Machine-annotated rationales: faithfully explaining machine learning models for text classification

Elize Herrewijnen

July 31, 2020

First supervisor: dr. Dong Nguyen
Second supervisor: prof. dr. Floris Bex
Daily supervisor: dr. Jelte Mense

Master Artificial Intelligence
Utrecht University
6323375

Abstract

Artificial intelligence is not always interpretable to humans at first sight. Especially machine learning models with hidden states or high complexity remain difficult to understand. Explanations for such machine learning models can be found, but are not always faithful: according to the actual reasoning that was done inside the model. Finding parts of the model input that contain signals for a classification can be a way of explaining model outputs. Natural language explanations are called rationales. Whoever annotated a part of the text as being an explanation (rationale), is called the annotator. Texts form decomposable sets of interpretable features, where selections of (sub-)sentences can be explanations for model predictions. To find explanations for machine model predictions in text classification, this study introduces machine-annotated rationales, which are natural language explanations from the input text for a model's prediction. Four different approaches to finding faithful machine-annotated rationales are proposed. Evaluation is done by measuring faithfulness, set similarity to human-annotated rationales, and through a user evaluation. Results show that faithful machine-annotated rationales can be found for the investigated machine learning models, but that there is a trade-off between faithfulness and end-user interpretability.

Contents

Acronyms

AI artificial intelligence. 12, 15, 18

AR annotator rationale. 44, 45, 59, 61

BNAll BagNetsTextAll. 32

BNRats BagNetsTextRats. 32

FE feature extraction. 36, 37, 48, 50–52, 56, 61, 63, 68, 70

LAO-confidence Leave-All-Out confidence. 46, 47, 52, 54, 65, 68, 69

LOO-method Leave-One-Out method. 37, 38, 46, 48, 50, 51, 54, 56, 64, 65, 68

MaR machine-annotated rationale. 13, 24, 33, 34, 39, 44, 45, 47, 48, 52, 59–61, 63, 66, 68–70

ML machine learning. 12–21, 23, 25, 30, 38, 56, 64–66, 70, 76

NN neural network. 12

RE rationale extraction. 13, 34–38, 40, 44, 47, 48, 54, 58, 59, 61, 64, 66, 68–70

RS RationaleSearch. 33, 61

SVM support vector machine. 16, 21, 30, 31, 38, 50

XAI explainable artificial intelligence. 12, 17, 20, 21

List of Figures

List of Tables

1 Introduction

Artificial intelligence (AI) can perform human-like tasks with promising results. Machine learning (ML) models are applications of AI that use data to learn to perform such tasks. However, deploying ML models in the work field can have undesired consequences. One example is using an ML model to de-pixelate a pixelated but recognizable photo of Barack Obama. The Face Depixelizer (Menon et al., 2020) produced an unexpected result: a photo of a white man (Truong, 2020). The ML model adopted the bias that was in the training data, which caused the model to behave racially biased. Such behaviour might not be acceptable in real-life situations. When the Face Depixelizer is used in applications like Photoshop, this bias does not impact the lives of people. If the model is used to identify missing persons, the bias can become problematic.

The above example is not the only ML model to behave unseemlily. Other applications of AI display racist or sexist behaviour, for example, associating European American names in text with 'pleasant' and African American ones with 'unpleasant' (Zou et al., 2018). Automatic hate speech detection using ML models can also show racial bias (Sap et al., 2019). And using a future crime risk assessment algorithm to determine who will be set free and who will go to prison shows racial bias in a publication by Julia Angwin et al. (2016).

1.1 Explainable artificial intelligence (XAI)

AI is not always completely understood by humans. While some ML models are based on simple algorithms, others are **black boxes** that seem to miraculously transform input to output, without giving clues about what went on inside the box. If the reasoning that was done to come to an output remains unknown, how can the model be trusted to be competent enough to predict correctly and based on the right grounds?

When the task of an ML model is to assess future crime risk, and the output of the model is consequently used to form verdicts, the model has a great impact on the lives of people. A (racial) bias in such a model is not desired, but it does not mean that the model is therefore not useful in the work field. As long as such a model can explain itself, and the explanation shows that the decision was not made with e.g. a racial bias, then the decision can be used as advice in real-life settings. This shows that the reason for such a model's decision or suggestion is just as important as the decision or suggestion itself.

When an ML model can explain how it came to an output, it is said to be explainable. The field of explainable artificial intelligence (XAI) focusses on finding human-understandable explanations for AI applications such as ML models. Explanations for ML models come in different forms, for example as a complete overview of a model's inner algorithm, a set of important parts of the input that influenced the prediction, or a human-provided natural-language text. The type of explanation depends on the type of ML model and the receiver of the explanation. A receiver can be a researcher with years of experience in AI, but also an end-user in a work field application, like a judge.

Machine learning models can be simple to understand to humans, or very complex and difficult to interpret. The term **transparency** is used to indicate how understandable a model's algorithm or parameters are to humans. **Opaque**ness is the opposite of transparency. A black box is a model that is very opaque: its algorithm or parameters are not understandable to humans (Lipton, 2016). An example of such an opaque model is a deep neural network (NN).

1.2 Faithfulness

Suppose that the Face Depixelizer (Menon et al., 2020) explains its prediction by pointing at the background colour of the image, which causes the model to believe that the picture is taken at night and thus outputting a picture of a white man. However, the (hypothetical) true reason for the prediction is that the dataset contains only pictures of white men in suits, and thus the model transforms the pixelated picture of Barack Obama wearing a suit into a picture of a white man wearing a suit. While the first explanation might make the classification simply unfortunate, the second explanation shows a flaw in the model itself. This flaw decreases the usefulness and reliability of the ML model in crucial work field tasks like identifying missing persons.

One important aspect of this work is the **faithful**ness of explanations: an explanation is faithful if it explains a model's output according to the algorithmic process of the ML model. In other words, the explanation is based on what exactly happened inside the model, and not on, for example, a guess. The first explanation in the above example is not faithful. When an explanation is not faithful, it is in fact not explaining a model's

prediction, and therefore does not contribute to the usefulness of the model in the work field. Therefore, it is important to verify the faithfulness of explanations to understand ML models.

1.3 Rationales

A **rationale** is a human-understandable natural language explanation for an action or decision (Ehsan et al., 2019). A rationale for a prediction from the Face Depixelizer example could be "Suits are only worn by white men and there is a suit in the input picture.". The ML model has learned this and therefore could explain itself along these lines.

When a human annotates a certain part of the text as a rationale for a certain classification made by the human, this rationale is called an annotator rationale (Zaidan et al., 2007). In this work, annotator rationales are used as a benchmark of valid rationales. A rationale is valid if it represents a signal in the input data that ML models (or humans) can use to correctly classify on.

1.4 Rationale extraction

Not only humans can annotate text as rationales, but machines can too. I introduce rationales that are annotated by ML models as machine-annotated rationales (MaRs). Such a rationale is (part of) an explanation for a classification made by an ML model, annotated by the model.

In this work, I try to find rationales for predictions by ML models that classify text documents. Machine-annotated rationales are extracted using different rationale extraction (RE) methods: approaches to finding rationales for ML models. Some of these approaches can be applied to all ML models and are *model-agnostic*, while other approaches are only applicable to specially designed models and are *model-dependent*. Because of this, different ML models for classifying documents are investigated in this work. Four rationale extraction methods are introduced: the Feature Extraction method, the Leave-One-Out method, the BagNetsText model with two implementations called BagNetsTextAll and BagNetsTextRats, and the RationaleSearch model.

1.5 Measuring explanation quality

To verify the explanation quality of MaRs, three aspects are taken into account in this study: faithfulness, similarity to annotator rationales, and user-evaluated quality. The faithfulness of an explanation is determined by measuring changes in model output when the MaRs are omitted from the input.

To find out how similar the found MaRs are to human-annotated rationales, a set of annotator rationales is used as a benchmark. The set of MaRs is compared to the set of annotator rationales for a classification using set theory. The more overlap the two sets have, the more similar the sets are.

To measure how useful explanations are to users in real-life settings, a user evaluation is carried out. Explanations are useful in real-life settings when they are complete and comprehensible. A **complete** explanation is one that contains enough information for the receiver to be able to base a classification on. An explanation is **comprehensible** when a receiver can understand it. In the user evaluation, users are tasked to make classifications based on explanations and are asked to identify correctly classifying models by their explanations.

1.6 Context

In this work, I attempt to find explanations for **text classification** ML models, with end-users from the work field as receivers. Different models with different levels of transparency and complexity are used to perform the text classification task. Predictions are explained by finding sections in the input text that form rationales for the classification. I assume that certain sections in the document are signals that lead the model to a certain classification. Since the signals are in natural-language textual form, they are rationales for the classification.

Originally, the goal of this study was to find explanations for decisions by an ML model for risk analysis at the Dutch Police. The end-users were decision-makers that estimate the risk of a person, based on large sets of documents. To reduce redundant reading, an explanation that refers to a part of the document can be useful. By providing such an explanation, the decision-maker can be pointed to (parts of) documents that are necessary for a decision. This allows more focused reading since not all documents need to be read to make a decision. Due to complications with data availability, I use movie reviews and their polarity (positive or negative) instead

of police documents and risk analyses, but the main idea still holds. The end-users of the new classification task are movie review readers in general.

1.7 Research questions

The main objective of this study can be expressed using the following research question:

> 'How can faithful machine-annotated rationales for text classification be found that are complete and comprehensible, and do they form explanations for end-users similar to annotator rationales?'

The text classification task here is determining the polarity of movie reviews, and the end-users are human readers without knowledge about the used ML models. Polarity classification is a rather simple task that does not require additional experience or knowledge about the field and therefore anyone that can read English documents can be an end-user. Explanations in the form of machine-annotated rationales that are part of the input documents are used.

To answer the research question, the following sub-questions are used:

SQ1. Feature Extraction method

- (a) Are machine-annotated rationales that were found using post-hoc feature extraction
  - (i) faithful?
  - (ii) similar to annotator rationales?

SQ2. Leave-One-Out method

- (a) Are machine-annotated rationales that were found using the Leave-One-Out method
  - (i) faithful?
  - (ii) similar to annotator rationales?

SQ3. BagNets-for-text models

- (a) Are machine-annotated rationales that were found using the BagNetsTextAll model
  - (i) faithful?
  - (ii) similar to annotator rationales?
- (b) Are machine-annotated rationales that were found using the BagNetsTextRats model
  - (i) faithful?
  - (ii) similar to annotator rationales?

SQ4. Annotator rationale search model

- (a) Are machine-annotated rationales that were found using the RationaleSearch model
  1. faithful?
  2. similar to annotator rationales?

SQ5. Machine-annotated rationale comparison

- (a) How do machine-annotated rationales that were found using the methods in SQ1, SQ2, SQ3, and SQ4 compare on sentence set overlap?

SQ6. User evaluation

- (a) Do machine-annotated rationales that were found using the methods in SQ1, SQ2, SQ3, or SQ4 form explanations for end-users that are
  - (i) complete?
  - (ii) comprehensible?
  - (iii) similar explanations compared to annotator rationales in a blind study?

1.8 Structure

This thesis is outlined as follows: In Section 2 some background and related work on explainable AI can be found. Then, in Section 3, I explain some terminology that is used in the subsequent sections. After that, the dataset used in this work is described in Section 4. The following Section 5 describes classification models and the different approaches to finding machine-annotated rationales. Every rationale approach and associated ML model has an unique colour. In Section 6 the results of the different rationale extraction approaches are discussed. Section 7 contains the methods and results of the user evaluation. After that, the discussion of all results can be found in Section 8. This work ends with a conclusion in Section 9.

2 Background

In the following section related work on explainable text classification is described. The section starts with ML models and the task of text classification in Section 2.1. Then, some elements of explainable artificial intelligence are discussed in Section 2.2. I go into interpretability (Section 2.2.1), explainability (Section 2.2.2), types of explainability (Section 2.2.3), and rationales (Section 2.2.4). The section ends with approaches to explainable text classification in Section 2.3.

2.1 Machine learning models

An ML model can be used for decision-making. The use of an ML model might be advantageous in certain scenarios, e.g. to reduce costs, work, or fraud. A model can perform tasks faster than humans, taking into account more data, without getting tired, and is always available. When an ML model is used in the work field, the purpose of the model usually is to support human decision-makers, by performing a difficult, redundant, time-consuming, or just inefficient task, thus in a sense performing (part of) the task instead of the human (Bekri et al., 2019). An ML model could also give some new insights regarding the decision process. One example is discovering new features that are important for the decision making process. A distinction between **supervised** and **unsupervised** models can be made, specifically models that learn using labelled training data and model that find structures without the label in the training data (Hu, 2018). This study focusses on supervised machine learning, as the goal of the classification task is to learn to correctly classify, and not to learn new structures.

Text classification

Text classification is the task of determining the class of a given document or text (Korde, 2012), using the underlying features (Aggarwal et al., 2012). Common approaches are creating decision-rules based on expert knowledge, training a neural network, or using a Naïve Bayes classifier.

The classes of documents can for example be whether a movie review is positive or negative. The main process of text classification consists of the following steps (Korde, 2012):

1. **Document collection**: gathering datasets and labels.
2. **Preprocessing**: tokenisation, stop-word removal, and stemming words.
3. **Indexing**: transforming the text to a model input representation, like a vector space model.
4. **Feature selection**: filtering out irrelevant words using importance measures.
5. **Classification**: learn to classify documents using labels in the training data (supervised), or with a small labelled set and larger unlabelled set (semi-supervised).
6. **Performance evaluation**: measuring performance by calculating the error, fallout and accuracy. Possibly also precision, recall and F1 scores.

One note is that a simple ML model might give accurate output, but might be reasoning in a way that could be described as 'cheating'. For example, a sentiment analysis ML model might associate words like 'good' with positive sentiment, but that does not mean that it classifies on the semantic meaning of the sentence (*'this movie can't be called good'*). It only looks at single words, while one might expect the model to reason based on for example the sentiment of whole sentences or paragraphs.

Simple machine learning models like **decision tree**s, support vector machines (SVMs), and **naïve Bayes** classifiers can be used for text classification. Such models are interpretable, as long as they do not become too large (e.g. decision trees). Simple ML models can reach high accuracies but might reason in ways that could be described as 'cheating', as mentioned in the previous section. For example, a model that classifies documents with names of popular actors as positive is often correct, but is classifying on unjustified grounds (thus 'cheating').

More complex ML models might find more or different signals than simple models. A neural network simulates the human brain to some extent and maps input units to output units (Korde, 2012). This approach can perform well, but is very opaque, making it hard for humans to explain or interpret. Some research has been done on interpreting neural networks, for example, the attention mechanism as described in the upcoming Section 2.3.1.

2.2 Explainable AI (XAI)

When automated systems and machines are used as decision-makers in the work field, some insight into their behaviour can be desired. In specific fields where legal and ethical issues apply, it is vital to understand why a machine makes a prediction (Preece, 2018). For example, a judge should not trust a prediction by a future crime risk analyses model blindly, but evaluate the grounds on which the prediction was made on, and then make a judgment of conviction. The field of XAI focusses on problems related to the explanation of ML models.

Good performance does not necessarily mean that a model performs the task correctly. Lapuschkin et al. (2019) show that even if ML models achieve high accuracy, their tactic can be unintelligent (i.e. cheating), or that a model behaves very different from a human. Such a model is then right for the wrong reasons (Ross et al., 2017). In certain fields, like the medical or financial domain, the (correct) reasons behind a decision are required to prevent misuse or errors. For example, a medical diagnosis should be made on correct symptoms and not on statistical coincidences.

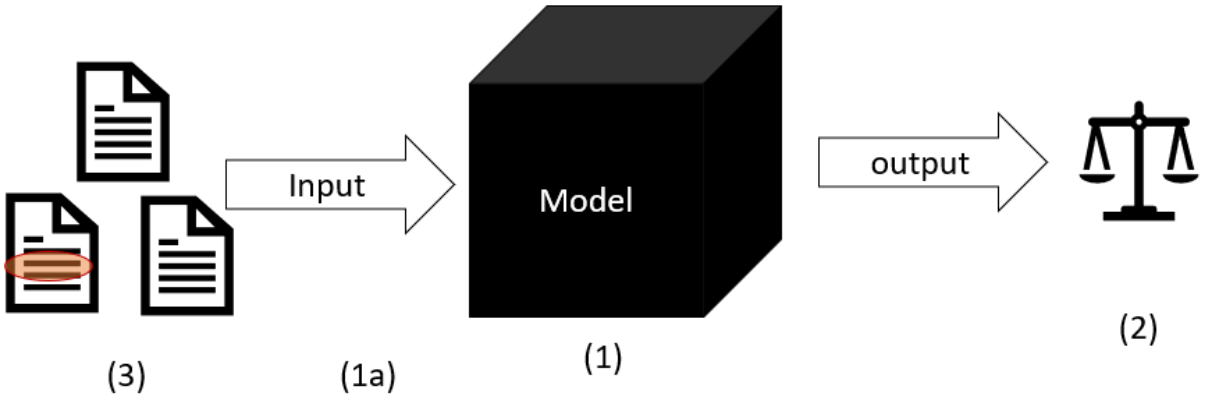
From a technical perspective, explainable artificial intelligence is also very useful for debugging models. Finding the critical points or cases where the model fails can point to technical incompetence or difficulties, and may give insight into the decision process.

In addition, I give three potential reasons for explaining models:

1. **Competency**: Showing that a model is competent enough to perform the task correctly.
   1.a Determining the inputs that cause the model to fail (or succeed) and why. This can be done when new information about the reasoning process of the model is found through the explanation.
2. **Proving output validity**: An output is correct because it can be explained.
3. **Gain insight into the classification task**: Explanations can be used for finding deciding features in the input data, and finding reasoning strategies for the classification task. If a simple ML model used for polarity classification is explained, and all explanations for positively classified documents contain certain words, a possible new causal relationship between those words and the positive class may be found.

An additional reason for explaining ML models can be increasing user trust in the model, or in the prediction (Ribeiro et al., 2016). In Figure 1 a visual overview of the above list is shown.

Figure 1: Potential reasons for explaining (black box) models



In the following sections, different aspects of XAI are discussed. Interpretability and explainability are described in Section 2.2.1 and Section 2.2.2. Then, in Section 2.2.3 I go into different types of explainability. After that, rationales are discussed in Section 2.2.4

2.2.1 Interpretability

Whenever a human user can understand how an ML model came to a certain prediction, the model is said to be interpretable. Miller (2017) describes **interpretability** as "*the degree to which an observer can understand*

the cause of a decision", thus enabling the observer to predict the decision based on the input. In this research, I will use this definition. Interpretability in AI is the extent to which a human can look at an algorithm or ML model and completely understand why certain inputs result in certain outputs. Thus, the observer (end-user) plays a major role in determining the interpretability of a model. Jacovi et al. (2020) address understandable ML models as 'inherently interpretable'.

When looking at how understandable ML models are to humans, a distinction between transparent models (very understandable) and black box models (not understandable at all) can be made (Lipton, 2016). When models are transparent, they are also interpretable, as long as they are not too large or too complex. A human needs to be able to grasp the concept of the model's reasoning. The term transparency is used to indicate how understandable a model's algorithm or parameters are to humans. Transparency consists of the following three levels (Lipton, 2016):

1. **Simulatability**: how well the human can reproduce the prediction given the model, input and parameters.
2. **Decomposability**: every calculation, input and parameter in the model has an explanation.
3. **Algorithmic transparency**: how well the inner workings of a model are understandable to humans.

When a model is the opposite of transparent, it is called opaque. Note that (complete) transparency is not required to interpret a model. Some general knowledge of the algorithm can also make it interpretable. For example, decisions made by humans might be interpretable, because usually an explanation can be provided for the decision (Lipton, 2016) (although not necessarily entirely truthful), but the inner workings of the brain remain unknown.

2.2.2 Explainability

An **explanation** can be defined as a justification of a certain action, that provides new information on the given action (Preece, 2018). An explanation has a sender and a receiver: the entity that is explaining a concept, and the entity that is the destination of the explanation. Senders for example can be ML models or humans, and receivers usually are humans.

In AI, **explainability** is used to describe the extent to which an ML model can be explained in human terms. The differences between interpretability and explainability is the level of understanding a concept. Interpretability focusses more on the general possibility of understanding a complete concept, not taking into account the level of complexity, while explainability focusses on explaining just enough to make a concept understandable. When a concept is too complex or opaque to be interpreted, it can still be explained. The explanation then uses relevant information to help the receiver of the explanation understand why event A caused event B. The amount of detail that the explanation contains, depends on the level of expertise of the receiver. For example, rain can be explained by understanding the entire interpretable meteorological process of rain, but also by the fact that it currently is the season for rain. Both explanations can be suitable for different situations.

In the field of AI, not all machine learning models are interpretable. To understand them, explainability needs to be added to such models. In the following sections, I go into transparent, post-hoc, local, and global explanations.

2.2.3 Types of explainability

A transparent ML model (see Section 2.2.1) is a model that is interpretable to humans (Doran et al., 2017). The explanation for such a model consists of the inner algorithm of the model, or an intuitive understanding of its reasoning (Lipton, 2016).

Post-hoc explanations An opaque ML model (black box) cannot be interpreted, but it could possibly be explained. When an explanation is not based on the model's inner algorithm (transparency) but found after a prediction is made, it is called post-hoc (Bekri et al., 2019). A **post-hoc** explanation is a more simplified version of an explanation based on the complete reasoning process of a model. A post-hoc explanation does add some new information on why the prediction is made. This new information tells something about the reasoning process of the model that was unknown before the explanation was given. An example of a post-hoc explanation is a human verbal explanation (Preece, 2018), which is given after an action is done and gives

insight into the cause. Post-hoc explanations can be used to explain black box models, as shown by Ehsan et al. (2019), Bekri et al. (2019) and Ribeiro et al. (2016).

Faithful explanations A post-hoc explanation might not explain faithfully. A faithful explanation is one that explains the prediction process according to algorithmic transparency (see Section 2.2.1), thus truthfully (according to the true reasoning of the model) and using all necessary information (Jacovi et al., 2020). When a post-hoc explanation does not explain according to what exactly happened inside the model, it is an unfaithful explanation. An explanation that is not **complete**, is one that does not contain all necessary information for a certain prediction. When post-hoc explanations are simplified versions of transparent explanations, they may not be complete, since not all information is included.

Post-hoc explanations are not ideal in situations where the model's inner algorithm needs to be interpreted or explained, because they can be misleading (Lipton, 2016). A post-hoc explanation can be **feasible**, giving insight, but not necessarily complete or truthfully so, thus misleading the receiver. An explanation can be very feasible, but not faithful (Lipton, 2016). A feasible explanation can for example not take into account certain information, or point to a correlation instead of a causal relation. Note that in some scenarios, a feasible explanation might be sufficient, for example, if it is one of the possible explanations for a decision. Jacovi et al. (2020) add another dimension, termed **plausibility**, which refers to how convincing an explanation is to a human. Such an explanation can be unfaithful but feasible. Jacovi et al. (2020) warn that feasible explanations may mislead future users of the system, creating false trustworthiness, which is undesirable in sensitive work fields like the legal field.

