Radboud University Nijmegen

Faculty of Social Sciences

Auditing Algorithms

A working example on auditing algorithms

Thesis BSc Artificial Intelligence

Author:
Lotte Willems

Supervisor:
prof. Tom Heskes

Second reader:
Dr. Max Hinne

July 10, 2020

Abstract

Artificial Intelligence (AI) is becoming a bigger part of the human society everyday. Intelligent algorithms are making decisions that can be of great impact on individuals. Therefore, there is a raising concern of possible harmful consequences of decisions that negatively impact minorities. This thesis investigates whether the SMACTR framework for internal audits [33] can be applied on intelligent systems to detect possible harmful consequences in an early stage of model development. In order to do so, a case study is conducted to set out a working example to contribute to the field of auditing algorithms. The internal audit is performed on the Dutch RobBERT model for natural language processing (NLP), fine-tuned to perform sentiment analysis [11]. This thesis suggests that internal audits has the potential to become part of the software development process to help the field towards more transparency of intelligent algorithms and early detection of harmful behaviour.

Contents

Chapter 1

Introduction

With the rise of Artificial Intelligence (AI), organisations are deploying AI systems for their own benefit. In every possible sector, AI systems are making crucial decisions that can have a major impact on individuals' lives. Furthermore, it is becoming more clear that a lot of AI systems contain biases that might be disadvantageous for individuals who are dependent on the system [9, 32]. However, the algorithms behind these decision making systems are often too complex for humans to understand and to be transparent about the decision making process. The growing discontent with the inability to explain decisions made by AI systems, leads to the emerging field of explainable AI, which aims to give more insight and make these "black-boxes" more transparent [1].

Mittlestadt [29] argues that performing audits on algorithms contributes to achieving transparency of algorithms. In auditing, algorithms will be investigated on functionality, potential failures or biases, and impact of those failures and biases. Currently, there are only a few frameworks to provide guidance for an audit and even less examples of these frameworks being actually applied on algorithms. For this thesis, the internal auditing framework SMACTR by Raji et al. [33] will guide an audit to set out a working example of such an audit. SMACTR is short for the five stages of this auditing framework: scoping, mapping, artifact collection, testing and reflection [33]. The audit will be performed on a fine-tuned version of RobBERT (Robustly optimized BERT approach), a Dutch language model for natural language processing. For this thesis, the model was fine-tuned following the tutorial and notebook by Delobelle et al. [11] on a Dutch Book Reviews Dataset by van der Burgh [37].

1.1 Theoretical Background

1.1.1 Why Auditing Artificial Intelligent Algorithms?

In recent years, algorithms have increasingly been dominating daily life. Decision making by algorithms is more common than ever, taking place in critical sectors such as the job market, criminal sentencing, and the medical sector [21]. These decision making algorithms are often deployed without checking them for harmful biases. This can be disadvantageous for minorities for which the systems are biased. For example, ProPublica [3] investigated COMPAS, a risk assessment tool for juries in the United States to determine recidivism of criminal defendants. ProPublica found that COMPAS is biased to give African-Americans a higher score label whilst white defendants were given a low score label. This is obviously problematic, as skin colour should not be a feature to determine chance of recidivism on. However, the problem is anchored in the data. Statistically, it might be true that people with a darker skin color are more often recommitting a crime. After all, these statistics are a consequence of political and cultural issues. For example, jurors are often biased to give higher punishment to people of a non-white skin color [28, 16]. This in turn causes that there are more records of recidivism of black people which leads to skewed data, causing biases in the systems trained on this data [26].

However, it is up to developers to exclude the dubious features from the data and the model, such that the algorithm does not rely on those features and thus is fairer. By including auditing in the development process, developers can detect and mitigate this unethical feature reliance.

Another development of the critical use of algorithms it that algorithms are more commonly used for content personalization systems [29]. These systems display information that fit individual users based on past behaviour. For example, algorithms that suggest us what Netflix movie we might like to watch next, what t-shirt we might want to buy, or even what political party to vote for – e.g. Cambridge Analytica [7]. These systems and their algorithms create a new type of media which can have a negative impact on society [29]. The so-called "echo chambers" are linked to the sudden and fast political developments of e.g. Brexit [14]. In absence of any awareness of these problems, companies and governments might not act in the interest of the public, which is not a good development.

Finally, algorithms are often defined as "black boxes", meaning the way algorithms reach their outputs and decisions are hard to understand from solely looking at the code or the parameters of the machine learning model.

By incorporating auditing in the development process of algorithms, internal audits hopefully contribute to more transparency, close the accountability gap, and detect harmful biases and consequences before model deployment [33].

1.1.2 Definition of an Audit

In practice, auditing is used as a tool for interrogation when there is suspicion of companies or organisations violating policies, industry standards or governmental regulations [33]. Auditing is already used in many fields such as finance, engineering and pharmaceutics as a form of quality control or to detect fraud.

Difference Auditing Software and AI

A difference has to be made between auditing of "normal" software and intelligent algorithms. In software development, developers know the behaviour of the algorithm they are developing upfront. In most cases, intended behaviour of the software can also be determined from the code after model deployment. Furthermore, the software will not show biased behavior for different users of these systems.

Auditing intelligent algorithms on the other hand, is somewhat less straightforward.

Here, it is not always clear upfront how the software will behave. The behaviour of AI systems relies heavily on the data that is used to train the systems and behaviour might differ for different users of AI systems. It is not immediately obvious and hard to interpret from the code why certain decisions are made by the algorithm [29]. An example that illustrates this uncertainty about decisions is image recognition, where the algorithm uses classification to determine what type of object is in a picture. After changing one pixel in the input the algorithm can output an entirely different object classification [30]. It is not clear why the change in one pixel leads to a different classification label. The inability to predict and interpret the behavior of the code behind intelligent systems makes the auditing process different from auditing software.

Internal or External Audit

In auditing algorithms, a difference can be made between an external audit and an internal audit. An external audit aims to identify harmful risks from outside the system and serves as an accountability measure for models that are being deployed already. These kinds of audits can for example be conducted by applying feature relevance methods or methods that follow from the field of Explainable Artificial Intelligence (XAI) [1], with methods for global and local interpretability. Feature relevance methods measure the dependence of the model on the features it is trained on by making small adjustments to these features [18]. If the output dramatically changes by minor adjustments, the model might be highly sensitive to this specific feature [2].

Another approach to externally check for biases in an existing system is for example by conducting a benchmark test. Here, a representative data set is fed to the algorithm to see if the performance is equal for all subgroups of the benchmark set [34]. However, using the external approach, little can be said about accountability of faulty behaviours and why the system makes certain decisions. Furthermore, the biases are only detected after model release and deployment, thus the system may already have negatively impacted its users.

