis also allows users to alter the topical distance measurement as well as the multidimensional scaling algorithm to produce the global topic view. Thus, it can be argued that the system offers potential for customization depending on its use and application in practice.

2.6 Word Cloud

Word clouds are another method for topic segment visualization, which provides a bird's eye view of what the major terms in the text are, calculated on the basis of term frequency. This is commonly used in combination with a topic modelling algorithm to showcase the most frequent words in a particular segment, in combination with the topic model that has been extracted for the same segment, which is a useful way to evaluate the performance of the modelling algorithm [31]. There is a variety of research, published on the topic of word clouds [31][32][33]. The process of generating word clouds and the software's command steps are featured in Breedvelt-Schouten's paper, which allows programmers to re-create the software within their systems, should they want to, driving the popularity of the use of this tool [30]. For example, Heimerl et al. have demonstrated this software tool's potential in creating powerful visualizations, but also implement the underlying reason of how word clouds are created in a software application's capability in the aspects of advanced natural language processing, sophisticated interaction possibilities, giving users a high level of control to provide support for different kinds of text analysis tasks [29]. In contrast, there is research that also argues against the use of word clouds as it destroys the function of traditional narrative, which can be harmful for scenarios where key information is located in aspects of the text, which might not be as frequently discussed [34]. The last research is considered pivotal for the use of word clouds as part of the current project, as although they might be useful for presenting some aspects of the information, it can be argued that they should not be used as a single tool for representing information from the fiscal reports, as this might hinder the

effectiveness of the application in uncovering text that speaks unfavourable for the organization, as it is assumed that such instances would be less frequently mentioned throughout the text, and thus will not appear on a word cloud visualization. Hence why topic modelling and sentiment analysis should be implemented, the academic literature for which is discussed in the next to section of this literature review.

2.7 Topic Modelling

Topic modelling is a technique used for text processing and text analytics which aims to overcome the overload of information in a given document and extract the patterns of the text data, i.e the topics [35]. As a results, the experience of reading a document is improved, as readers are able to navigate a given text or a collection of text quicker and seek out the information only for the topics that concerns them [36]. Traditionally, this type of analysis is categorized as an unsupervised technique, as it does not require previous labelling of data as part of training or validation of the model's performance. Instead, the program presents and output based on the text alone in a form of a summary of discovered themes [37]. There are two modes in which the detection of topics can be done - online or offline. Online detection has an element of time continuity, as it aims to discover dynamic topics as they appear, whereas offline detection is retrospective, considering the documents that the models is extracting topics from as a corpus [38]. There are four approaches to topic modelling: keyboard-based approach, probabilistic topic modelling, aging theory, and graph-based approaches [39]. The topic modelling that will be applied as part of the current research is offline, as the reports have been acquired prior to implementation of the topic modelling algorithm. The approach that will be implemented as part of the development will be probabilistic, namely Latent Dirichlet Allocation (LDA).

LDA can be defined as a Bayesian hierarchical probabilistic generative model and it collects discrete data, which means that it has a built-in assumption that words within the document are exchangeable, on the basis of which topic models are created [35].

LDA has been found to have superior performance when compared with competing topic modelling tools such as Latent Semantic Indexing (LSI), which uses linear algebra and vector representations for mapping words with a similar meaning [36]. LDA is also one of the most

commonly researched topic modelling algorithms in academic literature, which also offers the advantage for developers implementing it of having a variety of visualization and development tools and libraries to support its application as part of any developed program or software, with a prime example being LDAvis. Research has demonstrated that implementing LDA topic modelling can improve a company's competitive advantage for analysing data sampled from social media, when paired with sentiment analysis techniques [40]. In a financial context, the technique has been used to analyze consumer complaints in a financial protection bureau in the study by Bastani et al. [41].

It does have some limitations, the most commonly cited being the previously mentioned assumption of document exchangeability, which is unsuitable in a dynamic context where the meaning of a topic can change over time [42]. The model is also criticised for neglecting co-occurrence of relations across documents [43][44]. Finally, some scholars have suggested that as the model works with unlabelled data, there is potential for further automation when implementing it in an industry context [45].

With the prevalence of this model in literature, many academic scholars have proposed adaptations to the algorithm in order to address some of the criticisms illustrated above and to make LDA more suitable to a greater variety of tasks. These include adding a hierarchical dimension [46] or adding a Dirichlet multinomial mixture and a word vectorisation component [47].

2.8 Sentiment Analysis

Another common research problem in the field of NLP is sentiment analysis. This is defined as the process of extracting sentiment or semantic expressions from text. In technical terms, this is represented through classifying an opinion expressed in the text as either positive, negative or neutral [48]. Opinion extraction is superior to simple text mining or data mining, as it allows organizations, who utilise this software to gain insights into behaviour by analysing text related to the user or client base [46]. Sentiment analysis is also often discussed in association with another form of research - trend capturing, which aims to collect the output data from a variety of NLP tasks and collate this as input for predicting future behaviour [14]. The implementation of sentiment analysis and predictive behaviour mod-

elling techniques is considered a source of competitive advantage for organizations and is recommended by academic scholars [49].

There are some challenges with sentiment analysis, the main being language ambiguity or the fact that many times when language is spoken, there might be a presence of mixed semantic attributes, which makes it difficult for a classification algorithm to perform its function [50]. It is also difficult for people who will view the program's output and classification to categorize the context in which the semantics are giving, which might hinder the effectiveness of the classification overall and its usability in a real-world context [51]. Nonetheless, it is considered that these hindrances are outweighed by the benefits that sentiment classification offers in terms of speed in comparison with human evaluation and insight.

This has resulted in sentiment analysis being researched as a potential solution to a variety of business problems in various contexts, such as pattern recognition of social media for the detection of road accidents [52] or stock market prediction [53], to name a few.

Although most academic studies aim to detect sentiment polarity as one of three classes (e.g.[54][55]), this ternary approach can be considered a limitation of the field overall. Chaturvedi et al. [56] argue that sentiment analysis in its traditional form is unable to address the complexities of modern-day expression, as it fails to capture objectivity and subjectivity. The example of the study is where categorization is attempted between fake news and facts, which is proved impossible through traditional sentiment analysis. Some more advanced options use aspects of text such as affective manifestations, which indicate emotions [57], whereas other researchers have implemented deep learning algorithms, most commonly convolutional neural networks [58].

Sentiment classification models can be supervised, semi-supervised and unsupervised, with the first being challenging to obtain and cost-inefficient [59]. Classification can be performed at a word-, document- or sentence-level [60], depending on the data available.

2.9 Text Extraction

Text extraction, otherwise referred to as text mining, is a process of extracting text strings from a source that can later be used as input for machine learning algorithms [61]. Text mining has no formal definition, as this process can vary from organization to organiza-

tion depending on the context of the task, as well as the available sources to pull data from [59]. One of the key problems of text mining research is found to be information extraction, which aims to extract the components of the text that offer most insight in terms of semantics and text interference [59]. There have been applications developed for information extraction from documents, who used natural language. These applications use a combination of NLP-ML techniques, namely information extraction, classification/categorization of these documents, based on the language shown in them, automated electronic data transmission, processing and routing, and plain parsing [62]. Furthermore, the applications can handle both pre- and post-processing of the application data and enrich the information extracted. In documents where there is an abstract structure, such as in financial reports, information extraction can happen through a framework such as the one proposed by Smith and Lopez [63], where heterogeneous sources are paired together on the basis of their application domain, resulting in a 'tool which recognizes the implicit structure and identifies pertinent information, concepts, contained in semi-structured documents' (p. 141).

There are many approaches to text extraction, such as in image processing a pyramid approach for extracting text strings from an image that contains text, such as a road-map [64], or a connected-component approach, which groups small components of an image until larger ones are identified [65]. In light of the current research, it can be argued that this approach can be implemented as part of an advanced information extraction algorithm, which aims to identify if companies have used info-graphics or other types of graphical data to 'hide' concerning financial data. Other researchers have developed algorithms to extract text from handwritten documents [66], which can also be useful in the context of the current research, if documents are listed as part of the reports that contain hand-written notes (e.g. a note to shareholders).

The following chapter discusses design methodology, after which there will be an overview of technologies used as part of the current research and the iterations of development.

Chapter 3

Research Methodology

The research methodology chapter provides an overview of the methodology used as part of this research. Specifically, the Design Science research concept will be explored, an overview of which will be provided as part of Section 3.1, after which the design and creation processes will be explained, with reference to academic literature. Thus, in Section 3.2 the reasoning for selecting the various methodologies which were incorporated in the creation process of the artifact of this research application will be given. Section 3.3 discusses the data collection procedures. Section 3.4 discusses the role of the artifact in design science, as well as providing an overview of the artifact created as part of this research project. The end-to-end prototyping is exemplified as part of Section 3.5. In section 3.6 and 3.7 semi-structured and structured interviews are discussed and in the last section 3.8 the expert interview is explained.

3.1 Design Science Research

Design science is fundamental to engineering, architecture, and the arts, which leads to it being crucial for the area of Information Systems, as it enables the development of prescriptive theories [67]. Such theories can enhance the development process of practical and effective software applications which solve existing business problems [68]. Design science considers knowledge as a build project, rather than as an object [69], which shifts the focus of development onto how knowledge is produced. Some scholars have questioned how this concept can be implemented in software development, especially when considering that

some systems do not exist in any form prior to their conceptualisation [68], yet literature demonstrates that this can be achieved through developing new ideas and theories about knowledge. The fundamental component of design science is rooted in the process of theorising, as this leads to new ideas, concepts, frameworks, methods and models [70]. Thus, this concept of software modelling and creation considers the current state of things insufficient, but instead questions how it can be used for creation of new applications [71].

In traditional science, there are three methods of reasoning - the inductive, deductive, and abductive, with design science using the last one mentioned [72]. The abductive method consists of studying facts and proposing theories to explain them through a process of creating explanatory hypotheses for a given phenomenon or situation. Dresch et al. argued that this method is needed, for example, when the researcher is proposing solutions to unresolved problems [73]. Therefore, in design science, research usually starts from the need to design or build a given artifact, with the researcher demonstrating the need to develop an artifact through gaining an understanding of existing processes and observations of reality. At a later stage, other scientific methods can be used to test these hypotheses. Consequently, design science research is considered a scientific method, and it should be conducted in consideration with such principles, which means providing a clear emphasis on quantitative and logical reasoning. This demonstrates how traditional science and design science do not oppose, but rather compliment each other, yet take somewhat different approaches. Traditional science is more focused on the problem, whereas design science is focused on the solutions to a problem. In this regard, design science opposes the traditional notion of research that aims to explore, explain and describe a phenomenon, shifting the objectives of research to prescribing solutions and designing or formalising artifacts [74]. Therefore, as stated by Simon [75],

'fulfill-ment of purpose or adaptation to a goal involves a relation among three terms: the purpose or goal, the character of the artifact, and the environment in which the artifact performs' (p.28).

Therefore, a successful solution is the one that is sufficient for problems in which an existing solution is inaccessible or impractical for implementation [70]. This concept relates to the software development principle of the Minimum Viable Product (MVP), which mandates that a working prototype that solves an existing problem is produced at the first possible

instance, after which it is improved during later development iterations. This can be as either

- (1) a design artifact - (i.e. used for design idea visualisation, reflection on existing architectural designs, facilitation of creativity or clarifying user expectation-output mismatches)
- (2) a boundary spanning artifact - (e.g. bridging the gaps between business and technical thinking, between the inventor/entrepreneurial team and end user, and between the inventor and investors.
- (3) a reusable artifact (e.g. for documentation, as a growth hacking mechanism or as a bootstrapping tool) [76]

Therefore, a key aspect of the implementation of design science is the definition of the MVP requirements, otherwise referred to as the satisfactory result. This can be achieved by either achieving a consensus among the parties involved in the problem or an advancement of the current solution, compared to the solutions generated by previous artifacts.

The above-illustrated considerations from a design-standpoint contradict the principles of operational research, which aims to achieve an optimal outcome. Thus, design science is established as a method with pragmatic validity. To elaborate, it seeks to ensure that the solution proposed for solving the research problem is tested in the context of the problem and achieves the expected results [72]. Additionally, by approaching the research questions from a pragmatic standpoint, the researcher should ensure that the development process takes into account;

- the costs and benefits of the solution
- whether it meets the specificity of the environment/context in which it will be applied
- the needs of the parties interested in the proposed solution [72]

As a result, the implementation of the design science research approach is often driven by more than one scientific method and requires the researcher to apply creative reasoning for finding ways to solve the problem at hand.

From a research methodology standpoint, design science is categorised as an epistemological paradigm that advances the progression of knowledge in academic research, especially

in the field of Information systems, where the concept's application results in a faster development process overall [67][68]. This leads to the emergence of another research method, which is referred to as Design Science Research, which can solve an existing business problem. Specifically, a design result in the building of an artifact, which is what the current research will do. The concept of an artifact will be explained below, in Section 5.3

3.2 Design and Creation

The generic process of design has four stages: analysis, projection, synthesis and communication [2]. Each of these breaks down into four smaller processes (or micro-processes): research, analysis, synthesis and realisation. An image 3.1 to represent this is located below:

		Steps of the iterative micro process of learning / designing			
		research	analysis	synthesis	realization
Domains of design inquiry, steps / components of the iterative macro process of designing	ANALYSIS "the true" how it is today	How to get data on the situation as it IS? → data on what IS	How to make sense of this data? → knowledge on what IS	How to understand the situation as a whole? → worldviews	How to present the situation as IS? → consent on the situation
	PROJECTION "the ideal" how it could be	How to get data on future changes? → future-related data	How to interpret these data? → information about futures	How to get consistent images of possible futures? → scenarios	How to present the future scenarios? → consent on problems / goals
	SYNTHESIS "the real" how it is tomorrow	How to get data on the situation as it SHALL BE → problem data	How to evaluate these data? → problem, list of requirements	How to design solutions of the problem? → design solutions	How to present the solutions? → decisions about "go / no go"
	COMMUNICATION "the driver"	How to establish the process and move it forward? How to enable positive team dynamics? How to find balance between action/reflection? How to build hot teams? How to enable equal participation? → focused and efficient teamwork			

Figure 3.1: Design process model (generic) [2], (p.3)

There are a variety of research processes that address the design and creation process in design science. All of them can be broken down into three key stages: problem identification, solution design and evaluation [2]. The following table offers a summary of some of the methodologies that can be utilised, as categorised by Offerman et al. [2] (p.4).

	Peffer et al. 2008	Takeda et al. 1990 [77]	Nunamaker et al.1991[70]	March and Smith 1995[78]	Vaishnavi and Keuchler 2004/5[79]	Process presented here
Problem identification	* Problem identification and motivation * Define objectives for a solution	* Enumeration of problems	* Construct a Conceptual Framework		* Awareness of Problem	* Identify problem * Literature research * Expert interviews * Pre-evaluate relevance
Solution Design	* Design and development	* Suggestion * Development	* Develop a System Architecture * Analyze and Design the System * Build the system	* Build	* Suggestion * Development	* Design artifact * Literature research
Evaluation	* Demonstrate * Evaluation	* Evaluation to confirm the solution * Decision on a solution to be adopted	* Observe and Evaluate the System	* Evaluate	* Evaluation * Conclusion	* Refine hypothesis * Expert survey * Laboratory experiment * Case study / action research * Summarize results

The current research's development process has been conducted through the above-presented approaches due to them being affirmed by literature as suitable research approaches for the design science methodology for Information Systems research [2].

The artifact/application is developed in continuous collaboration with the case study organisation (DNB Liv), which mandates that all design and development processes are done in a collaborative manner. As a result of this, an agile methodology will be used. Agile is a software engineering methodology that is user-centric, with an iterative and incremental development approach that requires robust planning and management of self-organising and cross-functional teams [4]. Projects developed with this methodology can vary in terms of project life-cycle, as this is entirely guided by the product features, which in turn are guided by customer requirements. Such a method requires a high degree of collaboration, coordination and communication between team members [80].

As part of the agile methodologies, there are a variety different ones that have emerged as a result. This research will use a Scrumban method, which is a combination of Scrum and Kanban. Scrum is an agile iterative development process which also utilises sprints and frequent iterations [80]. The sprints are at default set to 14 days, with improvements being recorded at the end of each sprint, using a Kanban board on an online tool called Trello for keeping control of completed tasks and outstanding tasks. Additionally, a backlog has been created as well, the use of which is to assist with the conceptualisation and planning of future improvements [80]. All described creation procedures (i.e. a development, simple design, refactoring and rapid prototyping) are derived from the practical implementation guide of the agile development methodology in software system creation.

According to the research done by Gustavsson there are numerous benefits for applying the principles of agile development which are reported by numerous organisations involved in his study [81]. These benefits have been summarised in the following table 3.1.

Value from the agile manifesto	Corresponding reported benefit
Individuals and interactions over processes and tools	Better collaboration in the team Increased transparency and visibility Increased knowledge sharing Better focus Impediment removal process Increased individual autonomy Increased motivation Clear sense of progress Improved resource allocation
Working software over comprehensive documentation	Increased productivity and speed Increased quality
Customer collaboration over contract negotiation	Increased customer interaction Better understanding of goals/tasks/requirements Customer-centred value-add priority process Increased cross-organisational collaboration Decreased customer complaints
Responding to change over following a plan	Increased flexibility of coping with change

Table 3.1: Benefits of Agile Development [4]

His research has shown that there are several areas which are not even close to agile development where there is great interest in applying the principles of this methodology. These benefits were intended, which is the rationale for going for an agile creative process in this research.

3.3 Data Collection

Data collection for the purpose of process modelling will be done through informal meetings with the bank and its employees, the actuators. This will be done to understand how the actuaries work, and establish an overview of the day-to-day procedures their roles are involved in. The gathered insights will later be utilized in the process of modelling the requirements for the application, created as part of the research. So these semi-structured interviews were conducted in an informal meeting, with the actuaries responding to question about how they work, and giving an overview of what they usually do. No personal or identifying data

about the employees was recorded or requested.

Semi-structured interviews are a naturalistic, data-rich source of information which allows the subject to express a deep opinion. It offers the researcher the ability to gain a deep understanding of a topic, beyond what is being asked for, which also allows the researcher to get a feel for the expressed sentiment [82][83][84]. A limitation of this research approach is that the transcripts from such interviews are lengthy and time-consuming to analyze, with patterns amongst participants sometimes not emerging, which can hinder the research process [72][73]. In this case the semi-structured interviews were on the lighter side of structured and contained mostly handwritten notes from informal meetings with the bank employees.

Additionally, as part of the current process modelling, DNB Liv will provide client data from some organisations they have on file. This data includes information about how much sick leave a company has, what their average work injury stats are and how often their employees have to be let go or temporary laid off, all of which is currently used as part of the risk assessment process of the bank prior to selling insurance to any organization. The data will be used to gain insight into the current risk assessment processes as well as providing a foundation for the application creation, as it will be speculated how it can be used together with the analysis of fiscal reports.

A non-disclosure agreement (NDA) has been signed to eliminate the potential for releasing sensitive information as part of this research. Measures have also been taken to anonymize sensitive data as part of the text pre-processing procedures, and the list of companies which include the assessed risk factor is not included in the appendix.

3.4 Artifact

In the context of design science, artifacts are understood as things that are man-made, or as described by Simon [75], they are 'artificial things' (p. 28), which can be considered through characteristics of functions, goals, adaptation. Therefore, they are knowledge-containing. This knowledge can potentially range from the design logic, the construction methods and tools to the assumptions about the context in which the artifact is intended to function. It is considered that both the creation and the evaluation of artifacts is an important aspect of design science research, which affirms the importance of evaluative procedures.

Design science research artifacts range from models, methods, constructs to instantiations and design theories, and can even include social innovations, new explanatory theories and implementation methods. The artifact created as part of this research is an application for using machine learning and natural language processing to automate the analysis process of fiscal yearly reports for use in risk assessment. Some parts of the program has been developed using Python. The application has been conceptualised to have a user-friendly GUI that is suitable to the needs of bank employees, which has been conceptualised as wireframes. The system prototyping process will be detailed in the next section.

3.5 Design Prototyping

In recent years, there has been an upsurge in rapid prototyping both as an academic discipline, as well as an organisational and business's imperative, with companies being enabled to prove the design of their products a lot faster than ever before [85]. As there is a vast number of systems being built using rapid prototyping, there is great variability in the applications manufactured. This leads to efficient automation, as existing problems are solved quickly through solutions that are built in a rapid and adaptive manner, as well as with consideration of, and cooperation with the end user [79]. As discussed by Kapyaho et al., traditional rapid prototyping puts an emphasis on determining a set of requirements in a separate phase before starting actual software design and development [86].

Prototypes of the system will be refined through feedback from the meetings with bank employees, which is then used to improve the iterations of the systems in each subsequent iteration. The overall process that has been followed has been influenced by Kapyaho et al's research [86] and proceeds through the following stages:

- Eliciting Initial Requirements
- Building the Prototype
- Reviewing the Prototype

The first stage consisted of semi-structured interviews with the client, where they shared their ideas of the features they wanted implemented in the following iteration. These ideas

were typically discussed very informally and taken into consideration alongside the synthesised process models. The process models were developed on the basis of data that was provided by DNB Liv of their risk assessment procedures that concern how they review clients and from fiscal reports from Brønnøysundregistrene. The meetings ensured that there was an opportunity to discuss implementation issues, as well as gain insight into future development plans. Each meeting enabled a good enough picture of what should be done for the next stage of building the prototype system.