Proving that an explanation is *not* faithful is more easily done than proving that it is faithful. Jacovi et al. (2020) describe common assumptions in proving faithfulness: (1) models that predict similarly should give similar explanations, (2) similar input-outputs should have similar explanations and (3) **the Linearity Assumption**: different parts in the input are independent of each other and influence a prediction. Thus, erasing relevant (explaining) parts of the input will result in different (wrong) predictions. While these assumptions might be true, there is no universal proof for them. They might apply to certain ML models, but it remains unknown which ones exactly. Jacovi et al. (2020) argue that a definition of **sufficient faithfulness** is necessary to evaluate ML model explanations on their faithfulness. Such an evaluation should then indicate how faithfully useful the explanation is in a certain work field. Complete and verified faithfulness might not be necessary in for example recommendation systems.

In addition, Jacovi et al. (2020) propose five guidelines for evaluating faithfulness:

1. Be explicit in what you evaluate: distinguish between *faithful* and *plausible* explanations.
2. Faithfulness evaluation should not involve human judgement on the quality of interpretation: humans are not always capable of distinguishing between plausible and faithful explanations and should therefore not *assess* the faithfulness of explanations.
3. Faithfulness evaluation should not involve human-provided gold labels: using human-provided gold labels will shift the focus to *plausible* explanations instead of faithful ones.
4. Do not trust "inherent interpretability" claims: not all models that are inherently interpretable (understandable to humans) are faithful.
5. Faithfulness evaluation of Intelligent User Interfaces (IUI) systems should not rely on user performance: such evaluations only measure correlations between the plausibility of explanations and model performance.

While human judgement should not be used in faithfulness evaluation, human input can be used to evaluate the fidelity of explanations. **Fidelity** is the degree to which an explanation method can successfully mimic a model's predictions in terms of accuracy (Jacovi et al., 2020). The accuracy is measured as the number of times that an explanation can be used to come to the same prediction as the model. Using *forward simulation*, where users are asked to simulate a (to the user unknown) model by predicting its output using only inputs and explanations, the fidelity of an explanation method can be measured. Nguyen (2018) applied forward simulation to evaluate explanations and found moderate correlations between human-evaluated and automated measures of faithfulness.

Explanations are not always either transparent or post-hoc. In Rudin et al. (2018), a black box model is used to predict a scoring-system-like output that is similar to more transparent scoring systems, in a way providing an output that 'explains' the final prediction. The output consists of scores for different scoring features, and the sum of these scores in combination with a total score threshold is used to make a final classification. See Figure 2 for an example output. The model learns to assign the scores (or the risks in the advanced version), that can be combined to make a prediction. This solution is not transparent, but also not post-hoc, since it does not explain the model itself and does not generate an explanation after the prediction is made. Using the output, a decision can be made, making the prediction decomposable (see Section 2.2.1).

Figure 2: Example of a prediction containing explanation: scoring features (Rudin et al., 2018). The sum of points is used to make the final decision.

1.	Prior Arrests ≥ 2	1 point	$\cdots$
2.	Prior Arrests ≥ 5	1 point	$+ \cdots$
3.	Prior Arrests for Local Ordinance	1 point	$+ \cdots$
4.	Age at Release between 18 to 24	1 point	$+ \cdots$
5.	Age at Release ≥ 40	-1 points	$+ \cdots$
		SCORE	$= \cdots$

SCORE	-1	0	1	2	3	4
RISK	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Local and global explanations **Local explainability** of a model is the connection between a certain input and a certain output (Mohseni et al., 2018). Thus, a local explanation explains one prediction of the model based on one input. In Ross et al. (2017), local explanations are generated by shrinking irrelevant input gradients using the loss function. Local explanations can be **model-agnostic**, meaning that they can be used to explain predictions for different models. For example, LIME is a model-agnostic local explanation generator proposed by Ribeiro et al. (2016). An explanation that explains a model or a set of instances of a model, is called **global** (Mohseni et al., 2018). **Global explainability** applies to an entire model, and not to specific predictions.

2.2.4 Rationales

A rationale is a human-understandable natural language explanation for an action or decision (Ehsan et al., 2019). In XAI, rationales can be used to explain predictions by ML models. These rationales can be generated post-hoc (Ehsan et al., 2019), but can also be part of the model input, for example, part of the text used as input (Zaidan et al., 2007). Rationales can be local explanations, explaining outputs by inputs, but also more global, for example explaining model states.

Rationales can also be (post-hoc) generated using machine learning, for example by training a model on thinking-out-loud human sentences and machine decisions (Ehsan et al., 2019). Such rationales are not based on the decision process of the model (not faithful), since they do not represent the entire decision process, but do give a feasible explanation for a prediction that might be considered useful. For example, some form of human-like rationale could improve the trust of users in the competence of a model (Weitz et al., 2019), even it is not faithful.

Die hard 2 is an altogether unfortunate fiasco, inferior to the original in every respect. Place the blame squarely on the shoulders of Steven de Souza and Doug Richardson, who wrote the film's pathetic screenplay. Every line of dialogue reeks of either smarmy sap or forced humor.

The plot is altogether implausible; the convoluted story line involves a band of terrorists who take over Dulles airport and shut down the control tower, leaving a dozen planes stranded in the air waiting to land. The film has zero credibility, and all of the characters come off as cliched, cardboard cut-outs. So much for the script. How about the action? Well, let's put it this way : director Renny Harlin could learn a few things from john McTiernan, who directed the original, as well as the hunt for red October and predator–all standouts for their hair-raising suspense. By contrast, Harlin doesn't have a clue when it comes to choreographing action, and consequently, die hard 2 never picks up steam. Die harder is impossible to take seriously even for a minute. In fact, the movie often seems deliberately campy, and it almost reaches the threshold of being so bad it's good. You do laugh, but you laugh at the film, not with it. Die hard 2 should have never been cleared for takeoff.

Figure 3: Example of annotator rationales from the dataset used by Zaidan et al. (2007). The underlined (sub-)sentences are annotator rationales. The classification output is that the review is negative.

In Zaidan et al. (2007), annotator rationales are used to improve text classification using an SVM. These rationales are gathered from human users, that were asked to annotate 'why' a certain classification should be made. When a rationale is annotated in the text by a human, it is called an annotator rationale (Zaidan et al., 2007). In Figure 3 an example of annotator rationales for a 'negative' document classification is shown. In the SVM, these annotator rationales are then omitted from the original input, and the classification using that masked-out input should be less confident than the input with the rationales. This idea is then translated into adding a constraint that takes into account the added value of a rationale and enforces weights to adjust to the rationale values. Adding the constraint improves classification performance slightly, and removing rationales from the input decreases performance significantly.

Using annotator rationales as explanations can add interpretability to a model that is both human-understandable (text) and faithful, since they are signals that the prediction is based on. Therefore, annotator rationales are very useful for explaining text classification.

2.3 Approaches to Explainable AI

In the following sections, I give an overview of some approaches to explaining ML models. Approaches of XAI in general, faithful XAI, and XAI for text classification are discussed.

To explain ML models, different approaches can be taken. Some models satisfy transparency (see Section 2.2.1), like (small) decision trees, rule or decision-based models, linear models and Naïve Bayes classifiers. Such models are simple enough for humans to understand, but do not always perform as well as more complex models (e.g. neural networks).

Combining two models to add interpretability is a way of achieving XAI. **Hybrid explainable models** use a complex and a simpler model to perform a task. Using a k-nearest neighbour algorithm to find support for a complex-model prediction (Papernot et al., 2018), or using a deep neural network to learn features and using a softmax function to make the classification (Brendel et al., 2019) are examples. This last approach is adopted in this study, as further described in Section 5.2.2.

Another approach may be learning a more transparent (and thus interpretable) model from a more complex model, like learning a decision tree from a neural network (ANN-D, Schmitz et al. (1999)). In Caruana et al. (1999), a trained model is used to find training inputs similar to a new case, generating case-based explanations by pointing to similar cases in the training dataset. Such an explanation can be seen as an explanation based on experience from the past, which can be useful in for example the medical domain (Caruana et al., 1999).

Machine learning models can also combine explanation and prediction in the output. TED by Hind et al. (2018) is an example of such a model. The TED model trains on class labels and explanations and predicts both a class and an explanation. These explanations are not tied to the output class, and thus not faithful. There is also some research on looking into the different states or layers of models, and generating feature sets for them (Zhang et al., 2017). Attention mechanisms as described in Section 2.3.1 might give some insight into the deciding features of a neural network.

The final approach that I will mention in this section, is explaining a model by adjusting the model's inputs. Zaidan et al. (2007) show that adjusting weights for deciding input features improves classification. Masking certain input features and comparing the prediction can also be used to find deciding features. This method makes use of the Linearity Assumption and does not explain the inner workings of the model, but might give some insight into causal relationships, since inputs can be connected to outputs.

2.3.1 Attention

Some models are too complex to be interpretable to humans, like neural networks. One approach to interpreting black box models is the attention mechanism.

The attention mechanism, introduced by Bahdanau et al. (2014), was used initially for word-to-word alignments to translate texts to other languages. The model predicted words based on a subset of selected input words, which were selected by searching for words that had strong alignment with the next word to be predicted. Using this method, the words in the input that are responsible for the predicted output can be found. Thus, an explanation for the prediction can be generated, by looking into the model's algorithmic process. Note that this method is faithful according to the Linearity Assumption (see Section 2.2.3) since a relation between parts of the input and the output is assumed.

The attention mechanism is not restricted to models with textual inputs. Xu et al. (2015) introduce an image-to-text model that generates natural language sentences explaining images. Parts of the image are assigned to predicted words in the sentence and form an explanation for the word. Not all explanations are useful, however. For example, words like 'a' should not point to any part of the image.

The question of whether attention is explanation has been a discussion in literature Jain et al. (2019) and Wiegreffe et al. (2019). One of the arguments is that results of attention mechanisms are not always consistent and therefore not necessary faithful explanations. Looking from the viewpoint of explainable AI, attention mechanisms can be used as explanations for less strict definitions of explainability (plausible rather than faithful) (Wiegreffe et al., 2019). In this study, I do not use the attention mechanism for finding explanations, because of their possible unfaithfulness.

2.3.2 Faithful explainable AI

Some approaches to explaining black box models focused on faithfulness can be found in the literature. Lakkaraju et al. (2019) find interpretable, faithful and global explanations for black box model behaviour using if-then rules pre-defined by users. Faithfulness is regarded as fidelity by Lakkaraju et al. (2019) and is measured by counting the number of cases that the if-then rules do not apply to model outputs.

In Ross et al. (2017), faithful local explanations are found by shrinking irrelevant input gradients. These explanations are said to be faithful, since the explanations are found using the input gradients and thus make use of the model's inner algorithmic process.

Zhong et al., 2019 show that untrained attention mechanisms (see Section 2.3.1) do not faithfully explain, but that training an attention mechanism using annotator rationales (see Section 2.2.4) can add faithfulness. The attention of a model is made to match the attention of a human, using provided annotator rationales. The faithfulness is then measured by counting the number of words that received higher attention weights than the word that caused the highest change in prediction confidence (i.e. the most influential word). A more fine-grained measure also takes into account the total attention weights of the words ranked above the most influential word.

2.3.3 Explainable AI for text classification

The following section contains some approaches to explainable text classification found in literature.

Explainable text mining using deep neural networks has been investigated in Raaijmakers et al. (2017), where the k-nearest neighbour algorithm is used to find representative and explanatory training data to compare the new input to, based on the neural network layer actions, and semantic document similarity.

Liu et al., 2018 use a hybrid generative-discriminative method to generate fine-grained information alongside a prediction. This fine-grained information is used is a summary of an original textual document. These local explanations can help interpret the prediction, but are not faithful.

Clos et al., 2017 show that a text classifier based on learning lexicons and modifier terms can be used to classify texts, creating an interpretable white-box alternative to black-box classifiers. In a paper about the use of predictive coding to find rationales, multiple models are used to extract semantic and syntactical values from documents that form explanations (Chhatwal et al., 2018).

Zaidan et al. show in their 2007 (Zaidan et al., 2007) and 2008 (Zaidan et al., 2008) papers that annotator rationales can be used to improve classification using a support vector machine. In the 2007 paper, the weights of the model are adjusted to the annotator rationales, and in the 2008 paper, an improved generative approach is taken. Yessenalina et al. (2010) use an ML model to automatically generate annotator rationales.

In Robnik-Šikonja et al. (2008), local explanations are found by removing words from the input and measuring the effect on the prediction. Lei et al. (2016) propose an encoder-generator method for extracting rationales, where sentences are encoded according to their class. A subset of rationales is then extracted by the generator and used as explaining summary. Possible rationale-words are found without explicitly training on rationales, but by regularizing the model by desiderata for rationales. A similar two-model approach is used by Jain et al. (2020), where one model is trained for finding rationales and one for classification. The labels for rationales are given to the first model during the training process.

3 Definition of terms

In this section, a short summary of the terms used in this study is given. Some new terms are also introduced.

An **explanation** can be defined as a justification for a certain action, that provides new information on the given action (Preece, 2018). An explanation should be sufficiently explaining to the receiver. For this study, I suggest the following features for a **sufficient** explanation:

- **Sound**: no contradictions (Miller, 2017).
- **Complete**: all relevant causes are present (Miller, 2017).
- **Comprehensibility**: simple enough to be comprehensible for the receiver.

Furthermore, the goal of an explanation is to give the receiver insight into the causes of the event that is explained. In this study, the focus lies on complete and comprehensible explanations, because these features can be most concretely measured through a user evaluation.

An explanation can be based on **transparency**, meaning that it is based on the inner workings of a model. Such an explanation is called **faithful** for it is according to the algorithmic process of the model. **Post-hoc** explanations are created after the prediction has been made, by trying to derive what happened during the prediction process. A post-hoc explanation might not be faithful, as incorrect features of the model may be used in the explanation. An explanation is **local** if it explains a given prediction. A **global** explanation explains a whole model, not taking into account separate classification cases.

As described in Section 2.2.4, an **annotator rationale** is a human-annotated (sub-)sentence in a text that explains a classification. Annotator rationales are local explanations because they explain predictions. Rationales can also be annotated by a machine. I define a machine-annotated rationale (MaR) as an annotator rationale that was annotated by a machine for a given classification prediction. In this study, machine-annotated rationales are used as local explanations for text classification predictions. The explanation is local because parts of the input for a prediction are used in the explanation, thus making the explanation specific for that case. Both annotator rationales and machine-annotated rationales are a form of rationales, but with different annotators. In this study, I use the term rationale when referring to either subtype.

The input format of a text classification model may differ in size and content. The **input chunk size** can consist of paragraphs, sentences, sub-sentences and words.

3.1 Faithfulness

As described in Section 2.2.3, explanations are not always faithful. A transparent explanation is faithful because it is based on the model's inner algorithm and thus explains the reasoning behind a prediction. When explanations are post-hoc, it is more complex to determine whether they are faithful. Jacovi et al. (2020) warn that faithfulness should not be measured through user judgement, because in some settings users cannot distinguish between plausible and faithful explanations.

In this study, I adopt **the Linearity Assumption** (Jacovi et al., 2020): certain parts of the input are responsible for the output. For documents, this means that some sentences contain more relevant information than others. I assume that faithfulness of explanations for textual input can be measured by omitting (erasing) parts of the input and measuring the effects on the prediction. In addition, I assume that a prediction for textual documents does not depend on only one input feature, but on a combination of features. When the task is to determine whether a document is positive or negative about a subject, multiple positive and negative signals (features) might be included, but the document cannot be both positive and negative.

In this work, I measure the faithfulness of the explaining features by completely omitting features and re-predicting the class of the document. When the relevant features are not present any more, a model should predict differently if the explanation is faithful. This difference is expressed in a score and can be seen as a version of sufficient faithfulness as proposed by Jacovi et al. (2020) (see Section 2.2.3). In Section 6.3.2 this score is described in detail. In this work, I only use the score to measure how well the Linearity Assumption applies to an explanation. Features in the form of whole rationale sentences are used.

3.2 Rationale quality

The quality of an explanation is in part determined by the receiver: in case of this study, a human. In the above section, I suggested three criteria of good explanations: soundness, completeness and comprehensibility. These criteria are respectively measured through faithfulness, completeness, and interpretability in this work. The faithfulness of an explanation will ensure that it is sound, for a faithful explanation will support a classification and not contradict it. An interpretable explanation is one that a receiver understands well enough to come to the same classification result.

In this study, two approaches to measuring the quality of found machine-annotated rationales are taken. The first approach is described in Section 3.2.1 and uses a benchmark of annotator rationales as a comparison. The second approach measures subjective quality through human judgement and is described in Section 3.2.2.

3.2.1 Similarity to annotator rationales

In this study, the quality of machine-annotated rationales is measured by comparing them to annotator rationales. The annotator rationales are used as the state-of-the-art explanation for a document classification These rationales are not necessarily the only or most correct explanation available. This study focusses on explanations that are complete and comprehensible to users, and therefore annotator rationales are a good benchmark since they are rationales by humans for human receivers. It is assumed in this study that the annotator rationales are both sound, complete and comprehensible.

The rationales are compared to each other using set theory: the set of machine-annotated rationales is compared to the set of annotator rationales. The more similar the sets are, the better the machine-annotated rationales are. An ML model does not necessarily explain itself as a human would, so a dissimilarity between the two sets of rationales is not always a bad thing. A difference in rationales can show that a model classifies using different signals or a different number of signals. Such machine-annotated rationales can also explain the model and its prediction, and therefore can still be a good explanation. In Section 6.3.1 I go further into measuring the similarity of rationales.

3.2.2 Subjective quality of explanations

Apart from using annotator rationales as a benchmark for comparison, human judgement can be used to measure the (subjective) quality of machine-annotated rationales as explanations. How a human receiver views an explanation cannot be measured with some mathematical formula, but a couple of different scenarios can be considered. In this work, the following scenarios are taken into account:

1. **Good explanation**:
   (a) Correct classification: the explanation supports a classification that is correct.
   (b) Incorrect classification: the explanation supports a classification that is *not* correct.
2. **Incomplete explanation**: the explanation does not contain enough useful information to support any classification. The receiver does not understand why a classification would be made given that explanation since the explanation does not *support* any classification result in a human-understandable manner.
3. **Non-interpretable explanation**: the explanation does not correctly indicate why a classification is made. This is the case when an explanation does not show fidelity, meaning that a human would come to a different prediction than the model, based on the explanation. The explanation method does not mimic the ML model successfully. Such an explanation is not interpretable because it does not make any sense to the receiver and might even confuse the receiver. The explanation is then *misleading*. An explanation that is not interpretable might also not be faithful (see Section 3.1), in cases where the explanation is not conforming to the model's algorithmic process. This can usually not be discovered by the human receiver of the explanation when the explanation method is a black box. The main difference between non-interpretable and non-complete explanations is that a non-complete explanation will not give enough information, and a non-interpretable explanation will give misleading information. Note that if a model misclassifies and the explanation supports that prediction, the explanation is still adequate.

The criteria mentioned at the beginning of this section (sound, complete and comprehensible) can be distributed over the above scenarios: A good explanation is sound, complete and comprehensible. An incomplete explanation is not complete, and a non-interpretable explanation is either not sound or not comprehensible.

A sentence that is a rationale is naturally comprehensible, because of its natural-language form. It is more understandable to humans than for example an explanation in the form of a set of model state values. When a sentence does not contain information that points to a classification (not a rationale), it is not comprehensible. To be comprehensible, an explanation needs to explain an event.

4 Dataset

The dataset used for training and rationale extraction is the IMDB[1] movie review dataset enriched by annotator rationales by Zaidan et al. (2007)[2]. This dataset consists of 1000 positive and 1000 negative textual reviews on movies from the polarity dataset (v2.0) from the Movie Review Data set by Pang et al. (2004). The enriched dataset contains annotator rationales for every document. These annotator rationales are defined as follows:

> "Basically, 'rationales' are segments of the text that support an annotator's classification. Let's say we have a movie review that is labelled as positive (i.e. the writer has a favorable opinion of the movie). Then the rationales would be segments of the text that support the claim (by an annotator) that the review is, indeed, positive."(Zaidan et al., 2007)

The annotations were done by human 'rationale annotators', who were asked to highlight words and phrases that justified a given positive or negative classification. Only rationales for the requested classification were required. The number of rationales selected depended on the annotator, who was requested to mark enough rationales to provide convincing support for the class of interest. The selected number of rationales completely depends on the judgment of the annotator. In Figure 4 the rationale distribution and ratio of the documents is displayed. The average number of rationales is 8.55. In this study, the whole sentence around the rationale is used as rationales, reducing the mean to 8 rationales per document, which can be explained by the occurrence of multiple rationales in the same sentence. The test set only contains 7 rationales per document on average, and this number is used further in this study as a leading rationale set length.

The dataset is split up into the following sets:

1. Train set: 600 positive and 600 negative annotated movie reviews. Used for training models.
2. Tuning set: 100 positive and 100 negative annotated movie reviews. Used for model accuracy validation during training.
3. Test set: 200 positive and 200 negative annotated movie reviews. Used to evaluate a trained model.
4. User study set: 100 positive and 100 negative movie reviews. These reviews do *not* contain annotator rationales and are used in the user evaluation.

[1] Internet Movie Database (Miller et al., 2009)
[2] http://www.cs.jhu.edu/ ozaidan/rationales/

Figure 4: Rationale statistics for the original dataset (1800 documents) and re-annotated documents (30 documents).

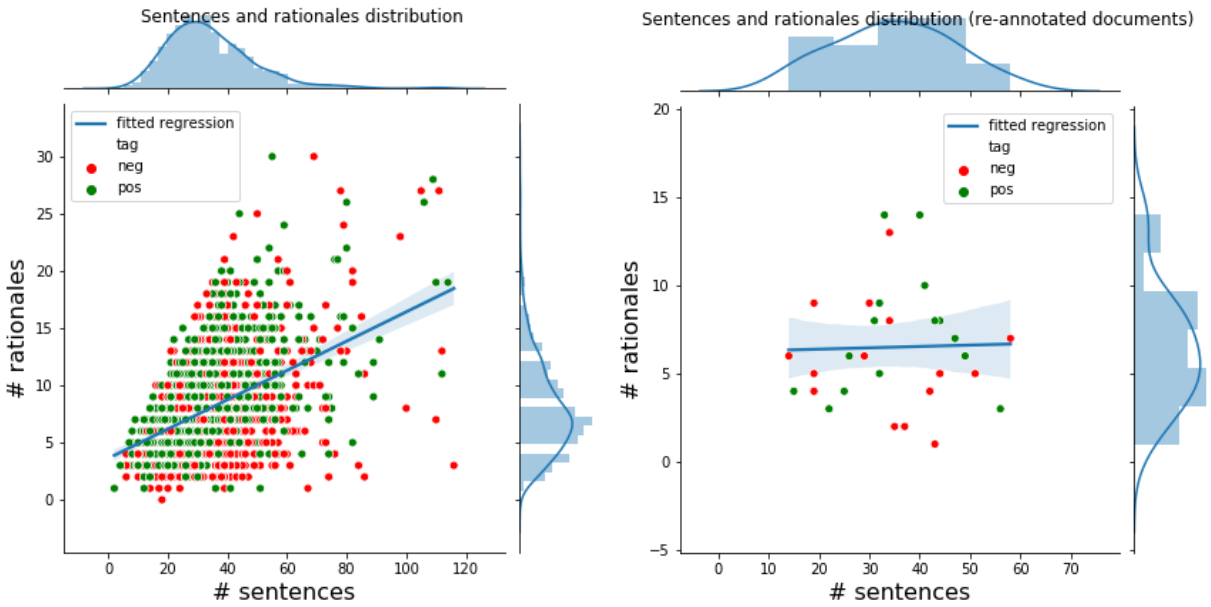


Figure 5: Distribution of $\frac{\#\ rationales}{\#\ sentences}$ ratio. The blue area around the regression line is the confidence interval.