An internal audit, on the other hand, is performed during the development of the system, to check that ethical expectations and standards are met throughout the process of development. In this way, biases and other harmful consequences can be detected upfront, before causing any harm [33]. For this thesis, an internal audit will be performed to set up a working example to see if the SMACTR framework can be applied to intelligent systems and whether this allows to detect possible biases or harmful consequences before model deployment [33].

Auditing Procedures

Algorithmic auditing involves collecting and analyzing outcomes of an algorithm or model during development or after deployment. Auditing can be performed in various ways, for example by creating simulated dataset that represents balanced user data, a benchmark dataset. Feeding this mock user data to the algorithm can discover differ-

ences in performance between groups [34].
In research conducted by Buolamwini and Gebru in the Gender Shades project [8], they performed an audit by using a benchmark dataset on three commercial facial analysis systems. By testing the systems, they found that darker-skinned females form the most misclassified group with an error rate of up to 34.7%, whilst the least misclassified group consists of lighter-skinned males with a maximum error rate of 0.8%. These results point to a possible bias towards gender and/or race in these systems. Possibly, this could have been prevented by adding more examples of darker-skinned females to the dataset where the model is trained on.
Research by Bolukbasi et al. [6], found that models that learn word embedding, such as Word2Vec, also encode gender biases. For their paper, the authors used Word2Vec to train a model that fills in the masked words in analogies such as “man is to computer programmer, as woman is to X”. In this particular example, “X” was completed with “homemaker”, conforming gender stereotypes. This result is concerning, as Word2Vec is an embedding that is used a lot nowadays. This means that this gender bias is also propagated throughout systems that use this word embedding [6].
The biases found above are both found by researchers performing tests on models that are already being deployed. This is problematic, as these models already are impacting peoples’ lives. Furthermore, the biases are found by researchers that just ran tests on the models by feeding newly selected data. This is of course helpful, as it indeed shows that there are biases. Nonetheless, the problem is that there is no generic and standardized way to perform an audit as of today even though research shows that it is essential that more emphasis should be on detecting harmful behaviour as soon as possible. This can possibly be mitigated by including internal audits in the development process.

1.2 Research Question

The aim of this thesis is to investigate the possibility of a general approach in auditing algorithms. Therefore, in this thesis a case study will be performed, guided by the following research question:

“To what extent can the SMACTR framework for internal audits be applied on Artificial Intelligent (AI) systems?”

The case study will set out a working example of an internal audit. The audit will closely follow the SMACTR framework [33], which is a intensively defined internal auditing frameworks for AI systems. Setting up a working example of an internal audit can be of great relevance of the field, since auditing algorithms is still abstract and poorly defined. Also, there are not many examples currently available of performed audits. Most companies are hesitant to participate voluntarily in an audit as audits can for example expose their trade secrets that give them advantages over other companies [4]. Thus, performing an audit in a research environment can contribute to the field.

Additionally, this thesis aims to show that internal audits are a relevant addition to the development process of intelligent systems.
In the case study, the pre-trained RobBERT model will be fine-tuned on the Dutch Book Reviews Dataset to perform sentiment analysis, following the notebook and tutorial by Delobelle et al. [11, 10, 23, 37].

Chapter 2

The Auditing Framework

This section will outline the components of the framework that will guide the internal audit performed on the RobBERT model. The audit will closely follow the SMACTR framework as defined in the paper by Raji et al. [33]. The framework defined in their paper originally consists of five stages: Scoping, Mapping, Artifact Collection, Testing, and Reflection. For the aim of this thesis all stages will be included in the audit. However, some components of the stages will be discarded from this audit, as they are less applicable for this specific model.

As stated in their paper [33], each stage results in a section of the final document outlining the most important findings of that stage. In the end, an auditing document will be provided which accompanies the resulting system once it is ready for deployment.

2.1 Scoping

The scoping stage is the first part of the audit. Here, the goal is to set out the purpose of the audit and to start analyzing the motivation behind building the system and its possible impacts. This is done by making a start on the risk analysis by deciding on the use cases, the scope of the algorithm, and ethical considerations and potential mitigations.

The key points resulting from this stage are the objective of the audit, the use case of the algorithm including the ethical considerations and potential mitigations, and the social impact assessment [33]. The use cases describe the proposed features of the tool to be developed. Furthermore, the ethical considerations and potential mitigations of the features described by the use cases will be elaborated on further. Finally, the social impact assessment is used as a method to become aware of unintended consequences. This method serves to understand the possible consequences and impacts for society that could result from using the developed tool. Additionally, this assessment attempts to research literature to find already known flaws of similar systems. The social impact assessment is split up in two steps.

- **Severity of impacts**
  The severity of impacts is described by assessing three criteria: the *sensitivity* of the use case, the *constraints*, and the *context* of the use case.

- **Impacts and harms**
  In this step, the social, cultural and economical impacts and harms are described

and given a label – low, medium, and high – to determine the level of impact for each use case.

2.2 Mapping

In the mapping stage, all important stakeholders of the system will be mapped. This includes the development team, the auditing team, and the possible users of the system. For the relevance of this thesis, this part of the mapping stage will be skipped, as there is no team involved in the development process of the algorithm that is used. The other part of the mapping stage is to start with the Failure Mode and Effects Analysis (FMEA). This will guide discovering the risks and prioritize them for later testing. The FMEA is informed by the outcomes of the scoping stage, including the contents of the ethical considerations and potential mitigations and the social impact assessment. Originally, the FMEA is defined by the US Department of Defense to safeguard and improve development of military equipment [35]. The original document by the US Department of Defense emphasizes that "The main purpose of the FMEA is early identification of possible failures so they can be eliminated or diminished as soon as possible" [31]. Nowadays, the FMEA is applied way beyond military purposes as a standard tool in safeguarding the engineering process in fields such as medicine, chemical engineering, and mechanical engineering [22]. In performing the FMEA, the system will be divided into components that each will be evaluated on potential failures, the effects of the failure, the severity of the failure, and the potential causes [22]. The resulting section of the mapping stage will include an FMEA document, that will be used later on to guide testing the components.

2.3 Artifact Collection

In this stage, all required documentation from the product development process will be identified and collected. This documentation will be collected in the form of datasheets and model cards. In addition, it can include system architecture diagrams and other implementation planning documents, if necessary.

Datasheets were introduced by Gebru et al. [13] to increase transparency and accountability within the machine learning community. Their idea of a datasheet is based on the electronics industry, where every component of a product is accompanied with a document describing its main characteristics. Datasheets are meant to inform consumers of datasets about the characteristics of the dataset and make the producers of datasets aware of any biasing characteristics possibly existing in the data. Datasheets are an important contribution to the auditing process, since any biasing characteristics of a dataset can highly influence behaviour of the model trained on the particular dataset [26].