The prototyping was fairly unstructured, and was started right after the initial requirements meeting. The process modelling and interviews have produced the vision and idea of the UI, while the literature review allowed the work to start on implementing and optimizing the text extraction that were applied. After the initial prototype was developed, the bank employees reviewed the design.

The review stage involved the stakeholders in the case organisation (DNB Liv), which was the actuators and the head of the department. In this stage, we went through the prototype features together and there were questions and discussions along the way. As the application developed was a MVP, there were changes proposed, which included tweaks, yet the system was shown as a potential solution to the problem.

3.6 Semi-structured Interviews

Semi-structured interviews are a style of less rigid questions posed to the interview subject, and is in between the styles of structured and unstructured interviews [87]. Interviews like these usually have a basic guideline to make sure the essential topics are covered and that answers are somewhat satisfactory [88]. A variant of this type of interview questions was done in a series of informal meetings with the bank employees of DNB.

3.7 Unstructured Interviews

Unstructured interviews are less controlled and more exploratory, and often go deep into a particular field. An agenda should be set for the interview and a plan on what topics to discuss should be set. New information uncovered during the interview should be followed

up upon and explored further. The benefit of using an interview style like this is that you could gather new knowledge about topics that were not already known to the interviewer [87].

Not all of the meetings with DNB had semi-structured interviews and also some mail correspondence answered some open-ended questions.

3.8 Expert Interviews

These types of interviews can follow any of the structured-, unstructured or semi-structured interview style, but are conducted with experts of a domain. Although they are experts in their field, one should be aware that their answers can be biased and can have a more critical viewpoint than a person less knowledgeable in that domain. However, they can add valuable insight when interpreted with caution. [88].

Chapter 4

Tools

This chapter will elaborate on what kind of tools, libraries and programming languages that have been used in this research project.

4.1 Programming Language and Libraries

Python

The Python is an interpreted, high-level, general-purpose programming language. It is often used for smaller projects since the language has a simpler syntax, but has proven to be powerful enough for big projects. There is a good selection of natural language and machine learning libraries and one of the reasons it was chosen. [89]

PyPDF2

A pure-Python library build as a PDF toolkit, this library is capable of extracting document information, splitting documents page by page, merging documents page by page, cropping pages, merging multiple pages into a single page and encrypting and decrypting PDF files [90].

From this library the modules, PdfFileWriter and PdfFileReader was used.

This was first used to access and read the PDF's, but after some testing 'textract' proved to be better for PDF's containing images.

textract

A Python library used for extracting text in Word documents, PowerPoints and PDF's [91]. This is a collection of libraries/packages used for text extraction, and the default for PDF's is 'pdftotext', but since many of the PDF's were scanned documents or contained images the 'tesseract-ocr' package was used instead.

tesseract-ocr

Tesseract OCR is a package that extract text from images, and it has unicode (UTF-8) support, which is needed since almost all of the PDF's contained Norwegian letters [92]. It was used as a method for extracting text in the 'textract' collection of packages library.

pandas

Open source library providing high-performance, easy-to use data structures and data analysis tools for Python [93].

Data-frames was used to take all the extracted data from the reports and put them in columns in single a combined .csv file . Also used to load the data from the .csv when doing frequency distribution, LDA visualization and word-clouds.

nlTK

A industry leading platform for building python programs in regards to work with human language data [94].

Used for tokenizing words, which divides strings into substrings by splitting on parts of the string you chose. Then removing stopwords for both English and Norwegian words. Stemming of words to remove/concatenate words with same meaning, like «report» and «reports». Lemmatization to change the words like «am, are, is» to «be». Stemming and lemmatization was tried, but removed due to the language in reports, it did not provide that much value.

Modules that were used:tokenize,corpus, WordNetLemmatizer, SnowballStemmer,FreqDist

gensim

It is a collection of python scripts for use in topic modeling [24][25]. Which was explained more in the theory chapter.

Models that were used: `simple_preprocess`, `corpora`, `models`, `STOPWORDS`, `TfidfModel`

FreqDist

This is included in the NLTK toolkit, and provides a way to encode «frequency distribution», so that it counts the number of times that a word occurs. [94]

wordcloud

Word cloud generator library for Python. Used to create wordclouds of the different categories of risk, from high, medium and low. It takes the words from the frequency distribution and shows them visually in a wordcloud, so that the most common words are the biggest. [33]

matplotlib

Matplotlib is a 2D plotting library used to make different figures in a variety of formats. It can be used python scripts, shells web apps but more notable for this use, in Jupyter notebook [95].

scikit-learn

scikit-learn is a machine learning library for Python. It contains classification, regression and clustering algorithms and works well with other scientifically and numerical libraries like NumPy and SciPy.

In the application this was used for creating a TFIDF Visualization of the reports in vector space. By using the modules: `TfidfVectorizer` and `TSNE` [96]

pyLDavis

Python library for interactive topic model visualization. pyLDavis is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization [97].

The visualization is intended to be used within an IPython notebook but can also be saved to a stand-alone HTML file for easy sharing

4.2 Applications

Jupyter

Jupyter is a server-client application that allows running and editing «notebook documents» in the web browser. It has a kernel that is a «computational engine» that executes the code inside the notebook. It can be done cell by cell, and the kernel is started automatically when we launch a notebook [98].

The interactive Jupyter notebook from IPython Interactive Computing was used to be able to run Python code line by line instead of running an entire script. Jupyter used to be included in IPython but has been separated from the main project. It is an interactive environment where we can combine code and rich text. This is a useful tool to keep an overview of the code base and easier to debug and troubleshoot the code. Also made it easier to go from one development environment to another, as I used a desktop with Windows and a laptop with Linux.

Anaconda

Anaconda is a distribution software that is used to install Python, and it contains a bundle of packages. Many which come pre-loaded. It comes with Jupyter integrated [99]. It also makes it easier to keep a python development environment the same form one computer to another, regardless of the OS the computer has.

Anaconda was used to install and have the same development environment on two different

computers, and in conjunction with Jupyter.

Git

Git is a version-control system that tracks changes in the source code during development. It is most commonly used to coordinate work between developers so that they can work on different areas of the software without disrupting or overwriting each others code [100]

Git was used it to track changes during the development stages and to keep all the source files in one place. It also made it easier to work on two different computers. The well known platform that host Git, called Github was used.

Trello

This is a Kanban list-making application that is hosted on the web. Made by the company Atlassian, and it's basic features is free to use [101].

This was incorporated into the project to keep control of tasks during the different iterations of my development, and to do the work in planned sprints.

Dropbox

Dropbox is a file hosting service that serves a workspace in the clouds, that makes it easy to access files between different workstations and computers. It also supports sharing between different people, and it comes with 2 GB of free storage [102].

In Dropbox the images, documents etc was stored, to make it easier to access it depending on whether the work was done on a laptop or desktop.

Wireframesketcher

Wireframesketcher is a very lightweight tool for creating wireframes, mockups and prototypes for web, destop and mobile [103].

It was used to create the first low-fidelity prototype of the application.

Adobe Xd

Adobe Xd is a vector-based UX tool, to help design web- and mobile apps. It supports the creation of wire-frames and creating immersive and interactive click-through prototypes. This allows for rapid prototype development [104].

Adobe Photoshop

Adobe Photoshop is a raster graphics editor that is one of the most well known photo editor software. It is such a leading product in the market that "photoshopping" images is being used as a verb to describe the editing of images [105].

Both Xd and Photoshop was used to create/edit medium- and high-fidelity wire-frames and images.

4.3 Summary

The selection of the different tools and libraries that were used in this project came after research and some trial and error. Some of the applications were chosen because they helped with a more productive workflow, and to be able to work on different computers.

The following chapter will illustrate the development that was done in this project, illustrating the various components of the system that were created as part of each of the four iterations. Followed by a evaluation of the overall research project in Chapter 6.

Chapter 5

Development

There are four iterations overall, each of which had a specific aim (listed in Table 4, below). As detailed in the Methodology chapter, each iteration was developed using the Scrumban, with one alteration made. Instead of the traditional for Scrum/agile development 14 days that are used for sprints, in this case the sprint lengths would vary, and would often be longer than the norm. This decision was driven by the meeting schedule with the bank, where the time between each meeting was used for development, which would sometimes be a month or more in-between.

Table 5.1: Iteration cycles

Iteration	Goal
1	Find requirements / Get reports (PDF)
2	Extraxt text from reports
3	Do analysis of the text
4	Prototype for a web application

As discussed in previous chapters, Trello was used as a digital time management and planning tool. The initial requirements for the project are illustrated in Figure 5.1. The web-based board was utilised during all iterations and was operated using a Kanban methodology. The following sections will provide an analysis of the work completed in each iteration, followed by a discussion of the results.

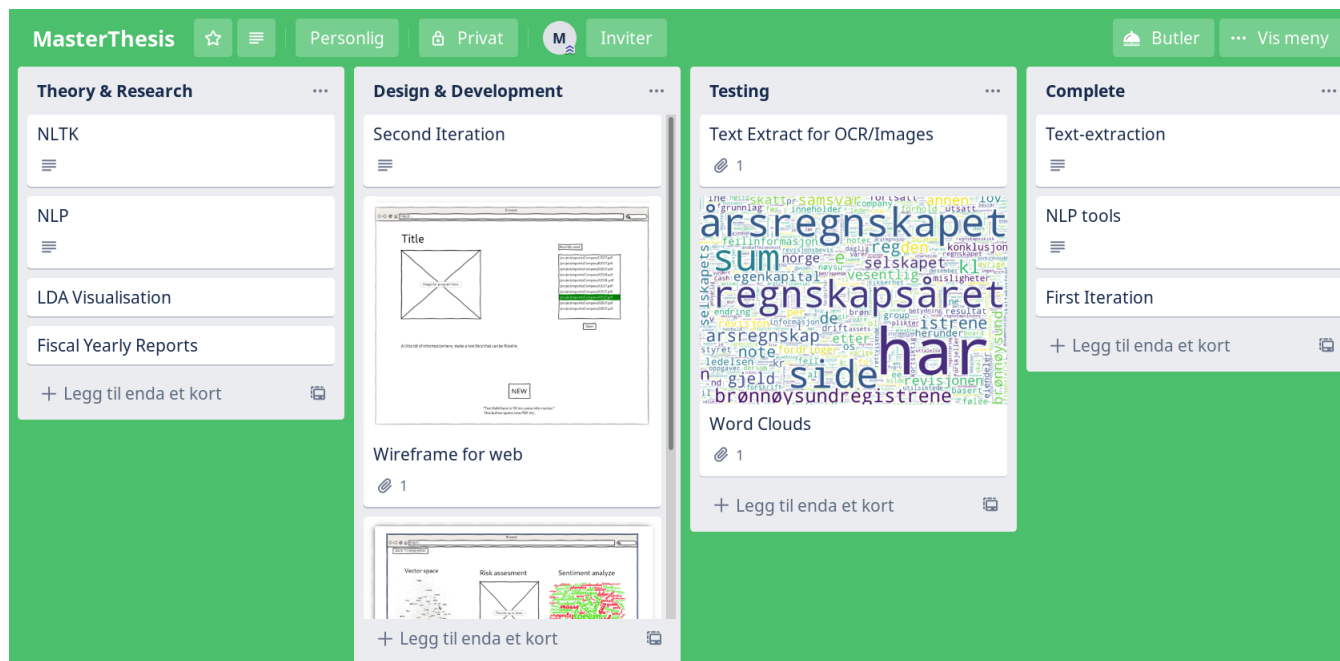


Figure 5.1: Kanban board on Trello

5.1 First Iteration

The first iteration followed the first meeting with the bank. This meeting was informal with some bank stakeholders, specifically the head of the insurance department and a consultant from the IT department. It also included a couple of actuaries which gave their input on research paths for work on analyzing fiscal reports. In this meeting the business need and user requirements were contextualised. The outcome of the meeting included some initial ideas for system requirements and pathways for development, as suggested by bank staff.

5.1.1 Initial Requirements

Semi-structured Interview

To figure out the requirements, some question were posed to the employees of the bank.

1. *Why do you analyze fiscal yearly reports?*
2. *How do you analyze fiscal yearly reports?*
3. *How do you feel the current process of analyzing a report is?*
4. *What improvements to the current process of report analyzing would you like to see?*

5. *How would you like them to be displayed to you visually?*

Following the discussion with the project's stakeholders, the primary need for the developed system was identified, namely creating a faster way to understand if a fiscal yearly report would indicate that the company they were doing a risk assessment on had a good or a bad year. Alternatively, if a conclusive assessment could not be reached based on the system's analysis, there was a need to create a report on any other identifier (or combination of identifiers) that could assist risk assessors making a decision on the risk assessment factor. Overall, any form of task automation was sought as a means of reducing the manual time for labour as the actuators of the bank had to spend a lot of time performing risk assessment manually, as some of these reports could be well over a hundred pages long. After a collaborative discussion and upon providing recommendations on the basis of their business problem's need, with positive sentiment being expressed for technologies such as Word-cloud, Graph Visualisations and Thumbs up/down or percentage based indicator as components of the end system.

Functional Requirements

When determining the functional requirements, it is important to understand what requirement needs the intended user has [87]. In the previous section, the needs of the user was discovered after a collaborative discussion about the business problem and brainstorming about solutions.

The functional requirements are as follows:

- Automated way to do the work, thus saving time for the workers to pursue other work tasks
- Provide data that is accurate and helpful for risk assessment
- Have an application that can combine all the results and visualize them

Non-Functional Requirements





Non-functional requirements are extra features that add to the aesthetics of the application, and can in some cases provide constraints on the system that is being developed [88].

The non-functional requirements are as follows:

- Look pretty and be visually pleasing
- Take user experience into consideration
- Work on different devices and browsers

5.1.2 Obtaining Data: Getting the Fiscal Yearly Reports

PDF versions of some reports were originally provided by the bank; however, the bank only had a few select reports on file. Nonetheless, fiscal yearly reports are available in the official registry (also known as Brønnøysundregistrene). The registry's operational system allows a user to search for the company by its identifier (organization number) and order a copy of the fiscal yearly reports for any company and any year, through a user-friendly interface, illustrated in Figure 5.2.

 **Brønnøysundregistrene** Språk  Søk  Meny 

[Forsiden](#) | [Produkter og tjenester](#) | **Bestilling av utskrifter, attester og kopier**

Bestilling av utskrifter, attester og kopier

Søk [»](#) **Velg** [»](#) [Detaljer](#) [»](#) [Logg inn](#) [»](#) [Bekreft](#) [»](#) [Betaling](#) [»](#) [Kvittering](#) [Logg inn](#)

Her er alle tilgjengelige produkter på
organisasjonsnummer 988 996 564
 BILFINGER INDUSTRIAL SERVICES NORWAY AS, Herøya Industripark Hydrovegen 55, 3936 PORSGRUNN Har du roller i det aktuelle selskapet,
 kan du bestille enkelte produkter gratis i
[Altinn](#)

Kryss av for produktene du vil bestille og trykk «Gå videre» nederst på siden

Kopi av årsregnskap fra Regnskapsregisteret - E-post(PDF)															
	2019	2018	2017	2016	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005
Årsregnskap (kr 0/år)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Utskrift og attester fra Foretaksregisteret, Partiregisteret, Enhetsregisteret og Frivillighetsregisteret		
	E-post (PDF)	Post
Rolleoversikt organisasjonsnummer (gratis)	<input type="checkbox"/>	<input type="checkbox"/>
Firmaattest vanlig (gratis/kr 181)	<input type="checkbox"/>	<input type="checkbox"/>
Firmaattest notarialbekreftet (kr 424)	<input type="checkbox"/>	<input type="checkbox"/>
Firmaattest på engelsk notarialbekreftet NB! Inneholder ikke vedtektsfestet formål (kr 424)	<input type="checkbox"/>	<input type="checkbox"/>
Firmaattest med historisk foretaksnavn (gratis/kr 181)	<input type="checkbox"/>	<input type="checkbox"/>
Registerutskrift fra Enhetsregisteret (gratis)	<input type="checkbox"/>	<input type="checkbox"/>

Utskrift og attester fra Løsøreregisteret og Konkursregisteret		
	E-post (PDF)	Post
Pantattest (gratis/kr 181)	<input type="checkbox"/>	<input type="checkbox"/>
Bekreftelse fra Konkursregisteret (gratis)	<input type="checkbox"/>	<input type="checkbox"/>

Figure 5.2: Brønnøysundregistrene (Official Registry) User Interface

As demonstrated by the Figure 5.2, the user can order a copy of the financial report by indicating in the check-boxes the desired year and the posting type. The interface operates similarly to a web shop, where the orders are placed in a shopping basket. Provided that reports are free for public access, however, the user is not charged with any price at checkout. Once the user confirms their order and checkout is completed, all requested reports are sent in a PDF format to the indicated email, associated with the order account. The download process requires opening a link, viewing and downloading the document.

The first iteration contained reports from the year 2017, as many companies had not filed their most recent reports yet. The process was heavily manual at first, however an automation workaround was implemented. Specifically, upon getting the report in the email, hyperlinks came as the report name, but the PDF was a generated number (e.g. 20190002024494-1.pdf).

The initial process followed for downloading and sorting the reports is illustrated on the left in Figure 5.3, whereas the more automated solution is illustrated on the right.

The first instance shows a process that resembles proximity to current practice, and is very manual and cumbersome; overall, unsuitable for implementation as part of a text analytics program or system. Thus, as demonstrated on the right, a more automated solution was created, which scrapes report links from the source code of the HTML page in the email to a text file, loops over each link and downloads it to a local directory, after which reports are sorted manually in the relevant risk factor categories. This improved the time cost for data gathering significantly as it eliminated the need to manually click “Save link to” and then finding the company in the excel file (which was provided by the bank and contained all companies and relevant risk factors assessments) and renaming the file to that company.

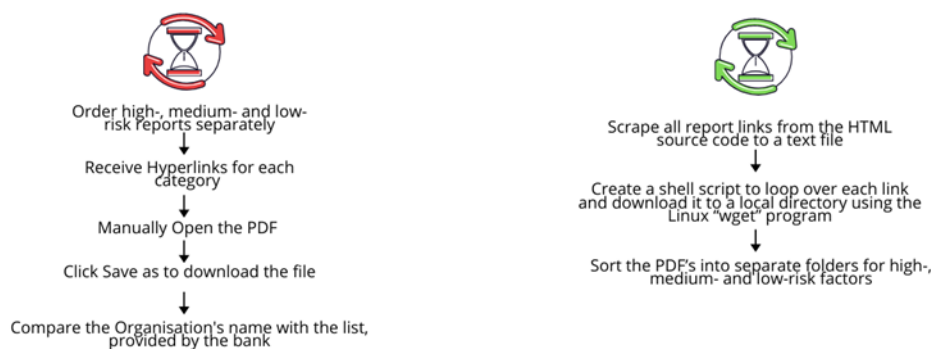


Figure 5.3: Initial (manual, left) versus Subsequent (semi-automated, right) process

Company Name	Employees	Disabled worker	Risk
[obfuscated]	213	3	MEDIUM
[obfuscated]	207	4	LOW
[obfuscated]	200	2	LOW
[obfuscated]	201	0	LOW
[obfuscated]	184	16	LOW
[obfuscated]	186	10	HIGH
[obfuscated]	193	0	LOW
[obfuscated]	176	0	LOW
[obfuscated]	174	1	LOW
[obfuscated]	156	5	LOW
[obfuscated]	159	1	LOW
[obfuscated]	159	0	LOW
[obfuscated]	151	0	LOW
[obfuscated]	147	0	LOW
[obfuscated]	136	7	MEDIUM
[obfuscated]	135	8	LOW
[obfuscated]	136	2	LOW
[obfuscated]	136	1	MEDIUM
[obfuscated]	131	3	MEDIUM
[obfuscated]	128	3	LOW
[obfuscated]	117	0	LOW
[obfuscated]	115	1	MEDIUM
[obfuscated]	104	5	LOW
[obfuscated]	105	1	HIGH
[obfuscated]	102	0	LOW
[obfuscated]	100	0	LOW
[obfuscated]	80	20	HIGH
[obfuscated]	98	1	LOW
[obfuscated]	95	3	LOW
[obfuscated]	87	9	HIGH
[obfuscated]	95	0	LOW
[obfuscated]	94	0	LOW
[obfuscated]	91	3	LOW
[obfuscated]	86	8	MEDIUM
[obfuscated]	91	2	LOW
[obfuscated]	91	1	LOW
[obfuscated]	88	0	LOW
[obfuscated]	78	9	HIGH

Figure 5.4: Excel Sheet of Companies and their Risk Factor

The excel sheet mentioned above, which was provided by the bank contained an overview of 500 companies with information about the number of their employees, risk factor and organization number. It is showcased as Figure 5.4. The risk factor of each company consists of a decimal number and has been categorized into three categories of low, medium or high. The actual numbers are not included, and the names of the companies have been obfuscated due to NDA.

5.1.3 Summary

From a requirements standpoint, the bank assisted the development process by being open to ideas and collaboration. Not many restrictions were set from a requirements standpoint, and the development task was simple and consistent throughout the discussion – finding any way to automate the task of reading fiscal yearly reports, aimed at saving actuators’ time from conducting manual work themselves.

The suggestions brought forward during the initial discussion were approved of and were carried forward as system requirements, e.g. text analysis, information extraction and different types of visualisation. The report extraction procedures from the official registry proved to be heavily-manual and cumbersome for integration in an automated system. Automation was implemented for some aspects of the work, yet some manual components remained.