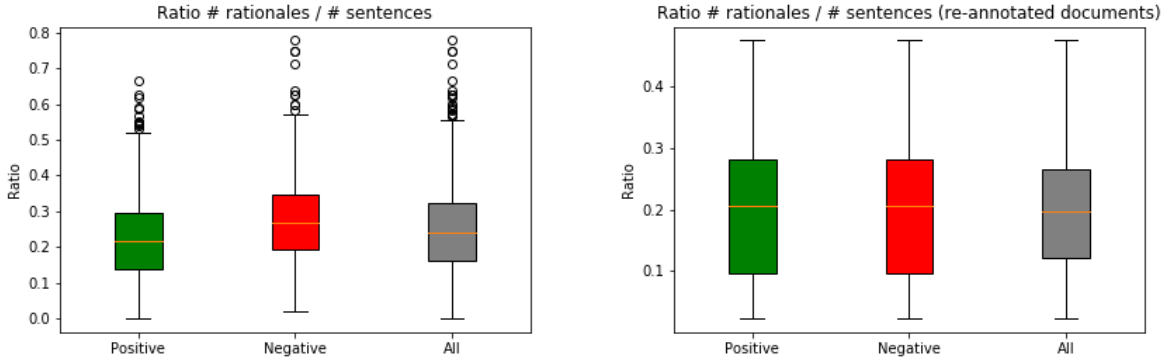


Figure 6: Ratio of $\frac{\#\ rationales}{\#\ sentences}$ boxplot.

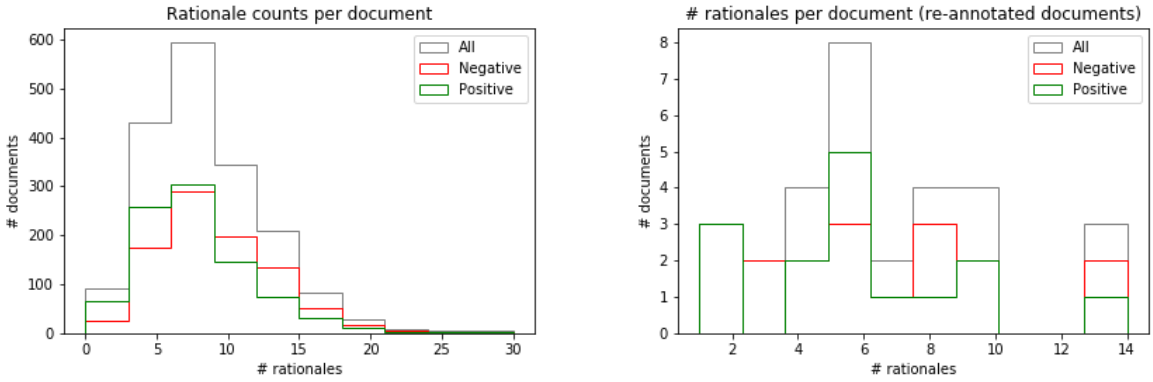


Figure 7: Histograms of rationale counts.

5 Methods

The goal of this study is to find faithful machine-annotated rationales, and can be divided into two steps:

1. **Classification**: Determine the class, positive or negative, of a given textual document using a machine learning model. Different types of models will be used to perform this step.
2. **Explanation**: A classification is based on signals in the input. These signals form an explanation for the classification. Whether such an explanation is intelligent and understandable to the receiver, depends on the situation. An example of an unintelligent but understandable explanation is that documents of a certain length are always positive. While such a model can be explained and might achieve a good accuracy, it is not the type of explanation this study focusses on. Instead, explanations in the form of rationales that are part of the input text are used. Such a rationale can give insight into which sentences, sub-sentences, or words had a (strong) influence on the classification. The rationales represent the areas in the input that contain signals pointing towards a class. In this study, rationales in the form of whole sentences are used for simplicity.

This section is structured as follows: In Section 5.1 the preprocessing of the dataset is described. Then in Section 5.2, the ML models used in the first step are introduced. Section 5.3 goes into different approaches to finding machine-annotated rationales.

5.1 Preprocessing

To use textual documents as input for a machine learning model, the documents need to be processed to a compatible format. In this work, a format that maintains sentence structure is used to keep the input decomposable. Documents are split into sentences using the NLTK (Loper et al., 2002) English punkt tokenizer[3]. The annotation tags ($< POS >< /POS >$ and $<$ NEG $><$ /NEG $>$) are removed from the text. All other words and punctuation except for repeating dots (...) are left in the documents.

Sentences are embedded using the Sentence-BERT (Reimers et al., 2019) embedding model. This embedding model encodes on sentence-level and not on word- or document-level like the regular Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018). Sentence-BERT uses the BERT (Devlin et al., 2018) model to encode text and applies pooling to the output to derive semantically meaningful fixed size sentence embeddings (Reimers et al., 2019).

Every document is padded with as many sentences as needed to create similar-size documents. All documents are transformed into a format based on the dimensions of the longest document in the dataset, and padding is added to every shorter document. See Appendix A.1 for more detail on the used padding.

5.2 Step 1: Classification

In this section, I will discuss all classification models used in step 1 of this study. Different models with different levels of complexity are examined. The more complex models (neural networks) might not necessarily perform better at classifying documents, but it might be more interesting to generate explanations for them, because of their opaqueness. All models (except for the LinearSVC model) use the Binary Cross-Entropy loss function (Plunkett et al., 1997) and the ADAM optimizer (Kingma et al., 2014) from the PyTorch library (Paszke et al., 2019).

5.2.1 Classification baselines: LinearSVC (LSVC) and SimpleNet

The movie review dataset by Zaidan et al. (2007) has been used in other work on text classification. Timmaraju et al. (2015) achieved 86.49%, 83.94%, and 83.88% accuracy using respectively a linear SVM model, a 2-layer neural network, and a recurrent neural network. Chintala (2012) reached an accuracy of 76.67% using a convolutional network. Narayanan et al. (2013) accomplished an accuracy of 88.80% using a Naive Bayes model with feature selection: removing redundant features from the input. Jain et al. (2020) reached an accuracy of 94% using a two-model method where annotator rationales were used to improve the classification accuracy.

[3] tokenizers/punkt/english.pickle

The Linear SVM and 2-layer neural network from above-mentioned literature are able to solve the classification step with an accuracy of $\geq 80\%$. I use both models as a baseline in this study, because these models are simple but accurate compared to other above-mentioned models.

Sklearn's CalibratedClassifierCV for LinearSVC (LinearSVC)

A Linear Support Vector Classification model from Sklearn (Pedregosa et al., 2011) called LinearSVC[4] is used as a simple interpretable baseline model. This model predicts a binary output by separating inputs on hyperplanes. Because the model learns linear relations between input features (words), it can be interpreted by inspecting the trained model's weights.

This model does not use the Sentence-BERT embedding as described in Section 5.1. Instead, a CountVectorizer class is used to encode the documents: all documents are converted to a matrix of token counts. Every word in the dataset has its own column, where the rows represent documents and the cells contain counts of words. The CalibratedClassifierCV object is used as a wrapper for the Linear Support Vector Classifier (LinearSVC) to find probabilistic outputs. The model is trained using the squared hinge loss function.

Simple neural network (SimpleNet)

The second baseline is a black box model, meaning that it is not interpretable and that the reasoning behind a prediction is opaque (see Section 2.2.1).

The SimpleNet model is a two-layer neural network consisting of an input of 768 (Sentence-BERT embedding) $\times$ 116 (sentences) dimensions and an output of 1 dimension. The input is passed through a layer that flattens the input, one linear layer ($768 \times 116 = 89,088$ to 1) and a sigmoid activation function. See Figure 8 for a visualisation of the SimpleNet model.

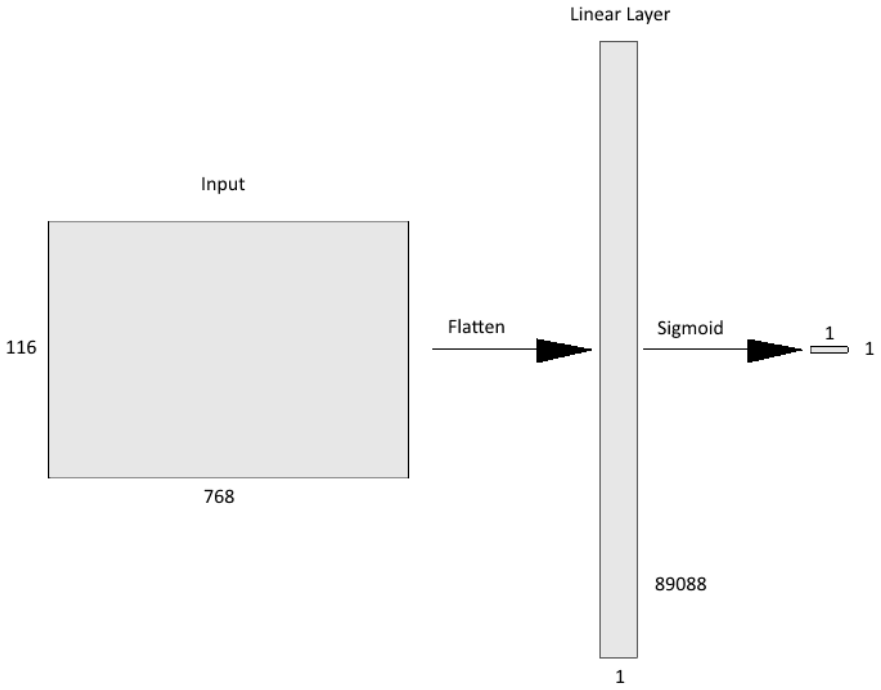


Figure 8: Visualisation of the SimpleNet model architecture.

5.2.2 BagNetsTextAll model

The two baseline models are either a black box (SimpleNet) or very simple (LinearSVC). To find transparent local explanations, I propose a new model. There are two variations on this model. This section introduces the model that uses all sentences (BagNetsTextAll) in the text, and the next section the model that uses a subset marked as machine-annotated rationales (BagNetsTextRats) to base a prediction on.

[4]Default parameters are used for the model.

The BagNetsTextAll (BNAll) model uses an architecture similar to the BagNets (Brendel et al., 2019) architecture: the classification is done by dividing the input into multiple chunks. Then, the outputs of all chunks are combined to find a final prediction. The original BagNets model was developed for image classification. I adapt the BagNets model to classify based on text, by using text chunks instead of image chunks. This model is called BagNetsText. For simplicity, the chunks are in the format of sentences. See Figure 9 for a visualisation of the BagNetsTextAll model.

In the BagNetsTextAll model, every Sentence-BERT embedded sentence in the document is passed through multiple convolutional layers, which are at the end passed through a linear layer. This linear layer gives one output for every sentence in the document. To make a prediction, the 116 outputs are put through a sigmoid function, and the average of those values is used to decide the class. Eq. (1) shows the formula for the final prediction μ by the BagNetsTextAll model, where S is the set of sentences, $\hat{S}$ is the output of the BagNetsTextAll model and $pred : s \to \hat{s}$ is the bijective function of the prediction $\hat{s} \in \hat{S}$ for sentence $s \in S$. Note that the padding is also included in S.

$$\mu = pred'(S) : \frac{\sum_{s \in S} pred(s)}{\|S\|} \tag{1}$$

Some adjustments are made in the training process of the BagNetsTextAll model. The BagNets model by Brendel et al. (2019) is trained to predict using small image chunks that are combined in the last step to form a prediction. Every chunk receives feedback during the training process. In the dataset used in this study, this information is not available for independent sentences, only for whole documents. Therefore, this feedback can not be provided in the training process. To solve this problem, the BagNetsTextAll model learns differently: the feedback is given after the final prediction after the sentence chunks are combined. Thus, the BagNetsTextAll learns the polarity of sentences, with the document class as a label. In a way, the model trains on noisy labels, because the class label does not apply to all input sentences, like padding or neutral sentences.

For the BagNetsTextAll model, the following steps are taken:

1. Input the Sentence-BERT embedded vectors.
2. For every sentence s, within the model, do:
   (a) Transpose the 116×768 vector to 768×116 vector for the convolutional layers.
   (b) Pass through 6 1D convolutional layers[5], consisting of 768 input channels and 768 output channels. The kernel size is 1 to maintain differentiation between sentences.
   (c) Transpose the 768×116 vector back to 116×768 vector for the linear layer.
   (d) Use a linear layer with 768 inputs and 1 output to generate a single output $\hat{s}$ for the sentence.
3. Use the sigmoid function on all 116 sentence outputs $\hat{s} \in \hat{S}$ to find a class value for every sentence.
4. Take the average μ of all sentence outputs $\hat{S}$.
5. Compare the μ with the class label and use this in the Binary Cross-Entropy loss function and ADAM optimizer to update the model's gradients. The μ is the final prediction.

5.2.3 BagNetsTextRats: BagNetsTextAll with restricted input

Not all sentences in a preprocessed document are useful for the classification. For example, a padding sentence does not contain any information, and some sentences can be neutral. I introduce the BagNetsTextRats (BNRats) model as a variation of the BagNetsTextAll model, where instead of using all sentences, only the sentences marked as relevant (rationales) are used to make the final prediction. When the neutral sentences are not taken into account for the final prediction, the prediction base will be less noisy, which reduces the loss of the final model output. After step 3 in the BagNetsTextAll model, continue as follows:

4. Take the average μ of all sentence outputs.

[5]The number 6 is chosen because of a time-accuracy trade-off during development.

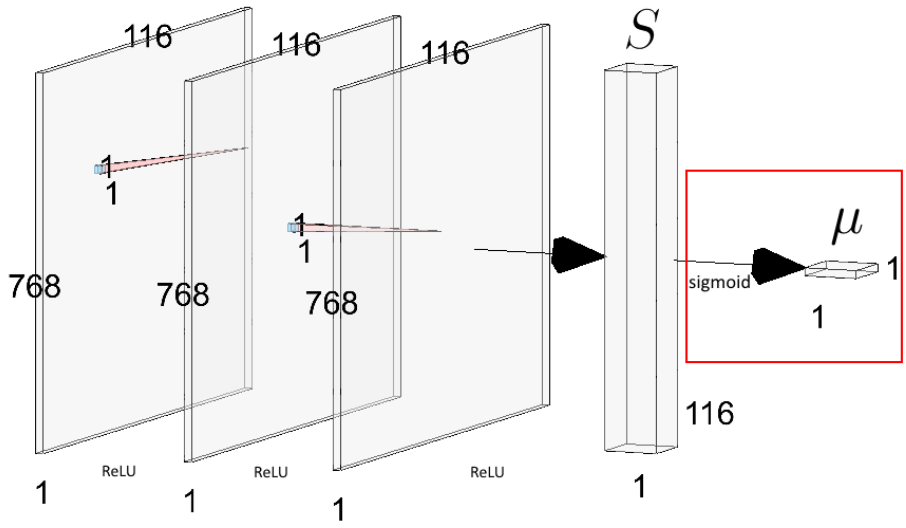


Figure 9: Visualisation of the BagNetsTextAll model architecture. The area in the red square marks the steps taken after the neural network's output is gathered (step 3).

   (a) Find sentences that support[6] the found rounded μ. These are the machine-annotated rationales R for the given prediction.
   (b) Take the average $\hat{\mu}$ of R.

5. Compare the $\hat{\mu}$ with the class label and use this in the Binary Cross-Entropy loss function and ADAM optimizer to update the model's gradients. The found $\hat{\mu}$ is the final prediction.

By adding these steps, the final prediction $\hat{\mu}$ will be closer to the class label than the prediction made by the BagNetsTextAll model, since only the sentences with values that (strongly) support the prediction μ are used. Sentences with different logits, like padding, will bring the prediction closer to 0.5, because they average the other sentences out. In Section 6.3.4, an example of the output logits per sentence is given in Figure 18.

See Eq. 2 for the formula used to calculate the final prediction $\hat{\mu}$, where S is the set of sentences, $\hat{S}$ is the output of the BagNetsTextRats model, $pred : s \rightarrow \hat{s}$ is the bijective function of the prediction $\hat{s} \in \hat{S}$ for sentence $s \in S$, and R is the set of selected rationales. The selection criteria of rationales is further described in Section 5.3.4. In Figure 10 an overview of the model's architecture is shown.

$$\hat{\mu} = pred''(S, R) : \frac{\sum_{s \in S}\{pred(s) | s \in R\}}{\|R\|} \tag{2}$$

5.2.4 RationaleSearch (RS)

Where the BagNetsTextAll model predicts based on the logits of separate sentences and BagNetsTextRats reduces the base of the final prediction to the found MaRs, the RationaleSearch (RS) model combines the two concepts. Two models are used to solve the classification step:

- **Rationale search sub-model**: for finding annotator rationales.
- **Classification sub-model**: for determining the class of a document, based on only those found rationales.

The Rationale search sub-model is trained specifically to find rationales using the annotator rationale set. These found rationales are then used as input for the classification sub-model. See Figure 11 for a visualisation of the complete RationaleSearch model.

[6] The sentences that fall in the top 20% of the distribution for the given class, see Section 5.3.4.

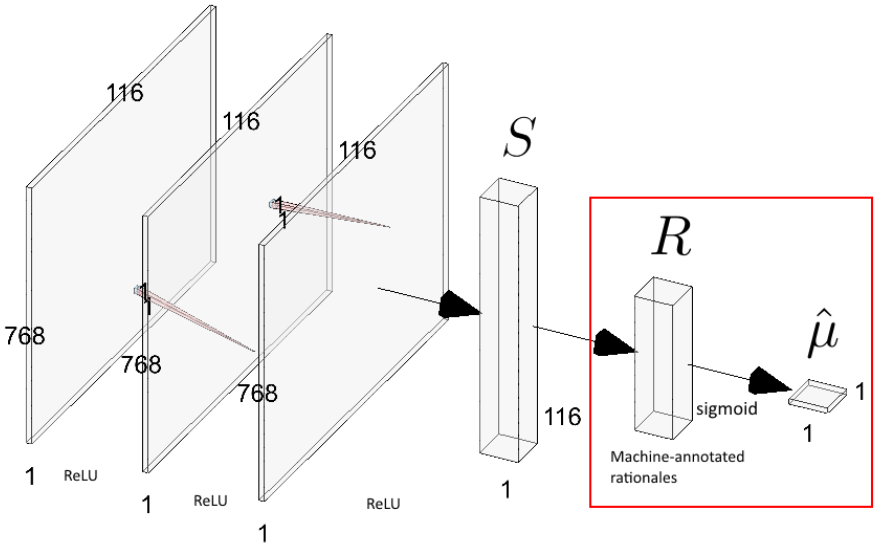


Figure 10: A visualisation of the BagNetsTextRats model architecture. The area in the red square marks the steps taken after the neural network's output is gathered (step 3).

The following steps are taken to make a prediction:

1. For every document, in the RationaleSearch model:
   (a) Input the Sentence-BERT encoded sentences
   (b) For every sentence, within the rationale search sub-model, use a linear layer with 768 inputs and 1 output to generate a single output for the sentence. The output consists of a value between 0 and 1, where 0 is a non-rationale and 1 is a rationale.
2. Compare the rounded output (0 or 1) of every sentence with the true/false (1/0) value label list on whether the sentence was annotated as an annotator rationale. Use a Binary Cross-Entropy loss function and ADAM optimizer to update the gradients.
3. Mask all sentences that were not marked as annotator rationales by replacing them by padding.
4. Pass the masked documents through the SimpleNet model from Section 5.2.1 and classify based on polarity.

This approach is similar to that of Jain et al. (2020), where rationales are found in one model and then used as input for a second model. One difference is that in the RationaleSearch model the task is not separated, but in one go. Two loss functions are used for both sub-models, but the classification sub-model learns simultaneously to the rationale search sub-model. Thus, the classification sub-model adjusts to the rationale search sub-model during training. Lei et al. (2016) use an encoder-decoder approach (hence also in one go) to find rationales, but without training explicitly on how to find rationales.

5.3 Step 2: Explanation through rationale extraction (RE)

In this study, explanations for model predictions are generated by finding machine-annotated rationales for a given model and prediction. I define the process of finding these MaRs with the term rationale extraction (RE). In this section, approaches to step 2 of this study (explanation) are described.

The following rationale extraction approaches are applied to the different models from Section 5.2:

- LinearSVC: Feature extraction (FE) method.

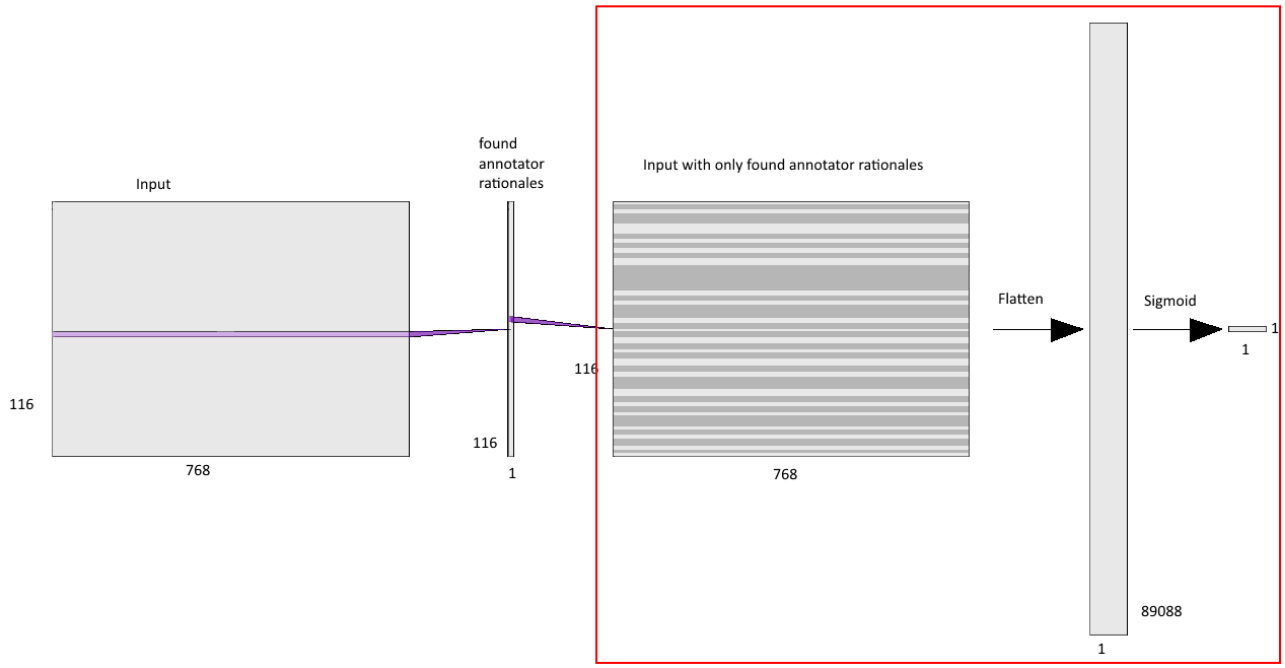


Figure 11: A visualisation of the RationaleSearch model architecture. Part of the model prediction process is the SimpleNet model as displayed in the red square.