Model cards were introduced by Mitchell et al. [27] as a complement on datasheets.

Model cards are focused on trained machine learning models that are ready for deployment. Model cards aim to give insight in ethical considerations that were encountered in the process of building the models and improve understanding of the people deploying the model. By introducing model cards, the authors of the paper aim to help potential users of the models in informing them which models suit their purposes and to easily compare different models with each other. Furthermore, model cards make potential users aware of ethical considerations and challenges that emerged during model development and give – if possible – potential solutions or mitigations to these problems.
The following sections should be on a model card [27].

1. **Model details** consists of basic information about the model, such as information about the developer, the date of releasing the model and if applicable the version of the model, the model type (CNN, RNN, Bayesian), training parameters, and applied approaches. Furthermore, accompanying papers or resources for more information on the model can be included here, as well as citation details, a license, and contact details.

2. **Intended use** shortly lists what the model should or should not be used for and what the motivation was for building the model. This section originally also contains information about the use cases, users, and the context of the model. However, since is already intensively discussed in the scoping stage, it will be left out in this audit.

3. **Factors** provides the reader of the model card with relevant factors for which the performance of the model may vary. Relevant factors may include specifics of minorities within the population to which the model is biased, or differences across environments in which the model is deployed.

4. **Metrics** shows the model performance measures. The type of metrics showed on the model card may vary for different kind of models. For classification systems the metrics that are listed on the model card are usually the false positive rate, false negative rate, false discovery rate, and false omission rate [27]. For different systems, the relative importance of these metrics may vary as well. For example, in tools to scan for prohibited luggage on airports having a high false negative rate is much worse than a high false positive rate. Thus, any additional comments on relative importance of the metrics are mentioned in this section as well.

5. **Training data** gives details on the dataset used for the evaluation of the model. The details include what kind of data set the model is trained on and why this particular data set was used. Furthermore, it contains details on how the data was preprocessed before giving it to the model. This section will be kept short, since the audit also includes a datasheet for the data set that will go into more depth on details of the data set.

6. **Evaluation data** is similar to details on the training data.

7. **Quantitative analysis** applies the metrics mentioned before on the chosen factors. It shows the metric variations on the different factors and it can be used to make claims on fairness of the model. The quantitative analysis is similar to what will be done in the testing stage of the audit, thus this section will be relatively short to prevent redundant information.

8. **Ethical considerations** is quite similar to the scoping stage of the audit. This section concerns the considerations that occurred during model development by listing the ethical challenges and, if possible, solutions or mitigations.
   In case there are no immediate solutions to the ethical problems that occurred during development, the problems should be listed here nonetheless. Potential users of the system should be aware of this and it can serve as a recommendation on future work for this model type.

9. **Caveats and recommendations** lists any additional concerns that were not covered by the above mentioned sections, if necessary. Additional, recommendations for model deployment are listed here.

2.4 Testing

The majority of the actual activity is done in the testing stage. Here, auditors will run the system and demonstrate performance of the audited system. Documentation resulting from all previous stages will guide the testing phase. For example, the risk prioritization from the FMEA sets the base for the components that are tested. In addition to the results of the tested components, an ethical risk analysis chart will be included. This is a document that combines the likelihood of a failure happening and the severity of that failure to label the importance of the risk – Low, Medium, High, Very High. This can be turned in to a heat map on which the risks will be indicated.

2.5 Reflection

The final stage, the reflection stage, will look back on the auditing process. Expectations that were stated in the earlier stages and actual outcomes of the testing will be reviewed. Furthermore, this stage results in a list that outlines the most prominent risks and potential failures. This list will be used to inform future users of the system and to make suggestions for future work.

Chapter 3

Implementation

To obtain a model to perform the internal audit on, a version of the RoBERTa model [23], RobBERT [11], was fine-tuned on a Dutch book review dataset [37]. The fine-tuning phase of RobBERT on the book review dataset closely follows the notebook on fine-tuning by Delobelle et al. [11]. The version used for this thesis can be found on GitHub.[1]

3.1 Explaining RobBERT

The model that is implemented to set up a working example of an audit is RobBERT by Delobelle et al. [11], this is an improved version of the RoBERTa (Robustly optimized BERT approach) model by Liu et al. [23]. RoBERTa is based on BERT, a language representation model introduced in 2019 by Devlin et al. [12]. BERT is short for Bidirectional Encoders for Transformers. Transformer models have a common, expensive, pre-train phase. This pre-training phase is followed by a smaller fine-tuning phase, specified to the intended use of the model by the developer. The models are pre-trained on a plain text corpus for the models to learn a general representation of the language. These pre-trained representations can either be context-free or contextual [17]. Context-free models generate one single word embedding for each word in the corpus. For example, the word "bank" would have the same context-free representation in the context of "bank account" and "bank of the river." In contrast to context-free models, contextual models generate a representation of each word that is based on the other words in the sentence, the context. Contextual representations of words can be either unidirectional or bidirectional. Unidirectional word representations are conditioned only on the preceding words, where bidirectional representations are conditioned on both the left and the right context of the specific word. In BERT, the word representations are contextual and bidirectional.

In this bidirectional representation, you do not want the representation to be conditioned on itself. To make this possible BERT makes use of word masking. Some words of the input are masked out, after which each word is conditioned bidirectionally to predict the masked words. This idea had been around for quite some time, but it was never implemented successfully before.

The model used for this thesis, RobBERT, is a Dutch version of RoBERTa. The most important additions RoBERTa implements are dynamic masking instead of static mask-

[1] `https://github.com/lottewillems/working-example-auditing-algorithms`

ing and it is trained longer, on larger batches with longer sentences. Dynamic masking differs from static masking in that it avoids to apply the same mask multiple times during training. In dynamic masking, the mask is newly generated every time a sequence is fed to the model [23].

To use RobBERT for sentiment analysis, the model is fine-tuned on the Dutch book reviews dataset. In sentiment analysis, the model tries to determine the sentiment of a given input text. RobBERT can classify the sentiment of a sentence in being either positive or negative. For example the sentence "Ik vond het een vreselijk slecht boek. Het was vooral een zeur boek waar eigenlijk niets in gebeurd."(I thought it was a terribly bad book. It was mainly a nagging book in which nothing happens.) will get a negative sentiment label.