5.2 Second Iteration

The second iteration involved text extraction from the reports. Insight was provided by the bank for the process of report analysis and risk assessment that is currently implemented in the company by actuaries, working in the departments, specifically what they identify as risk concerns when reading the reports.

5.2.1 Input from Bank

As part of each meeting with the bank employees, a summary of the development progress was provided, which is consistent with agile development principles. Within the second meeting a summary of the report retrieval process was provided, including detailing some of the challenges faced in automating the process of report classification. Furthermore, the format of reports was discussed, specifically it not being in plain text and containing images, which was identified as an analysis challenge for an NLP program. The required input for the second iteration involved identifying specific words or phrases that actuaries were looking for when performing manual risk assessment. The bank provided sample reports of three different companies from public websites, which were annotated by an actuator, which highlighted words and sentences, which they consider telling of potential risk concerns. An example of the annotated report is provided as Figure 5.5.

Additionally, the bank provided a list of words and phrases that they identify as being categorically positive or negative. This classification sample is attached, as Figure 5.6.



Figure 5.5: Sample Annotated Report

Spesielt positive signaler:	Spesielt negative signaler
<ul style="list-style-type: none"> - Ingen ulykker - Få uføre - Godt arbeidsmiljø - Lavt sykefravær - Aktiv bedriftshelsetjeneste - Omfattende HMS-arbeid - God økonomi - Vekstambisjoner - Oppbemanning - Investering - Innovasjon - Oppkjøp - Tilfredse tillitsvalgte - Overskudd - Oppskalering - Tildelt langtidskontrakter - Fornøyde medarbeidere - Bonus til de ansatte - Omstillingsprogram med mål om å ivareta de ansatte - Forebygge - Nytenkende - God score på kundetilfredshetsindeks - Nytenkende 	<ul style="list-style-type: none"> - Ulykker - Økt antall uføre - Utfordrende arbeidsmiljø - For høyt sykefravær - Tiltak for å redusere sykefraværet - Har bedriftshelsetjeneste - Nevner ikke HMS-arbeid - Utfordrende økonomi - Underskudd <ul style="list-style-type: none"> o Ikke tilfredsstillende soliditet o Ikke tilfredsstillende likviditet - Emisjoner - Nedbemanning - Investering - Fare for nedleggelse - Konflikt med tillitsvalgte - Underskudd - Nedskalering - Forhandling om oppkjøp av bedriften - Tapt langtidskontrakter og andre anbud - Misnøye blant medarbeidere - Hjørnesteinsbedrift - Lokalisering utenfor bynære strøk - Mangelfull kompetanse - Nødvendig omstilling - Krav om ny kompetanse - Negativ score på kundetilfredshetsindeks - Digitalisering av produksjonsprosesser

Figure 5.6: Positive and Negative Words/Phrases

5.2.2 Text Extraction

Initially, text extraction was done using the PyPDF2 library, which reads and writes PDF's; however, this approach did not effectively extract text from images. This was troublesome in terms of process efficiency as most documents were scanned, thus the entire PDF was read as a combination of images. As the task required analysis of text alone, the need for processing data from tables or numeric information was non-existent, which shifted the focus of the work.

A solution to this issue, was the textract library, which is normally used for extracting text in a variety of formats, most notably PDFs. The default extracting method used is called “pdfto-text”, but since we are dealing with images, the “tesseract-ocr” method was used. Tesseract OCR is a package that extract text from images, and it has Unicode (UTF-8) support, meaning it can simultaneously extract text from images and include the Norwegian letters “æøå”.

Stopwords were handled using a variety of functions. A Norwegian stopwords list was downloaded, stored and integrated in the following function 5.1, which opens the list encoded in latin-1 as this encryption enables the extraction of extract “æøå” and ‘r’ which is the default option for reading a file in “open”. The stopwords are then put to lower case with the “lower()” method and split on empty space with the regular expression “n”.

```
1 with open('norstop.txt','r',encoding='latin-1') as f:  
2     stopwords = f.read().lower().split('\n')
```

Listing 5.1: Stopwords

The clean_nor function (listing 5.2) uses regular expressions to return a string that has a carriage return or a new line (r) OR (n). It decodes in UTF-8 and turns all to lowercase. It uses NLTK's word_tokenize to turn all sentences into words. It transforms all punctuation characters into commas, after which is strips all characters and splits on whitespaces, removing unused commas. Then it loops over all words and returns the ones that are not in the NLTK' stopwords.

```
1 def clean_nor(text, stop_rem = True):  
2     cl = re.sub('\r+|\n+', ' ', text.decode('utf8')).lower()  
3     cl = word_tokenize(cl)  
4     ## remove punctuations  
5     cl = cl.translate(str.maketrans('', '', string.punctuation))
```



```

6     cl = cl.strip().split()
7     if stop_rem:
8         cl = [i for i in cl if i not in stopwords]
9     return cl

```

Listing 5.2: Clean Function

Subsequently, a table is created, which loads 8 random files from either of the three risk-factor directories, using the code provided as listing 5.3.

```

1     directories = []
2     ## eight randomly selected files from each directory
3     topn = 8
4     for directory in ['HIGH', 'LOW', 'MEDIUM']:
5         for files in os.listdir(directory)[:topn]:
6             file_path = os.path.join(directory, files)
7             directories.append(file_path)

```

Listing 5.3: Dictionary creation and risk factor sorting

The `process_pdf` function 5.4 first checks if the file category HIGH-EXTRACTED, MEDIUM-EXTRACTED or LOW-EXTRACTED exists in the `os` directory, else it makes them. Then it starts the extraction process using the `tesseract` method with the Norwegian language selected as a parameter. The function then uses the `clean_nor` function (detailed previously), after which it saves the files with an appended '-EXT' after the name of the file and changes the text from `.pdf` to `.txt` in lowercase.

```

1     def process_pdf(path):
2         print('the path is : ', path)
3         file_category = path.split('\\')[0]
4         if not file_category in os.listdir():
5             print('making : ', file_category)
6             os.makedirs(file_category)
7
8         text = tesseract.process(path, method='tesseract', language='nor')
9         print('done extracting this pdf')
10        text = clean_nor(text)
11
12        save_name = path.lower().replace('\\', 'EXT').replace('.pdf', '.txt')
13        .replace(' ', '')

```

```
14     with open( os.path.join(file_category,save_name) , 'w', encoding = '
        latin-1' ) as f:
15         print('Writing =>> ', save_name )
16         f.write(' '.join(text))
17         f.close()
```

Listing 5.4: Processing PDF

As reports were placed in different folders based on their risk factor, the text extraction code extracted the text from the PDF reports and put them in folders with the same name, but with "-EXTRACTED" suffixed (e.g MEDIUM-EXTRACTED). This enabled carrying the classification schema of high-, medium- and low-risk factors forward, with the analysis component of the program being able to target specific folders, as it could distinguish their classification based on the folder's name.

An identified limitation of the «extract» method is that it is slow when operating on large PDF's with lots of images. This was overcome by implementing multiprocessing and batch operation principles to extract text. After performing experimentation with processing higher numbers for batches, the command shell froze. The optimal processing speed was identified when processing 8 reports at a time, using the following function 5.5.

```
1  if __name__ == "__main__":
2      processes = []
3      batch_size = 8
4      num_workers = mp.cpu_count()
5      pool = mp.Pool(num_workers)
6
7      for i in range(0, len(directories), batch_size):
8          print('\n\n PROCESSING A BATCH \n\n')
9          for path in directories[i:i+batch_size]:
10             pool.apply_async( process_pdf, args = (path,))
11             pool.close()
12             pool.join()
13             print('Finished one batch of 'batch_size', continuing to next.'
)
)
```

Listing 5.5: Multiprocessing using batches

Following the implementation of all functions detailed above, extracted text was saved into different folders, depending on its category. A sample of extracted text is illustrated as Figure

5.7.

```

<
lowEXT811600792-elteraas.txt

7 brønnøysundregistrene årsregnskapet regnskapsåret 2017 – generell informasjon enheten organisasjonsnummer organisasjonsform
foretaksnavn forretningsadresse regnskapsår årsregnskapets periode konsern morselskap konsern konsernregnskap lagt
regnskapsregler regler små foretak benyttet benyttet utarbeidelsen årsregnskapet selskapet årsregnskapet fastsatt kompetent
organ bekreftet representant selskapet dato fastsettelse årsregnskapet grunnlag avgivelse år 2017 årsregnskapet elektronisk
innlevert 811 600 792 aksjeselskap eltera as fekjan lib 1394 nesbru 01012017 –31122017 ja regnskapslovens alminnelige regler
arne riise 18042018 år 2016 tall hentet elektronisk innlevert årsregnskap 2017 krav at årsregnskapet mv sendes
regnskapsregisteret undertegnet kontrollen at dette utført ligger hos revisorenhetens øverste organ sikkerheten ivaretas at
innsender har rollerettighet innsending årsregnskapet via altinn at bekreftes at årsregnskapet fastsatt kompetent organ
brønnøysundregistrene 08052019 brønnøysundregistrene postadresse 8910 brønnøysund telefoner opplysningstelefonen 75 00 75 00
telefaks 75 00 75 05 e-post firmapostbrrreg no internett www brrreg no organisasjonsnummer 974 760 673 brønnøysund reg istrene
årsregnskap regnskapsåret 2017 811600792 resultatregnskap belop nok note 2017 2016 resultatregnskap inntekter salgsinntekt 115
171 000 127 908 000 annen driftsinntekt 2 492 000 1 000 sum inntekter 117 664 000 127 909 000 kostnader varekostnad 5 46 632 000
56 060 000 lønnskostnad 1 41 229 000 39 776 000 avskrivning varige driftsmidler immaterielle eiendeler 3 45 000 annen
driftskostnad 1 14 732 000 15 444 000 sum kostnader 12 102 592 000 111 325 000 driftsresultat 15 072 000 16 584 000
finansinntekter finanskostnader inntekt investering datterselskap tilknyttet selskap 7 955 000 1 050 000 annen renteinntekt 27
000 12 000 annen finansinntekt 86 000 65 000 sum finansinntekter 8 068 000 1 126 000 annen rentekostnad 16 000 26 000 annen
finanskostnad 1 000 sum finanskostnader 16 000 27 000 netto finans 8 052 000 1 099 000 ordinært resultat før skattekostnad 23
124 000 17 683 000 skattekostnad ordinært resultat 2 3 878 000 4376 000 ordinært resultat etter skattekostnad 19 245 000 13 307
000 årsresultat 10 19 245 000 13 307 000 årsresultat etter minoritetsinteresser 19 245 000 13 307 000 totalvesultat 19 245 000
13 307 000 08052019 kl 1826 brønnøysundregistrene side 1 22 brønnøysund reg istrene årsregnskap regnskapsåret 2017 811600792
resultatregnskap belop nok note 2017 2016 overføringer disponeringer utbytte 15 000 000 12 000 000 konsernbidrag 605 000
overføringer tilfra annen egenkapital 3 641 000 1 307 000 sum overføringer disponeringer 19 245 000 13 307 000 08052019 kl 1826
brønnøysundregistrene side 2 22 brønnøysund reg istrene årsregnskap regnskapsåret 2017 811600792 balanse belop nok note 2017
2016 balanse – eiendeler anleggsmidler immaterielle eiendeler usatt skattefordel 2 705 000 817 000 sum immaterielle eiendeler
705 000 817 000 varige driftsmidler tomter bygninger annen fast eiendom maskiner anlegg skip rigger fly lignende driftsløsøre
inventar verktøy kontormaskiner lignende 32 000 32 000 gs m os os sum varige driftsmidler finansielle anleggsmidler investering
datterselskap investering annet foretak konsern lån foretak konsern ll lån tilknyttet selskap felles kontrollert virksomhet 11
investeringer aksjer andeler 4 597 000 2 891 000 fordringer 7 11 sum finansielle anleggsmidler 4 597 000 2 891 000 sum
anleggsmidler 5 302 000 3 741 000 omløpsmidler varer sum varer 5 500 000 500 000 fordringer kundefordringer 6 11 19 195 000 16
615 000 fordringer 7 11 2 123 000 1 888 000 konsernfordringer 11 12 278 000 1 327 000 sum fordringer 7 33 597 000 19 830 000
investeringer aksjer andeler foretak konsern 4 08052019 kl 1826 brønnøysundregistrene side 3 22 brønnøysundregistrene
årsregnskap regnskapsåret 2017 811600792 balanse belop nok note 2017 2016 bankinnskudd kontanter lignende bankinnskudd kontanter
lignende 8 12 927 000 21 103 000 sum bankinnskudd kontanter lignende 12 927 000 21 103 000 sum omløpsmidler 47 024 000 41 433
000 sum eiendeler 52 325 000 45 174 000 balanse – egenkapital gjeld egenkapital innskutt egenkapital selskapskapital 9 10 50 000
50 000 beholdning egne aksjer 9 10 –4 000 overkurs 10 10000 10000 annen innskutt egenkapital 10 4 033 000 686 000 sum innskutt
egenkapital 4 093 000 743 000 opptjent egenkapital fond 10 annen egenkapital 10 5 578 000 6 088 000 udekket tap 10 sum opptjent
egenkapital 5 578 000 6 088 000 sum egenkapital 10 9 671 000 6 830 000 gjeld langsiktig gjeld usatt skatt 2 annen langsiktig
gjeld gjeld kredittinstitusjoner 7 øvrig langsiktig gjeld 7 11 sum langsiktig gjeld 0 0 kortsiktig gjeld 08052019 kl 1826
brønnøysundregistrene side 4 22 brønnøysund reg istrene årsregnskap regnskapsåret 2017 811600792 balanse belop nok note 2017
2016 leverandørgjeld 11 7 592 000 6311 000 betalbar skatt 2 3 575 000 4 437 000 skyldige offentlige avgifter 5 305 000 6 502 000
utbytte 15 000 000 12 000 000 kortsiktig konserngjeld 11 796 000 annen kortsiktig gjeld 11 10 387 000 9093 000 sum kortsiktig
gjeld 42 655 000 38 344 000 sum gjeld 42 655 000 38 344 000 sum egenkapital gjeld 52 325 000 45 174 000 08052019 kl 1826

```

Figure 5.7: Sample Extracted Text

5.2.3 Wireframes

During the second meeting with the bank, some crude low-fidelity mock-ups of an application was conceptualized. One main/landing page (Figure 5.8) where you could load new reports and the results would show up in separate summary page (Figure 5.9). This main page shows the name of the application, some information about it and you can load a new report with the "New" button. In the summary screen, we can see some of the analysis that has been done post-processing and it is displayed in various visualizations.

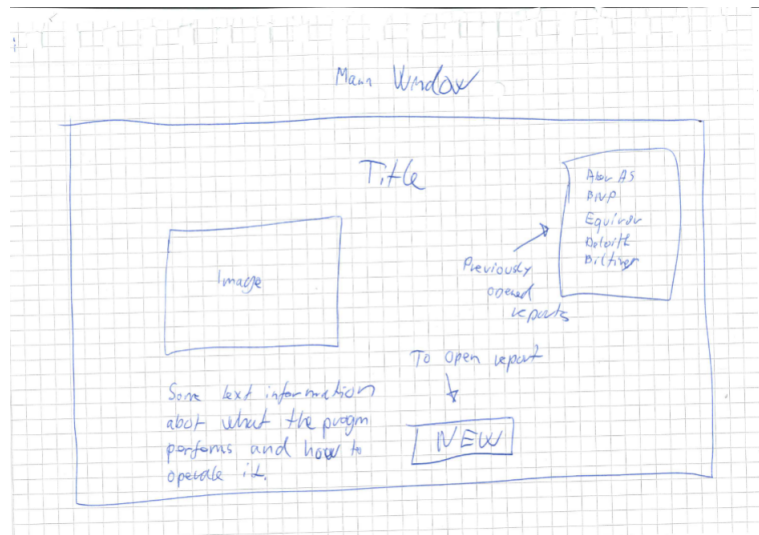


Figure 5.8: Main screen, hand-drawn

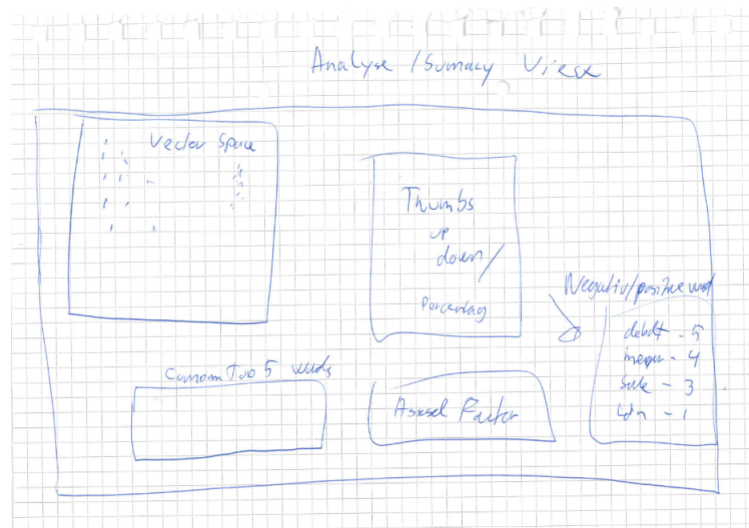


Figure 5.9: Summary screen, hand-drawn

After the meeting with the bank, some further iterations of the wireframes were drawn in the lightweight program Wireframesketcher [103]. Which we can see in the figures 5.10, 5.11.

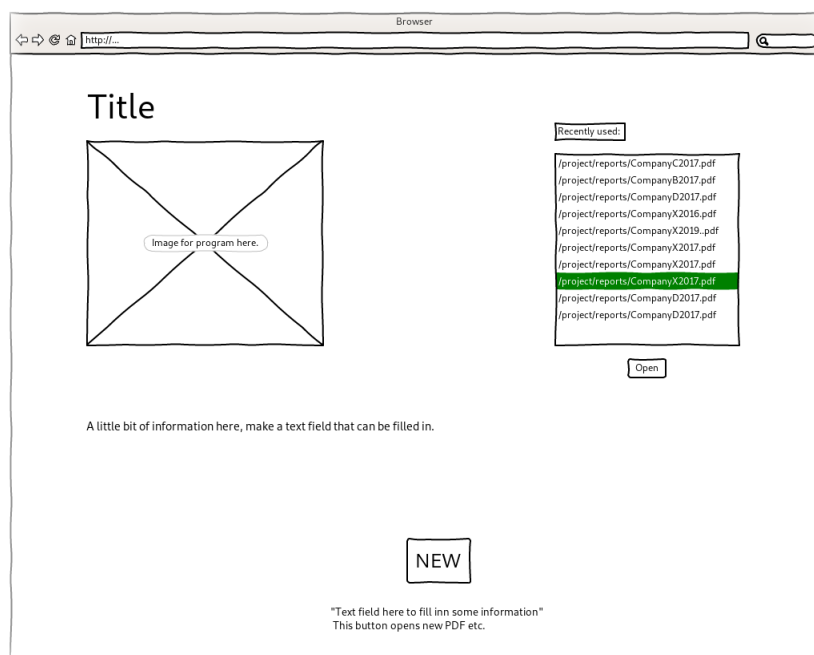


Figure 5.10: Main screen, low-fi wireframe

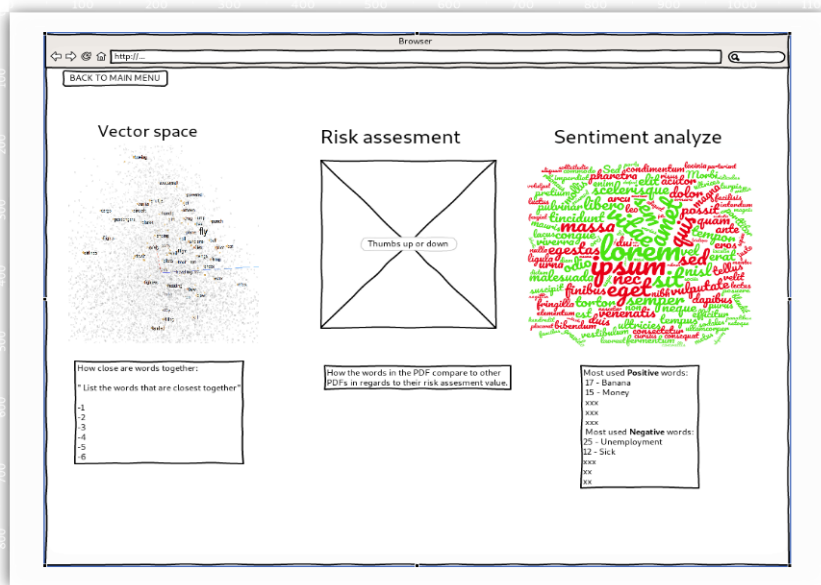


Figure 5.11: Summary screen, low-fi wireframe

Expert Interview

A domain expert in the finance industry with a background in IT was interviewed to evaluate the current design of the wireframes. The expert responded with ideas for directions to take in regards to a user-experience standpoint and the current features. The expert said that easy of use was important, since this is used by employees in their workday. But, a bigger factor is that it can provide the data that is needed, and that is displayed in a manner that is understandable. Also, security is something that should be considered, since there is sensitive data involved. The work should require as little input from the user as possible, and most of the tasks should be automated. It should be able to view previously seen data, and it could also be a good option to view the report before it is processed. The last thing mentioned was that the application should have helpful information, or a small guide of how to use it.