- SimpleNet: Leave-One-Out (LOO) method.
- BagNetsText (BNAll & BNRats): model itself.
- RationaleSearch: model itself.

In this work, the terms 'method' and 'model' are used to point at the source of the machine-annotated rationales. This is done to emphasize their application: rationales extracted by methods are applicable to multiple (types of) models (i.e. model-agnostic), like the feature extraction and Leave-One-Out method. The rationales found by the BagNetsTextAll, BagNetsTextRats, and RationaleSearch models can only be extracted using those particular models, and are therefore specific to that model (i.e. model-dependent). In Table 1 an overview of the approaches and their explanation types is given. The following subsections describe the approaches in more detail.

	transparent	post-hoc	local	global	type
Feature extraction (FE)		X		X	method
Leave-One-Out (LOO)		X	X		method
BagNetsText (BNAll & BNRats)	X		X		model
RationaleSearch (RS)		X	X	X	model

Table 1: Different explanation types of rationale extraction methods.

5.3.1 Selecting rationales

The upcoming RE-methods use an alternative method to select a flexible number of machine-annotated rationales. To determine the sentences that need to be selected as machine-annotated rationales, multiple approaches can be taken. One could select the expected number of rationales, for example by using the average number of annotator rationales in all documents in the train dataset, or a fixed percentage of the sentences in the document. For example, Jain et al. (2020) use the top 30% sentences as rationales.

In this study, a different approach is taken: when selecting sentences as machine-annotated rationales, their values are compared to the complete distribution of the values of all sentences in the document. This is done by

creating a histogram of the distribution with n bins, and only selecting sentences in the leftmost or rightmost bin as rationales. Both sides of the histogram represent a class in a binary classification task. Note that this method needs to be adjusted to be applicable to a multi-class classification task. The number of bins is determined by the **bin-size**: $\#bins = 1/\text{bin-size}$, and can be adjusted per model. For example, a bin-size of 0.1 would result in 10 bins. What the values of the sentences entail differs per RE-method.

Using this approach, only sentences that show a significant support for a class are selected as rationales, instead of a fixed set of more-or-less supporting sentences. For example, if only 2 sentences in a 20-sentence document are (annotator) rationales for a negative classification, and the top 30% is selected as machine-annotated rationales, this would mean that $6 - 2 = 4$ selected machine-annotated rationales are not actually rationales. By selecting the sentences that fall in the leftmost or rightmost bins of the distribution, only the two sentences will be selected as rationales. See Figure 12 in Section 5.3.3 for an example of a histogram used in this approach.

5.3.2 Feature extraction

For the LinearSVC model, information from inside the model can be used to find machine-annotated rationales. After the LinearSVC is trained, the weights of trained model can be used to find words that the model found to have a strong linear relation to a given class. These words are signals for a classification. These signal words could be rationales on their own, but since this study focusses on rationales in the form of sentences, whole sentences that contain (mostly) these words are used. I refer to the process of extracting these words with the term feature extraction (FE).

The found machine-annotated rationales are extracted after the prediction is made and are an explanation of how the model works, and not for a specific prediction. Therefore, feature extraction is a *post-hoc* method of finding *global* explanations. Explanations are generated in the form of sentences that contain words that the whole model uses to distinguish between classes. While the features are extracted using the model's weights and thus using the model's reasoning process, the feature extraction method is not transparent. Sentences that contain many signal words are selected as rationales, so some of the signal words are disposed.The sentences are post-hoc interpretations of the explanation given by the model (the signal words) because they do not exactly represent the model's inner reasoning. These rationale sentences are thus only *indirectly* based on the model's inner reasoning process. Furthermore, the explanation is given for a prediction (thus local), but because the explanation is based on a global explanation (the global signal words), the FE-method is global. An explanation for a prediction only reflects the global explanation when using this RE-method.

Two different approaches to extracting features are used to find the words in a document that are indicators for a given classification. In the following sections these approaches are described. The main difference is that the first approach does not use the model's weights, but tries to find a relation between words and classes that the model predicted. The second approach uses the model's weights to find these words.

Custom extraction. As a 'naive' baseline, a custom method of finding class-related words is used. Words that uniquely occur in all documents that were classified with one of the two classes are treated as rationale-words, and sentences that contain those words are selected as machine-annotated rationales for the given class. Words that occur in both positive and negative classified documents are not taken into account.

The steps of this custom extraction method are:

1. Load the complete (training and testing) dataset with embedded (using the CountVectorizer) sentences.
2. Load the pre-trained model for the positive/negative classification task.
3. For every document in the training set, predict the output class using the pre-trained model.
4. Divide the training set into positively and negatively predicted documents.
5. For both sets, find the n (500)[7] most common words by the following steps[8]:
   (a) Append all sentences from all documents in the dataset to create a corpus.
   (b) Remove punctuation.

[7]The number 500 is chosen arbitrary but resolved in a sufficiently large final word set.
[8]These preprocessing steps are done exclusively for the feature extraction method, and not for other methods.

(c) Tokenize into words.

(d) Remove single-character words.

(e) Remove numbers.

(f) Remove stopwords.

(g) Create a FreqDist[9] object using the words. This object counts the number of times that each outcome of an experiment occurs by encoding frequency distributions.

(h) Select the n most common words for a given class according to the FreqDist object.

6. Remove words that occur in the negative word set from the positive word set, and vice versa for the negative set.

Model extraction. The second method uses the trained model weights to extract words that indicate a certain class. This RE-method is applicable to machine learning models with interpretable weights.

1. Load the complete (training and testing) dataset with embedded (using the CountVectorizer) sentences.
2. Load the pre-trained model for the positive/negative classification task.
3. Map the weights of the model to the words used in the vectorizer. Every word receives a weight.
4. Order the words by their weights in the model.
5. Take the top n words as positive words and last n words as negative words.

Selecting sentences as rationales for the FE-method. To find the machine-annotated rationales, sentences that contain the most words from the class word set are selected as rationales. For both methods, proceed after step 5 (model extraction) or step 6 (custom extraction) with:

7. For every document:

   (a) Use one of the above methods to find word sets for positive and negative classes. These are the *positive word set* and the *negative word set*.

   (b) Use the pre-trained model to make a prediction for the document.

   (c) If the prediction is positive, select the n (maximum of 7[10] and number of original annotator rationales for the given document, $\max(7, \#\text{ annotator rationales})$) sentences that contain the most words from the positive words set. If negative, do the same for the negative words set. When the dataset does not contain annotator rationales, the number 7 is used.

5.3.3 Leave-One-Out

A prediction by a machine learning model is based on certain signals in the input. According to the Linearity Assumption (see Section 2.2.3), the areas that influence the prediction can be discovered by removing parts of the input and measuring the effect on the prediction: if the prediction changes significantly[11], that part of the input is annotated as a rationale. This approach to finding local explanations had been used by (Robnik-Šikonja et al., 2008) to find words that contribute to a prediction.

In this study, instead of words, sentences are omitted from the input. This is done to measure the effect of a whole rationale, and since rationales are stored in the format of whole sentences, they also need to be omitted in that format to measure the effect. I define the Leave-One-Out method (LOO-method) as a rationale extraction method for finding post-hoc local explanations in the form of machine-annotated rationales by omitting sentences from the input, measuring their effect on the prediction and selecting the ones with the

[9] A frequency distribution class from NLTK, see http://www.nltk.org/api/nltk.html#nltk.probability.FreqDist

[10] The number 7 is chosen because it is the average number of annotator rationales per document in the test dataset, as described in Section 4.

[11] Significant change can mean a change in predicted class or a different output value, depending on the classification model and task.

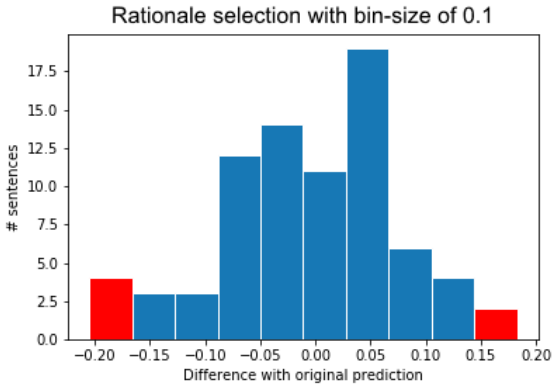
most significant supporting effect. The explanations found by this method are *local* and *post-hoc* because the machine-annotated rationales apply to a specific prediction after the prediction is made.

The selection of rationales is done using the method described in Section 5.3.1, with a bin-size of 0.3[12]. This RE-method can also be used on interpretable models, like support vector machines (SVMs).

The steps of finding machine-annotated rationales using the LOO-method are:

1. Load dataset with Sentence-BERT embedded sentences.
2. Load pre-trained model for positive/negative classification task.
3. Generate a masked sentence (vector of 768 0's).
4. For every document in the loaded dataset:
   (a) Predict the output p for the document using an ML model.
   (b) For every sentence in the document:
      i. Predict the output $\hat{p}$ using a document with that sentence replaced by the masked sentence vector. Store $\hat{p}$ in masked predictions list $\hat{P}$.
      ii. Calculate the difference δ between the original prediction and the masked prediction $(p - \hat{p})$. Store δ in the list of differences Δ.
   (c) Create histogram bins (intervals) based on the distribution of the differences Δ. See Figure 12 for such a distribution with 10 bins. The number of bins is dependent on the bin-size.
   (d) The sentences in Δ that considerably support the original prediction p are selected. Considerably supporting sentences are the sentences in the leftmost bin (negative δ) for a positive classification, and sentences that are in the rightmost bin (positive δ) for negative classifications. The red bins in Figure 12 are the bins for positive (left) and negative (right) predictions.

Figure 12: Example of rationale selection based on a bin-size of 0.1 using the LOO-method. The left red bin contains rationales for a positive classification and the right red bin for a negative classification.



5.3.4 BagNetsTextAll and BagNetsTextRats

The two methods described in the previous sections are both post-hoc, meaning that explanations are extracted after the predictions have been made. A new method to find transparent local explanations is by using information from inside a model during the prediction process. The machine-annotated rationales in the BagNetsTextAll and BagNetsTextRats models are extracted during the prediction process itself. The explanation is based on information from inside the model, during the prediction process for a specific prediction, and thus is a *transparent* and *local* explanation.

This method does not require many computations like the LOO-method and can extract the machine-annotated rationales during the prediction process.

[12]This values proved to be most precise during the development stage. See Section 6.3.4 for more detail on how this value was chosen.

The following steps are taken to find machine-annotated rationales:

1. Train the BagNetsText model on the movie review polarity classification task.
2. For output, use a sigmoid activation function to find a value between 0 and 1 for every sentence.
3. Compare the outputs to the average of all outputs (the prediction). Select all sentences that have an output that is lower or equal (positive prediction), or higher or equal (negative prediction) than the defined threshold. Those sentences are treated as the machine-annotated rationales. The threshold is found using the distribution of the logits in the model's output and a bin-size of 0.2[13], as described in Section 5.3.1.

5.3.5 RationaleSearch

The first sub-model in the RationaleSearch setup extracts annotator rationales from documents. Thus, the sub-model learns to recognize sentences that humans annotate as rationales, which can be seen as a global explanation for the classification task done by humans. The explanations are global, because they give information about the whole classification task, and not for specific cases (predictions). The found rationales are not an explanation for a prediction, but a filtered input consisting of typical sentences that might contain signals for the classification task. The rationale search sub-model identifies sentences as annotator rationales, and annotator rationales usually contain signals for a classification result.

These rationales can also be used as machine-annotated rationales: when the found rationales are used as input for the classification model, they can be seen as local machine-annotated rationales for the prediction, because they are (exactly) the base on which the model made the prediction. The machine-annotated rationales do not give information on how the model's inner reasoning process came to predict a class, since they are only sentences that look like annotator rationales. Because the MaRs do not give any information about how the classification model reasons, the explanation is not transparent. The explanation is not post-hoc either, since the found machine-annotated rationales are generated during the prediction process and not after.

Since the output of the rationale search sub-model is a set of found machine-annotated rationales, no additional steps need to be taken to extract rationales.

[13]The number 0.2 is chosen arbitrarily.

6 Results

As described in Section 5, the task of this study is divided into two steps: classification and explanation. In the following section, the evaluation methods for both steps and their metrics are described. In Section 6.1, I begin by explaining how results are compared to each other using significance testing. Then the classification step results are described in Section 6.2 and the results of the explanation step are described in Section 6.3. To gather some insight into the annotator rationales from the dataset by Zaidan et al. (2007), some of the documents were re-annotated. Their results are described in Section 6.3.3.

6.1 Significance testing through permutation testing

To determine whether the output of two models or methods is significantly different, permutation testing (Dror et al., 2018) is used. This test does not require information about the distribution of the dataset. If two models or methods show different averages on a certain metric but are not significantly different according to the permutation test, this means that they are in fact not significantly different. Knowing whether two results are significantly different is important when selecting the best-performing model or RE-method.

Let n be the number of paired results for systems A and B, α be the test level, F_A be the output for system A and F_B the output for system B, and θ the average difference between F_A and F_B. Then, find θ_0 by doing 2^n permutations on the paired results. The null hypothesis (Eq. (3)) is that the systems A and B do not differ significantly and can be interchanged without changing the distribution. The p-value is the number of times the difference between the two systems is greater in the permuted dataset than in the non-permuted dataset. If the p-value is less than α, we reject the null hypothesis.

$$H_0 : F_A = F_B \tag{3}$$

$$H_1 : F_A \neq F_B \tag{4}$$

When the number of results (n) is very large, it becomes computationally expensive to perform 2^n permutations. A solution is the Monte Carlo Permutation test (Kovacs, 2014), which only performs a random subset of permutations. Not all permutations are tried, the output of the test consists of an interval for the p-value with a lower and upper bound, and a confidence value. For example, the output of $0.001, 0.003, 0.99$ means that the test is 99% certain that the p-value lies between 0.001 and 0.003. In this study, the upper bound is used.

To preserve the reliability of the test, the α can be set to 0.05 or 0.01, depending on the size of the dataset and the number of permutations chosen. In this study, an α of 0.05 and 10,000 permutations are used, as is advised by Marozzi (2004).

6.2 Step 1: Classification

In the following sections the results for different models for the classification step are described. The models are compared using the metrics described in Section 6.2.1. Then, an overview of all models is given in Section 6.2.2. After that, the results of all models are described in the sections 6.2.2 to 6.2.2.

6.2.1 Evaluation metrics

For the classification task, different machine learning models are used to perform the same task: classify documents on polarity. The classification models are evaluated using the following metrics:

1. Accuracy: the fraction of correct predictions among all predictions.
   accuracy $= \frac{\text{\# correctly classified documents}}{\text{\# documents}}$
2. Loss: an indicator of how far away a machine learning model's prediction lies from the class labels. A high loss indicates that a model predicts far away from the correct class label and is likely to make inaccurate predictions. Binary Cross-Entropy (BCE) loss (Plunkett et al., 1997) is used to determine how far off the prediction is from the given label.

3. Precision: the precision metric is used to measure how well a model can recognize a given class through the fraction of retrieved relevant (correctly classified documents with given class) instances among retrieved instances (all documents classified with given class).
   $\text{precision} = \frac{\#\text{ correctly classified document with class}}{\#\text{ documents classified with class}}$
4. Recall: the model's ability to recognize a given class in the dataset, through the fraction of relevant retrieved instances (correctly classified documents with given class) among relevant instances (all documents with given class).
   $\text{recall} = \frac{\#\text{ correctly classified document with class}}{\#\text{ documents with class}}$
5. F1-score: the harmonic mean between precision and recall.
   $\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

The precision, recall, and F1-score are calculated for both the positive and negative classes to determine which class the model is better at recognizing.

6.2.2 Classification models

The different machine learning models used for the classification step show similar accuracy scores, with only the RationaleSearch model as a worse-performing exception. In Table 2 the classification accuracy scores of all models are shown.

To determine whether models reason differently, their outputs are compared. See Table 3 for a complete overview of the differences in prediction between all models. Both rounded and non-rounded predictions are compared, where a non-rounded prediction is the class output (positive or negative).

	LinearSVC	SimpleNet	BagNetsTextAll	BagNetsTextRats	RationaleSearch	
					search	classification
Train						
epochs		5	30	30	30	
accuracy	0.845	0.830	**0.895**	0.860	0.946	0.800
loss[14]		0.418	0.612	0.449	0.549	0.489
Test						
accuracy	0.815	0.800	**0.888**	0.847	0.947	0.777
loss		0.401	0.622	0.422	0.556	0.457

Table 2: Training accuracy and loss for all models. Best performing model scores are in bold.

	LinearSVC	SimpleNet	BagNetsTextAll	BagNetsTextRats	RationaleSearch
LinearSVC		0.002	0.013	0.001	**0.405**
SimpleNet	0.032		**0.403**	**0.321**	0.019
BagNetsTextAll	**0.506**	0.002		**0.055**	**0.145**
BagNetsTextRats	0.001	0.028	0.001		0.002
RationaleSearch	**0.781**	0.026	**0.364**	0.001	

Table 3: Significant differences in model predictions on the test set with $\alpha = 0.05$ and a 99% confidence interval. The values shown are the p-values of the upper bound. Non-significant values are in bold. Left bottom half: non-rounded predictions. Right top half: rounded predictions.

Classification baselines: LinearSVC (LSVC) and SimpleNet

The LinearSVC and SimpleNet models are used to find a classification accuracy baseline. Both models reach a similar accuracy (± 0.810). An accuracy score of 0.8 is used as a baseline for comparing the models in further sections. The results of the LinearSVC and SimpleNet models are discussed in more detail below.

LinearSVC. The LinearSVC model reaches an accuracy of 0.815, with a slightly higher F1-score for positive documents. See Table 4 for all scores for the model. The LinearSVC uses a different loss function than the other models (squared hinge), and therefore no loss metrics are included.

LSVC model performance metrics on the test set						
overall	positive			negative		
accuracy	precision	recall	f1-score	precision	recall	f1-score
0.815	0.794	0.850	0.821	0.839	0.780	0.808

Table 4: LinearSVC Test Metrics

Simple neural network (SimpleNet). The SimpleNet model reaches an accuracy of 0.800, with a higher F1-score for negative documents. Training the SimpleNet model on 30 epochs results in a training loss of 0.354 and accuracy of 0.8 on the test set. In Table 5 an overview of the metrics for the SimpleNet model trained on 5 epochs on the test set is shown. Accuracy and rationale quality metrics increase in the first couple of epochs and stabilize or improve after this initial increase. The loss increases around epoch 5, and therefore the SimpleNet model trained on 5 epochs is used in the rest of this study. See Figure 26 in the Appendix for an overview of the training statistics of the SimpleNet model.

SimpleNet model performance metrics on the test set with 5 epochs							
Overall		Positive			Negative		
accuracy	loss	precision	recall	f1-score	precision	recall	f1-score
0.800	0.354	0.861	0.715	0.781	0.756	0.885	0.816

Table 5: SimpleNet Test Metrics

BagNetsTextAll (BNAll)

The BagNetsTextAll model reaches a higher accuracy of 0.888 after 30 epochs, which is higher than the baseline (0.810). The model reaches a higher F1-score for positive documents. See Table 6 for the performance metrics of BagNetsTextAll on the test set.

After the first couple of epochs, the BagNetsTextAll model reaches a mostly stable rationale quality score. The accuracy does increase over 30 epochs, and the loss continues to decrease. See Figure 27 in the Appendix for the training statistics.

The BagNetsTextAll model predicts very closely to the decision boundary (0.5), resulting in a higher loss than the SimpleNet model (0.354). This high loss is caused by the substantial difference between the prediction (± 0.5) and the class value (0 or 1). The averages for positive (0) or negative (1) predictions by the BNALL model are 0.485 and 0.522 respectively. Since the final prediction is the average of all sentences in the document, including neutral ones (non-rationales), the final prediction lies close to 0.5.

BNAll model performance metrics on the test set with 30 epochs							
Overall		Positive			Negative		
accuracy	loss	precision	recall	f1-score	precision	recall	f1-score
0.888	0.622	0.787	0.905	0.842	0.888	0.755	0.816

Table 6: BagNetsTextAll Test Metrics

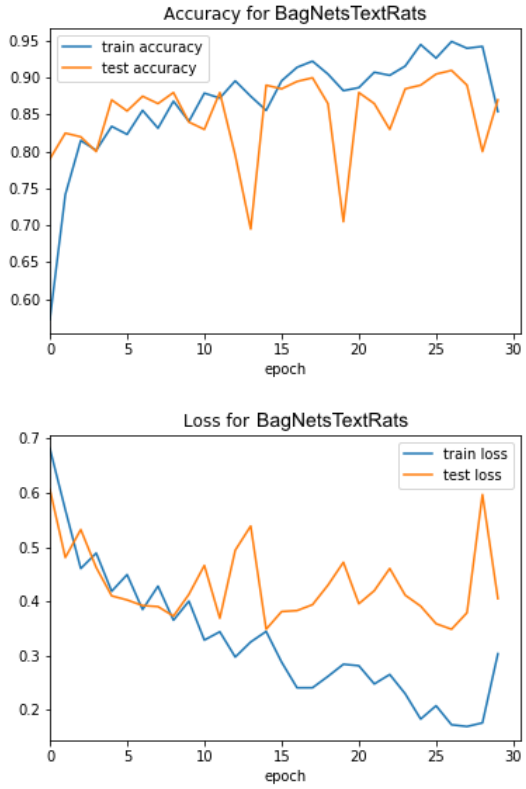
BagNetsTextRats (BNRats)

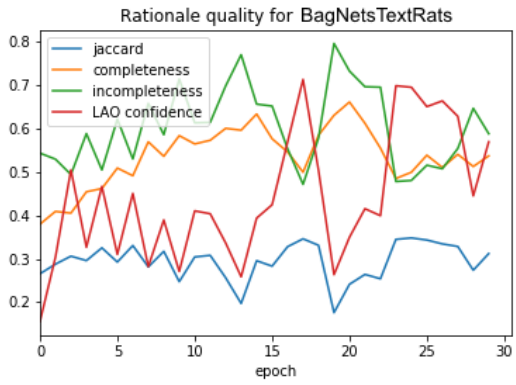
The BagNetsTextRats model reaches an accuracy of 0.847, which is higher than the baseline. The loss after 30 epochs is 0.442 for the BNRats model, which is lower than the BagNetsTextAll loss (0.622). Where the

BNALL model has a higher F1-score for positive documents, the BNRats model reaches a higher F1-score for negative documents. Moreover, there is a significant difference between the predictions of the models, but only on non-rounded outputs (see Table 3).