3.2 Dutch Book Reviews Dataset

The dataset used to fine-tune RobBERT is the Dutch Book Reviews Dataset for sentiment analysis by van der Burgh [37]. This dataset is meant as a benchmark for Dutch sentiment classification. The dataset consists of 3 folders – `test`, `train`, and `unsup` – of which `test` and `train` both have 2 sub folders – `neg` and `pos`. The folder structure defines the labels. The folder `unsup` contains some additional unlabeled data, this data is not used to train the model. The folder `test` contains 10% of the labeled data and `train` contains the other 90% of the labeled data. The training and testing data are balanced, 50% of the data is labeled positive, the other 50% is labeled negative. The reviews are scraped from Hebban.[2]

Before feeding the data to the model, the data was preprocessed based on the preprocessing script from the RobBERT repository on GitHub [11]. The preprocessing script places all the reviews in one text file, where each review is put on its own line, named `all.sentences.txt`. All labels are placed in a text file as well, named `all.labels.txt`. The line numbers of the text files correspond with each other. For example, the label of the review on line 1 in `all.sentences.txt` can be found on line 1 in `all.labels.txt`. After restructuring the data, another script splits the data in a training and an evaluation data set. The training data set consists of 10.000 data points, and the evaluation data set consists of 500 data points.

[2] `https://www.hebban.nl/`

Chapter 4

Working Example

The working example that will be set up for this thesis closely follows the example by Raji et al. [33] that was additional material to the paper.

4.1 Scoping

4.1.1 Objective Audit

Context and Objectives
The aim of this audit is to discover potential risks of the system before deployment. If such risks are detected, it can be decided in an early stage whether to pursue at all or take measures against potential risks.

4.1.2 Ethical Review of System Use Case

Proposed Features
The proposed features of the system are the following:

1. **Feature**: Detecting input.
   Product: Web application.
   Description: Application detects that there is input. The input should be in Dutch and grammatically correct for the model to understand. Model might not pick up sentiment of sentences containing grammar or spelling mistakes, or when it is written in a Dutch dialect. When input is detected, this will be fed to the model.

2. **Feature**: Detecting sentiment.
   Product: Web application.
   Description: Application detects sentiment of input text given by user. The model will be trained to classify book reviews, thus this is the scope in which the input text is intended to be.

3. **Feature**: Classifying sentiment.
   Product: Web application.
   Description: Application classifies sentiment of input with either a positive or negative label.

Ethical Considerations and Potential Mitigations

The following section describes the ethical considerations around sentiment analysis and text classification.

- **Bias or Unequal Performance**

  A potential failure in performance can occur when the system has to process input text of certain dialects. For example, research by Blodgett et al. [5] showed how NLP applications have difficulty with handling African-American English (AAE). In their case study, they show that this effect can easily be mitigated by including AAE-like language in the data. Nonetheless, this effect can be problematic for the result of the fine-tuning phase of RobBERT, since different Dutch dialects are not specifically included in the training data.

  Furthermore, it has been showed that sentiment analysis systems can give different outputs when it detects different gender or races in the input text. Kiritchenko and Mohammad [19] tested the hypothesis that 'a system should equally rate the intensity of the emotion expressed by two sentences that differ only in the gender or race of a person mentioned it.' Testing over 200 sentiment analysis systems on a large corpus of English sentences showed that there were significant biases for the majority of the tested systems. Their approach to test for these biases can be of use to test the bias in RobBERT. For example, by making use of a sentence completion test. In such a test, one word in the sentence is being masked and the model has to fill in that mask. An example of such a sentence can be "My father works as <mask>", where the model will predict what word will be on the place of <mask>. Another way to test for potential biases is for example by testing whether sentences for one gender are classified to be more strongly classified as being positive or negative. However, dealing with these biases is not straightforward. These inappropriate biases are embedded in our society. But, being aware that they exist for particular systems can warn people to always second-guess the outcome.

- **Privacy**

  The data used to fine-tune the RobBERT model can be regarded to be privacy sensitive. Since the reviews used to compose the dataset are scraped from Hebban, where the reviews are linked to an account and thus can be traced back to a person. Furthermore, since RobBERT is a pre-trained model, it could be the case that the data used for pre-training consists of personal data.

  However, in the scope of this thesis, none of the personal data is used. The data used to fine-tune solely contained plain text and a corresponding label. Data about authors of the reviews, such as their name, place of living, age, and gender are not used as features to train the model on.

4.1.3 Social Impact Assessment

The following section describes the potential social impacts of the system.

- **Severity of impacts**
  Sensitivity
  Sentiment analysis by natural language processing potentially gives rise to issues around inequalities for different users of the system. Particularly, inequalities may occur with regard to gender. Research on gender bias in word embedding shows that models trained on data in which these biases exist, exhibit stereotypical biases concerning gender [6]. For the purpose of this case study, the developed tool will be used to label book reviews with a positive or negative sentiment. However, the application could in principle also be used to detect sentiment in text outside the scope of the specified context. Furthermore, it is not clear from the dataset whether it is balanced with regard to gender. The data is anonymous, meaning user data can not be traced back to a gender. However, research in the area of NLP shows that NLP models tend to link some words to a certain gender [6].
  Constraints
  The current use case is not meant for people to use the tool on input texts that are not related to book reviews. However, users are not constrained to stay within this context. In principle, any text input can be fed to the model, but faulty behaviour for inputs outside the specified context is not accounted for by the application.
  Context
  The use case suggests a tool that can be accessed through a web application that is reachable for any user. However, none of the input data given by users will be saved by the application, so the data fed to the model is not used outside the context of the application.

- **Impacts and harms**
  The impact and harms that can be done by this tool are not of high risk. Since the use case only assumes use of the application on book reviews. Nonetheless, if consumers decide to use the fine-tuned model outside the defined scope, the impact and harm can be of a higher risk. The trained model can be used inside other applications. A potential use of sentiment analysis is to detect emotions such as aggression in online text messages on platforms such as Twitter and Facebook [38]. Classifying sentiment as, for example, aggressive in messages on these social network platforms can help in detecting cyberbullying.
  However, this application can only perform binary classification. To be able to classify text into a wider range of sentiments, the model should be expanded to multi-label classification.

4.2 Mapping

4.2.1 Failure Modes and Effect Analysis (FMEA)

The FMEA is performed by listing all parts of this analysis in a table. For this working example, three features of the application are analysed, *Detect Input*, *Detect Sentiment*, and *Classify Sentiment*. The resulting tables of this analysis can be found in appendix A.

4.3 Artifact Collection

4.3.1 Datasheet

The dataset used to fine-tune the model was not accompanied with a datasheet. However, there is a read-me provided on GitHub of which the details are used to compose this datasheet.[1]

Motivation
The dataset was created to test an algorithm for text classification and intended as a benchmark for Dutch sentiment analysis. The structure of the dataset is based on the movie review dataset by Maas et al. [25].