5.2.4 Summary

Upon experimentation with various libraries for text extraction from images, the textract tool and its tesseract-ocr module proved to be most efficient in extracting text from images. Nonetheless, the library offered a challenge of slow operation, which was overcome by implementation of a multiprocessor, which worked in batches of 8, which sped-up the work. 200 PDFs were extracted for around 5-6 hours using the multiprocessor. 1023 PDFs were extracted in total, split as follows: 320 in the high-risk category, 376 for medium-risk and 327 for low-risk. Wireframes were drawn as low-fidelity mock-ups to set the working base of a prototype application, and an interview with an expert in the domain was done for some extra input on the direction of the application.

5.3 Third Iteration

In this iteration data analysis and visualisation was completed, along with relevant data preparation procedures, such as cleaning the extracted text.

5.3.1 Input from bank

As part of this meeting, the initially-proposed approaches were discussed in greater depth, including the use of word-clouds, LDA and frequency distribution for visualization. This enabled the transparency of the approached and mitigated the risk of having an expectation-reality gap, as it provided an opportunity for the stakeholders to discuss and agree the suitability of the implemented technologies.

5.3.2 Analysis and Visualisation

The analysed reports usually start with information about the company, including company name, address, representative and so on, after financial and accounting information is attached. The part suitable for NLP analysis is the annual report or “Årsberetning” (Appendix B.1) in Norwegian; however, the company does not always write an annual report. In some cases, they just include the independent accountant report, or “Uavhengig revisor beretning” (Appendix B.2). Both examples are illustrated in appendix section B. These two parts contain the text needed for cleaning and extraction as they contain human language suitable for NLP analysis. As discussed in previous chapters, some reports feature financial data first, which creates challenges for NLP processing of the fiscal reports.

The report data was cleaned and stored in a file which is a comma separated value file (.csv), where a column for the text was inserted, and another for the risk classification “low”, “med” or “high”, using the function, listed as 5.6.

```
1 def load_data(path_to_folder):
2     texts = []
3     for i in os.listdir(path_to_folder):
4         with open( os.path.join(path_to_folder, i ) , 'r', encoding='utf
5             -8' ) as f:
6             text = f.read()
7             texts.append(text)
8     return texts
9
10 ## Each of them contains list of cleaned text
11 low_clean = load_data('LOW-EXTRACTED/')
12 med_clean = load_data('MEDIUM-EXTRACTED/')
```

```
11 high_clean = load_data('HIGH-EXTRACTED//')
```

Listing 5.6: CSV Creation

Subsequently, lists are loaded with three different classification categories in listing 5.7. After which stopwords removal was implemented, preparing the files for LDA analysis. Also in the `split_clean` function we extract the text that comes after the word 'årsberetning' and a year from 2014-2018 since we had reports from those years. If the report contained an independent account report that was extracted with 'uavhengig revisors beretning'.

```
1 def split_clean(text):
2     if '\årsberetning 2014' in text:
3         #print('Fiscal Report')
4         text = text.split('årsberetning 2017',1)[1].strip()
5     elif 'årsberetning 2015' in text:
6         text = text.split('årsberetning 2017',1)[1].strip()
7     elif 'årsberetning 2016' in text:
8         text = text.split('årsberetning 2017',1)[1].strip()
9     elif 'årsberetning 2017' in text:
10        text = text.split('årsberetning 2017',1)[1].strip()
11    elif 'årsberetning 2018' in text:
12        text = text.split('årsberetning 2017',1)[1].strip()
13    elif 'uavhengig revisors beretning' in text:
14        #print('Independent report')
15        text = text.split('uavhengig revisors beretning',1)[1].strip()
16        text = re.sub('\d+| ', '', text)
17        text = text.translate(str.maketrans('', '', string.punctuation)).
strip()
18        return text #.split()
19
20 low_clean = list(map(split_clean, low_clean))
21 med_clean = list(map(split_clean, med_clean))
22 high_clean = list(map(split_clean, high_clean))
23
24 combined = low_clean + med_clean + high_clean
25 category = ['low'] * len(low_clean) + ['med'] * len(med_clean) + ['high'
    ] * len(high_clean)
26 print('We have now {} total documents for analysis'.format(len(combined)
    ))
27
```



```

28 docs = pd.DataFrame(data = combined, columns=['report_text'] )
29 docs['cat'] = category
30 docs.head()
31 print('Writing to the output ... final_prepared.csv')
32 docs.to_csv( 'final_prepared.csv', index = False, header = True )

```

Listing 5.7: Loading of text into classification categories

Topic modelling was done using `gensim`, `nltk` and `pandas` to do a LDA analysis 5.8. A `pandas` dataframe was created in a similar fashion as when cleaning the text and sorting it in the `.csv` file, following which the documents were processed, sorting them in two columns – one, containing the document text and another for the classification. The `gensim` method `dictionary()` was implemented to create a dictionary, mapping the words between their integer ids. Subsequently, the text was tokenized, using «filter-extremes» `gensim` method to filter out tokens based on their frequency. Different thresholds were experimented with, upon which words were discarded.

```

1  ## These values can be tweaked for different results
2  top = 5 ## Topics
3  below_thresh = 15 ## Words whose frequency is below 15 is discarded
4  above_thresh = 0.5 ## Words whose occurrences is in more than 50 percent
   of text discarded
5
6  ## Remove those english stopwords as well
7  from nltk.corpus import stopwords
8  stop_list = stopwords.words("english")
9  def rem_stop(text):
10     return ' '.join([i for i in text.split() if i not in stop_list])
11     docs['report_text'] = docs['report_text'].apply(rem_stop)
12
13
14     processed_docs = docs['report_text'].apply(lambda x : x.strip().
split() )
15     processed_docs[:10]
16     dictionary = gensim.corpora.Dictionary(processed_docs)
17     dictionary.filter_extremes(no_below=below_thresh, no_above=
above_thresh, keep_n=100000)
18
19     bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]

```

```
20 tfidf = models.TfidfModel(bow_corpus)
21 corpus_tfidf = tfidf[bow_corpus]
22
23 lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=top,
id2word=dictionary, passes=5, workers=8)
24
25 print('USING METHOD ONE')
26
27 for idx, topic in lda_model.print_topics(-1):
28     print('Topic: {} \nWords: {}'.format(idx, topic))
29
30 lda_model_tfidf = gensim.models.LdaMulticore(corpus_tfidf,
num_topics=top, id2word=dictionary, passes=2, workers=8)
31
32 print('\n\nUSING METHOD 2')
33 for idx, topic in lda_model_tfidf.print_topics(-1):
34     print('Topic: {} Word: {}'.format(idx, topic))
35
36 # Visualization part
37 print('\n\n\nNow creating visualizations for topic amount : ', top)
38
39 import pyLDAvis.gensim
40 import gensim
41 if not 'vis' in os.listdir():
42     os.makedirs('vis')
43
44 pyLDAvis.enable_notebook()
45 data = pyLDAvis.gensim.prepare(lda_model, bow_corpus, dictionary)
46 pyLDAvis.save_html(data, 'vis/lda_graph_M1_for_topic_n_{}.html'.
format(top))
47 data = pyLDAvis.gensim.prepare(lda_model_tfidf, corpus_tfidf,
dictionary)
48 pyLDAvis.save_html(data, 'vis/lda_graph_M2_for_topic_n_{}.html'.
format(top))
```

Listing 5.8: Implementation of Gensim LDA

A bag of words corpus was created using the gensim method doc2bow, which was implemented in a term frequency inverse document frequency (TF-IDF) model from the gensim

library, used to generate the LDAMulticore model. PyLDAvis was used for topic model visualization, which provided the following visualization 5.12. The generated findings from the topic modelling algorithm although offering immense value from a visualization standpoint, arguably fail to generate value on their own, i.e. without interpretation or context-specific knowledge. This is observed in academic literature as lack of topic coherence, which is a limitation usually faced when processing microblogs, otherwise – texts that are shorter (e.g. paragraphs), as the LDA algorithm fails to extract relational data from the text in such instances [106]. Another reason why this has occurred is because the data still contains noise, even following the data cleaning procedures applied, which can impact the model's performance [107].

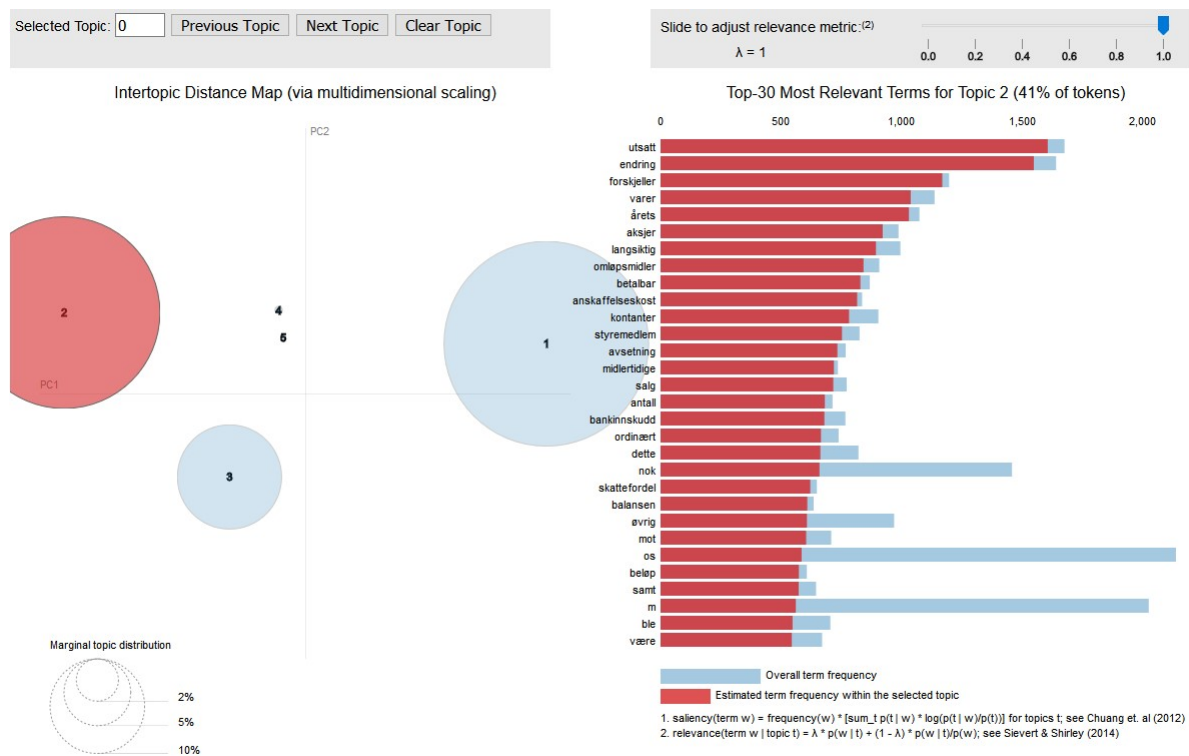


Figure 5.12: LDA visualisation

Wordcloud and frequency distribution graphs were created using nltk, with the following images generated (see figures 5.13, 5.14, 5.15). The created graphs are valuable from a visualisation standpoint [108], however, as they are based on the term frequency of the category collection, they fail to provide insight on specific reports.

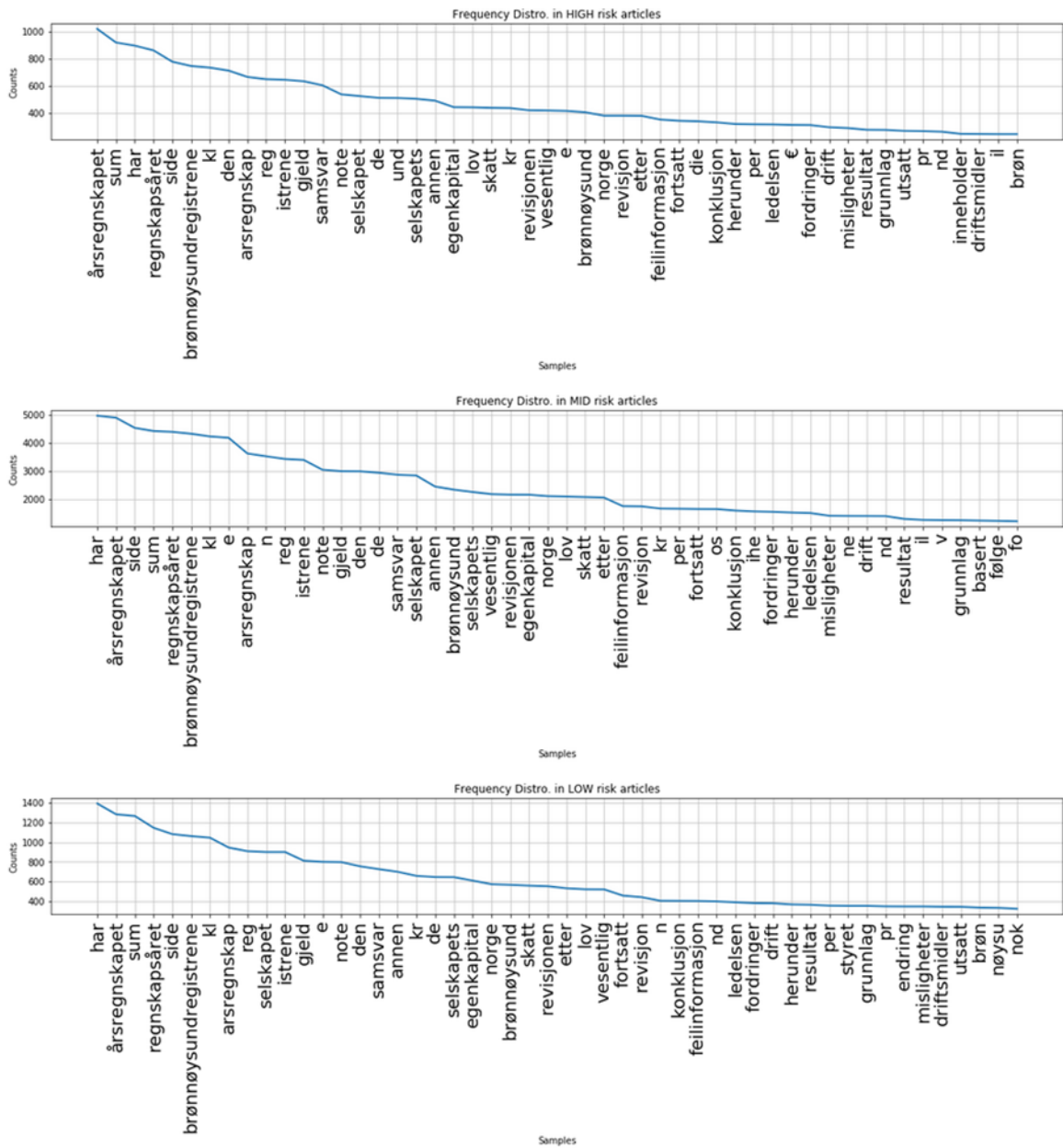


Figure 5.16: Term Frequency Distribution

For TF-IDF visualisation, in 2D vector space, the TfidfVectorizer was used, which uses for following equation to perform calculations:

$$tf(t, d) = \frac{\text{number of occurrences of term in document}}{\text{total number of all words in document}} \quad (5.1)$$

The visualization of the 2-dimensional mapping is shown in Figure 5.17. As we can see in the mapping, there is no conclusive results that groups either of the risk categories into clusters. The high risk classification have the least amount reports that are by themselves and out of bigger clusters.

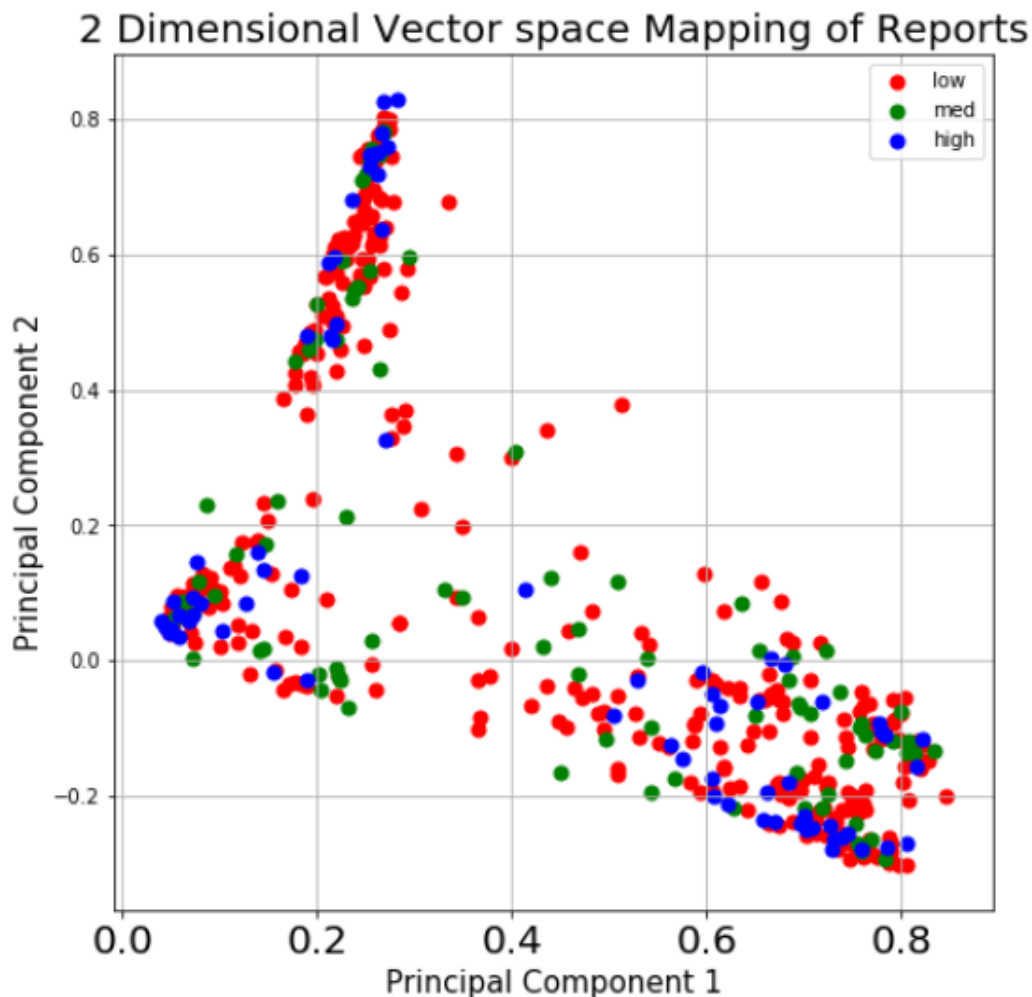


Figure 5.17: TF-IDF 2-D vector space mapping of reports

5.3.3 Wireframes

Using the application Adobe Xd [104], the wireframes design was altered a little by getting a menu on the left side of the window. A purposed way to traverse the application was thought of, as indicated by the arrows on figure 5.18.

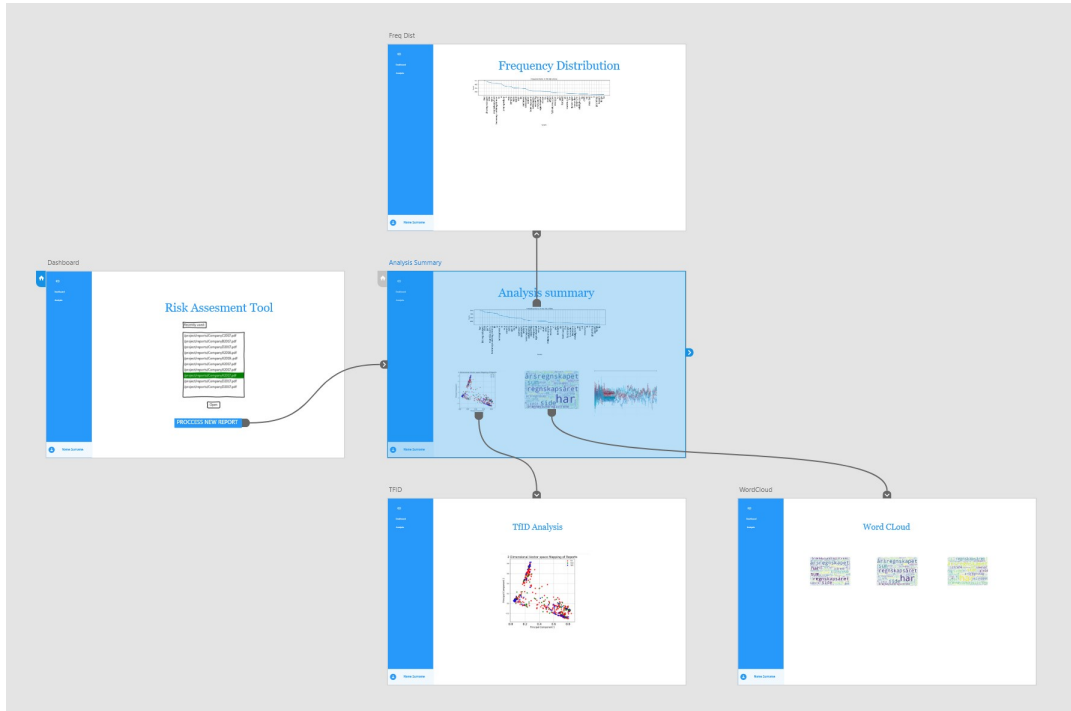


Figure 5.18: Overview of medium fidelity wireframes

5.3.4 Summary

Although a variety of visualization methods and tools were utilized as part of the analysis, no tool could provide definitive results of the risk factor of a given report. This demonstrates that the process of risk assessment cannot be fully automated using the NLP-ML technologies available or tested as part of this research, unless more cleaning of data and other methods had been tried. The process did nonetheless demonstrate the potential of implementing these technologies for assisting the decision-making of actuators. Work was also done to take the wireframes from low-fidelity to medium-fidelity.