The BagNetsTextRats training statistics in Figure 13 show that the training process is less stable than that of the BagNetsTextAll. The BagNetsTextRats model predicts closely to the positive (0) or negative (1) class labels (average of 0.052 and 0.940 respectively). This is because only the found rationales are used in the final prediction, making the prediction more extreme. Figure 13 shows that around epoch 14, 19, and 28, the wrong rationales are used to base the predictions on, increasing loss, and decreasing accuracy and corresponding rationale quality. The metrics for rationale quality are introduced in Section 6.3.1. See Table 7 for performance metrics on the test set.

Figure 13: Training statistics for the BagNetsTextRats model





BNRats model performance metrics with 30 epochs		
	accuracy	loss
	0.847	0.422
Positive		
precision	recall	f1-score
0.960	0.725	0.826
Negative		
precision	recall	f1-score
0.779	0.970	0.864

Table 7: BagNetsTextRats Test Metrics. The top right figure shows the rationale quality over epochs. The metrics for rationale quality are introduced in Section 6.3.1.

RationaleSearch (RS)

The RationaleSearch model scores below the classification baseline of the LinearSVC and SimpleNet models (0.810) with an accuracy of 0.777 for classification. The F1-score for positive and negative classes is nearly identical after training. Table 8 shows classification scores for the test dataset.

The RationaleSearch model consists of two sub-models, the rationale search sub-model and the classification sub-model. The rationale search sub-model converges in the first two epochs (see Figure 28 in the Appendix). The accuracy for the rationale search sub-model lies around 0.95 after 30 epochs. The classification sub-model has a slower learning curve and stabilizes after 14 epochs. The rationale quality scores stabilize with the classification accuracy score.

Since the classification sub-model is a differently trained SimpleNet model as used in Section 6.2.2, it should be pointed out that the RationaleSearch model predicts significantly less accurate than the SimpleNet model. The main difference here is the input: full documents for the SimpleNet and rationale-only documents for the RationaleSearch model. This is the same difference as the BNAll and BNRats models, but with a supervised

rationale search for the RationaleSearch model.

RationaleSearch model performance metrics on the test set with 30 epochs									
Overall				Positive			Negative		
search		classification							
accuracy	loss	accuracy	loss	precision	recall	f1-score	precision	recall	f1-score
0.947	0.556	0.777	0.457	0.776	0.780	0.778	0.779	0.775	0.777

Table 8: RationaleSearch Test Metrics

6.3 Step 2: Explanation

The first step in the task of this study is classification, and the second step is finding explanations. Explaining is done by rationale extraction. The following section describes the results of different RE-methods and how to compare them. First, formulae that can be used for comparing machine-annotated rationales to annotator rationales are described in Section 6.3.1. Second, in Section 6.3.2, a measure of faithfulness, called the Leave-All-Out-confidence is described. Then the results from the re-annotated documents are discussed in Section 6.3.3. After that, I go into the results of every rationale extraction method in Section 6.3.4.

6.3.1 Similarity metrics formulae

The following similarity metrics are used for evaluating machine-annotated rationales. The machine-annotated rationales are compared to annotator rationales, where high similarity shows that the explanation of the model is similar to a human explanation.

The **Jaccard index** is used to measure the overlap between two sets of rationales. The formula for calculating the Jaccard index is shown in Eq. (5), where X and Y are the two sets to be compared. A value of 0 indicates no overlap and a value of 1 complete overlap. The more overlap, the more similar two sets are.

The **Completeness index** can be calculated using the formula in Eq. (6). This index measures the number of annotator rationales (AR) present in the machine-annotated rationale (MaR) set. A value of 0 indicates that there are no annotator rationales (ARs) in the MaR set, and a value of 1 that all ARs are in the MaR set. The Completeness index is similar to the precision metric used in the classification step (see Section 6.2.1) and indicates how much relevant information (i.e. complete) is in the set of found machine-annotated rationales. A low (but non-zero) Completeness score may indicate that the number of machine-annotated rationales that are not annotator rationales is large, meaning that there is an explanation, but that the explanation differs from the human-provided one. This means that a model explains differently than a human.

The **Incompleteness index**, calculated by the formula in Eq. (7), measures the number of missing annotator rationales (AR) in the machine-annotated rationale (MaR) set. This is similar to an inverted recall score. The Incompleteness index shows how much relevant information is missing (i.e. incomplete) from the found explanation, compared to the human explanation. A value of 1 indicates that all ARs are missing in the MaR set, and a value of 0 that all MaRs are ARs. A high Incompleteness score thus indicates that a model does not use the same sentences as a human would to make a classification.

Note that the denominator for the Incompleteness index the number of ARs (see Eq. 7), instead of the number of MaRs as used in the Completeness index (Eq. 6). Therefore, the Completeness and Incompleteness index do not sum up to 1.

The Completeness and Incompleteness can be used alongside the Jaccard index for more insight into rationale quality. While a high Jaccard index can only be achieved alongside a high Completeness and a low Incompleteness index, a low Jaccard index does not always mean that a machine-annotated rationale set is very non-similar to the annotator rationale set. Table 9 contains examples of the concurrence between Jaccard, Completeness and Incompleteness indexes. The fourth example in Table 9 shows that a low Jaccard index and

a high Incompleteness index can still mean that all ARs are included in the MaR set.

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \tag{5}$$

$$completeness(MaR, AR) = \frac{|MaR \cap AR|}{|MaR|} \tag{6}$$

$$incompleteness(MaR, AR) = 1 - \frac{|MaR \cap AR|}{|AR|} \tag{7}$$

Jaccard	Completeness	Incompleteness	Set description
0.167	0.167	0	Large MaR set (18) and small AR set (3), but all ARs are in the MaR set.
0.667	0.8	0.2	The MaR and AR set are the same size and differ with only 1 rationale in both sets.
0.100	0.167	0.800	The MaR and AR set are the same size and only one AR is in the MaR set.
0.143	1	0.857	The AR set is much larger than the MaR set, but all MaRs are ARs.
1	1	0	The AR and MaR sets are identical.

Table 9: Examples of concurrence between Jaccard, Completeness and Incompleteness indexes.

In Figure 14 the development of all indexes for a document with 15 annotator rationales and different numbers of machine-annotated rationales is shown. In the first 15 steps of the x-axis, one AR is added to the MaR set at a time. After 15, non-ARs are added to the MaR set.

The Jaccard index steadily grows as more ARs are selected as MaRs. The Jaccard and Completeness indexes show the same decrease when non-annotator rationales are selected as MaRs. The Completeness index shows an increase from 0 to 1 when the first rationale has been found, and only drops after MaRs that are not ARs are selected (> 15 on the x-axis). The Incompleteness index decreases as more ARs are selected as MaRs and stabilizes when all annotator rationales are in the machine-annotated rationale set.

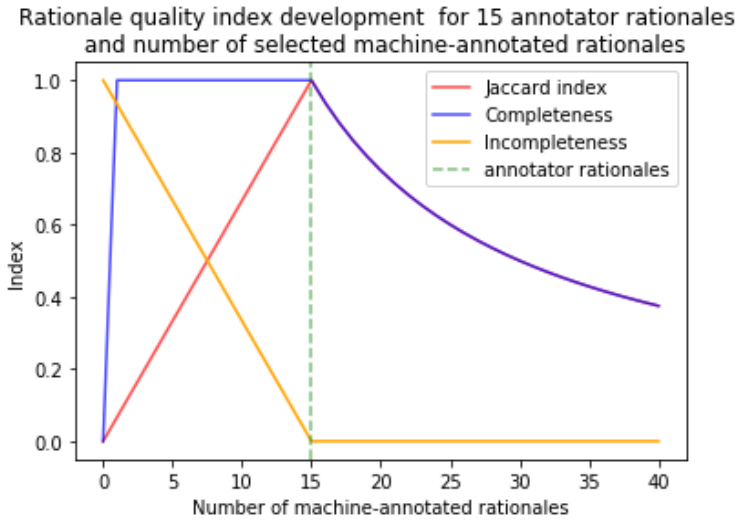


Figure 14: Development of the different similarity metrics for a document that contains 15 annotator rationales and different numbers of selected machine-annotated rationales.

6.3.2 Leave-All-Out (LAO) confidence

To gather a (post-hoc) measurement of faithfulness, a new metric inspired by the LOO-method is used. The metric is called Leave-All-Out confidence (LAO-confidence), and compares the original document prediction to a prediction where all found rationales are removed from the document. Intuitively this would result in a different or less strong prediction for the original predicted class because omitting the rationales would result in a document with fewer signals for a certain class. This method is post-hoc, because the effect of machine-annotated rationales is measured after the original prediction is made. Every sentence that is labelled as a machine-annotated rationale is replaced by padding (see Section 5.1), and the resulting document is used as input for the new prediction.

The difference between the original prediction and the adjusted prediction is called the LAO-confidence. See Eq. (8) for the formula, where *pred* is the prediction function, *docs* are the documents to be predicted, N is the number of documents, and $MaRs$ are the machine-annotated rationales for the documents. A value close to 0 indicates that omitting the found rationales from the document does not change the prediction, and a value of 0.5 or higher indicates that the prediction changes drastically enough to switch to another class. While it is possible in theory that the LAO-confidence is close to 1, it would mean that a prediction changes from 0 (very positive) to 1 (very negative), or vice versa. This is not likely, because predictions often lie closer to the classification boundary (0.5).

$$\text{LAO-confidence}(pred, docs, MaRs) = \frac{\sum_{i=1}^{N} \mid \text{pred}(docs^i) - \text{pred}(docs^i \setminus MaRs^i) \mid}{N} \tag{8}$$

6.3.3 Custom re-annotated rationales

To verify the quality of the annotator rationales in the Zaidan et al. (2007) dataset, a small subset of documents has been re-annotated in this study. Three new annotators were tasked to select a number of rationales close to the original number of annotator rationales (a difference of no more than 2) supporting the given document and classification.

Apart from selecting rationales that support the classification, rationales for other classifications could also be selected. These rationales are called **anti-rationales**, meaning that they support another classification. A subset consisting of 30 documents (15 positive and 15 negative) from the test set was used. See Table 10 for a comparison between the annotator rationales for the original and re-annotated (custom) annotator rationales. The distribution of selected rationales compared to the original dataset is shown in Figure 4 in Section 4.

The scores show that annotator rationales in the original dataset are not always a ground-truth explanation, but that other explanations also exist. The average Jaccard index of the original and re-annotated annotator rationales is 0.278, which shows that there is a significant difference in explanations by humans. Therefore, it might be almost impossible to achieve high values for the Jaccard Index when the explainer is different.

In Table 31 in the Appendix, the rationale quality scores per document are displayed. Using a permutation test, the difference in the number of selected rationales is not significant (p-value between 0.497 and 0.523 with a 0.99 confidence rate).

	original dataset	custom re-annotated dataset
# rationales	7.1	6.5
Jaccard index	0.278	
Completeness	0.403	
Incompleteness	0.557	

Table 10: Comparison between original and re-annotated (custom) annotator rationales

6.3.4 Rationale extraction methods

In the following section, the results of different rationale extraction methods are described. When comparing RE-methods, the focus lies on the quality of the found machine-annotated rationales compared to annotator rationales. The test set, which contains annotator rationales (as described in Section 4), is used in the evaluation. See Appendix A.5 for documents annotated by the different RE-methods.

As a baseline, a random subset of the sentences in a document is selected as rationales. This is done for all documents in the test set. The number of random sentences is calculated using $rand(0.1 \times num\_sentences, 0.9 \times num\_sentences)$, where $rand(a, b)$ is a function that selects a number between a and b, and $num\_sentences$ is the number of sentences in a document. The average number of rationales selected in the random rationale set is 17.095. The rationale quality scores for the random rationale baseline are 0.171 for the Jaccard index, 0.226 for the Completeness index, and 0.495 for the Incompleteness index.

In Table 13, the rationale quality scores for the different rationale extraction methods are shown. The BagNetsTextAll, BagNetsTextRats and RationaleSearch models perform best on the Jaccard and Completeness indexes. The scores are not significantly different, and therefore the models share first place. The BagNetsTextRats model achieves the highest score for the LAO-confidence.

Incompleteness: Number of rationales selected. None of the RE-methods performed better than the random rationale baseline on the Incompleteness index. This can be explained by the number of selected rationales: the average number of rationales in the random rationale set was 17.095. Most RE-methods select a number of rationales that is close to the average number of annotator rationales in the dataset (7 rationales). The more sentences selected, the higher the chance of selecting annotator rationales, and the lower the Incompleteness index is. Since the random rationale set contains more rationales on average, it scores better in the Incompleteness index than the RE-methods. The number of rationales selected per method is shown in Table 11.

The number of rationales selected differs per RE-method. The RationaleSearch, BagNetsTextAll and BagNetsTextRats methods select the smallest sets of rationales (≤ 5). The BagNetsTextAll and BagNetsTextRats select the largest sets of rationales (≥ 5) as well. The Feature Extraction method extracts most sets consisting of 7 elements, but this number is was explicitly specified as a minimum in the method (see Section 5.3.2). In Figure 15 a histogram of the number of selected rationales per document for all rationale extraction methods is shown.

	# rationales				
	mean	median	mode	min	max
Random	17.095	15	10	0	68
AR	7.09	7	7	1	22
FE	8.3	7	7	4	22
LOO	6.4	6	7	1	22
BNAll	6.11	5	3	1	25
BNRats	5.47	5	4	1	24
RS	5.25	5	3	0	20

Table 11: Average number of rationales selected per RE-method for the test set.

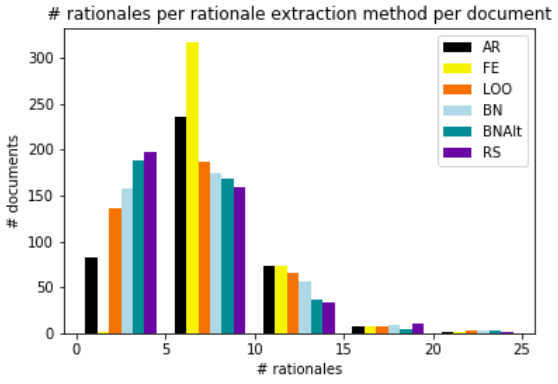


Figure 15: Histogram of the number of rationales selected per document for RE-methods for the test set. Used bins are $\{0, 5, 10, 15, 20, 25\}$.

Comparison between rationale extraction methods. Machine-annotated rationales found by different rationale extraction methods show overlap. When two sets of MaRs overlap, this means that the explanation is similar. In Table 12 the Jaccard indexes of the rationale extraction methods are shown. The BagNetsTextAll and BagNetsTextRats methods overlap the most with a Jaccard index of 0.678. Both methods also overlap similarly with the RationaleSearch model (0.338), while the RationaleSearch model has little overlap with other

methods. The two rationale sets with the least overlap are the sets generated by the LOO-method and the FE-method.

Jaccard indexes				
	FE	LOO	BNAll	BNRats
LOO	0.131			
BNAll	0.169	0.255		
BNRats	0.163	0.239	0.678	
RS	0.160	0.161	0.338	0.338

Table 12: Jaccard indexes of different rationale extraction methods compared to each other.

RE-methods and RE-models. Machine-annotated rationales found using models are more similar to annotator rationales than MaRs found by model-agnostic rationale extraction methods. As shown in Table 13, the Jaccard and Completeness indexes are higher for the RationaleSearch, BagNetsTextAll and BagNetsTextRats models. This shows that explanations that are generated by models that are trained to be self-explaining, explain more similar to humans than model-agnostic RE methods.

Global and local explanations. The rationale quality of explanations for incorrect model-classifications by the RationaleSearch and feature extraction methods is higher than the rationale quality of explanations for incorrect model-classifications by the BagNetsText implementations and LOO-method. In other words, the explanations by the RationaleSearch and feature extraction methods for *incorrect* classifications by the model, are more similar to human explanations for *correct* classifications by humans than the other RE-methods. So even though the model misclassified, the explanation supports the correct classification and is thus not supporting the model's classification. The explanation is similar to a human explanation, but misleading. It does not explain the prediction but gives more of an overall summary of which sentences might be relevant for a classification. Thus, an explanation based on global information about the model (LinearSVC) or task (RationaleSearch) does not explain local predictions as well as local-information explanations (BagNetsText implementations and LOO-method).

The subsections that follow go into the individual details of different methods.

Test					
	# documents	LAO-confidence	Jaccard Index	Completeness	Incompleteness
Random rationales					
total	400		0.171	0.226	**0.495**
Feature Extraction (FE) & LinearSVC					
total	400	0.274	0.227	0.323	0.597
correct	326	0.276	0.245	0.346	0.574
incorrect	74	0.264	0.148	0.219	0.698
Leave-One-Out (LOO) & SimpleNet					
total	400	0.351	0.192	0.370	0.698
correct	320	0.336	0.235	0.455	0.636
incorrect	80	0.408	0.022	0.031	0.945
BagNetsTextAll (BNAll)					
total	400	0.025	**0.326**	**0.546**	0.562
correct	355	0.025	0.361	0.603	0.517
incorrect	45	0.020	0.052	0.102	0.915
BagNetsTextRats (BNRats)					
total	400	**0.566**	**0.316**	**0.550**	0.595
correct	339	0.542	0.362	0.629	0.540
incorrect	61	0.702	0.060	0.110	0.902
RationaleSearch & SimpleNet					
total	400	0.352	**0.328**	**0.586**	0.578
correct	311	0.311	0.370	0.647	0.529
incorrect	89	0.495	0.180	0.372	0.749
Significant?		Not RS & LOO (0.947)	Not LOO & BN, LOO & LSVC, BNAll & LSVC (0.090), BNRats & RS (0.411) BNAll & BNRats (0.48)	Not BNRats & RS (0.101) BNAll & BNRats (0.88)	Not LSVC & BNRats (0.92), LSVC & RS (0.272), BNRats & RS (0.331) BNAll & BNRats (0.06)

Table 13: Rationale quality for rationale extraction methods on the test set. Bold values perform best on the index. If multiple values are bold, the two methods do not significantly differ.

Feature Extraction (FE)

is a post-hoc method of finding words that signal towards a certain class, as described in Section 5.3.2. Rationales are found by selecting sentences that contain those words. The custom feature extraction method, where words that are common for classified documents are selected as features, scores significantly worse on all rationale quality metrics compared to the other methods. The feature extraction method based on the model's weights scores best on the Jaccard-index, with a score of 0.227. In Table 14 the results for the feature extraction methods on the LinearSVC model are shown. There is no substantial difference in rationale quality for explanations for correctly and incorrectly classified documents.

Custom word extraction. The custom word extraction method selects words that are common in classified documents for a certain class as features, and selects sentences that contain these words as machine-annotated rationales. Wordlist 1 and Wordlist 2 in the Appendix contain the positive and negative words extracted by the custom rationale extraction. The words are ordered based on their weights, meaning that the words 'jackie', 'war', 'perfect', 'beautiful', and 'throughout' are very positive and the words 'worst', 'supposed', 'attempt', 'save', and 'stupid', are very negative. Some of the words in the lists are not necessarily positive or negative but occur often and uniquely in either class.

Model word extraction. The model word extraction method selects words from a trained model's weight as features for a given class, and again selects sentences that contain these words as machine-annotated rationales. In Wordlist 3 the positive words and in Wordlist 4 the negative words extracted using the weights of the model are shown in the Appendix. The words are ordered based on their weights, meaning that the words 'great', 'quite', 'excellent', 'hilarious', and 'fun' are very positive and the words 'completely', 'women'[15], 'crap', 'don', and 'cheap' are very negative. The word 'don' here is the first part of 'don't' in a sentence. Similar to the custom extraction method, not all words in the word lists are necessarily positive or negative, but the model does associate them with one of the classes.

Test					
	# documents	LAO-confidence	Jaccard Index	Completeness	Incompleteness
FE model[16]					
total	400	**0.274**	**0.227**	0.323	0.597
correct	326	0.276	0.245	0.346	0.574
incorrect	74	0.264	0.148	0.219	0.692
FE custom					
total	400	0.152	0.198	0.290	0.645
correct	326	0.153	0.207	0.303	0.636
incorrect	74	0.148	0.160	0.234	0.685
LOO					
total	400	**0.292**	0.173	**0.583**	**0.256**
correct	326	0.285	0.191	0.467	0.721
incorrect	74	0.324	0.091	0.199	0.846
Significant?		Not LOO and model FE (0.077)	Yes	Yes	Yes

Table 14: Results for feature extraction methods and LOO-method on the LinearSVC model

Both the feature extraction and LOO-method are applied to the LinearSVC model to gather insight into the difference between global and local explanations. The FE-method explains locally using transparent global information, and there is not much of a difference in performance compared to a post-hoc local explanation

[15]The word 'women' among the most negative words could indicate that this SVM behaves with some sexist bias.

method like the LOO-method (see Table 14). Using the FE-method to extract machine-annotated rationales resulted in a higher Jaccard index, but the LOO-method performed better on both Completeness and Incompleteness indexes. The LOO-method selected fewer rationales than the FE-method, which explains the difference in Jaccard indexes. These results show that global (FE) and local (LOO) explanations are not necessarily different.

The feature extraction method used for evaluation (see Table 13) is the method based on the trained model's weights (FE model). The FE model method is chosen because the method scores highest on Jaccard Index, and not significantly different with the LOO-method on LAO-confidence (and thus shares first place). The LOO-method selects fewer rationales than both FE-methods. See Table 15 for the number of rationales selected for the different methods.

# rationales					
	mean	median	mode	min	max
FE model	8.3	7	7	4	22
FE custom	83	7	7	4	22
LOO LSVC	5.4	4	3	1	21

Table 15: The average number of rationales selected per method for the LinearSVC model and the test set.

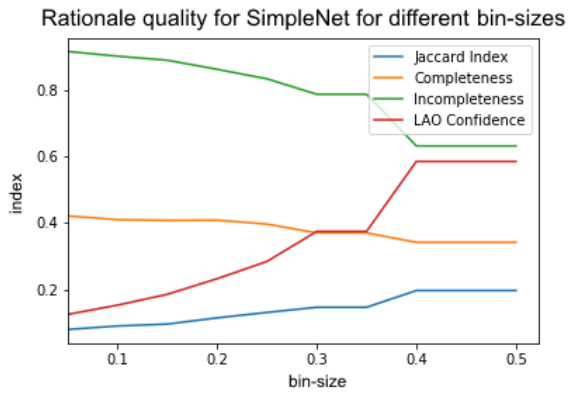
Leave-One-Out (LOO)

With a Jaccard index of 0.192, the LOO-method does not score significantly higher than the FE-method. The method does score better on Completeness and LAO-confidence (0.351 and 0.370 respectively, see Table 13).

Applying the LOO-method on the LinearSVC (see the previous section) significantly improves on all rationale quality metrics except for the Jaccard Index (see Table 14). The rationale quality scores (except for the LAO-confidence) differ for correctly and incorrectly classified documents, which was not the case for the FE-method.