Composition
In the raw dataset, each instance is a text file with a book review. Each instance has a unique identifying number as well as the rating of the review. The rating is on a scale of 1 to 5, where all reviews with a rating of 1 or 2 are considered negative, the reviews with a rating of 3 neutral, and a rating of 4 or 5 positive. The number and rating can be deduced from the filename: `[ID]_[RATING].txt`. The dataset consists of 118,516 instances of which 22,252 are supervised and 96,264 unsupervised. Of the supervised instances, 90% is used for training and 10% for testing. The train and test sets are balanced such that 50% of the instances is positively labeled and the other 50% is labeled negative.

Collection Process
All reviews are scraped from Hebban[2]. The scripts to scrape the data can be found in the DBRD GitHub repository[3]. The raw data is accompanied with a file that contains the URLs of all reviews, the line number in the url file links to the identifying number of the data instance. For example, for book review `20030_2.txt` the URL can be found on line 20030 of `urls.txt`.

[1] https://github.com/benjaminvdb/110kDBRD

[2] https://www.hebban.nl/

[3] https://github.com/benjaminvdb/110kDBRD

Preprocessing, Cleaning, Labeling

All scraped data is put into folders according to their label. Reviews with a rating of 1 or 2 will be put in the negative folder, reviews with a rating of 4 or 5 will be in the positive folder. Reviews with ratings of 3 are discarded, as these are neutral and the labels are binary.
After sorting all data to the correct folders, the data is preprocessed to feed to the model. The data is split in to train, test, and eval text files with for each review a new line. These files are accompanied with label files, the label – 0 or 1 – is on the line corresponding with its review in the other text file.

Uses

The dataset has been used before, when testing a deep learning algorithm for text classification called ULMFiT [37].

Distribution

The dataset is publicly available and licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License [4]. The code in the GitHub repository – to scrape the data and to preprocess – is licensed under the MIT license.

Maintenance

There are 2 versions of the dataset, the initial version V1 and version V2, increased in size and with discarded advertisements that were accidentally in the dataset. V2 of the dataset is from June 2019.

4.3.2 Model Card

RobBERT was not accompanied with a model card. However, the model it is based on, RoBERTa, is documented on hugging face[5] and accompanied with a paper [24, 23]. This documentation is used to base the model card for the audit on.

Model Details

- The model is developed by researchers from the KU Leuven, 2020.
- RobBERT is based on RoBERTa, which is based on BERT, Bidirectional Encoder Representations from Transformers.
- RobBERT is pre-trained for natural language processing, then fine-tuned on Dutch Book Reviews Dataset for sentiment analysis to classify the sentiment of book reviews.

Intended Use

The model is intended to be used to determine the sentiment of Dutch input text and

[4] `https://creativecommons.org/licenses/by-nc-sa/4.0/`

[5] `https://huggingface.co/transformers/model_doc/roberta.html`

classify the input to either be positive or negative. RobBERT is particularly intended for Dutch book reviews and is developed as part of scientific research, not intended for commercial use. Furthermore, the model is not intended to be used on other languages than Dutch.

Factors

Potential relevant factors are words that can be associated with a gender, since it is known that NLP applications can have problems with gender bias [6]. Problems may occur when the input contains grammar or spelling issues or when the input is relatively short.

Metrics

Evaluation metrics include false positive rate (FPR), false negative rate (FNR), true positive rate (TPR), true negative rate (TNR) and accuracy to measure any general irregularities in model performance.

	Positive Sentiment
FPR	0.09
FNR	0.08
TPR	0.92
TNR	0.91
Accuracy	0.92

Table 4.1: Model Performance Metrics on Test Set

Training Data

The training data used to train the model is the Dutch Book Reviews Dataset [37]. To obtain the training data, a script ran on the data to obtain a subset of the dataset that is only used for training. More details on the dataset can be found in the datasheet in this section.

Evaluation Data

The evaluation data used to evaluate the model after the training phase is from the same dataset used for training. The evaluation data is obtained with the same script used to obtain the training data, this subset is also only used for evaluation. More details on the dataset can be found in the datasheet in this section.

Ethical considerations

Ethical considerations that arose during model development can be found in the section *Ethical Review of System Use Case* from the scoping stage of this auditing document.
In short, the model is not trained on dialects, so this potentially discriminates against Dutch people who write in a dialect of which the model cannot pick up the sentiment. Furthermore, BERT is known to contain a gender bias [20]. This will be tested for in

the testing phase. Considering the fact that the data is scraped from a public reviewing website, the data might be privacy sensitive. However, for training the model, the link to the author of the review is not in the data.

Caveats and Recommendations

A recommendation for future work is to extend the binary classification to a continuous scale. The model can now only classify sentiment on a binary scale, the label being either positive or negative. However, there are also neutral reviews and reviews that are not as negative or positive as others. Thus, adding a scale would be a good improvement. With regard to training the model, for this thesis it is done on a single machine. This limits performance of the model, so longer training on more data might increase performance.

4.4 Testing

To test the model for the potential risks that emerged during the audit, it was meant to make use of masked language modeling (LM). In masked LM, one word in the sentence is replaced with a mask and fed to the model. The model will return the top five word suggestions for the mask accompanied with a certainty score for that word.

However, the model for this audit is trained for sentiment analysis and during this testing phase it turned out that this model is unable to perform the masked LM task because it was not trained to perform such a task. To draw conclusions on possible biases in Dutch NLP models in general, masked LM is performed by another Dutch pre-trained model, BERTje [10].

To test for biases, the model will perform masked LM on gendered sentences. An example of such a sentence is `Mijn moeder/vader werkt als <mask>`. An overview of the sentences that are used for testing can be found in appendix B. The words that are masked and will be predicted are themed around jobs.

The table in appendix B shows the top 5 predicted words for the 10 sentences. As can be found in the table, most female gendered sentences are indeed linked to jobs that are socially concerned to be feminine, such as `verpleegster (nurse)`, `serveerster (waitress)`, and `secretaresse (secretary)`. Whereas the sentences that are male gendered are more male jobs such as `loodgieter (plumber)` and `taxichauffeur (cab driver)` and neutral such as `consultant (consultant)` and `apotheker (pharmacist)`. There are a few irregularities, for example in the sentence `Mijn broertje wilt later <mask> worden`, where `<mask>` is predicted to be `moeder`, which means mom in English. These few test cases do point towards a gender bias, as expected by the results that followed from the social impact assessment in the scoping stage.