5.4 Fourth Iteration

The web application planning was done originally using a paper-based approach as seen in the second iteration, which was later translated in a digital format in the third iteration. In the last iteration the wireframes are now made using both Adobe Photoshop for creating and editing images and using Adobe Xd to make the wireframes interactive.

5.4.1 Input from Bank

Some changes were made since the previous iteration, and the expert interview also revealed some useful insight of things to include. One of the changes that were made, was an option for users to log in. So that sensitive data can be more secure. A more streamlined UI, with the menu on the left side. Useful information on the start screen, which can include a guide for how the user can operate the system. The application requires few steps for the user to reach the analyzed data that they want to see. After loading up a new report with "ADD A REPORT" the application processes the data, and shows it in the "Dashboard" screen. Here the user have access to the processed data, and can further click in to get a more detailed overview of that particular result.

Other visualisation that were included were words close together in vector space, the most commonly used positive and negative words and a risk assessment grid for where the reports would be placed in regards to their perceived risk factor.

5.4.2 Interactive Wireframes

In figure 5.19 we see the proposed home screen of the application, where the user will be greeted with information about the tool, and a button to add a new report. There is also a panel on the right side with access to previously viewed reports.

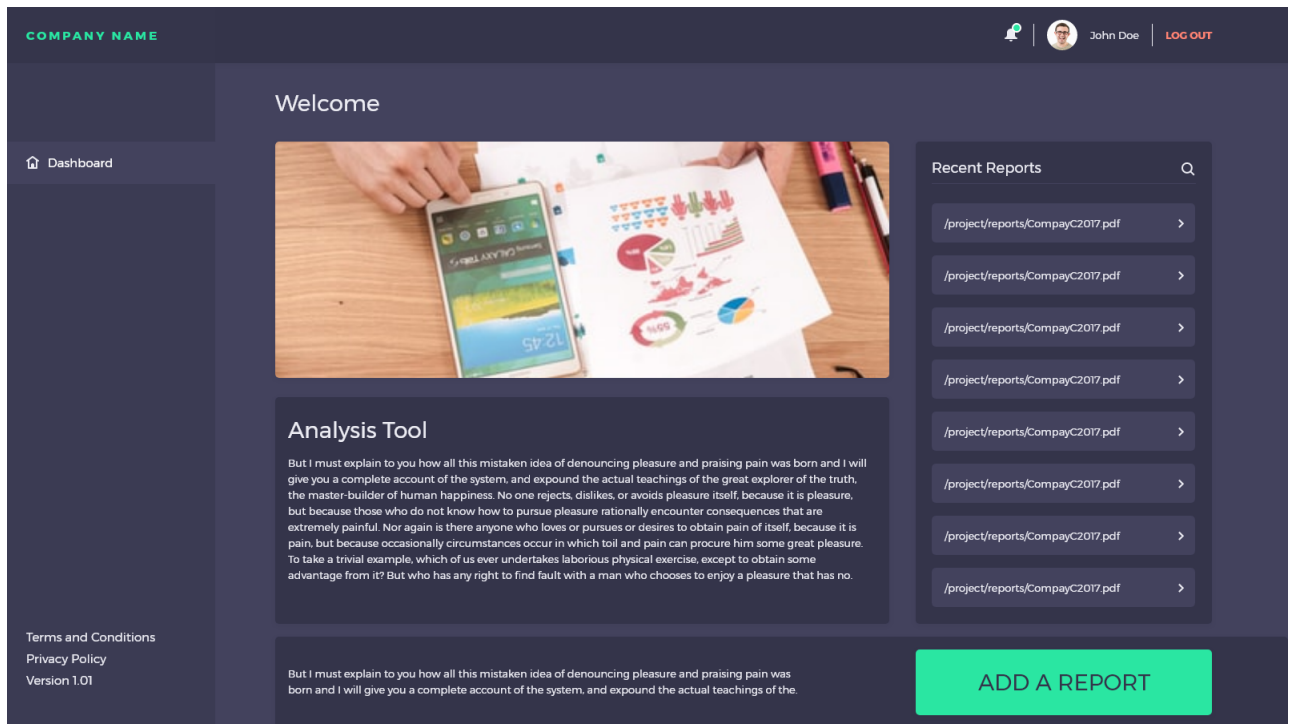


Figure 5.19: Landingpage wireframe

In figure 5.20 we see the dashboard screen. After a report is loaded, processed and analyzed the results come up here. Here we have some different components.

- Frequency Distribution
- Vector space TFIDF and Word vectors
- Risk Assesment
- Proccesed text
- Most commonly used positive and negative words
- Sentiment Analysis (Which could be done in a further iteration)
- Read Report - Option to read the full PDF of the report

The rest of the wireframes are included in Appendix B.

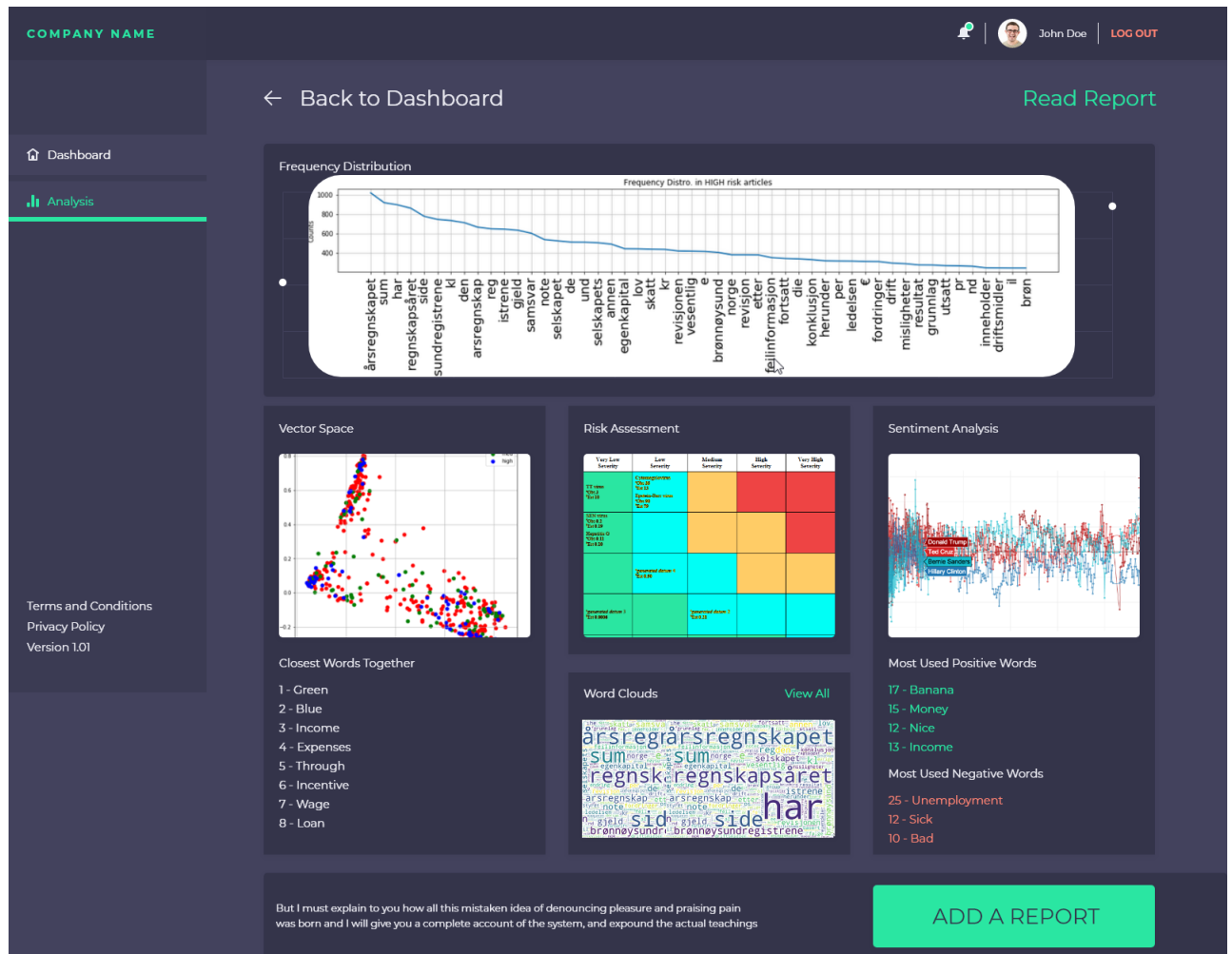


Figure 5.20: Dashboard wireframe

5.4.3 Summary

In the fourth iteration interactive wireframes of the proposed application was made. They take the user through the various section of the application and they can view and analyze data about the report.

Chapter 6

Prototype Evaluation

This chapter describes the evaluation of the developed application prototype. The testers consist of three domain expert and one IT expert, where they tested the application with a subsequent semi-structured interview. During the testing, the testers were observed and the observations were logged. The interactive prototype came with pre-loaded data and visualization, as that process is not fully implemented so it cannot process data inside the application.

The testing environment for the prototype was a laptop with Windows 10 and 1920x1080 resolution.

6.1 Observation

During the testing of the prototype notes were taken during the observation. The testers navigated the application and explored the application components. Navigation to the home menu from the dashboard, and then adding a new report was the only thing that one tester had trouble with.

6.2 Semi-structured Interviews

For the testing of the prototype, semi-structured interviews of the testers were conducted after the testing of the prototype, and the questions can be found in Appendix C.

The testers thought the design of the UI was very pleasing, and that it was looking like a tool that they could use in their work. One of the testers said that it was annoying to have to log-in, but if this only had to be done on first use and after a log-out, it would not have a negative impact on the experience of the application. One tester said that it could be annoying to have an introduction about the program on the landing page, since that is the first screen they see every time they use the application. This information could be placed in a "help" or "about" menu. Overall the color scheme and design was pleasing, although one tester mentioned that it was a little too dark, but the contrast made it okay.

It was very good that there was an option to access the recent reports, instead of having to reload previous reports with the same procedure as new reports. It was mentioned that there was no need to have directory names included in that list, but instead have the name of the company that the report was linked to or just the name of the report. Two of the testers agreed that the left side panel could contain more components and not just the Dashboard and Analysis menu options.

All the experts agreed that the dashboard looked visually pleasing, and the information displayed did not seem overwhelming. The testers liked that risk assessment was placed in the centre of the screen. It was interesting to see how the words and reports were clustered together, which the experts deemed valuable. All of the experts found positive and negative words to be very interesting, and that sentiment analysis is something they would explore more. Testers mentioned that the word cloud was a good feature. The option to read a report inside the application was considered a good solution, since otherwise they would have to find it in their file explorer.

6.3 IT Expert Interview

The IT expert had a good first impression of the design of the prototype, and said that it was up to industry standard. He questioned how this would be responsive on other devices than a laptop. The user login was pointed out to be a good feature, as it is important to have some sort of security in any application that handles sensitive data. When looking at the analyzed screen, the visualisation was said to be pleasing to look at, but should have contained more information of value.

Chapter 7

Discussion

The findings from the analysis demonstrate that it was feasible to implement procedures and automatize analysis of the risk factor categories in individual reports used in banking for insurance purposes assuming that the report data was correct. The prototype evaluation demonstrated that the application this artifact produced, was understandable, useful and had a potential to provide business value to the company if the components of the proposed system were deemed to be an improvement of the currently used algorithms and better procedures for noise cleaning of the text data prior to analysis are implemented.

The following chapter will provide a general discussion, and each of the research questions posed at the start of the thesis will be answered on the basis of the findings from the completed project. Several aspects of the overall research project will be discussed, namely the suitability of design science for this type of research, the efficiency of design prototyping for solving the problem at hand and the result of the testing methodology.

7.1 Answering the Research Questions

In Chapter 2, three research questions were introduced, namely:

- **RQ1:** *How can we use fiscal yearly reports to assess the risk of a company by analysing the natural language in it?*
- **RQ2:** *Does the fact that fiscal yearly reports do not have a standard structure affect data extraction and natural language analysis?*

- **RQ3:** *How can we visually present the result of a report analysis to the end user?*

These research questions will now be addressed considering the results from the completed project.

RQ1 has been addressed through various discussions in Chapter 3 (Theory), where recent developments in the field of NLP were presented and discussed in relation to developing an application for the analysis of annual financial reports. Specifically, developments in sentiment analysis, topic modelling and extraction of text were discussed taking into account both the textual aspects of such reports, as well as the extraction of insight from other report components such as images and infographics. As demonstrated by the subsequent system development, fiscal reports can be used for risk analysis when data is extracted, cleaned and processed for machine learning. However, the artifact in this research project extracts data from a small data set of reports, and the extracted text contains noise. In this stage of research it is hard to say if that is a consequence the limited sample data set or whether the cleaning of the text needs further improvement.

Regarding RQ2, the lack of a standardised structure of reports did not appear to affect data extraction and natural language analysis. Nonetheless, it could be emphasised that the results demonstrated a need for improving the data cleaning procedures prior to implementing the system for automated analysis. Specifically, it can be argued that the implementation of a more robust architecture for pattern recognition, given the report structure has a potential to substantially improve the natural language analysis.

RQ3 has been answered through evaluation of the prototype, where data extracted from the text and then visualized was shown to the three field experts and one IT expert. The test users approved of the application in regards to the components and UI, and agreed that it would provide valuable information with accurate data. The IT expert has found the application to be up to the standard of banking applications.

7.2 Semi-structured Interviews

Semi-structured interviews were used in the first iteration to set the initial system requirements. This was important to do early in the the development process, to understand the field of research and what directions to take. Interviews were also used for the evaluation of

the fourth iteration of the prototype. This provided the researcher with valuable knowledge and feedback about the developed application, what the future research should be focused on.

7.3 Prototype Development

As detailed in Chapter 5, Section 5.1 presented the first iteration of the development, where the initial requirements for the research project were established. Work was also done on gathering fiscal yearly reports from Brønnøysundregistrene. This was initially a very manual task, but was later rectified through a semi-automated process where the URLs were gathered and downloaded automatically.

The second iteration involved the work of text extraction, with guidance from DNB as to what important key words to look for in the text. Some low-fidelity mock-ups of an application were drawn on paper and using the application called Wiresketcher, as described in Section 5.2.

In Section 5.3, the third iteration involved cleaning the extracted data and visualizing it using LDAvis, wordcloud, TF-IDF 2d vector space representation and frequency distribution graphs. The results contained some noise, and could have benefited from more cleaning in the future iterations as discussed in Section 7.4 and mentioned in Section 8.2.

Finally, in the fourth iteration the interactive prototype was developed and can be found in Section 5.4. This work was done using Adobe Xd and Photoshop, taking into consideration the previous iterations and the IT-expert interview. The prototype was tested by experts and the results of that evaluation can be found in Chapter 6. The conclusion was that the prototype was feasible and could implement some of the initial requirements and the evaluation showed perceived usefulness of the prototype. However to get the prototype to the level of the product, robust procedures for data cleaning must be implemented in addition to carrying out a comprehensive sentiment analysis on a larger set of reports and not just a sample size.

7.4 Evaluation

The benefits of evaluation with intended users of the developed software ensured that the system was usable and user-friendly. Demonstrating the prototype throughout the iterations also provided feedback that would improve the prototype in subsequent steps.

In the observation of the evaluation sessions, and the semi-structured interviews that followed the evaluation subjects all expressed a positive attitude towards the application, but mentioned that more informative data resulting from it would give it more value. The evaluation subjects experienced a few minor problems with operating the prototype, and hoped for further iterations. The IT expert mentioned the security aspect, and noted that it would be a beneficial feature to include.

In regards to evaluation of the prototype, having an expert panel can help identifying various crucial usability problems, at low cost of time. In further iterations, lab experiments and field user studies should be considered to further improve the evaluation in regards to usability [109]. However, this was not needed at this stage of research.

7.5 Design Science Research

During this research project the Design Science research methodology has been used by following the guidelines [88]:

Design as an artifact which means that the research developed with the design science research method must produce a viable artifact in the form of a construct, model, method or instantiation [88]. In this research the artifact is a prototype application for a risk assessment tool.

Problem relevance which means the purpose of design science research is to develop solutions to solve important and relevant problems for organizations [88]. The proposed solution for the artifact developed was to provide DNB Liv or any other company that sells insurance, a way to help automate their risk assessment processes with automated tools.

Design evaluation says the utility, quality, and efficiency of the artifact must be rigorously demonstrated via well-executed evaluation methods [88]. By using research methods such as literature review, observation, semi-structured interviews and expert interviews the arti-

fact has been evaluated and tested.

Research contribution means that research conducted by the design science research method must provide clear and verifiable contributions in the specific areas of the developed artifacts and present clear grounding on the foundations of design and design methodologies [88]. The artifact developed for this research project has contributed to the research of NLP through analyzing and text extraction of fiscal yearly reports.

Research rigor says that research should be based on an application of rigorous methods in both the construction and the evaluation of artifacts [88]. The artifact was made through four iterations, and was tested and evaluated as a high-fidelity prototype with semi-structured interviews that included a panel of four experts.

Design as a research process means that the search for an effective artifact requires the use of means that are available to achieve the desired purpose while satisfying the laws governing the environment in which the problem is being studied [88]. This research project starts with the assumption that the work of manually reading fiscal reports can be automated, and the resulting data can be displayed visually to provide value to the task of risk assessment when setting the price for insurance being sold to companies. The research uses literature review and evaluation of a prototype to argue that this is possible.

Communication of the research says that research conducted by design science research must be presented to both an audience that is more technology-oriented and one that is more management-oriented [88]. The artifact created by this research has been presented to audience that is relevant to the domain of the scope of the research in both orientations. To meet the demand of technical oriented people the Development process has been discussed in Chapter 4 and Chapter 5. As to the management oriented audience, the evaluation in Chapter 6 is of interest.

7.6 Design Prototyping

Implementing a rapid prototyping methodology is considered an important component of the success of the system development initiative in the conditions of the current research. As demonstrated by academic research, many engineering processes commonly lack two things:

- the formal capacity to rapidly develop prototypes in the rudimentary stage of the project
- the ability to transition requirements into architectural design, evaluate the created designs and manage the product life-cycle in an efficient manner [110].

Creating a prototype system early-on and using it as a foundation for client discussions, testing and integration of client requirements helped generate client feedback early and identify, examine and mitigate risks in the project, which would have otherwise resulted in a greater expectation-reality gap. This observation matches the literature on prototype development [86]. Overall, the observed benefits of prototyping include:

- increased demonstrated usability of the developed system
- improved communication and collaboration with stakeholders and potential system users
- a comparatively leaner approach to traditional waterfall software development
- a more efficient use of time as a result of the lack of time spent developing requirements and describing the system development process using paper-based approaches.

7.7 Limitations

The research does have some limitations. The first is sentiment analysis done on the data extracted from a limited sample of available reports. This could have provided more valuable information to the end-user for their work with risk assessment. This was not prioritized at the start of the project, as there was a small amount of data to work with. When working on sentiment analysis it requires a rather large corpora of words, and in this project there was initially very few reports. Many of them also contained very similar wording. I believe that with a larger data-set, sentiment analyses would have been a good direction to further explore.

Furthermore the extracted data could have been more comprehensively pre-processed to decrease the amount of less valuable results. Some of the words that are extracted give little nuance to risk assessment.

The following conclusion chapter provides an overview of the research, which will be used as a foundation to elaborate on the possibilities for future software development and academic work stemming from this project.

Chapter 8

Conclusion

8.1 Conclusion

In Section 1.1, five goals were presented for the current research. Firstly, the research aimed to analyse the work processed DNB used in risk assessment. The analysis demonstrated the lack of efficiency of some aspects of current processes, namely the possibility of human error and the time-intensive process of manual inspection of the fiscal reports. A need for implementing a streamline process was identified. In order to achieve this, an implementation of NLP techniques was required, namely data extraction and topic modelling. The aim was to extract information from textual data in the fiscal reports used by employees for risk assessment in an automated manner that could also provide a deeper insight into the data. The use of the LDA topic modelling algorithm for this project has been affirmed through academic research. This approach was selected as the most appropriate purpose to answer the research question. The resulting prototyping has also confirmed feasibility of the application that it is easy to use and capable of visualizing topic modelling in a web-based software (i.e. LDAvis).

Initial sentiment classification was done on the limited sample of reports which did not yield actionable results but rather showed some deficiencies, namely language ambiguity of text which could be a hindrance for pattern recognition in the text.

Using a design science approach to knowledge generation and an agile development methodology, a prototype solution has been developed with following main functionalities such as

text extraction and visualisation of topic modelling, TF-IDF vectorization, wordcloud generation and frequency distribution of which were fully functional as separate components.

The prototype was feasible to meet some of the initial requirements and tested in a session with three domain experts and one IT expert. Evaluation results are showing perceived usefulness, relevance to the intended application, understandability, practicality and the ability to produce some relevant results. Feedback from the experts will be used in the next development iteration. Furthermore, the development need to be done within the organisation that will use the application. Lifting the prototype to product needs to include data cleaning, adding additional reports and carrying out a proper sentiment analysis.