Selecting rationales. The number of rationales selected depends on the chosen bin-size. The bin-size affects the rationale quality, as shown in Figure 16. In case of the SimpleNet model, a bin-size of 0.3 or higher does not improve the scores. A bin-size of 0.3 means that 60% ($30 \times 2 = 60\%$) of the sentences that support the prediction are selected as rationales. Using a higher bin-size improves the Jaccard, Incompleteness and LAO-confidence scores, but decreases the Completeness, as shown in Figure 16. An explanation is that more sentences are selected and thus more non-annotator rationales are selected, decreasing the Completeness index. Selecting more sentences does not necessarily increase rationale quality.

Figure 16: Average rationale quality for given bin-sizes using the LOO-method and the SimpleNet model.



BagNetsTextAll (BNALL)

The machine-annotated rationales found by the BagNetsTextAll model are extracted during the prediction process, as described in Section 5.3.4. The found MaRs score better than the FE and LOO methods on all indexes except for the LAO-confidence. The model scores 0.326 on the Jaccard index, 0.546 on the Completeness index, and 0.562 on the Incompleteness index. The LAO-confidence for this model is 0.05, which is much lower than the previously discussed methods. The explanations for correctly classified documents have higher rationale quality scores than explanations for incorrectly classified documents.

Selecting rationales. The bin-size for selecting rationales is set to 0.2, meaning that a histogram with 5 bins is created, where the sentences in the leftmost or rightmost (positive or negative) bin are selected as machine-annotated rationales. In Figure 17 a histogram for the BagNetsTextAll classification of a negative document is displayed. The red line is the decision threshold, which indicates that all sentences with a higher output value than 0.89 are selected as machine-annotated rationales. The value 0.89 here is the boundary of the rightmost bin in the histogram. The document in the example (negR_868.txt) only contains 10 sentences, which are all on the right side of the histogram. The leftmost side of the histogram is populated by the padding. These sentences all have the same (empty) content and are tagged with a value very close to 0.5. The selected rationales all have a very high value (≥ 0.9). See Figure 18 in the following section for the annotated document.

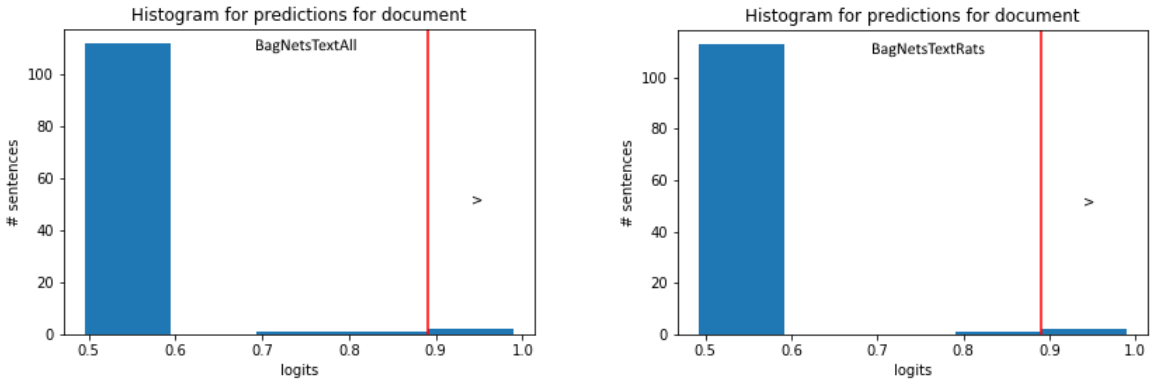


Figure 17: Histogram with bins for sentence logits for negative document negR_868.txt. The red line is the decision threshold. Left: The BagNetsTextAll model. Right: The BagNetsTextRats model.

BagNetsTextRats (BNRats)

The LAO-confidence for BagNetsTextRats increases with 0.25 to 0.566, compared to the BagNetsTextAll model. This LAO-confidence score is also the overall best. The BNRats machine-annotated rationales do not score differently on the other rationale quality indexes than the BagNetsTextAll model's machine-annotated rationales.

The differences in rationale quality for correct and incorrect classifications are similar to those of the BagNetsTextAll model, but a clear difference in the LAO-confidence (0.136) is visible (see Table 13).

In Figure 17 a histogram of the logits for a prediction by the BagNetsTextRats model is displayed. The distribution of the logits is similar to BagNetsTextAll, but slightly more extreme (closer to class labels). In Figure 18 the document with predictions and selected machine-annotated rationales is shown. The second sentence in the document receives a more neutral value (0.5097) from the BagNetsTextRats model than from BagNetsTextAll model (0.7809), which is also visible in the histogram.

Document	Tag	# sentences
negR_868.txt	1.0 (negative)	10

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	3		1.000				
	BNAll	2	True	0.514	0.008	0.667	1.000	0.333
	BNRats	2	True	0.981	0.145	0.667	1.000	0.333

Table 16: Rationales for negative document negR_868.txt using BagNetsTextAll and BagNetsTextRats.

Aspiring broadway composer Robert (Aaron Williams) secretly carries a torch for his best friend, struggling actor Marc (Michael Shawn Lucas).(0.5163) (0.4925) The problem is, Marc only has eyes for "perfect 10s," which the geeky, insecure Robert certainly is not. (0.7809)(0.5097) Meanwhile, marc's spoiled (hetero) female roommate, Cynthia (Mara Hobel), spends her days lying about their apartment and harrassing magazine editor tina brown. (0.5089) (0.4932) Writer-director Victor Mignatti's "very romantic comedy" (as the ad campaign states) is supposed to be (pardon the pun) a gay ol' romp, but it's hard to have much fun with these annoying, self-absorbed characters and their shallow personal problems: Marc and Cynthia have sitcom-level domestic "crises" (such as trying to kill bugs–how hilarious); Robert and Marc go to acting class (how riveting); the zaftig Cynthia goes on eating binges (how original). (0.5496)(0.5146) But more than anything else, the three whine. (0.8038)(0.8354) Constantly. (0.5382)(0.5076) Marc whines about his turbulent romance with an apparent "10," David (Hugh Panaro), the hunky musician from across the way; Robert whines about not being able to find the right guy; Cynthia whines about having to find a job (horrors). (0.4987)(0.4948) The terrible trio whine their way to a happy ending that is wholly undeserved. (0.9890)(0.9893) Add in overly broad performances and some laughable lipsynching by Panaro, and you're left with one astonishing piece of cinematic damage. (0.9149)(0.9715)

Figure 18: Annotated negative document negR_868.txt. The coloured underlined sentences are the rationales for the given method. See Table 16 for the rationale quality, predictions, and colour scheme. The logits value is added between the parentheses, where the first value is for BagNetsTextAll and the second is for BagNetsTextRats model.

RationaleSearch (RS)

As shown in Table 13, machine-annotated rationales found by the RationaleSearch model have slightly higher Jaccard (0.328) and Completeness (0.586) index scores than the other RE-methods. The BagNetsTextAll and

BagNetsTextRats models show similar scores that are non-significantly different. The LAO-confidence for the RationaleSearch machine-annotated rationales is not significantly different from the LOO-method.

Similar to the LinearSVC model, the difference in rationale quality scores for correctly and incorrectly classified documents is smaller than the other RE-methods. In 11 out of 400 cases, the RationaleSearch model selected 0 sentences as machine-annotated rationales, thus not being able to provide an explanation for a classification. As displayed in Table 11, all other methods manage to provide at least one machine-annotated rationale.

LAO-confidence. The LAO-confidence scores for the RationaleSearch and LOO-method machine-annotated rationales are not significantly different. Both methods use the same model for classification (SimpleNet) and this shows that the LAO-confidence is model-dependent. The RationaleSearch model predicts on only found annotator rationales and does not give information on the model's inner reasoning and is therefore not faithful. The LOO-method does take into account the model's inner reasoning process, since it measures changes in prediction for different inputs. And still, both methods show similar LAO-confidence scores, but the found rationales overlap with a Jaccard index of 0.161 (see Table 12), which indicates that the rationale sets are quite different. The average number of rationales selected as machine-annotated rationales is comparable (5.25 and 6.4, see Table 11). The RationaleSearch model does show a significantly worse classification accuracy. Combining all above-mentioned observations indicates that the two versions of the SimpleNet model reason and explain differently, but that this cannot be determined by the LAO-confidence as it is model-depended.

6.3.5 Comparison to custom-annotated documents

As described in Section 6.3.3, a small subset of documents was re-annotated. This was done to find an alternative benchmark that included anti-rationales. The re-annotated dataset reaches a score of 0.278 for the Jaccard index, 0.403 for the Completeness index, and 0.557 for the Incompleteness Index with the original dataset by Zaidan et al. (2007) (see Table 10 in Section 6.3.3). This indicates that different explainers give different explanations that do not necessarily overlap. In the following section, the results of all RE-methods are compared to the re-annotated rationales (rAR) and re-annotated anti-rationales (rAAR). See Table 17 for the similarity scores of the different RE-methods with the re-annotated dataset.

General rationales. When comparing machine-annotated rationales to re-annotated rationales, the best-scoring RE-methods on the Jaccard, Completeness, and Incompleteness indexes are the BagNetsTextAll and BagNetsTextRats models. The RationaleSearch model, which scores best for the regular annotator rationales as described in Section 6.3.4, is not among the best-scoring models compared to re-annotated rationales. This shows that the RationaleSearch model specifically explains similar to the annotator rationales in the Zaidan et al. (2007) dataset, since it is trained on finding that kind of annotator rationales. The BagNetsText implementations explain more using the model's inner reasoning and find more general rationales.

Sensible classifications. When documents are incorrectly classified, the re-annotated anti-rationales (rAAR) can be used to find out whether a model made a sensible classification. A classification is **sensible** when the model makes a classification on signals that humans would also use to come to that classification, thus making the classification correct regarding those signals. The re-annotated anti-rationales serve as an explanation for an alternative classification, which in this study is marked as an incorrect classification. The rationale quality indexes for incorrectly classified documents by the RationaleSearch model are better than the indexes for correctly classified documents when compared to anti-rationales. This shows that an explanation generated by the RationaleSearch model does support the decision of the model, in the same manner as humans would explain it.

Test							
	# documents	Jaccard Index		Completeness		Incompleteness	
Original dataset							
total	30	0.278		0.403		0.557	
FE							
		rAR	rAAR	rAR	rAAR	rAR	rAAR
total	30	0.102	0.050	0.160	0.057	0.798	0.772
correct	11	0.098	0.049	0.167	0.052	0.829	0.758
incorrect	19	0.104	0.050	0.156	0.060	0.780	0.781
LOO							
		rAR	rAAR	rAR	rAAR	rAR	rAAR
total	30	0.132	0.033	0.256	0.049	0.741	0.928
correct	26	0.150	0.038	0.291	0.056	0.706	0.917
incorrect	4	0.017	0	0.031	0	0.969	1
BNAll							
		rAR	rAAR	rAR	rAAR	rAR	rAAR
total	30	**0.243**	0.021	**0.383**	0.030	**0.642**	0.792
correct	28	0.260	0.023	0.410	0.032	0.616	0.812
incorrect	2	0	0	0	0	1	0.500
BNRats							
		rAR	rAAR	rAR	rAAR	rAR	rAAR
total	30	**0.240**	0.022	**0.371**	0.030	**0.660**	0.792
correct	28	0.250	0.024	0.380	0.032	0.645	0.812
incorrect	2	0.100	0.000	0.250	0.000	0.875	0.5
RS							
		rAR	rAAR	rAR	rAAR	rAR	rAAR
total	30	0.160	**0.104**	0.279	**0.142**	0.761	**0.550**
correct	23	0.180	0.083	0.298	0.094	0.718	0.558
incorrect	7	0.091	0.174	0.214	0.299	0.901	0.524

Table 17: Rationale extraction methods compared to re-annotated annotator rationales (rAR) and re-annotated annotator anti-rationales (rAAR). Best-performing methods are in bold.

7 User Evaluation

In Section 3.2.2, I described some possible scenarios of how explanations can be received by humans: explanations can be good, subjectively incomplete, or non-interpretable. Using the user evaluation, the subjective quality of machine-annotated rationales can be measured. In the following sections, the setup and results of this evaluation are described.

7.1 Setup

Human users are asked to perform different tasks to measure the interpretability and completeness of machine-annotated rationales. Machine-annotated rationales for classifications on 15 distinct documents from the user study dataset are classified by the users. The machine-annotated rationales represent an explanation that forms a base for the users to classify on. Machine-annotated rationales found by the FE-method for LinearSVC, LOO-method for SimpleNet, BagNetsTextRats, and RationaleSearch models are used. The BagNetsTextAll model is not included in the comparison because of its similarity with BagNetsTextRats, and to keep the user evaluation concise. Apart from documents from the user study dataset, the user evaluation also contained 7 distinct documents from the test dataset with annotator rationales as an explanation, because the user study dataset does not contain annotator rationales. These annotator rationales are used in the blind study to get insight into how the machine-annotated rationales and annotator rationale compare.

The user evaluation consists of the following three tasks:

1. **Sanity check**: Users are asked to mark all sentences in a document that they think are annotator rationales. This is a check on whether the user understands what annotator rationales are, to ensure that the gathered data is not arbitrary. If a user does not understand the definition of annotator rationales, the given answers for the other tasks are not used. See Figure 19 for the task. Users need to select enough correct annotator rationales to reach a positive Jaccard index (a measure of overlap between two sets, see Section 6.3.1) to pass the sanity check. This means that only one correct rationale needs to be selected, but the set of correct rationales is constructed so that it contains rationales that could be explanations independently.
2. **Blind study**: The goal of this task is to compare machine-annotated rationales between models and to annotator rationales, on comprehensibility and completeness. Users are asked to classify a document solely based on a given explanation for a classification. These explanations are given in the form of machine-annotated rationales from the feature extraction method, Leave-One-Out method, BagNetsTextRats, and RationaleSearch models, and annotator rationales from the original dataset by Zaidan et al. (2007). Note that classifications for this task are done by both ML models and users. I will further refer to these classifications as *model*-classifications and *user*-classifications. In Figure 20 an example of the task is shown. Users have three options for classifying a document after being presented with a set of annotator or machine-annotated rationales:
   (a) Positive
   (b) Negative
   (c) I need more information.

   The first two answers will indicate whether the rationales are indeed an interpretable explanation for the correct classification of the document. If a user chooses the correct polarity, this shows that the model reasoned and explained in a way that is interpretable to humans. If the user chooses incorrectly, this indicates that the model did not base the prediction on signals that are similar to signals that humans would use to classify the document, thus making the explanation not interpretable. The third answer will be a direct indicator of an incomplete explanation. If a user can correctly classify a document based only on the provided rationales, the rationales are a comprehensible and complete explanation. If the user chooses 'I need more information', the explanation is incomplete.

   Note that this notion of incompleteness is subjective to the judgement of the user and cannot be expressed as a mathematical formula. This therefore cannot be related to the incompleteness metric defined in Section 6.3.1. I will refer to this notion of incompleteness as **subjective incompleteness**, meaning that the explanation does not contain enough information *for the receiver* to base a classification on.

   A mix of 15 positive and negative documents is used for the blind study task. To gather insight about how a wrong classification by the model affects the explanation, the set contains machine-annotated rationales

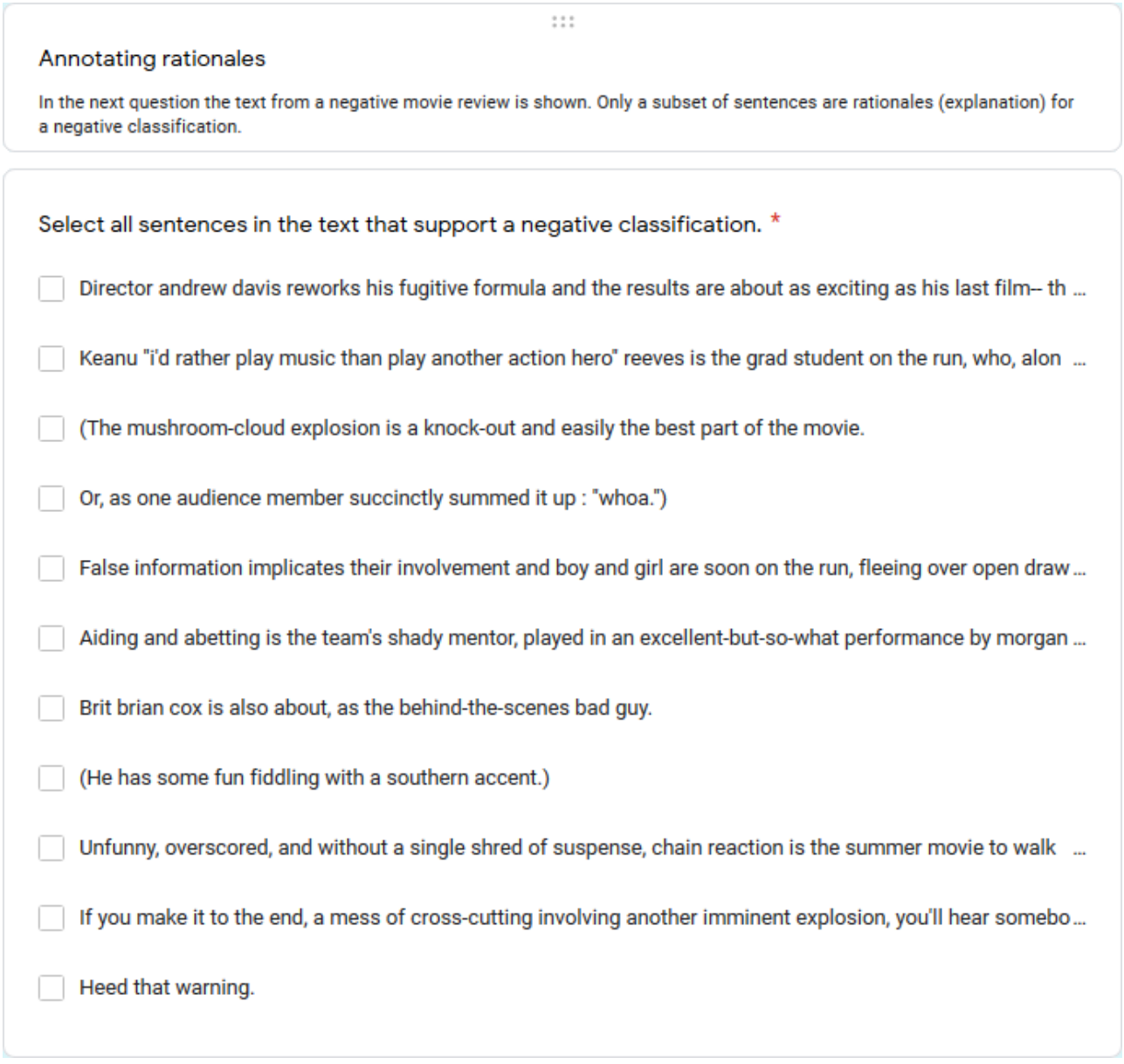


Figure 19: The sanity check in the user evaluation. The correct answer should include the top sentence and/or bottom three sentences.

for 185 correct and 48 incorrect classifications. If an explanation is in line with the model's prediction, the user's prediction will also align with the model's prediction. If not, the explanation is misleading (and thus non-interpretable, see Section 3.2.2).

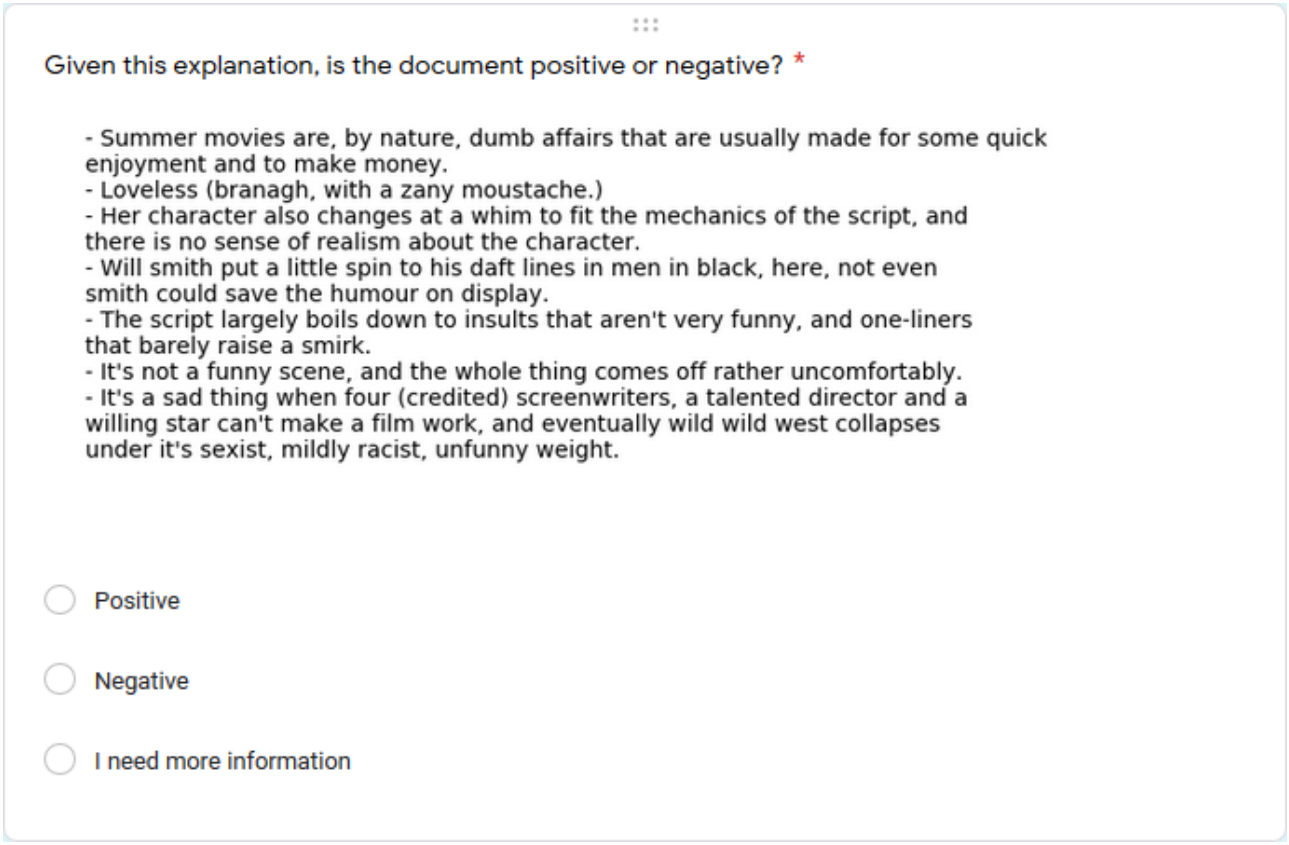


Figure 20: A question from the blind study in the user evaluation.

3. **Model quality task**: The goal of the model quality task is to find out which of the RE-methods explains models best: if a model is explained well, a recognizable difference between machine-annotated rationales for a correct and incorrect classification of the same document should be observable to a user. When an explanation reflects the inner workings of the model well enough so that users can tell the difference between a deteriorated and correctly functioning model, the explanation adds interpretability to the model. This is because the explanation gives insight into the inner reasoning of the model that can be understood by the user, thus making the model more interpretable.