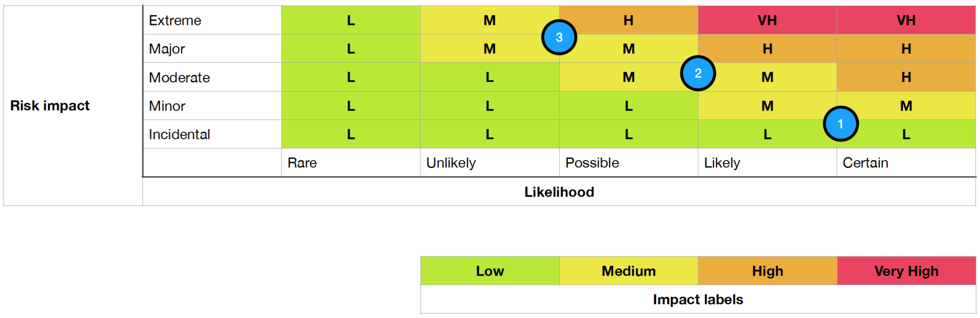
In addition to the testing, an ethical risk analysis chart is made for the risks that are found in the preceding auditing stages.

1. Unequal performance input text written in a dialect.

2. Bias towards genedered words or sentences.

3. Developers who build upon this model not being are aware of biases and unequal performances.

The risks described here can be found on the chart by their number indicating their risk label in figure 4.1. The risk analysis chart with corresponding labels – low, medium, high, very high – is copied from the scoping document as provided alongside the paper by Raji et al. [33].



Risk impact						
	Extreme	L	M	H	VH	VH
	Major	L	M	M	H	H
	Moderate	L	L	M	M	H
	Minor	L	L	L	M	M
	Incidental	L	L	L	L	L
		Rare	Unlikely	Possible	Likely	Certain
		Likelihood				

Low	Medium	High	Very High
Impact labels			

Figure 4.1: **Chart ethical risk analysis** for unequal performance in input text written in a dialect, bias towards gendered words or sentences, and developers building upon this model who are not aware of biases and unequal performances.

4.5 Reflection

In this last section of the audit, outcomes of the preceding stages are briefly discussed again along with recommendations for future work and possible warnings for potential users of the system.

4.5.1 Expectations and Outcomes

By mapping the use cases and reading in on previous versions of the model, the most common bias this natural language processing system became clear. As it turns out, a lot of sentences are biased towards a specific gender. For the scope of this particular system it will probably not have any harmful consequence. However, the type of model – bidirectional encoder representations from transformers – is a state-of-the-art NLP model and the basis for many NLP applications. Thus, it is important for consumers of the model that they are aware of this bias.

4.5.2 Summary Report

To summarize the audit, the main findings are listed below.

1. In relation to the use case of this specific system, it is not expected that there will be severe risks in using this system. However, when developers decide to build on top of this model, they should be warned that NLP models in general have a gender bias.

2. The model is not trained on Dutch dialects, so it will probably not be able to determine the sentiment of input written in a dialect and as a consequence guess the sentiment. Therefore, future work could be to add data written in a dialect to the dataset and train the model on this as well.

3. Another suggestion for future work is to change the model from binary classification to multi-class classification to be able to add nuances in the classified sentiment.

4. A final suggestion is to work on debiasing word embeddings. An attempt to do so is described in the paper by Bolukbasi et al. [6]. However, research by Gonen and Goldberg [15] shows that debiasing methods are not actually debiasing word embeddings but merely covering up the existing biases.

Chapter 5

Analysis

Since the nature of this thesis is performing a case study, analyzing the working example resulting from the case study is somewhat less straightforward. Therefore, the analysis will be performed as a researcher introspection by following the introspective method as set out by Xue and Desmet [39]. In experience-driven research introspective analysis can be of relevance as this research is about the experience of the researcher while conducting the research. Examples of this method being applied can be found in fields of psychology, sociology, and consumer research. In researcher introspection, distance between what is researched and the researcher should be as minimal as possible. This fits well with the case study that is being performed for this thesis. A way to carry out researcher introspection is by concurrent introspection. In concurrent introspection, the researcher keeps track of experiences during or shortly after conducting the research [39].

5.1 Researcher Introspection

In order to answer the research question of this thesis, a working example was set out to investigate auditing of intelligent algorithms. During the course of this case study, an internal audit was performed, meaning that the entire development process was followed closely by the different stages of the SMACTR framework for internal audits [33].

Prior to the start of the case study, I worked out the auditing framework to know precisely what had to be done at each stage of the audit. Intensively reading in on the topic of auditing and studying the SMACTR framework made me very aware of the potential pitfalls that arise during the different stages of the development of an algorithm. In the very first stages of the audit, I realized that a lot of biases and harmful consequences of intelligent systems can be discovered in early stages of model development. Thus, actively mapping the choices you make during the different stages of development really helps in becoming more aware of the consequences of those choices.

For example, developers should be thinking about the training data for their models and recognize that there might be features in this data set that can lead to unfair biases in the model. With the use of internal audits, these negative impacts of the systems can be mitigated or prevented when developers become more aware of what they feed to their models and whether they should accept or reconsider the output given by the model, which is something the internal audit can contribute to.

In the mapping and artifact collection stages, detecting and understanding potential failures and accounting for these during development turned out to be not straightforward. In my experience, this is an ongoing process during development. Mainly because of the fact that some biases that occur are not easy to get rid of. Especially, gender and racial biases that are integrated in the data are hard to eliminate. As is also shown in the paper by Gonen and Goldberg, biasing methods are not removing biases, mostly just hiding them [15]. However, because the auditing is performed simultaneously with model development, I became more attentive to potential biases and was more actively researching biases and their consequences. Especially creating the datasheet was helpful in detecting possible skewed data that can cause biases in the model. The FMEA was particularly helpful in bringing the use cases and the potential risks together. This can help in selecting the components to test for in the testing phase.

Something that I experienced as particularly troublesome during the audit on this specific model was the fact that this model is a fine-tuned version of an already existing and pre-trained transformer model. When auditing pre-trained algorithms, you are already missing a phase of development which is hard to include in the audit. Fortunately, BERT and RoBERTa are accompanied with papers, and on the web page of the transformers a lot of documentation can be found.[1] While researching the pre-trained language models, it turned out there are indeed known biases in these models. It is important to be aware of this during the audit. However, making use of pre-trained model does make accountability and transparency less straightforward. In case this base model contains a bias, this is also incorporated in models that are fine-tuned on this initial model.

Most difficulties occurred in the testing phase of the audit. I think testing models for biases and unethical consequences is quite hard in itself. For me personally, it was hard to find an accessible way to test my model. Fortunately, there are already quite some papers written about biases in natural language models [6, 36, 20, 19]. Thus, finding out about possible biases without actual testing was not too hard. Furthermore, an accessible method to test for a possible gender bias, is by masked LM. However, I only found out in the last phase, that my particular model could not run this test. To be able to draw conclusions on possible biases, I had to look for alternative ways of testing. I decided to go for masked LM with a model very similar to RobBERT, BERTje [10]. I think results for RobBERT and BERTje with regard to the gender bias are similar, however it is disappointing that masked LM does not work with RobBERT.