Although feedback has been recorded, and the prototype has been improved within the final stages of development, it has not been implemented in the organisation, which is consistent with academic literature on software prototyping, where it is stated that many developments fail to reach implementation in a practical setting [111].

8.2 Future Work

There are two directions of development that can secure improvement of the current prototype.

From a software development standpoint, a full implementation and integration of the prototype system can be made. In order for this to be achieved further customization and personalization of the system's components and UI is required to meet user needs. Additionally, it is recommended that such an initiative would take into consideration the current software that can be improved and tested before releasing it as a product to replace an existing process.

Sentiment classification should receive much attention, given its important for the domain application. The system can be dramatically improved if a more advanced classification approach is implemented, for example utilising existing semantic lexicons or presenting comparative results from word-, document- and sentence-level sentiment analysis.

From an academic research standpoint, it is considered that the system can be improved through a comparative analysis of a variety of techniques. For example, while the LDA topic

modelling algorithm has provided a good foundation for developing the prototype system, there are many scholars that believe that other methodologies, such as Latent Semantic Analysis (LSA) yield better performance [112], [113]. Future work can thus focus on comparing the performance of LDA and LSA topic modelling algorithms for fiscal report analysis. This can be done using the current research as a starting point, utilising aspects of the work such as text extraction procedures, data cleaning and preparation for analysis, as well as the results obtained from the LDA analysis from the data-set. It is considered that such a comparative approach to research can be insightful for academics, as well as business executives, as previous work on LSA demonstrates that it is more suitable for building simple and scalable models from large corpora quickly and efficiently [114]. Therefore, it can be argued that it might offer organizations a stronger foundation (from a performance standpoint) for making decisions to implement such a technology in their organizations.

References

- [1] Ahmad Fathan Hidayatullah and Muhammad Maarif. Road traffic topic modeling on twitter using latent dirichlet allocation. pages 47–52, November 2017.
- [2] Philipp Offermann, Olga Levina, Marten Schönherr, and Udo Bub. Outline of a design science research process. page 7. ACM, January 2009.
- [3] Mychajlo Lobur, Andriy Romanyuk, and Mariana Romanyshyn. Using nltk for educational and scientific purposes. *11th international conference the experience of designing and application of CAD systems in microelectronics (CADSM) 2011 Feb 23. IEEE*, pages 426–428, November 2011.
- [4] Asra Khalid and Sobia Zahra. Suitability and contribution of agile methods in mobile software development. *International Journal of Modern Education and Computer Science*, 6(1):56–62, February 2014.
- [5] Guowen Li, Xiaoqian Zhu, Jun Wang, Dengsheng Wu, and Jianping Li. Using lda model to quantify and visualize textual financial stability report. *Procedia Computer Science*, 122:370–376, January 2017.
- [6] Pushpak Bhattacharyya. Natural language processing: A perspective from computation in presence of ambiguity, resource constraint and multilinguality. pages 1(2):1–3, November 2019.
- [7] Saturnino Luz. Machine learning for nlp: Supervised learning techniques. November 2019.
- [8] Najah Attig, Sadok El Ghoul, Omrane Guedhami, and Jungwon Suh. Corporate social responsibility and credit ratings. *Journal of Business Ethics*, 117:(4):679–94, February 2013.

- [9] Jennifer Altamuro and Anne Beatty. How does internal control regulation affect financial reporting? *Journal of Accounting and Economics*, 49(1):58–74, 2010.
- [10] Daron Acemoglu and Pascual Restrepo. Artificial intelligence, automation and work. *SSRN Electronic Journal*, January 2018.
- [11] Irmgard Nübler. *New Technologies, Innovation, and the Future of Jobs*, pages 46–75. October 2018.
- [12] Prakash Nadkarni, Lucila Ohno-Machado, and Wendy Chapman. Natural language processing: An introduction. *Journal of the American Medical Informatics Association : JAMIA*, 18:544–51, September 2011.
- [13] Alan Ritter, Colin Cherry, and William Dolan. Data-driven response generation in social media. pages 583–593, January 2011.
- [14] Michael Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales, and marketing support*. April 2004.
- [15] Fengyu Wang, Kathleen Carley, Daniel Dajun Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83, April 2007.
- [16] Chandra Amaravadi, Subhashish Samaddar, and Siddhartha Dutta. Intelligent marketing information systems: Computerized intelligence for marketing decision making. *Marketing Intelligence & Planning*, 13(2):4–13, March 1995.
- [17] Hsiu-chin Chen, Roger Chiang, and Veda Storey. Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36:1165–1188, December 2012.
- [18] Kazuo Nakatani and Ta-Tao Chuang. A web analytics tool selection method: an analytical hierarchy process approach. *Internet Research*, 21(2):176–86, January 2011.
- [19] Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat, and Priyanka Badhani. Study of twitter sentiment analysis using machine learning algorithms on python. *International Journal of Computer Applications*, 165(9):29–34, May 2017.
- [20] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *CoRR*, cs.CL/0205028, July 2002.

- [21] Norwegian wordnet. *Kaldera språkteknologi AS*, February 2016. <https://www.nb.no/sprakbanken/>.
- [22] Robert. Nltk vs. spacy: Natural language processing in python, 2016. <https://blog.thedataincubator.com/2016/04/nltk-vs-spacy-natural-language-processing-in-python/>.
- [23] Michal Ptaszynski, Fumito Masui, and Pawel Lempa. A modular system for support of experiments in text classification. *Technical Transactions*, July:229–243, July 2015.
- [24] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. <http://is.muni.cz/publication/884893/en>.
- [25] Radim Rehurek. Scalability of semantic analysis in natural language processing. 2011.
- [26] Jin-Young Min, Sung-Hee Song, HyeJin Kim, and Kyoung-Bok Min. Mining hidden knowledge about illegal compensation for occupational injury: Topic model approach (preprint). 05 2019.
- [27] Timothy Fogarty. Financial accounting standard setting as an institutionalized action field: Constraints, opportunities and dilemmas. *Journal of Accounting and Public Policy*, 11(4):331–355, December 1992.
- [28] Michael Ettredge, Vernon Richardson, and Susan Scholz. The presentation of financial information at corporate web sites. *International Journal of Accounting Information Systems*, 2(3):149–168, September 2001.
- [29] Jill Collis, Andrew Holt, and Roger Hussey. Conceptual framework for financial reporting. *FASB*, (8):3, 2010.
- [30] Carson Sievert and Kenneth Shirley. Ldavis: A method for visualizing and interpreting topics. pages 63–70, June 2014.
- [31] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. pages 1833–1842. IEEE, January 2014.

- [32] inventors; International Business Machines Corp Breedvelt-Schouten IM, True JA. Generating word clouds. March 2019.
- [33] WORDCLOUD, a word cloud generator in python. https://github.com/amueller/word_cloud.
- [34] Jacob Harris. Word clouds considered harmful. October 2011.
- [35] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2013.
- [36] David Blei and John Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1:17–35, August 2007.
- [37] Tak Yeon Lee, Smith Alison, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28–42, March 2017.
- [38] Qian Chen, Xin Guo, and Hexiang Bai. Semantic-based topic detection using markov decision processes. *Neurocomputing*, 242:40–50, February 2017.
- [39] Qi Dang, Feng Gao, and Yadong Zhou. Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. *Expert Systems with Applications*, 57:285–295, April 2016.
- [40] Wenxin Wang, Yi Feng, and Wenqiang Dai. Topic analysis of online reviews for two competitive products using latent dirichlet allocation. *Electronic Commerce Research and Applications*, 29:142–156, April 2018.
- [41] Kaveh Bastani, Hamed Namavari, and Jeffrey Shaffer. Latent dirichlet allocation (lda) for topic modeling of the cfpb consumer complaints. *Expert Systems with Applications*, 127:256–271, March 2019.
- [42] Wilhelm Uys, Niek Du Preez, and E.W. Uys. Leveraging unstructured information using topic modelling. pages 955–961. IEEE, August 2008.

- [43] Chen Zhang, Hao Wang, Liangliang Cao, Wei Wang, and Fanjiang Xu. A hybrid term-term relations analysis approach for topic detection. *Knowledge-Based Systems*, 93:109–120, November 2015.
- [44] Stephan Curiskis, Barry Drake, Thomas Osborn, and Paul Kennedy. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, April 2019.
- [45] Hai Dohaiha, PWC Prasad, Angelika Maag, and Abeer Alsadoon. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Systems with Applications*, 118:272–299, October 2018.
- [46] Dongjin Yu, Dengwei Xu, Dongjing Wang, and Zhiyong Ni. Hierarchical topic modeling of twitter data for online analytical processing. *IEEE Access*, 7:12373–85, January 2019.
- [47] Jia Yu and Lirong Qiu. Ulw-dmm: An effective topic modeling method for microblog short text. *IEEE Access*, PP:884–893, December 2018.
- [48] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–35, January 2008.
- [49] AS Al-Sukkar, AH Hussein, and MM Jalil. The effect of applying artificial intelligence in shaping marketing strategies: Field study at the jordanian industrial companies. *International Journal of Applied Science and Technology, Jordan, Global Society of Scientific Research and Researchers (GSSRR)*, 3(2):1–11, January 2008.
- [50] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. *Mining Text Data*, pages 415–463, August 2013.
- [51] David Omand, Jamie Bartlett, and Carl Miller. Introducing social media intelligence (socmint). *Intelligence and National Security*, 27(6):801–823, December 2012.
- [52] Carlos Sureda Gutierrez, Paulo Figueiras, Pedro Oliveira, Ruben Costa, and Ricardo Jardim-Goncalves. *An Approach for Detecting Traffic Events Using Social Media*, volume 647, pages 61–81. Springer, Cham, June 2016.

- [53] Arman Khadjeh Nassirtoussi, Sr Aghabozorgi, Teh Wah, and David Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, November 2014.
- [54] Chen Chih-Hao, Lee Wei-Po, and Jhih-Yuan Huang. Tracking and recognising emotions in short text messages from online chatting services. *Information Processing & Management*, 54(6):1325–1344, November 2018.
- [55] Weiguo Fan and Michael Gordon. The power of social media analytics. *Communications of the ACM*, 57(6):74–81, June 2014.
- [56] Iti Chaturvedi, Erik Cambria, Roy Welsch, and Francisco Herrera. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44:65–77, December 2017.
- [57] Valentina Sintsova and Pearl Pu. Dystemo: Distant supervision method for multi-category emotion recognition in tweets. *ACM Transactions on Intelligent Systems and Technology*, 8(1):13, August 2016.
- [58] Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. Semantic clustering and convolutional neural network for short text categorization. pages 352–357, January 2015.
- [59] Nadia Felix, Luiz Coletta, Eduardo Hruschka, and Estevam Hruschka. Using unsupervised information to improve semi-supervised tweet sentiment classification. *Information Sciences*, 355:384–365, February 2016.
- [60] Claudia Diamantini, Alex Mircoli, Domenico Potena, and Emanuele Storti. Social information discovery enhanced by sentiment analysis techniques. *Future Generation Computer Systems*, 95:816–828, February 2018.
- [61] Jing Jiang. *Information Extraction from Text*, pages 11–41. Springer US, Boston, MA, 2012.
- [62] Hampp-Bahnmueller Thomas inventors; International Business Machines Corp Johnson, David E. Architecture of a framework for information extraction from natural language documents. April 2013.

- [63] Dan Smith. Information extraction for semi-structured documents. November 2003.
- [64] C.L. Tan and P.O. Ng. Text extraction using pyramid. *Pattern Recognition*, 31(1):63–72, January 1998.
- [65] Honggang Zhang, Kaili Zhao, Yi-Zhe Song, and Jun Guo. Text extraction from natural scene image: A survey. *Neurocomputing*, 122:310–323, December 2013.
- [66] Yao Pu and Zhixin Shi. A natural learning algorithm based on hough transform for text lines extraction in handwritten documents. *Advances In Handwriting Recognition*, pages 141–150, June 1999.
- [67] J.G. Walls, Joseph G., Widmeyer, George R., Omar Sawy, and Omar A. Building an information system design theory for vigilant eis. *Information Systems Research*, 3(1):36–59, March 1992.
- [68] Shirley Gregor. Building theory in the sciences of the artificial. page 4. ACM, January 2009.
- [69] Marie-Françoise Legendre. Le constructivisme: Tome 1: Des fondements, tome 2: Des epistemologies. esf. *Revue des sciences de l'éducation*, 22:197, January 1996.
- [70] Jay Jr, Minder Chen, and Titus Purdin. Systems development in information systems research. *Journal of Management Information Systems*, 7(3):89–106, January 1991.
- [71] Krsto Pandza and Richard Thorpe. Management as design, but what kind of design? an appraisal of the design science analogy for management. *British Journal of Management*, 21(1):171–186, February 2010.
- [72] B van Wyk. Research design and methods part i. *University of Western Cape*, 2012.
- [73] Aline Dresch, Daniel Lacerda, and José Jr. *Design Science—The Science of the Artificial*, pages 47–65. Springer, Cham, August 2015.
- [74] Anne Huff, D. Tranfield, and Joan van Aken. Management as a design science mindful of art and surprise: A conversation. *Journal of Management Inquiry*, 15(4):413–424, January 2006.
- [75] Simon Herbert A. The sciences of the artificial 2nd edition. 1996.

- [76] Anh Nguyen and Pekka Abrahamsson. Minimum viable product or multiple facet product? the role of mvp in software startups. pages 118–130. Springer, Cham, May 2016.
- [77] Hideaki Takeda, Tetsuo Tomiyama, Hiroyuki Yoshikawa, and Paul Veerkamp. Modeling design process. *AI Magazine*, 11:37–48, October 1990.
- [78] Salvatore March and Gerald Smith. Design and natural science research on information technology. *Decision Support Systems*, 15:251–266, December 1995.
- [79] Vijay Vaishnavi and Bill Kuechler. Design research in information systems. 2004.
- [80] Schwaber Ken. Scrum development process. *Business object design and implementation*, pages 117–134, 1997.
- [81] Tomas Gustavsson. Benefits of agile project management in a non-software development context - a literature review. pages 114–124. Latvijas Universitate, April 2016.
- [82] Uwe Flick. An introduction to qualitative research. Sage Publications Limited;, 2018.
- [83] Michael Patton. Two decades of developments in qualitative inquiry: A personal, experiential perspective. *Qualitative Social Work*, 1(3):261–283, September 2002.
- [84] Norman Denzin and Yvonna Lincoln. Handbook of qualitative research. *Denzin & Lincoln*, January 2000.
- [85] Spencer Onuh and K.K.B. Hon. Integration of rapid prototyping technology into fms for agile manufacturing. *Integrated Manufacturing Systems*, 12(3):179–186, June 2001.
- [86] Marja Kapyaho and Marjo Kauppinen. Agile requirements engineering with prototyping: A case study. pages 334–343. IEEE, August 2015.
- [87] Helen Sharp, Yvonne Rogers, and Jennifer Preece. *Interaction Design. Beyond Human-Computer Interaction*. 01 2007.
- [88] Alan Hevner and Samir Chatterjee. *Design Science Research in Information Systems*, volume 28, pages 9–22. 06 2010.
- [89] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

- [90] PyPDF2, A Pure-Python library build as a pdf toolkit. <https://pypi.org/project/PyPDF2/>.
- [91] TEXTTRACT, extract text from any document. <https://texttract.readthedocs.io/en/stable/>.
- [92] TESSERACT-OCR, extract text from images. <https://github.com/tesseract-ocr/tesseract/>.
- [93] Wes McKinney. PANDAS: a Foundational Python Library for Data Analysis and Statistics. <https://pandas.pydata.org/>.
- [94] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.
- [95] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007. <https://matplotlib.org/>.
- [96] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011. <https://scikit-learn.org/>.
- [97] PYLDAVIS, a library for topic model visualization. <https://pypi.org/project/pyLDavis/>.
- [98] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. <https://ipython.org>.
- [99] ANACONDA, distribution for python and r programming language. <https://www.anaconda.com/>.
- [100] GIT, a version-control system. <https://git-scm.com/>.
- [101] TRELLO, kanban list web application. <https://trello.com/>.
- [102] DROPBOX, file hosting service. <https://www.dropbox.com/>.
- [103] WIREFRAMESKETCHER, simple. <https://wireframesketcher.com/>.

-
- [104] ADOBE XD, program to make interactive wireframes. <https://www.adobe.com/products/xd.html>.
- [105] ADOBE PHOTOSHOP, raster graphics editor. <https://www.adobe.com/no/products/photoshop.html>.
- [106] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. pages 889–892, 07 2013.
- [107] David Newman, Edwin Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. pages 496–504, 01 2011.
- [108] Musa Jafar. Decision-making via visual analysis using the natural language toolkit and r. *Journal of Information Systems Applied Research (JISAR)*, 7:33, 02 2014.
- [109] Georgios Fiotakis, Dimitrios Raptis, and Nikolaos Avouris. Considering cost in usability evaluation of mobile applications: Who, where and when. volume 5726, pages 231–234, 08 2009.
- [110] Nitish Devadiga. Tailoring architecture centric design method with rapid prototyping. pages 924–930, 10 2017.
- [111] Dursun Delen and Haluk Demirkan. Data, information and analytics as services. *Decision Support Systems*, 55:359–363, 04 2013.
- [112] Nina Rizun, Yurii Taranenko, and Wojciech Waloszek. The algorithm of modelling and analysis of latent semantic relations: Linear algebra vs. probabilistic topic models. pages 53–68, 11 2017.
- [113] Sonia Bergamaschi and Laura Po. Comparing lda and lsa topic models for content-based movie recommendation systems. pages 247–263, 12 2015.
- [114] Gabriel Recchia and Michael Jones. More data trumps smarter algorithms: Comparing pointwise mutual information to latent semantic analysis. *Behavior research methods*, 41:647–56, 09 2009.

Appendix A

Fiscal reports

Here are two examples of how a fiscal yearly report would look like. The first is the one where a company employee writes the status quo from the year, and the other is an independent accountant.

A.1 Yearly report

A.2 Independent accountant report



Brønnøysundregistrene Årsregnskap regnskapsåret 2017 for 810859482



ÅRSBERETNING 2017

VIRKSOMHETEN

Brunstad AS driver produksjon, markedsføring og salg av stoppede møbler. Hovedkontor og produksjonsanlegg ligger i Sykkylven. Fabrikklokalene eies av Brunstad Invest AS.

FORTSATT DRIFT / FINANSIELL RISIKO

Årsregnskapet er avlagt etter forutsetning om fortsatt drift. Det er styrets oppfatning at det er godt grunnlag for fortsatt drift i Brunstad AS. Prognoser og strategiplaner er utarbeidet for de nærmeste årene, og danner grunnlaget for kontinuitet og langsiktighet. Selskapet har etter styrets vurdering en sunn økonomisk og finansiell stilling.

Da selskapet har små langsiktige forpliktelser er det liten risiko knyttet til økning i rentenivået. Selskapet importerer og eksporterer i ulike valutaer, og valutaendringer medfører derfor en viss risiko. Selskapet har de senere år hatt beskjedne tap på utestående fordringer. Alt eksportsalg skjer gjennom factoring eller mot kjeder som stiller sikkerhet for betaling.

ORGANISASJON, PERSONALE OG ARBEIDSMILJØ

Antall ansatte utgjorde ved årets slutt 88, tilsvarende 84,5 årsverk. Pr. 31.12.2017 var det ansatt 48 kvinner og 40 menn. Sykefraværet i 2017 var på 9,47 %. Langtidsfraværet utgjorde den største andelen av fraværet. Det var ingen personskade i bedriften i 2017.

Selskapet praktiserer full likestilling mellom kjønnene, og mellom utenlandske og norske medarbeidere. Vi er opptatt av at diskriminering ikke skal skje på noe område.

Arbeidsmiljøutvalget har hatt regelmessige møter gjennom året. Styret vurderer arbeidsmiljøet som godt. God oppfølging fra bedriftshelsetjeneste og stor grad av fleksibilitet i tilrettelegging av arbeidsplasser bidrar til dette.

MILJØRAPPORTERING

Etter det styret kjenner til, forekommer det ikke utslipp fra selskapets virksomhet som forurenser det ytre miljø. Bedriften har konsesjon til utslipp fra forbrenningsanlegg. Det arbeides systematisk med å redusere omfang av avfall og for å gjennomføre tiltak som bidrar til økt resirkulering av papir og plast. Avfallet leveres til godkjent deponi. Selskapet er medlem i Grønt Punkt Norge AS. Selskapets energiforbruk til oppvarming og belysning, samt forbruk av vann er under løpende oppfølging.

BRUNSTAD AS
Sykkylvsvegen 415, N-6230 Sykkylven
T +47 70 24 60 00

BANK 6559 05 04658
VAT NO 810 859 482 MVA
brunstad.no

Appendix B

Wireframes

Appendices of wireframes from low-, medium- and high-fidelity.