   In this task, users are asked to identify two models based on machine-annotated rationales and choose the model that will make the correct given prediction. The correct class of the document is given beforehand. Two models for the same rationale extraction method are compared, where one is deteriorated and the other correctly functions.

   A **deteriorated model** is a model that has been artificially deteriorated to misclassify (make the wrong classification). Models are deteriorated by training from scratch with randomly shuffled class labels. This results in a model that classifies differently from a correctly trained one. See Figure 21 for one of the questions from the model quality task. The process of deteriorating models is described in Appendix A.3.4.

   When comparing models, I use machine-annotated rationales for a correct classification by the correctly functioning model and an incorrect classification by the deteriorated model, for the same document. A total of 8 positive documents is used for this task[17].

In Table 18 a short overview of the user evaluation tasks is given.

[17]Only positive documents were misclassified by all deteriorated models and correctly classified by all correctly functioning models.

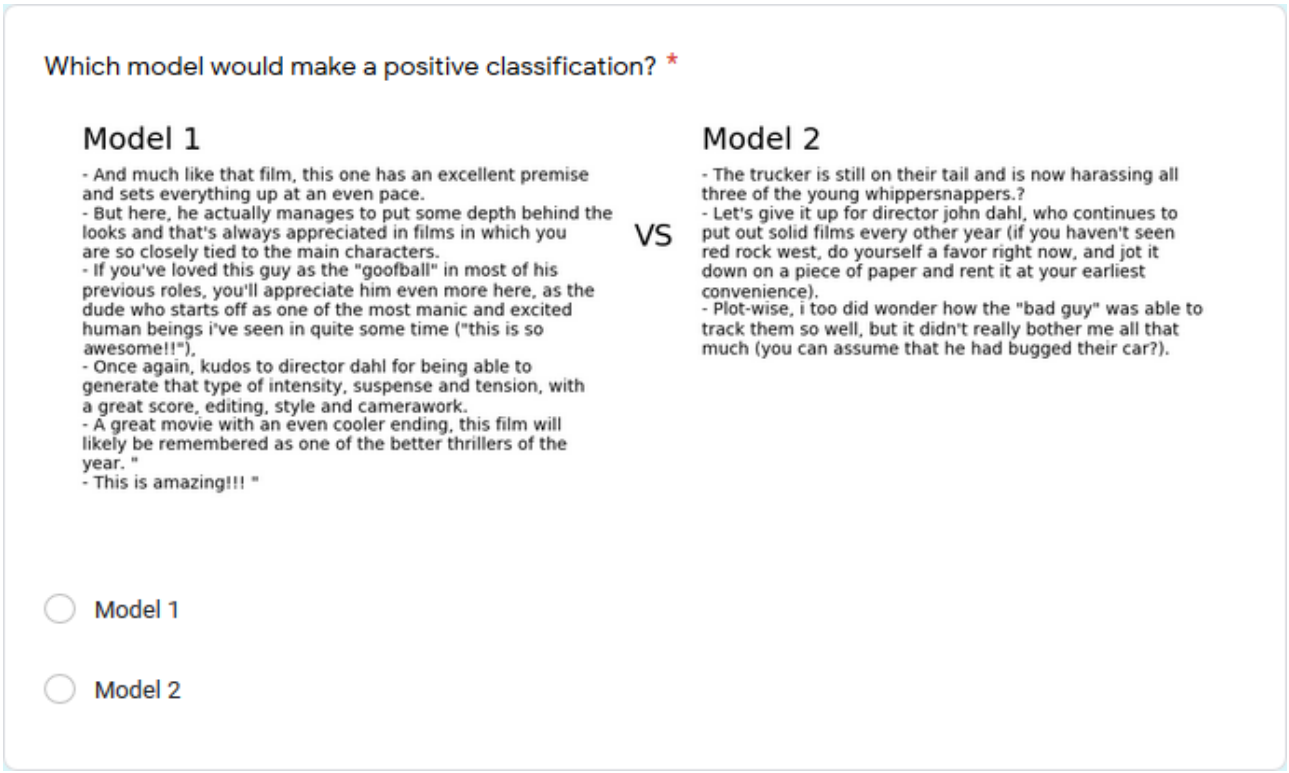


Figure 21: A question from the model quality task in the user evaluation.

	Input	Goal	User task
Sanity check	1 document from the test dataset.	Check whether a user understands the concept of annotator rationales.	Select all sentences in the given document that support a given prediction.
Blind study	MaRs for classifications by different models and RE-methods[18] for 8 distinct documents from the user study dataset, and 7 ARs from the test dataset.	Compare rationales from different sources on completeness and interpretability.	Classify a document on the given rationales, without knowing the source of the rationales.
Model quality	For every FE-method, the MaRs for classification by a correctly functioning and a deteriorated model, for 8 documents from the user study dataset.	Gather insight into whether a model is explained well by a RE-method.	Determine which of the two models is the correctly classifying one, given the model's MaRs.

Table 18: Overview of the user evaluation tasks, and their content and goal.

7.2 Results

The user evaluation was done by 45 users through an online survey. Only one participant did not pass the sanity check by selecting the sentence *'Or, as one audience member succinctly summed it up: "whoa."'* as the only rationale. All responses that had selected rationales with a Jaccard index greater than 0 with the rationale benchmark shown in Figure 22, were selected as useful. The average Jaccard index of the useful responses with the specified benchmark was 0.58. See Table 32 and Table 33 in the Appendix for an extensive overview of the results of the user evaluation. In the following two sections, the results for the blind study and model quality task are described.

Director Andrew Davis reworks his fugitive formula and the results are about as exciting as his last film– the dreadful comedy steal big, steal little– was funny. Keanu"i'd rather play music than play another action hero" Reeves is the grad student on the run, who, along with his superfluous sidekick (Rachel Weisz), has been framed for a sabotaged science experiment that vaporized eight Chicago city blocks. (The mushroom-cloud explosion is a knock-out and easily the best part of the movie. Or, as one audience member succinctly summed it up : "whoa.") False information implicates their involvement and boy and girl are soon on the run, fleeing over open drawbridges, across icy lakes, and through the corridors of power at a top-secret, underground energy facility. Aiding and abetting is the team's shady mentor, played in an excellent-but-so- what performance by Morgan Freeman. (Brit Brian Cox is also about, as the behind-the-scenes bad guy. He has some fun fiddling with a southern accent.) Unfunny, overscored, and without a single shred of suspense, chain reaction is * the * summer movie to walk out on. If you make it to the end, a mess of cross-cutting involving another imminent explosion, you'll hear somebody say "i guess it's time to go." Heed that warning.

Figure 22: Annotator rationales used for the sanity check in the user evaluation. All sentences that contain information that points towards a negative classification are selected. These annotator rationales are selected especially for the sanity check task in the user evaluation and do not come from the dataset by Zaidan et al. (2007). The set should represent a very lenient set of annotator rationales for a negative classification.

7.2.1 Blind study

In the blind study task, users were asked to classify a document based on a given explanation in the form of a set of machine-annotated rationales. A total of 233 classifications based on rationales were done by users. In Table 19 the distribution of classifications by users from the blind study is shown. Users had the additional 'I need more information'-option, to mark a document as subjectively incomplete. I refer to classifications by models as *model*-classifications, and to classifications by users as *user*-classifications.

The RationaleSearch model's MaRs proved to be slightly more interpretable than the annotator rationale benchmark. This can be seen in Figure 23, where the distribution of user classifications based on machine-annotated rationales is displayed. The user-classified documents based on MaRs by the RationaleSearch model are correctly classified in 97.4% of the cases, while the documents based on annotator rationales are correctly

classified in 92.5% of the cases. The user-classifications based on annotator rationales are a bit more often subjectively incomplete (7.5% for ARs and 2.6% for RS MaRs). Note that annotator rationales classifications are always correct, because of how they were collected (see Section 4).

user-classifications / model-classifications	incorrect model-classifications	correct model-classifications	total
correct	22 (45%)	158 (85%)	180
incorrect	6 (13%)	7 (4%)	13
subjectively incomplete	20 (42%)	20 (11%)	40
total	48	185	233

Table 19: Distribution of user and model classifications for the blind study. A model-classification is incorrect when the prediction is not the same as the class label of the document.

Incorrect model-classifications Some of the model-classifications were incorrect. Whenever a user correctly classified a document based on an explanation for an incorrect model-classification, the user in fact *disagreed* with the model. The explanation is then misleading to the user because it does not align with the model-classification. Note that this is an indication that the explanation is unfaithful (see Section 3.1) or non-interpretable (see Section 3.2.2).

Incorrect model-classifications were more often marked as subjectively incomplete by users than correct ones: 11% of the explanations for correct model-classifications were marked as subjectively incomplete, and 42% of the explanations for incorrect model-classifications was marked as subjectively incomplete (see Table 19).

The increase in subjective incompleteness for incorrect model-classified documents does not apply to machine-annotated rationales from the FE-method, which achieved a 100% correct user-classification rate. This shows that the MaRs for this method are misleading since they do not explain the *model's* classification. Moreover, the FE-method is able to find the relevant signals in the input for the *classification* task but does not explain the individual predictions.

For the set of model-misclassified documents (incorrect classifications), the RationaleSearch model showed the highest percentage of user predictions that aligned with the model output (i.e. a misclassification). This performance shows that the RationaleSearch does explain its decisions very well to users since they make similar classifications based on the explanations. So even if the RationaleSearch model does not faithfully find machine-annotated rationales, it does explain its decisions in a manner that is interpretable to humans, and that aligns with the inner reasoning of the model. The explanations by the RationaleSearch model may not be faithful, but they are not misleading either.

Some documents in the blind study are marked as subjectively incomplete, or classified incorrectly by users more often than others. See Figure 24 for the distribution of subjectively incomplete and incorrect user-classified documents. The number of sentences or different words in the documents does not seem to influence how the users classify based on MaRs. In Appendix A.7.1, some documents classified by users are shown.

7.2.2 Model quality task

The model quality task focusses on finding RE-methods that explain models: the explanations given by two models should be interpretable enough to be able to identify the correctly functioning and the deteriorated model.

The RationaleSearch model is the most difficult to identify for users since the machine-annotated rationales by this model are most often incorrectly identified by users. In Figure 25 the distribution of model identifications by users per RE-method is shown. This difficulty in identification is caused by the fact that the machine-annotated rationales by the RationaleSearch model are not based on the classification sub-model's inner reasoning process. Furthermore, the part of the model that searches for rationales is not deteriorated. Rationales are still extracted from the documents, but only the classification model is deteriorated, which results in a model that can still correctly find rationales, but not classify correctly based on them. In some cases, the sets of rationales for the deteriorated and correctly functioning model were identical.

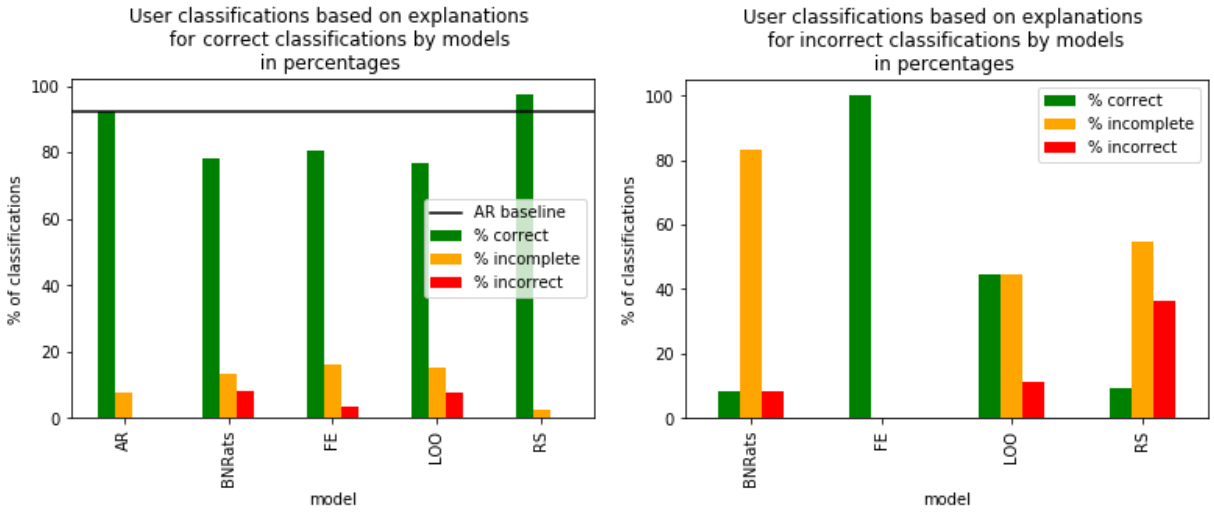


Figure 23: User-classifications based on machine-annotated rationales. Left: correct model classification. The percentage of correct classifications by users should be as high as possible. Right: incorrect model classifications. The percentage of correct classifications indicates that the given explanation does not reflect the model's decision. An incorrect classification shows that the explanation does reflect and support the model's decision. A high percentage of classifications marked as subjectively incomplete shows that the given explanation does not contain enough information for the user to base a classification on. The higher the percentage of incorrect classifications, the better, since it indicates that the explanation supports the model's prediction.

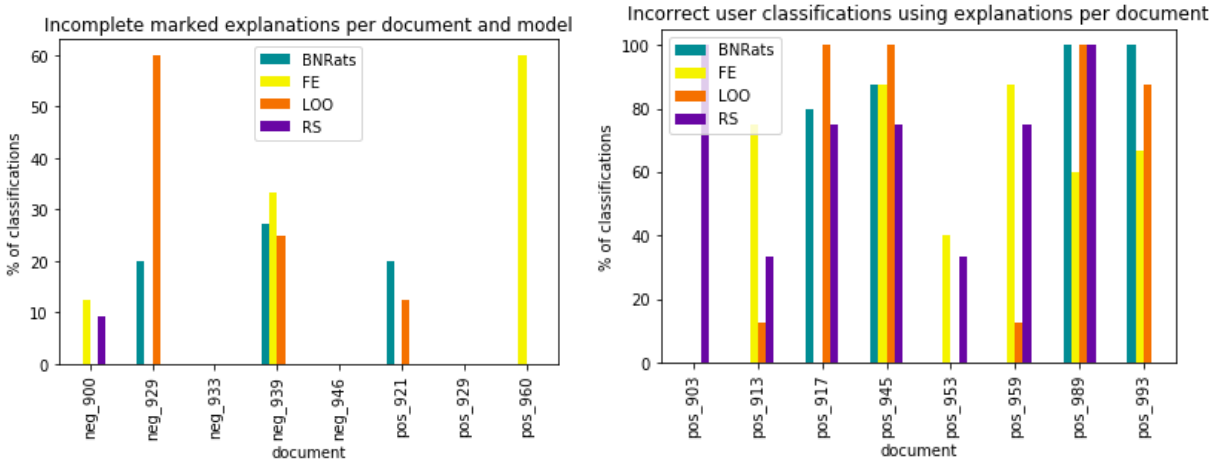


Figure 24: Subjectively incomplete and incorrect classified documents by users in the blind study per RE-method. Left: subjectively incomplete user-classifications. Right: incorrect user-classifications.

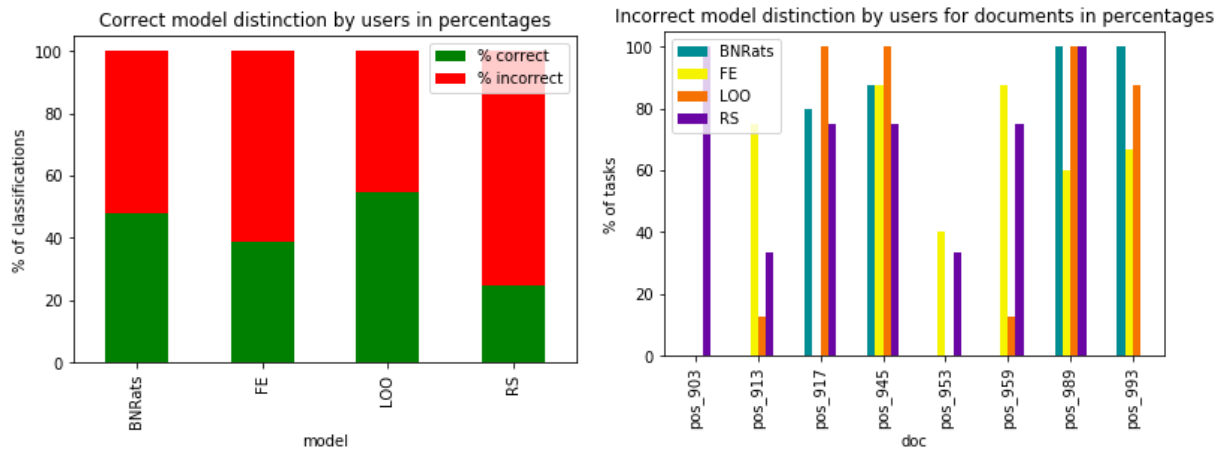


Figure 25: Left: correct and incorrect deteriorated model identifications by users in the model quality task. A model was correctly identified when the user chose the correctly classifying model based on provided MaRs. Right: incorrectly identified models per document and RE-method.

Global and local explanations. As displayed in Figure 25, the deteriorated models are more often correctly identified using the BagNetsText implementations and LOO-method machine-annotated rationales. Thus, MaRs by the BagNetsText implementations and LOO-method explain local predictions better than RationaleSearch and FE-method MaRs. This is expected, as the RationaleSearch model and FE-method base their explanations on global information about which parts of the input are relevant, and therefore do not explain predictions locally like the BagNetsTextRats model and LOO-method do.

8 Discussion

In this study, different approaches to finding faithful machine-annotated rationales for text classification have been investigated. First, some general insights are discussed. In Section 8.1, I go into the comparison of machine-annotated rationales to annotator rationales. Then, in Section 8.2, some insights on faithfulness are described. After that, some reflection on selecting rationales is given in Section 8.3.

There is no outstanding rationale extraction method that is superior to other methods on faithfulness, similarity to annotator rationales, and user evaluated-quality. The different approaches finding machine-annotated rationales do show advantages. Depending on the focus of the explanation, different RE-methods can be used to find explanations. Two (non-exhaustive) focus settings are:

1. When the focus lies on faithfulness, models specifically designed to be self-explaining can be used. An example of such a setting is when a model classifies on biased grounds, and the validity of a prediction needs to be confirmed. In future work, models like BagNetsTextAll and BagNetsTextRats can be extended to more complex classification tasks. It is important to be able to ensure the transparency of the model. One possible way of doing this could be to expand the models in iterations (by adding layers) and proving the faithfulness for every iteration. Some additional measure of faithfulness, for example the LOO-method, could be used to strengthen the proof.
2. When the focus lies on explaining similarly to humans, models that are trained for that specific task, like the RationaleSearch model, can be used. An example setting is finding summaries of movie reviews documents. While models like the RationaleSearch model explain their classifications in a way that is comprehensible to human receivers, they do not explain faithfully and do not explain the classification model. Only explanations similar to annotator rationales are found. Some faithfulness could be added to the explanations by creating a hybrid version of the RationaleSearch model with for example the LOO-method. By doing this, the model's inner reasoning is taken into account, as well as human-focused explaining. The order of selecting rationales (RationaleSearch first or LOO-method first) might have different effects and is a question for future work.

Simple and black box models. Explaining black box models proved to be a more complex task than explaining simple models, but the machine-annotated rationales for black box models do show more overlap with annotator rationales. Thus, explanations for predictions generated for and by complex ML models are more similar to human explanations than explanations for simple ML models and can be used in future work on explaining model predictions.

Model-agnostic and model-dependent RE-methods. The machine-annotated rationales by the model-agnostic RE-methods (feature extraction method and Leave-One-Out method) are less similar to annotator rationales and less faithful according to the LAO-confidence metric than model-dependent machine-annotated rationales (RationaleSearch, BagNetsTextAll, and BagNetsTextRats). This indicates that extracting explanations during the prediction process results in explanations more similar to human explanations.

RationaleSearch training process. The RationaleSearch model did show a stabilisation in rationale quality parallel with classification accuracy, and not with rationale search accuracy. This might be a coincidence, but it might also show that the quality of the explanations does improve with the model's ability to classify, thus adding some faithfulness to the explanation. The relation needs to be looked further into in future work.

Input chunk size and computational cost. The LOO-method proved to be very computationally expensive, given that for every sentence in the document, a new prediction needs to be made. In case of the largest document in the dataset, this means that 116 predictions had to be made. The machine-annotated rationales found by the BagNetsTextAll and BagNetsTextRats models are extracted during the prediction process and are thus less computationally expensive. Using rationales in the form of sentences results in human-comprehensible explanations, but using different an input chunk size (words or sub-sentences) might improve the clarity of the explanation, since rationales can also be given in sub-sentence form. This would require more preprocessing to find these sub-sentence structures, however, and the computational complexity of extracting rationales would increase.

8.1 Annotator rationale comparison

Comparing the found machine-annotated rationales to annotator rationales using set theory proved to be only somewhat suitable as a measurement of explanation quality. One of the issues was that the Jaccard index, which measures the overlap of the two sets, resulted in unexpected scores. For example, the re-annotated rationale set did not score very high on the Jaccard index (0.278, see Section 6.3.3). Since the set of re-annotated rationales are also annotated by humans, these explanations are of the same quality as the annotator rationale benchmark.

Some of the found machine-annotated rationales reached low Jaccard indexes, but the explanations were not necessary of low quality, only different in quantity. In other words, the explanations of the model were more concise than the human explanations, but did form a base for users to classify on. Another issue was that some machine-annotated rationales did contain some information that (weakly) pointed towards a certain classification, but were not annotated by the human in the dataset. Determining whether a sentence is a good rationale for a classification is not easily done. One method could be to use human judgement to determine whether a rationale supports a given classification. If the model is not completely transparent, it might not be possible for a human to truly understand the reasoning behind the classification, making it difficult to know whether a rationale is truly explaining the classification. Some future work on measuring the quality of explanations is therefore needed.

Annotator rationales can be used to determine whether an explanation is somewhat explanatory to humans. A useful baseline is a set of random sentences since the Jaccard and Completeness indexes with the annotator rationale set are low for such sets. In future work, a notion of rationale-strength could be added to the dataset, where every sentence receives some score of the amount of relevant information pointing to a classification that it contains. Such scores could be in the form of the output for the BagNetsText ML model as shown in Figure 18 (values between 0 and 1, where 0.5 is a neutral sentence). Using the information, a weighted Jaccard index could be created.

Another note is that the rationale quality metrics for the found machine-annotated rationales are calculated by taking the average of all classifications and their explanations. When a model made a wrong classification but does explain itself well, this still decreases the average index value. One solution for this is only taking the correctly classified documents into account for evaluation. Another, more insightful solution could be to add anti-rationales, rationales that support another classification, as was done for a small subset of the dataset (see Section 6.3.3). That way, the *support* of a rationale, whether it has a causal relation to the output, can be measured more accurately.