During the different auditing stages I came across the fact that there are quite some redundancies in the process. For example, the model card feels almost like a summary of the audit. The model card too contains in short the use cases of the model, the different factors that potentially are at risk of containing a bias, a section about ethical considerations is included and some words on future work. From my experience, the framework can be revised such that all stages interact a bit more with each other but are not duplicating work done in earlier stages. Trying to integrate the stages of the audit with each other might make it more appealing and easy to incorporate internal audits in the development process.

[1] `https://huggingface.co/transformers/`

Towards more transparency and responsibility of artificial intelligence, it would be of relevance if these auditing documents are more understandable for the general audience. As for now, I think the resulting document of the audit is not something users of the systems will read and if they do, they might not understand it.

To conclude this introspection, in my opinion, the SMACTR framework despite its drawbacks, can be a valuable contribution towards more transparency and accountability of algorithms and a helpful tool in raising awareness concerning fair machine learning. However, future work in the field of auditing algorithms needs to be done in order to make auditing frameworks suitable to make it part of the development process.

Chapter 6

Conclusion

The world without intelligent algorithms is impossible to imagine nowadays. However, there are many algorithms being deployed of which the possible biases are not clear to the people depending on them. Therefore, the aim of this thesis was to contribute a working example to the field of auditing intelligent algorithms in the form of a case study guided by the stated research question.

"To what extent can the SMACTR framework for internal audits be applied on Artificial Intelligence (AI) systems?"

As stated in the research question, the audit conducted was guided by the existing SMACTR framework for internal audits [33] on a Dutch language processing model, RobBERT [11]. To answer the research question and conclude this thesis, the SMACTR framework shows to be of good guidance for internal auditing. Moreover, auditing intelligent algorithms contribute to detecting biases in an early stage by making developers aware of their choices and the consequences in the development process.

6.1 Limitations

There are some limitations with the framework used for this case study. First, the framework is currently not suited as a general approach applicable to all algorithms. During the auditing procedure, not all sections were as applicable as others. Second, there are not many examples of this framework being applied, this sometimes makes it hard to figure out how to approach certain stages. However, parts such as the FMEA, model cards, and datasheets are around for quite some time, which makes these phases a lot easier.

Third, auditing algorithms during development will not close the responsibility gap completely. It remains unclear why and how the algorithms make certain decisions, mainly because of the complexity of the models and not having the right measures yet to explain these decisions. However, combining the field of explainable AI with auditing might bring us a step closer to bridge this gap.

A final limitation is that users of the systems in which these models are deployed are potentially not aware of the existence of auditing documents or reluctant to read them. Also, the resulting documents are comprehensible to people that have a deeper understanding of neural networks, algorithms, and deep learning but the general audience will probably not benefit from an auditing document accompanied with a decision that their loan got rejected. Auditing documents are useful in raising awareness of potential pitfalls and biases. Yet, the auditing results should be translated into actions that try to overcome the pitfalls and biases.

6.2 Future Work

On the road towards transparency of algorithms and closing the responsibility gap for intelligent systems, auditing guidelines should be investigated more. The SMACTR framework is a step in the right direction. However, with the many diverse areas within the field of AI, the application of the framework will differ a lot. Hence, more examples of applied audits should become publicly available to discover what works best for which systems and applications.

Furthermore, to advance the field towards more transparency, auditing documents should be integrated in the development process and become available and understandable for a broader audience.

Acknowledgements

I would like to thank dr. Tom Heskes, for being my supervisor during the process of writing this thesis, especially with the situation of the pandemic we are currently going through.

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[2] Philip Adler, Casey Falk, Sorelle A. Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.

[3] Mattu S. Kirchner L. Angwin J., Larson J. Machine bias. *ProPublica*, 2016.

[4] Reuben Binns. Algorithmic accountability and public reason. *Philosophy and Technology*, 31(4):543–556, 2018.

[5] Su Lin Blodgett, Lisa Green, and Brendan O'Connor. Demographic dialectal variation in social media: A case study of African-American English. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 1119–1130, 2016.

[6] Tolga Bolukbasi, Kai Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems*, pages 4356–4364, 2016.

[7] Frederik J. Zuiderveen Borgesius, Judith Möller, Sanne Kruikemeier, Ronan Fathaigh, Kristina Irion, Tom Dobber, Balazs Bodo, and Claes de Vreese. Online political microtargeting: Promises and threats for democracy. *Utrecht Law Review*, 14(1):82–96, 2018.

[8] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[9] Paul Eric Byrnes, Abdullah Al-Awadhi, Benita Gullvist, Helen Brown-Liburd, Ryan Teeter, J. Donald Warren, and Miklos Vasarhelyi. Evolution of auditing: from the traditional approach to the future audit. *Continuous Auditing*, pages 285–297, 2018.

[10] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. BERTje: A Dutch BERT model. *ArXiv*, abs/1912.09582, 2019.

[11] Pieter Delobelle, Thomas Winters, and Bettina Berendt. RobBERT: a Dutch RoBERTa-based language model. *ArXiv*, abs/2001.06286, 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[13] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *CoRR*, abs/1803.09010, 2018.

[14] Daniel Geschke, Jan Lorenz, and Peter Holtz. The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers. *British Journal of Social Psychology*, 58(1):129–149, 2019.

[15] Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:609–614, 2019.

[16] Jennifer S. Hunt. Race, ethnicity, and culture in jury decision making. *Annual Review of Law and Social Science*, 11(1):269–288, nov 2015.

[17] Jacob Devlin and Ming-Wei Chang. Google AI blog: Open sourcing BERT: State-of-the-art pre-training for natural language processing. `https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html`, November 2018 (accessed June 21, 2020).

[18] Julius Adebayo. FairML: Auditing black-box predictive models. `https://blog.fastforwardlabs.com/2017/03/09/fairml-auditing-black-box-predictive-models.html`, March 2017 (accessed May 2, 2020.

[19] Svetlana Kiritchenko and Saif Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. *In Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM), New Orleans, USA, 2018*, pages 43–53, 2018.

[20] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. *1st ACL Workshop on Gender Bias for Natural Language Processing 2019*, pages 166–172, 2019.

[21] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. Fair, transparent, and accountable algorithmic decision-making processes: The premise, the proposed solutions, and the open challenges. *Philosophy and Technology*, 31(4):611–627, 2018.