B.1 Low Fidelity

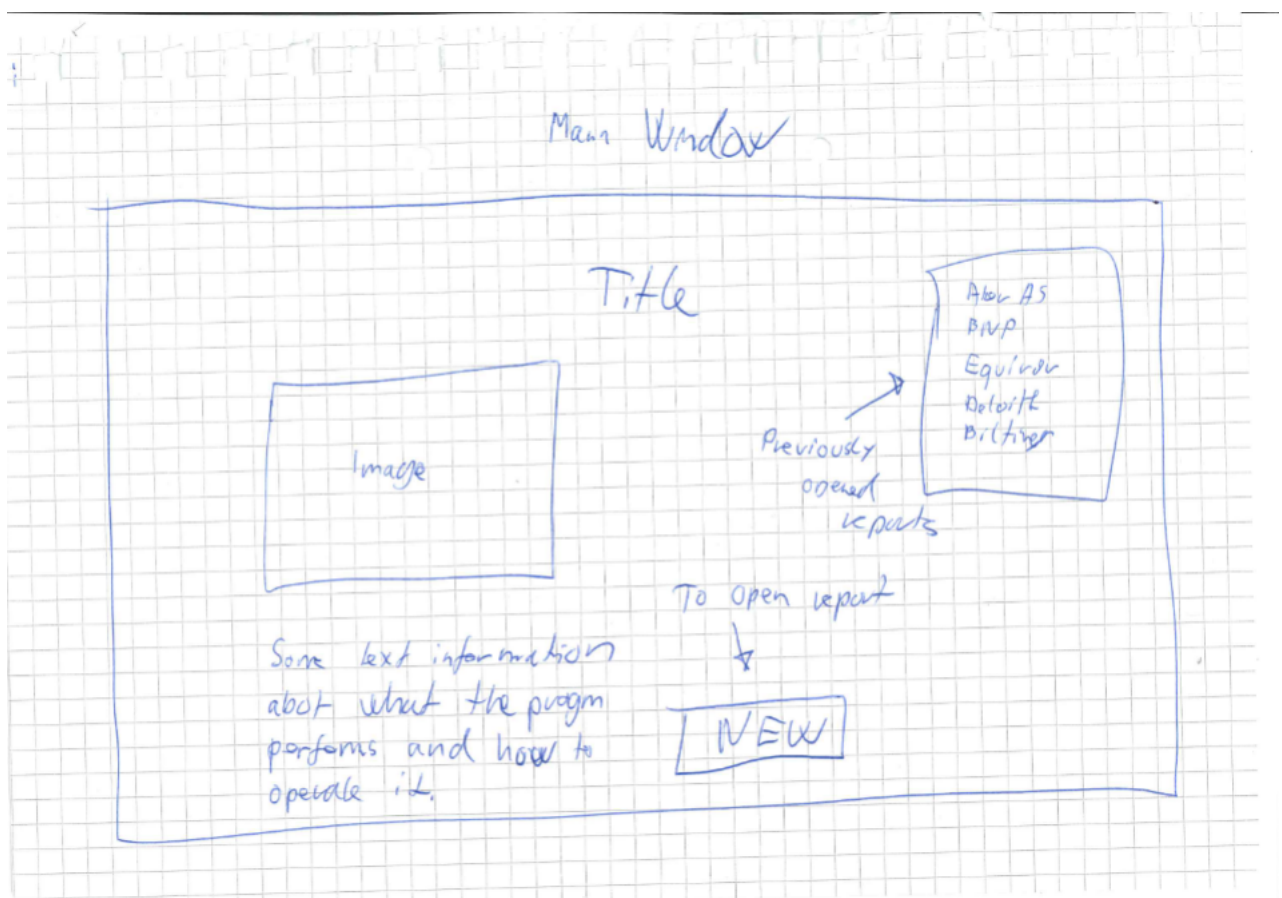


Figure B.1: Main window

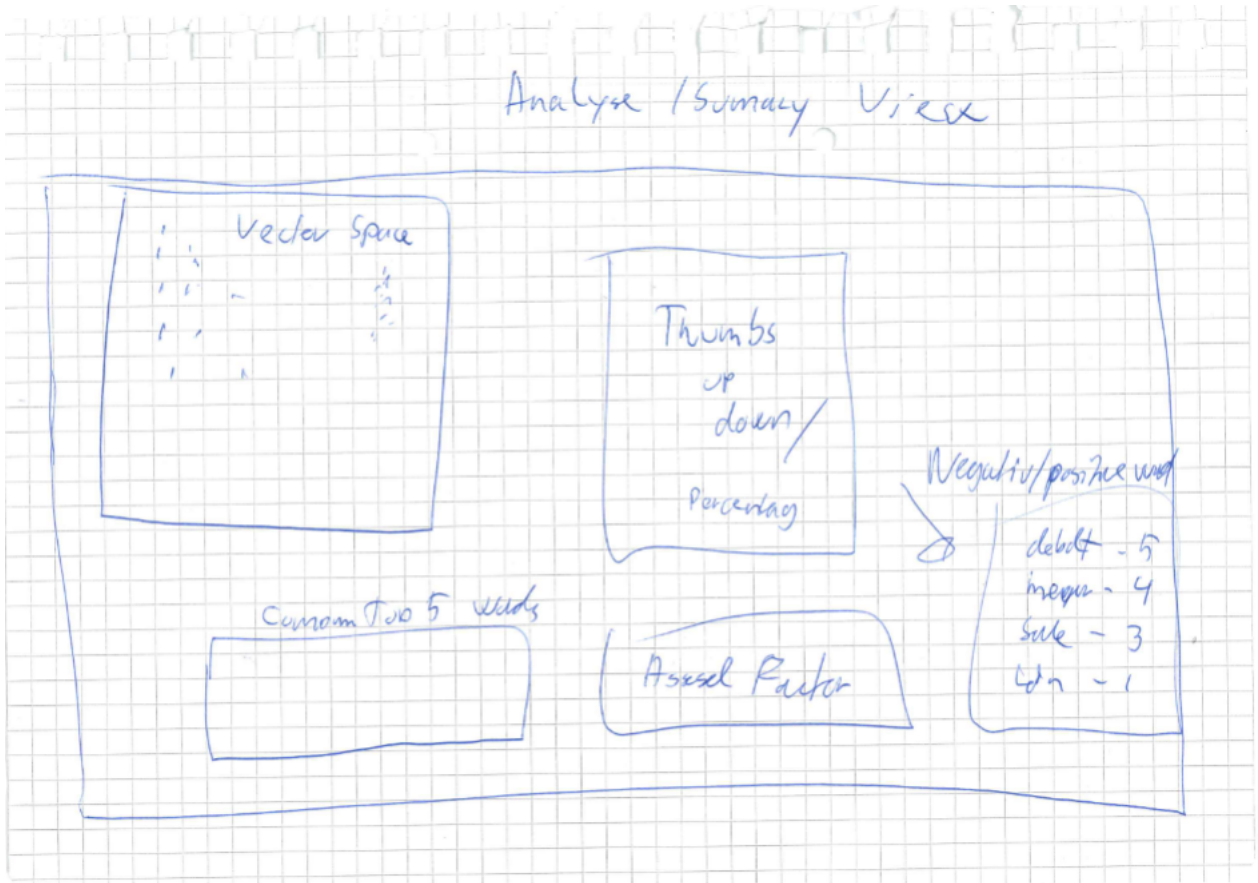


Figure B.2: Summary of analyzes

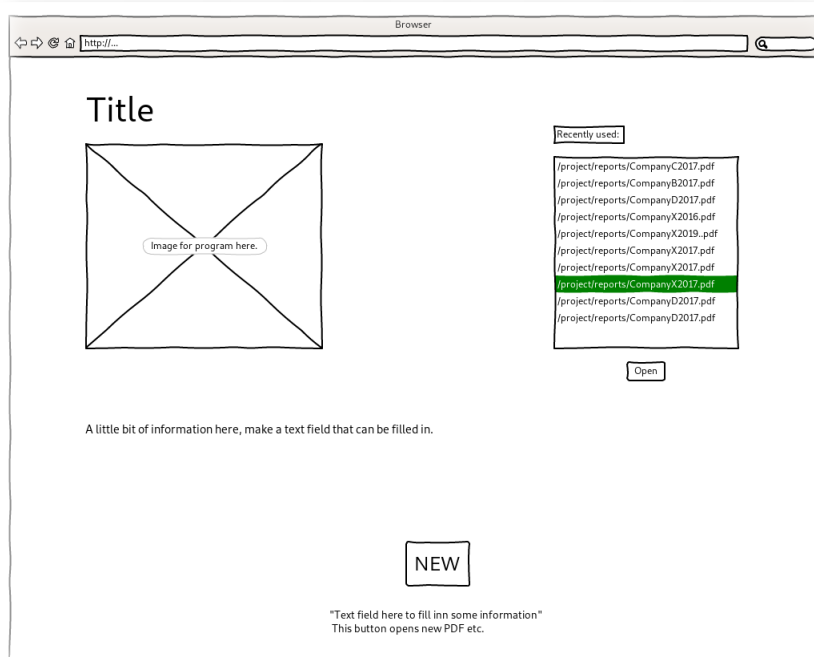


Figure B.3: Lo-fi Main Window made in Wiresketcher

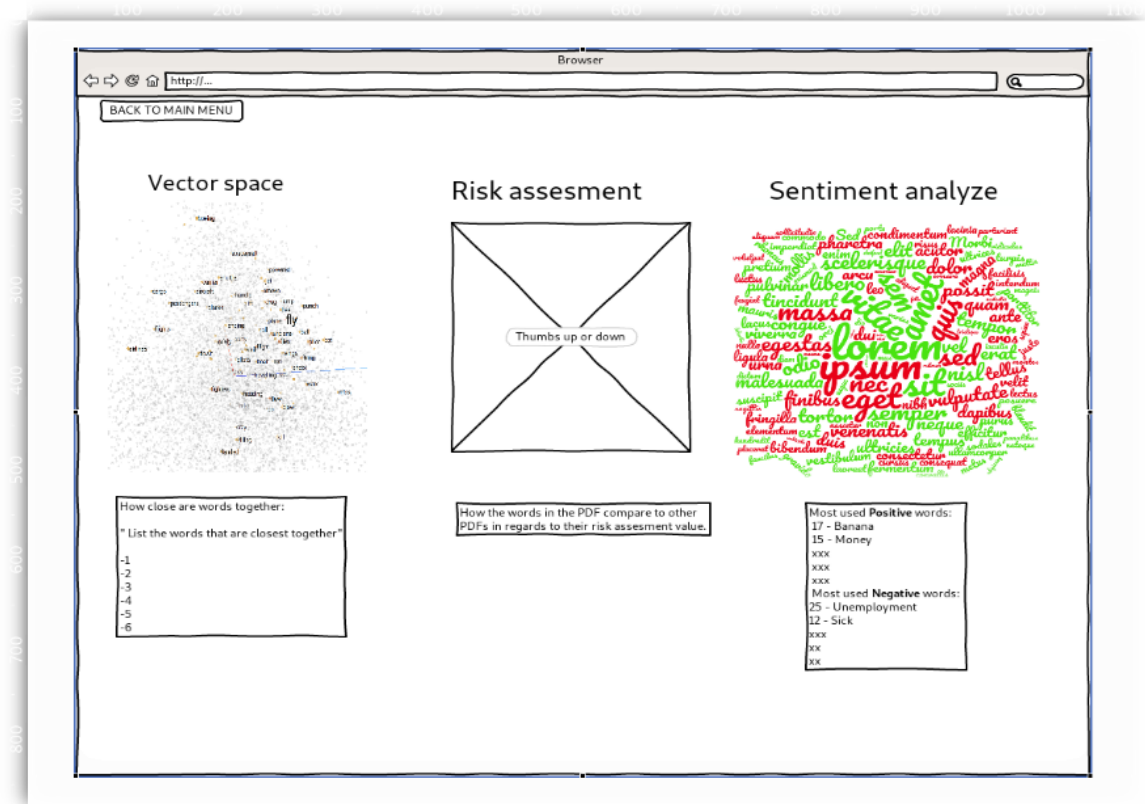


Figure B.4: Lo-fi Summary of Analyzes made in Wireshketcher

B.2 Medium Fidelity



Figure B.5: Dashboard Iteration 1

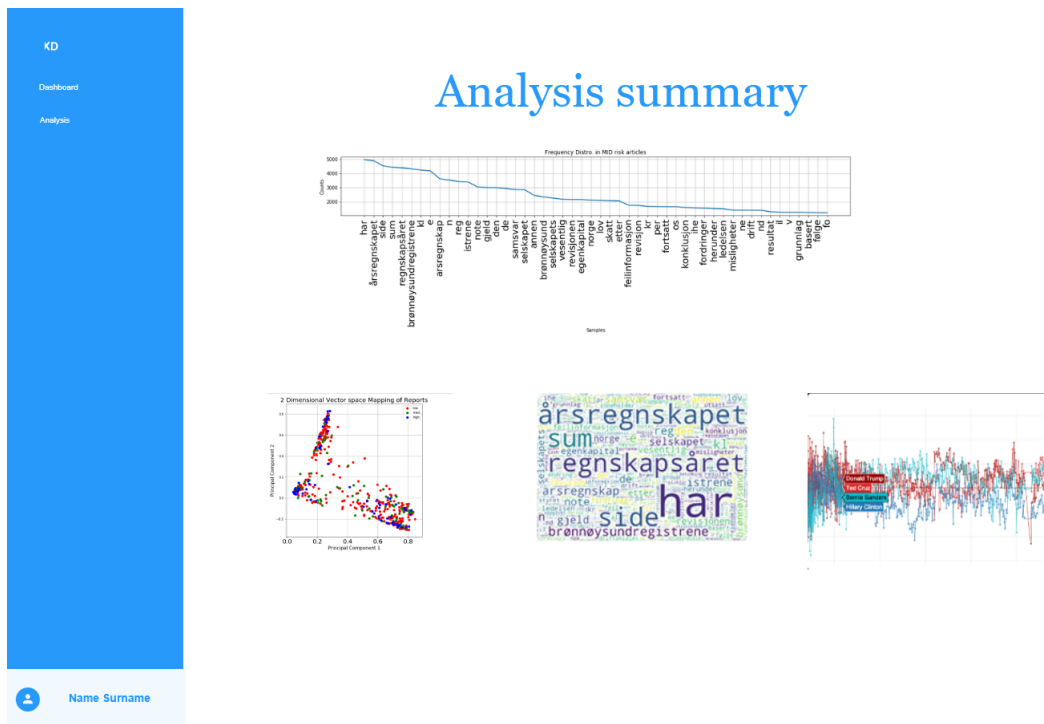


Figure B.6: Summary Iteration 1

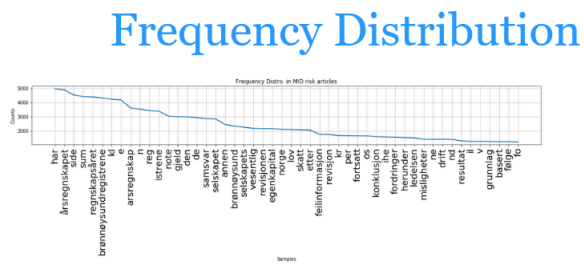
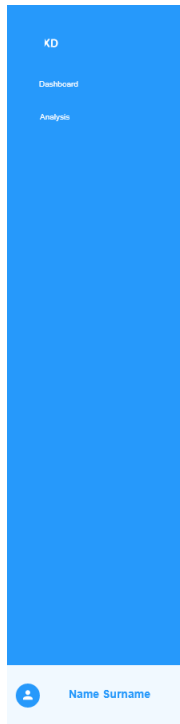