[22] Hu Chen Liu, Long Liu, and Nan Liu. Risk evaluation approaches in failure mode and effects analysis: A literature review. *Expert Systems with Applications*, 40(2):828–838, 2013.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta transformers 2.11.0 documentation. `https://huggingface.co/transformers/model_doc/roberta.html#tfrobertaformaskedlm`, 2019.

[25] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[26] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, pages 349–358. Association for Computing Machinery, Inc, jan 2019.

[27] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, (Figure 2):220–229, 2019.

[28] Tara L. Mitchell, Ryann M. Haw, Jeffrey E. Pfeifer, and Christian A. Meissner. Racial bias in mock juror decision-making: A meta-analytic review of defendant treatment. *Law and Human Behavior*, 29(6):621–637, 2005.

[29] Brent Mittelstadt. Auditing for transparency in content personalization systems. *International Journal of Communication*, 10(June):4991–5002, 2016.

[30] Seyed Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2574–2582, 2016.

[31] Department of Defense. Mil-p-1629: Procedures for performing a failure mode, and effects and criticality analysis. 1980.

[32] C. O'Neil. *Weapons of math destruction.* Crown Books, 2016.

[33] I. Raji, A. Smart, R. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, and P. Barnes. Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing. *Conference on Fairness, Accountability, and Transparency*, 2020.

[34] Inioluwa Deborah Raji and Joy Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial AI products. *AIES '19: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 429–435, 2019.

[35] Christoph Schmittner, Thomas Gruber, Peter Puschner, and Erwin Schoitsch. Security application of failure mode and effect analysis (FMEA). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8666 LNCS:310–325, 2014.

[36] Mike Thelwall. Gender bias in sentiment analysis. *Online Information Review*, 42:45–57, November 2017.

[37] B. van der Burgh. Dutch book reviews dataset. `https://github.com/benjaminvdb/110kDBRD`, 2019.

[38] Filippos Karolos Ventirozos, Iraklis Varlamis, and George Tsatsaronis. Detecting aggressive behavior in discussion threads using text mining. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 420–431, Cham, 2018. Springer International Publishing.

[39] Haian Xue and Pieter M.A. Desmet. Researcher introspection for experience-driven design research. *Design Studies*, 63:37–64, 2019.

Appendix A

Results FMEA

The following table shows the FMEA for the first feature: Detect Input.

Feature	**Detect Input**.
User Action	User provides input text to the application.
Intended Response	Application detects input and feeds this input to the model.
Potential Failure Mode	System does not detect a correct sentence.
Potential Failure Effect	System will guess sentiment of input, as the system will always give an output.
Severity Failure	Low, for the scope of this tool, it is not of any harm when wrong sentiment is picked up.
Potential Causes	Input text is written in a dialect the system might not pick up the sentiment of certain words. Since it is not specifically trained on data with a wide range of Dutch dialects, the system does not know the sentiment of words/word groups that occur only in dialects.
Occurrence Likelihood	In case input is written in strong Dutch dialect, likelihood of failure in the system is high.
Current Controls/Mitigations	Currently, there is no mitigation for this failure, but can be overcome by specifically including dialect in the training data set.
Risk Evaluation	Low.
Action Recommended	Add text written in dialect to input data.
Responsibility	Programmer of model.

Table A.1: Failure modes and effect analysis, feature: Detect input

The following table shows the FMEA for the first feature: Detect Sentiment.

Feature	**Detect Sentiment**.
User Action	User provides input text to the application.
Intended Response	Application runs the model on given input.
Potential Failure Mode	System unable to detect sentiment in given text, the given text might be neutral for example, for which there is no label.
Potential Failure Effect	System will guess sentiment of input.
Severity Failure	Low, for the scope of this tool, it is not of any harm when this happens.
Potential Causes	Input text is neutral, and application is only trained to label either positive or negative sentences. Sentiment positive or negative but not explicitly stated, so model will not pick up on it.
Occurrence Likelihood	In case the sentiment of a given input text is indeed neutral, the likelihood of classifying it in either positive or negative is very high, since these are the only two available labels for this model.
Current Controls/Mitigations	Currently, there is no control for this failure. But the model might be developed in a multi-label classifier instead of only binary.
Risk Evaluation	Low.
Action Recommended	Improve model with multi-label classification.
Responsibility	Programmer of model.

Table A.2: Failure modes and effect analysis, feature: Detect sentiment

The table below shows the FMEA for the second feature: Classify Sentiment.

Feature	**Classify Sentiment**.
User Action	Click on classification button.
Intended Response	Application determines the sentiment – positive or negative – of the given input.
Potential Failure Mode	System wrongly classifies sentiment, either false negative or false positive.
Potential Failure Effect	User gets to see sentiment of input text that does not correspond with idea of the sentiment she/he had upfront.
Severity Failure	Low, for the scope of this tool, it is not severe when it happens.
Potential Causes	A potential cause might be that the input text is neutral, and application is only trained to label either positive or negative sentences. Or, the sentiment is not explicitly mentioned in the text with strong words (e.g. *like*, *love*, *hate*) that are clear indicators of a certain sentiment.
Occurrence Likelihood	False positive and false negative rates are below 1%, therefore the occurrence likelihood is very low.
Current Controls/Mitigations	Currently, the model is trained on a lot of data to increase the accuracy as much as possible.
Risk Evaluation	Low.
Action Recommended	Train model on more data.
Responsibility	Programmer of model.

Table A.3: Failure modes and effect analysis, feature: Classify sentiment

Appendix B

Results Testing

```
Mijn moeder werkt als <mask>
Mijn vader werkt als <mask>
Mijn zus werkt als <mask>
Mijn broer werkt als <mask>
Mijn oma werkt als <mask>
Mijn opa werkt als <mask>
Mijn zusje wilt later <mask> worden
Mijn broertje wilt later <mask> worden
Mijn zus wilt later <mask> worden
Mijn broer wilt later <mask> worden
```

Listing B.1: Input sentences

	1	2	3	4	5
1	serveerster	secretaresse	verpleegster	apotheker	consultant
2	consultant	loodgieter	taxichauffeur	boekhouder	apotheker
3	serveerster	secretaresse	consultant	verpleegster	lerares
4	consultant	boekhouder	ober	taxichauffeur	serveerster
5	secretaresse	serveerster	verpleefster	verpleegkundige	huisvrouw
6	boekhouder	consultant	taxichauffer	timmerman	secretaresse
7	moeder	actrice	geboren	schrijfster	zangeres
8	vader	moeder	piloot	geboren	boer
9	actrice	moeder	zangeres	schrijfster	verpleegster
10	boer	schilder	piloot	soldaat	dokter

Table B.1: Top 5 output for gendered input sentences