Figure B.7: Frequency Distribution Iteration 1

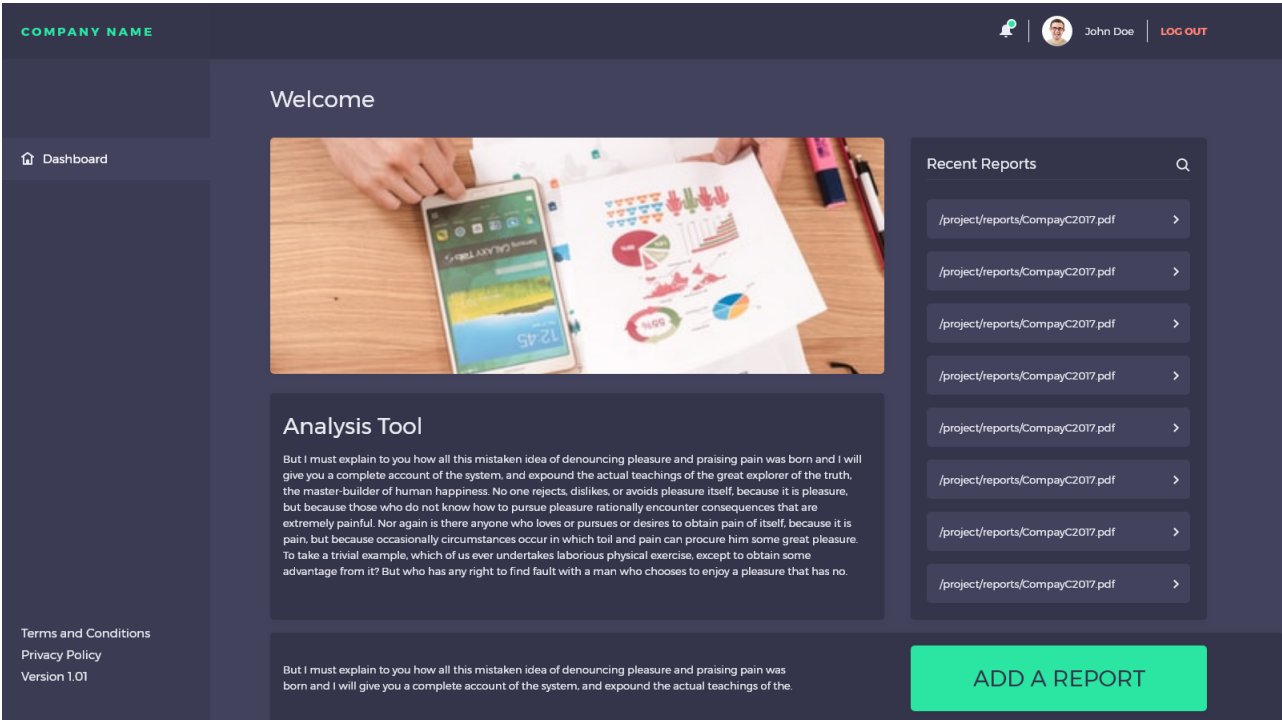


Figure B.8: Landing screen Iteration 2

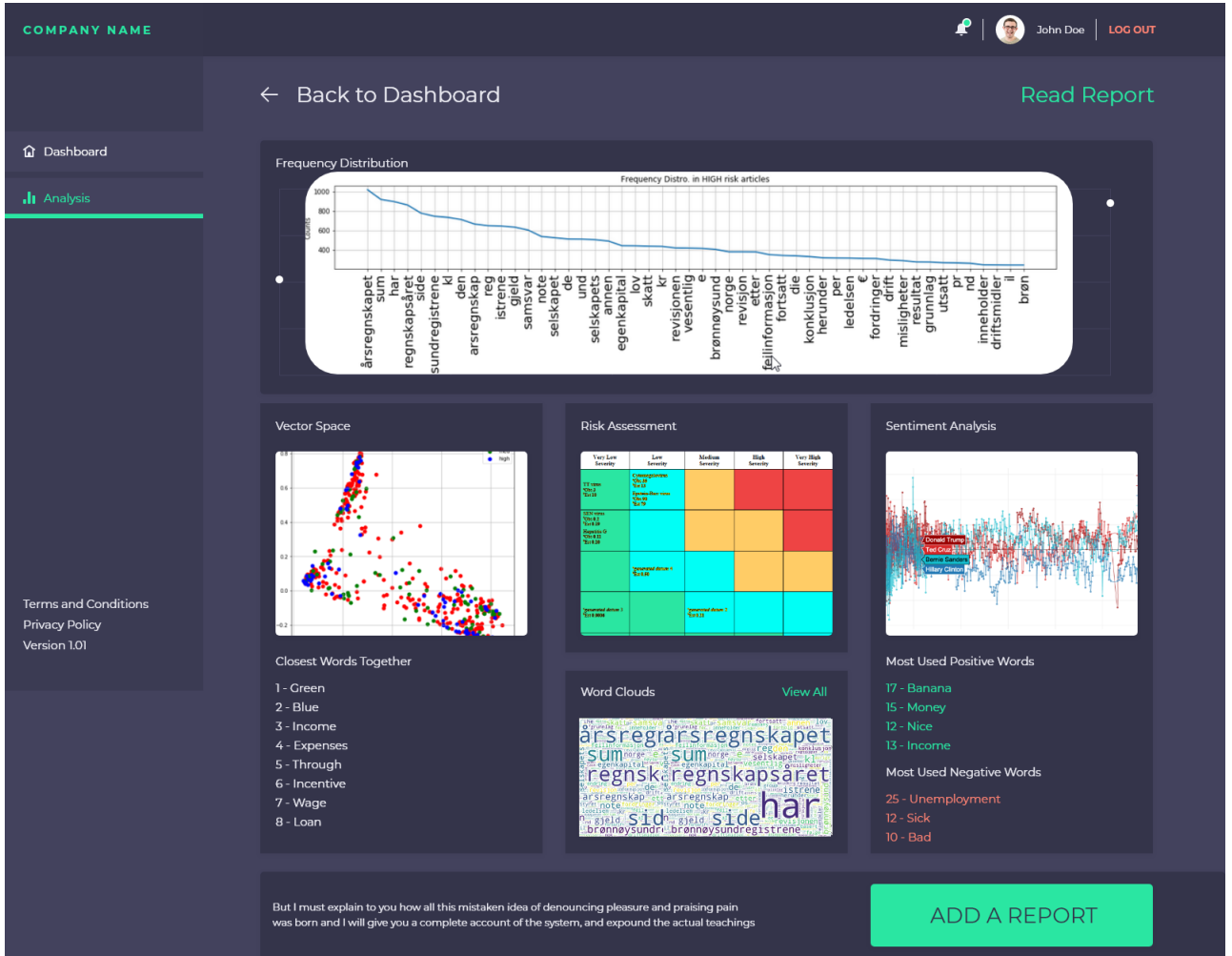


Figure B.9: Dashboard Iteration 2

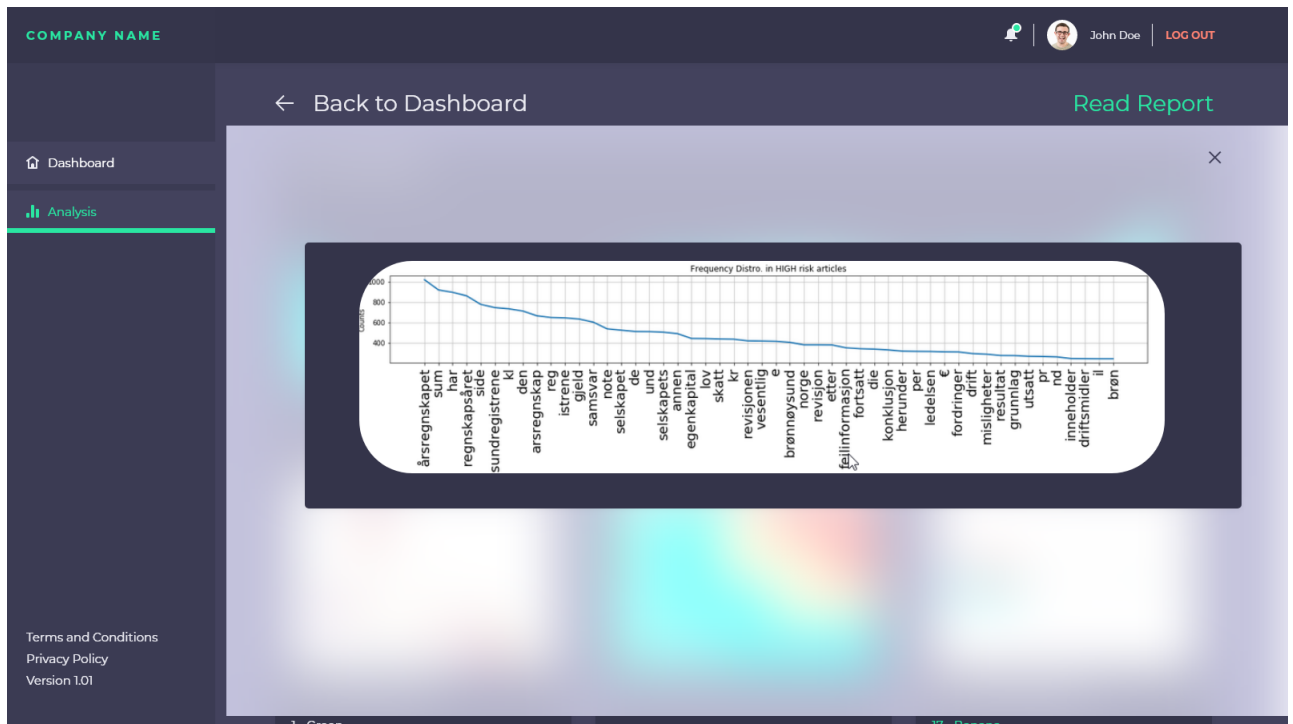
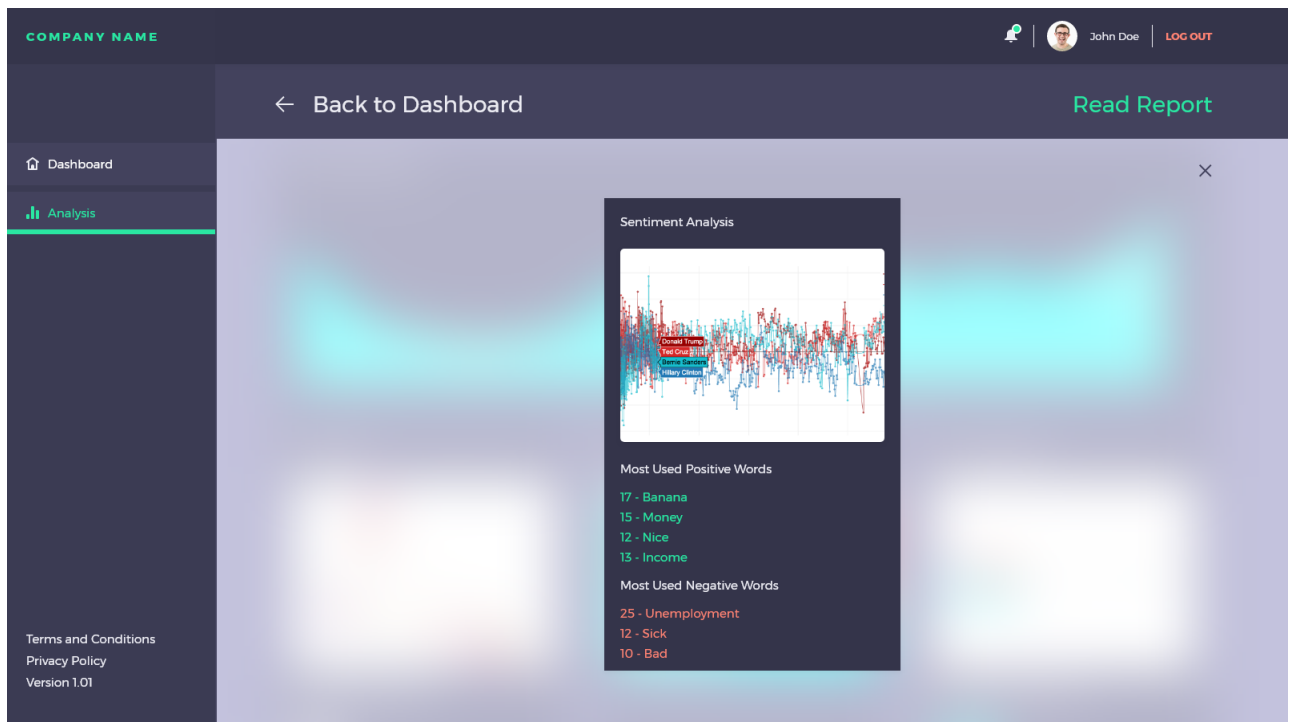


Figure B.10: Frequency Distribution Iteration 2



Figure B.11: Risk Assessment Iteration 2



Sentiment Iteration 2



Frequency Distribution Iteration 2

Appendix C

Semi-structured interview

C.1 Initial requirements

1. Why do you analyze fiscal yearly reports?
2. How do you analyze fiscal yearly reports?
3. How do you feel the current process of analyzing a report is?
4. What improvements to the current process of report analyzing would you like to see?
5. How would you like them to be displayed to you visually?

C.2 Prototype testing

1. What is your first impression of the application?
2. What are some of the features you like or dislike?
3. Is the user interface easy to understand?
4. Is the data presented valuable to you in your work?
4. What do you think of the different selections of visualisation of data?
5. Is this an application you could consider using as a work tool?

C.3 Evaluation questions

UI - UX

How was the use of the system in regards to the UI?

Is the design relevant for your work?

What do you think about having to log in?

Data and Visualisation

Was the data relevant?

How can this contribute to your daily work?

What do you think about the data of:

- Vectoring
- Word clouds
- Frequency Distribution
- Positive/Negative Words
- Risk assessment

Is it interesting to be able to read the full report in the application?

Appendix D

Visualization

D.1 Frequency distribution

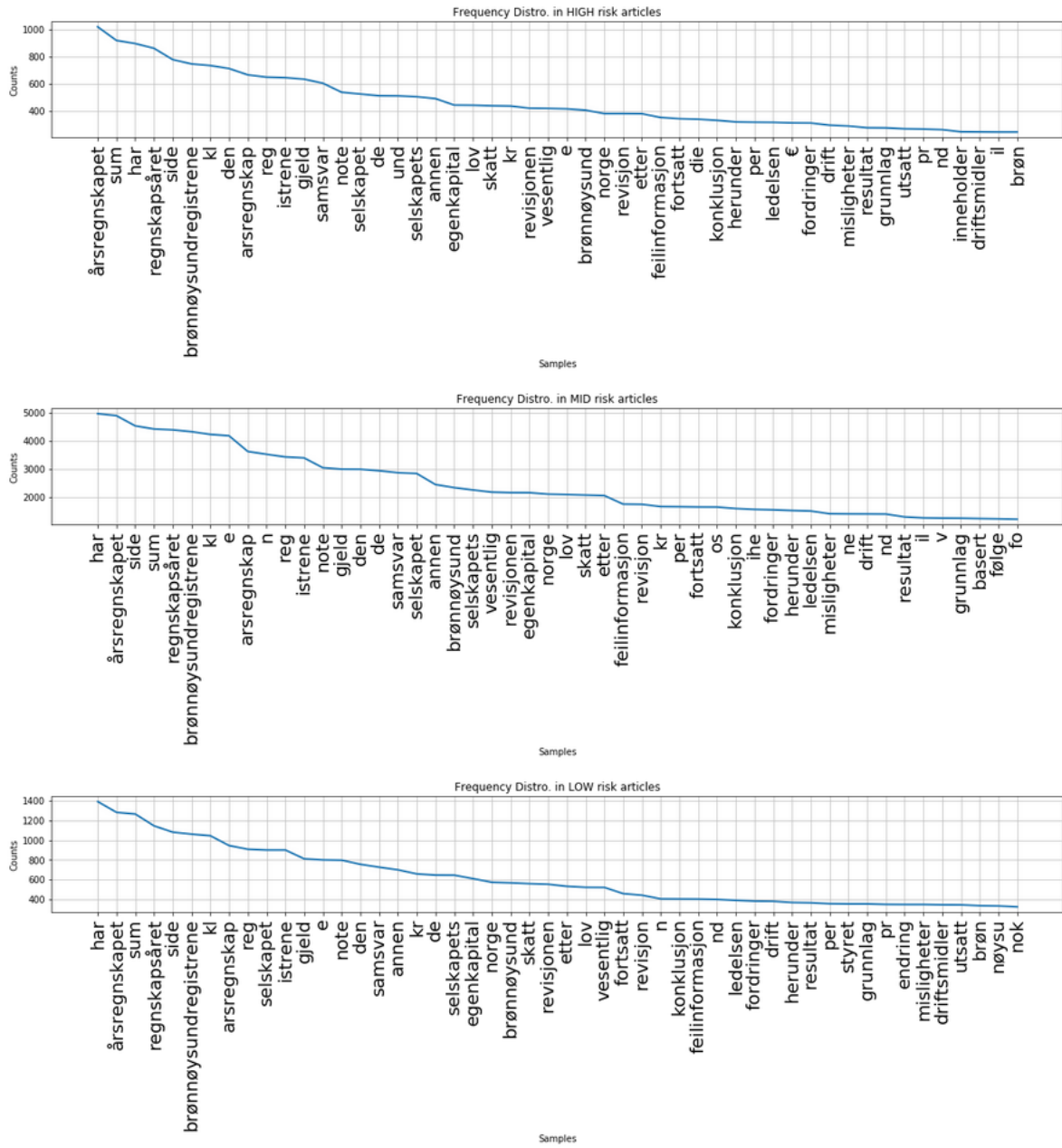


Figure D.1: Frequency distribution, initial reports

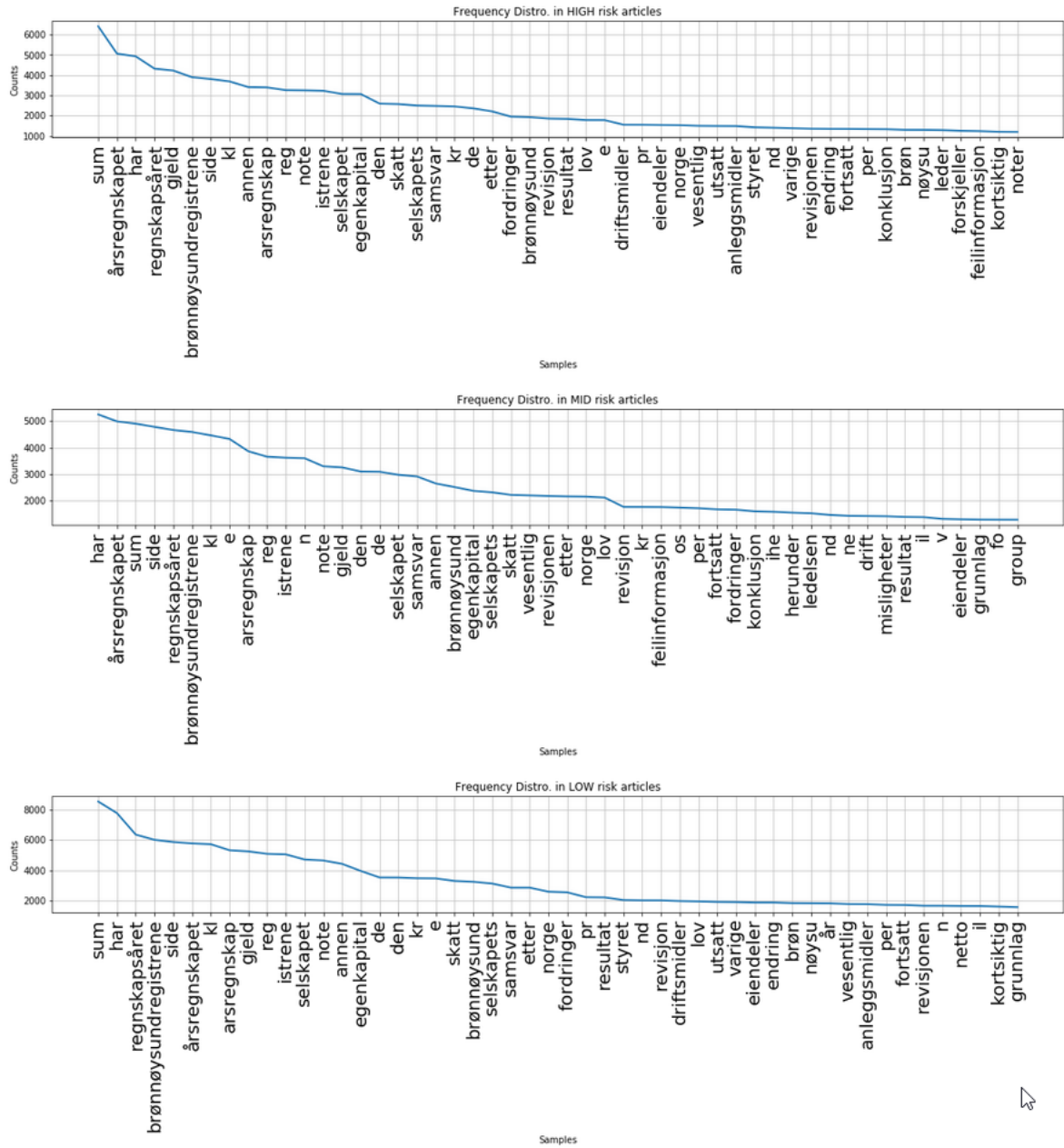


Figure D.2: Frequency distribution, all reports

D.2 TF-IDF

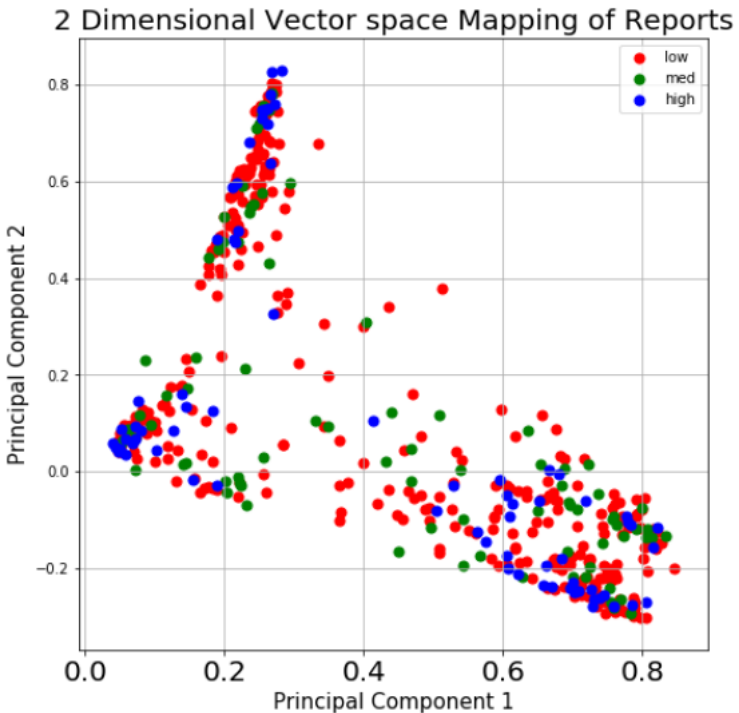


Figure D.3: TF-IDF, initial reports

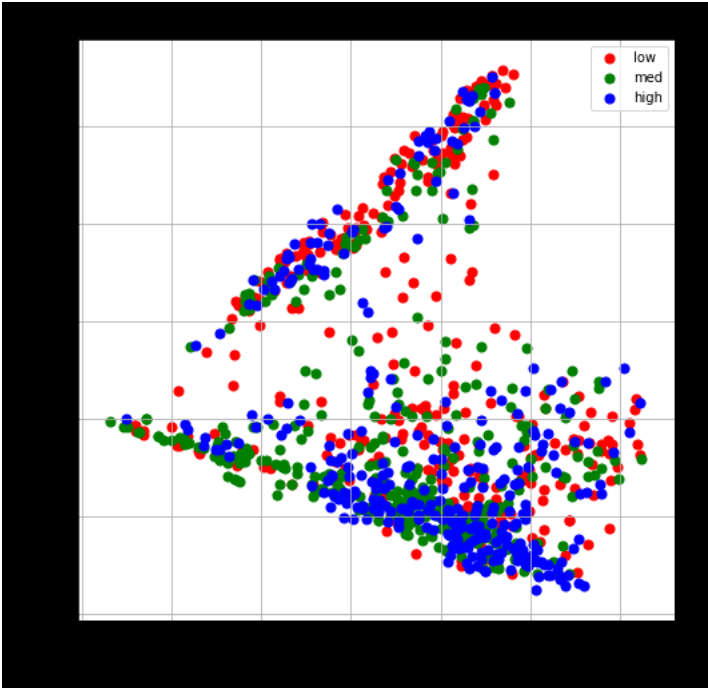


Figure D.4: TF-IDF, all reports

D.3 LDA

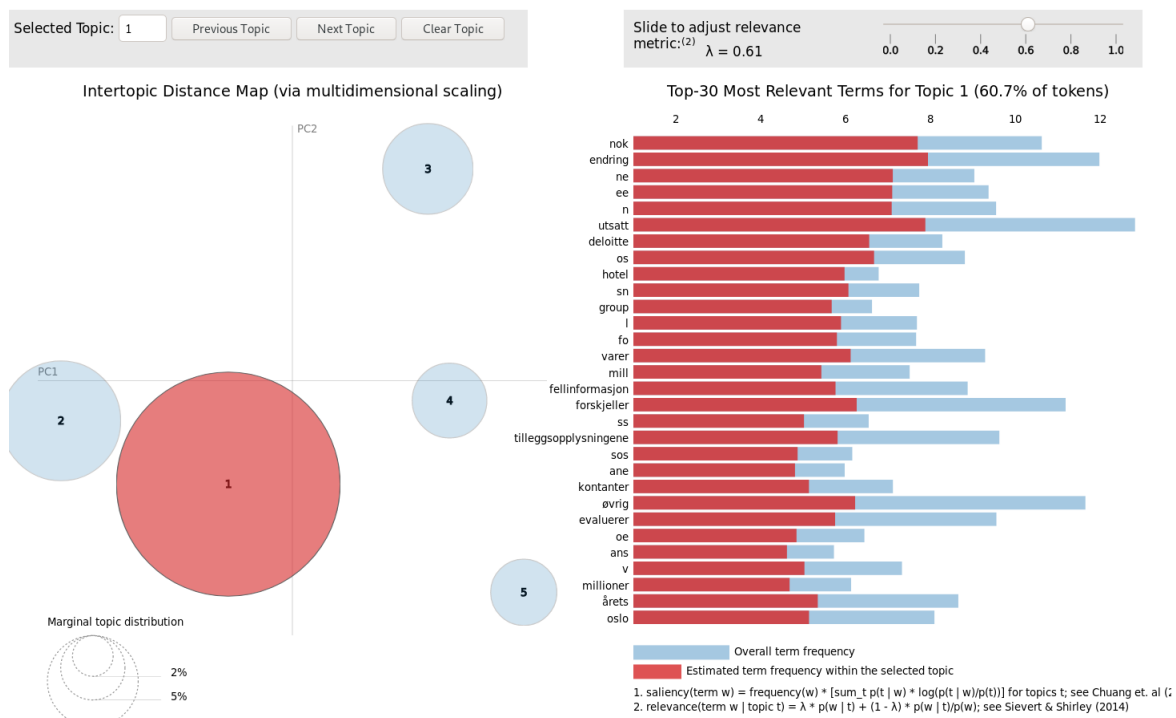


Figure D.5: LDA, all reports

Appendix E

Source code

The code for the master project will be released as open source on GitHub in January 2020.

The source code can be found on this web-page: <https://github.com/mtenmann/MasterFinal>