8.2 Faithfulness

The only transparently explaining rationale extraction-method used in this study is the BagNetsText model, with the BagNetsTextAll and BagNetsTextRats implementations. This is reflected in a high LAO-confidence (0.566, see Table 13) for the BagNetsTextRats model. The BagNetsTextAll scores very low on the metric (0.025), but this can be explained by the fact that the model predicts very close to the decision boundary (0.5). This shows that the current LAO-confidence is only applicable to models that predict confidently, meaning that they predict close to class labels. A normalized version of the LAO-confidence might solve this problem.

As described in Section 6.3.4, the RationaleSearch and SimpleNet models reason and explain differently, but use the same underlying model for classification. The LAO-confidence for the explanations for both models is similar. Since the models reason differently and explain differently, the similarity in LAO-confidence scores is not necessarily an indication towards unfaithfulness, but it remains difficult to determine the true level of faithfulness. More research on determining the faithfulness of explanation by input features is necessary.

All in all, I would say that as long as the Linearity Assumption (see Section 3.1) holds, the LOO-method and the RationaleSearch models are sufficiently faithful when using the LAO-confidence score as a measure. I leave the question of whether the Linearity Assumption is correct to future research. One possible extension to the LAO-confidence could be the Leave-All-In confidence, which erases all non-relevant sentences from the input and then measures the effect on the prediction. Another addition to the metric could be the percentage of cases that the prediction actually changes when the machine-annotated rationales are omitted.

8.3 Selecting rationales

Machine-annotated rationales are selected by looking at the values relative to the distribution of all sentences, as described in Section 5.3.1. By using this method, only the sentences that have a clear influence on the prediction are selected, which causes the machine-annotated rationale set to contain very strongly influencing rationales. The number of rationales selected on average by the different RE-methods lies around 6 (see Table 11), which is slightly lower than the average number of annotator rationales in the dataset by Zaidan et al. (2007) (7). This shows that a similar number of signals are used by the ML models and human annotators. The user evaluations showed that certain sets of machine-annotated rationales are very small, which increases the chance of an incomplete explanation.

In this work, the bin-size has not been explicitly tuned, but I expect that it will be dependent on the model which bin-size is the most beneficial. The number of rationales selected for the BagNetsTextRats model proved to be occasionally too concise in the user evaluation, and a higher bin-size for that model might improve the completeness of BagNetsTextRats MaRs. A higher bin-size value increases the chance of selecting non-annotator-rationales, but those sentences might be possible machine-rationales. Therefore, if the focus lies on extracting human-like annotator rationales, a low bin-size should be used. If the focus lies on finding rationales used by the machine, a higher bin-size value should be chosen.

8.4 User evaluation

Jacovi et al., 2020 warned about using user judgement for faithfulness evaluation by interpretation (see Section 2.2.3), but in this study, some new insights on the faithfulness of explanations were found using indirect human-judgement. In the blind study, a small set of documents was misclassified by the ML models, and users were asked to classify them based on the given explanations. By comparing the user-classifications to model-classifications, an indication of how faithful the explanations were could be gathered. When a user's prediction aligns with the model's prediction, the explanation supports the model's prediction and is at least faithful enough to not be misleading. If a model does not reason similarly to humans, the question of faithfulness is nevertheless raised again, since the explanation might be faithful but not interpretable to humans. In future work, a notion of faithfulness that *an explanation that does not contradict a model's prediction* (according to users), might be a form of sufficient faithfulness as proposed by Jacovi et al. (2020).

The polarity of a document might also influence the explanation quality, for example, that negative documents are more easily classified by ML models. In this study, no distinction between positive or negative documents is made in the user evaluation. Future work with a larger set of documents for the user evaluation could give new insights.

8.5 Subjective rationale quality

In Section 3.2.2, I made some assumptions about the subjective quality of explanations. I made a distinction between interpretable and subjectively incomplete explanations, viz, an incomplete explanation will not give enough information and a non-interpretable explanation will give misleading information. This distinction is not exhaustive, or even too harsh, since a subjectively incomplete explanation can also indicate that the signals are not interpretable to the user, but complete for the model (meaning that the model can use it as a classification base). From a user's viewpoint, the explanation is subjectively incomplete, and therefore I made the distinction in this study.

Another note is that the comprehensibility of an explanation is not easily measured, because there are different levels of understanding a concept. A user can understand why a prediction is made based on only the input, but a user can also know and understand the complete reasoning process of the model. By asking the user how comprehensible the prediction is, this level cannot precisely be estimated. In future work, I would advise focusing on a specific level of comprehensibility, like a user's ability to reproduce a prediction, a complete understanding of the model's reasoning process, or how useful an explanation is to a user when evaluating a prediction.

9 Conclusion

In this research, different approaches to finding faithful machine-annotated rationales were examined and evaluated. The evaluation was done by estimating faithfulness using the LAO-confidence, comparing explanations to annotator rationales, and through a user evaluation.

For evaluation by annotator rationale similarity, a benchmark of annotator rationales was used. The baseline of randomly selected rationales had the following rationale quality scores: 0.171 for the Jaccard index, 0.226 for the Completeness index and 0.495 for the Incompleteness index. In the following sections, these values are used to give an insight into the similarity of different results. The annotator rationale benchmark contained 7 rationales per document on average.

SQ1. Feature Extraction method

Finding machine-annotated rationales using the weights of a Linear classifier (see Section 5.3.2) proved to be a simple and fast way of finding post-hoc explanations.

Are machine-annotated rationales that were found using post-hoc feature extraction faithful?

The accessible and interpretable weights of the LinearSVC model make the model 'inherently interpretable', thus the base of the explanation (the words) is faithful. The faithfulness of this method is weakened by selecting the sentences that contain *most* of the extracted words as rationales because not all words are included in the selected rationales. The low LAO-confidence score (0.264) shows this weakened faithfulness too. The conclusion is that the FE-method explains more faithful than feasible, but not completely faithful. The explanations are an altered version of the signals used in the classification and are thus not based directly on the transparent reasoning of the model.

Are machine-annotated rationales that were found using post-hoc feature extraction similar to annotator rationales?

The Jaccard, Completeness, and Incompleteness indexes for the FE machine-annotated rationales compared to annotator rationales are 0.227, 0.323, and 0.597 respectively. The number of rationales selected on average is 7. This shows that even though a similar number of rationales as the annotator rationale benchmark is selected, the MaRs are only slightly better than the random rationale baseline. Therefore they are not very similar to annotator rationales.

SQ2. Leave-One-Out method

The Leave-One-Out method as described in Section 5.3.3 proved to be a computationally expensive method of extracting post-hoc machine-annotated rationales, that made heavy use of the Linearity Assumption (see Section 2.2.3).

Are machine-annotated rationales that were found using the Leave-One-Out method faithful?

As long as the Linearity Assumption holds, this RE-method is faithful. The LAO-confidence score of 0.351 also shows that the explanations are more faithful than the feature extraction method MaRs.

Are machine-annotated rationales that were found using the Leave-One-Out method similar to annotator rationales?

With Jaccard, Completeness and Incompleteness indexes of 0.192, 0.370, and 0.698 respectively, the machine-annotated rationales by the LOO-method are not very similar to annotator rationales. With an average of 6.4, the method selects fewer rationales than the annotator rationale benchmark. This is also reflected in the high Incompleteness index.

SQ3 (1). BagNetsTextAll

As described in Section 5.2.2, extracting machine-annotated rationales using the BagNetsTextAll model was done during the prediction process, making the method a fast and transparent way of extracting local machine-annotated rationales.

Are machine-annotated rationales that were found using the BagNetsTextAll model faithful?

Since the machine-annotated rationales are selected using values from inside the model, the BagNetsTextAll model is faithful by design. Nonetheless, the LAO-confidence score is very low (0.025). This can be explained by the small variance in outputs as described in Section 6.3.4, and in this particular case is not an indicator of a lack of faithfulness.

Are machine-annotated rationales that were found using the BagNetsTextAll model similar to annotator rationales?

The BagNetsTextAll machine-annotated rationale sets contain 6.11 rationales on average, thus fewer than the human annotators in the annotator rationale benchmark. With a Jaccard index of 0.326, Completeness index of 0.546, and Incompleteness index of 0.562, these machine-annotated rationales are more similar to annotator rationales than the previously mentioned RE-method MaRs.

SQ3 (2). BagNetsTextRats

The BagNetsTextRats method of finding machine-annotated rationales, as described in Section 5.3.4, is similar to the BagNetsTextAll model, but the BagNetsTextRats showed some more fine-tuned values in both the rationale-values and predictions.

Are machine-annotated rationales that were found using the BagNetsTextRats model faithful?

The machine-annotated rationales found are similarly faithful to the previously mentioned BagNetsTextAll model. The outputs of the BagNetsTextRats model are calculated differently, which results in a much higher LAO-confidence than the BagNetsTextAll model, namely 0.566. This shows that the LAO-confidence in the current form is very model-dependent.

Are machine-annotated rationales that were found using the BagNetsTextRats model similar to annotator rationales?

The BagNetsTextRats model selects 5.47 MaRs on average, which is fewer than the BagNetsTextAll model. The rationale quality indexes are 0.316, 0.550, and 0.595 for Jaccard, Completeness, and Incompleteness. Even though fewer machine-annotated rationales are selected than the annotator rationale benchmark, the sets are quite similar.

SQ4. Annotator rationale search model (RationaleSearch)

The RationaleSearch machine-annotated rationales explain predictions using global information about the form of annotator rationale sentences used in the classification task compared to neutral sentences. This method is neither post-hoc nor transparent but can be seen as a combined task of classifying documents and finding rationales in documents.

Are machine-annotated rationales that were found using the RationaleSearch model faithful?

The machine-annotated rationales are not faithful, because they are not based on the algorithmic process of the model. The LAO-confidence score of 0.352 does show that the machine-annotated rationales are pointing in the right direction nonetheless.

Are machine-annotated rationales that were found using the RationaleSearch model similar to annotator rationales?

With a Jaccard index of 0.328, a Completeness index 0.586, an Incompleteness index of 0.578 and an average machine-annotated rationale set size of 5.25, the found RationaleSearch MaRs are more concise but still similar to annotator rationales.

SQ5. Machine-annotated rationale comparison

How do machine-annotated rationales that were found using the methods SQ1, SQ2, SQ3, and SQ4 compare on sentence set overlap?

The machine-annotated rationales found by the BagNetsTextAll and BagNetsTextRats are most similar with a Jaccard index of 0.678. The RationaleSearch MaRs are also relatively similar to the BagNetsTextAll and

BagNetsTextRats rationale sets, with a rounded Jaccard index of 0.338 for both methods. The FE-method MaRs are most dissimilar from other RE-methods, with a Jaccard index of ±0.15. Overall, the different RE-methods find similar explanations.

SQ6. User Evaluation

Do machine-annotated rationales that were found using the methods SQ1, SQ2, SQ3, or SQ4 form complete explanations for end-users?

All machine-annotated rationales except RationaleSearch MaRs were marked subjectively incomplete in similar quantities (±10%) in the blind study. The RationaleSearch MaRs were marked subjectively incomplete only 2.6% of the cases, indicating that almost all sets of machine-annotated rationales contained enough information to base a classification on. Explanations for incorrect classifications by the ML models are more often marked subjectively incomplete, except for machine-annotated rationales by the FE-method.

Do machine-annotated rationales that were found using the methods SQ1, SQ2, SQ3, or SQ4 form comprehensible explanations for end-users?

As discussed in Section 8.5, the notion of comprehensibility is quite hard to measure, since one cannot simply ask an end-user 'do you understand this?', because any answer would not truly be an answer. In this study, I measure the comprehensibility by the last two tasks in the user evaluation, because they indirectly show how well the users understood the explanation.

Rationale extraction methods that are trained to explain similar to humans are more comprehensible than methods that just explain the models. The explanations by the RationaleSearch model are most similar to annotator rationales (by design), and thus are most comprehensible for users. In the blind study in the user evaluation, this is also shown. The BagNetsTextAll and BagNetsTextRats model machine-annotated rationales have similar rationale quality scores but do not perform as well in the user evaluation. There appears to be a trade-off between faithfully and comprehensibly explaining ML models.

Do machine-annotated rationales that were found using the methods SQ1, SQ2, SQ3, or SQ4 form similar explanations for end-users compared to annotator rationales in a blind study?

Machine-annotated rationales by the RationaleSearch model were slightly more useful than annotator rationales in the blind study. Other methods are more often marked subjectively incomplete or not interpretable, but perform only ±15% worse than annotator rationales. Taking into account that ML models reason different from human decision-makers, this difference can be expected. Machine-annotated rationales explain similar to annotator rationales, especially when an ML model reasons somewhat similar to humans.

The focus and goal of the explanation are important to keep in mind. If the focus lies on explaining a model and its reasoning, a transparent model like BagNetsTextAll or BagNetsTextRats can be used. If the goal is to find rationales that are similar to human explanations, like annotator rationales, a model that is trained to explain in that manner, like the RationaleSearch model, can be used.

Research question: How can faithful machine-annotated rationales for text classification be found that are complete and comprehensible, and do they form explanations for end-users similar to annotator rationales?

Faithful machine-annotated rationales for text classification can be found using post-hoc rationale extraction methods or by specifically designed self-explaining machine learning models. Which method is most useful, depends on the situation, the requirements, and the goal of the explanation task. Machine-annotated rationales are usually less complete than annotator rationales, but not necessarily too incomplete. Machine-annotated rationales are also less comprehensible than annotator rationales, but this might only be caused by the difference in human and machine learning model reasoning. Overall, machine-annotated rationales are not very similar to annotator rationales from a perspective of set theory. Machine-annotated rationales are useful in the work field, especially to explain machine learning models. The most faithful explanation method found in this study is the method using the BagNetsTextAll and BagNetsTextRats models, which are designed to be self-explaining. If the focus lies on finding explaining parts of the input for a classification task, a model that is trained using human-provided explanations, like the RationaleSearch model, is more suitable.

Glossary

algorithmic transparency The degree to which the inner workings of a model are understandable to humans. 18, 19

annotator rationale Rationales gathered from human users, who were asked to annotate 'why' a certain classification should be applied. 13, 14, 21–25, 30, 33–35, 37, 39, 40, 44–48, 54, 56, 59–61, 64–66, 68–70

anti-rationale a rationale that supports a different class than the documents class. 46, 54

BagNetsTextAll A model classifies documents on polarity and can be used to extract machine-annotated rationales during the prediction process. See Section 5.2.2. 32, 33, 64

BagNetsTextRats A model similar to BagNetsTextAll that classifies documents on polarity based on the found machine-annotated rationales (and can be used to extract machine-annotated rationales during the prediction process). See Section 5.3.4. 32, 33, 64

bin-size Value that determines what percentage of the distribution of a set of sentences estimated as rationales by a rationale extraction-method are selected as machine-annotated rationales. 36, 39

black box Something that is opaque, meaning that the internal process and reasoning is unknown. 12, 31

Completeness index An index that measures the number of annotator rationales (AR) present in the machine-annotated rationale (MR) set. 44

decision tree An interpretable classifier that classifies using a tree-like structure. 16

decomposability The degree to which a model can be taken apart so that every calculation, input and parameter has an explanation. 18

deteriorated model A model that has been artificially deteriorated to misclassify. 58, 78

explainability How well a concept can be explained. In AI, it describes the extend to which an ML model can be explained in human terms. 18

explainable artificial intelligence Artificial intelligence that can be explained or interpreted to humans. 12, 17

explanation A justification of a certain action, that provides new information on the given action. 18–20, 22, 24

faithful A faithful explanation is one that explains the decision process according to algorithmic transparency, thus truthfully and complete. 12, 14, 19, 24, 64, 70

feasible Not faithful, but plausible. (See faithful). 19

feature extraction Rationale extraction using the trained LinearSVC model's weights. 36, 48, 50, 51, 56, 64, 68

fidelity The degree to which an explanation method can successfully mimic a model's predictions in terms of accuracy.. 19, 25

global explainability An explanation for an ML model that explains the model or a set of models. 20

hybrid explainable model A complex black box and a simpler transparent model combined to perform a task. 21

Incompleteness index An index that measures the number of missing annotator rationales (AR) in the machine-annotated rationale (MR) set. 44

interpretability The degree to which an observer can understand the cause of a decision. 17

Jaccard index An index that is used to measure the overlap between two sets. In this work, it is usually used to compare machine-annotated rationales to annotator rationales. 44

Leave-All-Out confidence A measurement of faithfulness. The difference between the prediction based on a complete document and a prediction were the machine-annotated rationales are removed from the document and replaced by padding. 46

Leave-One-Out method A method that finds machine-annotated rationales by leaving out parts of the input and measuring the affect on the model. 37, 56, 64

local explainability Explanation by connecting input (features) to outputs. 20

machine learning The application of AI to create systems that automatically learn from experience. 18, 30, 37, 40

machine-annotated rationale Rationales gathered from a machine, that form an explanation for the classification decision. 13–15, 24, 25, 30, 31, 33–39, 44–48, 50–54, 56, 58, 60, 61, 63–66, 68–70, 78

naïve Bayes A classifier that classifies using a priori and a posteriori probabilities. 16

neural network A network that simulates the human brain to some extent and can be used to perform classification tasks. 12, 16, 30

opaque See black box. 12, 18, 30, 31

post-hoc An explanation that is not a representative of the model's inner working, but instead gives insight into the prediction or model process. 18, 24

rationale An human-understandable natural language explanation for an action or decision made by a machine. 13, 20, 24

rationale extraction The process of finding an explanation for a prediction of a given model in the form of machine-annotated rationales. 13, 15, 34, 37, 44, 47, 48, 58, 64, 65

RationaleSearch A model that combines the **annotator rationale search model** and **SimpleNet**. See Section 5.2.4. 33, 64

simulatability The degree to which a human can reproduce the prediction given the model, input and parameters. 18

subjective incompleteness An explanation is subjectively incomplete when it does not contain enough information to the receiver to base a decision on. 56, 61

support vector machine A classifier that classifies using decision boundaries in the feature space. 16, 23

text classification The task of determining the class of a given document or text. 13, 14, 16, 21, 64, 70

the Linearity Assumption The assumption that certain predictor (input) variables explain the dependent (output) variable, as is seen in linear regression models. 19, 22, 24, 37, 65, 68

transparency The interpretability of a model's algorithm or parameters, by the simulatability, decomposability, and algorithmic transparency of a model. A completely transparent model is the opposite of a black box model. 12, 18, 21, 24

References

Aggarwal, Charu C and ChengXiang Zhai (2012). "A survey of text classification algorithms". In: *Mining text data*. Springer, pp. 163–222 (cit. on p. 18).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (cit. on p. 24).

Bekri, Nadia, Jasmin Kling, and Marco Huber (May 2019). "A Study on Trust in Black Box Models and Post-hoc Explanations". In: pp. 35–46. ISBN: 978-3-658-07615-3. DOI: 10.1007/978-3-030-20055-8_4 (cit. on pp. 18, 20, 21).

Brendel, Wieland and Matthias Bethge (2019). *Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet*. arXiv: 1904.00760 [cs.CV] (cit. on pp. 23, 34).

Caruana, Rich et al. (Feb. 1999). "Case-based explanation of non-case-based learning methods". In: *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pp. 212–5 (cit. on p. 24).

Chhatwal, Rishi et al. (Dec. 2018). "Explainable Text Classification in Legal Document Review A Case Study of Explainable Predictive Coding". In: pp. 1905–1911. DOI: 10.1109/BigData.2018.8622073 (cit. on p. 25).

Chintala, Soumith (2012). "Sentiment Analysis using neural architectures". In: *New York University* (cit. on p. 32).

Clos, Jérémie, Nirmalie Wiratunga, and Stewart Massie (2017). "Towards explainable text classification by jointly learning lexicon and modifier terms". In: *IJCAI-17 Workshop on Explainable AI (XAI)*, p. 19 (cit. on p. 25).

Devlin, Jacob et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL] (cit. on p. 32).

Doran, Derek, Sarah Schulz, and Tarek R Besold (2017). "What does explainable AI really mean? A new conceptualization of perspectives". In: *arXiv preprint arXiv:1710.00794* (cit. on p. 20).

Dror, Rotem et al. (July 2018). "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1383–1392. DOI: 10.18653/v1/P18-1128 (cit. on p. 42).

Ehsan, Upol et al. (2019). "Automated rationale generation: a technique for explainable AI and its effects on human perceptions". In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. ACM, pp. 263–274 (cit. on pp. 15, 21, 22).

Hind, Michael et al. (2018). *TED: Teaching AI to Explain its Decisions*. arXiv: 1811.04896 [cs.AI] (cit. on p. 24).

Hu, Jin (Oct. 2018). "Explainable Deep Learning for Natural Language Processing". MA thesis. KTH royal institute of technology, school of electrical engineering and computer science (cit. on p. 18).

Jacovi, Alon and Yoav Goldberg (2020). *Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness?* arXiv: 2004.03685 [cs.CL] (cit. on pp. 20, 21, 26, 68).

Jain, Sarthak and Byron C. Wallace (2019). "Attention is not Explanation". In: *CoRR* abs/1902.10186. arXiv: 1902.10186 (cit. on p. 24).

Jain, Sarthak et al. (2020). "Learning to faithfully rationalize by construction". In: *arXiv preprint arXiv:2005.00115* (cit. on pp. 25, 32, 36, 37).

Julia Angwin Jeff Larson, Surya Mattu and Lauren Kirchner (May 2016). *Machine Bias. There's software used across the country to predict future criminals. And it's biased against blacks.* (Visited on 07/02/2020) (cit. on p. 14).

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG] (cit. on p. 32).

Korde, Vandana (Mar. 2012). "Text Classification and Classifiers:A Survey". In: *International Journal of Artificial Intelligence & Applications* 3, pp. 85–99. DOI: 10.5121/ijaia.2012.3208 (cit. on p. 18).

Kovacs, Balazs (Mar. 2014). "A Monte Carlo permutation test for co-occurrence data". In: *Quality & Quantity* 48. DOI: 10.1007/s11135-012-9817-x (cit. on p. 42).

Lakkaraju, Himabindu et al. (2019). "Faithful and customizable explanations of black box models". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 131–138 (cit. on p. 24).

Lapuschkin, Sebastian et al. (2019). "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn". In: *CoRR* abs/1902.10178. arXiv: 1902.10178 (cit. on p. 19).

Lei, Tao, Regina Barzilay, and Tommi Jaakkola (2016). "Rationalizing neural predictions". In: *arXiv preprint arXiv:1606.04155* (cit. on pp. 25, 36).

Lipton, Zachary C. (2016). *The Mythos of Model Interpretability*. arXiv: 1606.03490 [cs.LG] (cit. on pp. 14, 20, 21).

Liu, Hui, Qingyu Yin, and William Yang Wang (2018). "Towards explainable NLP: A generative explanation framework for text classification". In: *arXiv preprint arXiv:1811.00196* (cit. on p. 25).
Loper, Edward and Steven Bird (2002). "NLTK: The Natural Language Toolkit". In: *CoRR* cs.CL/0205028 (cit. on p. 32).
Marozzi, Marco (Jan. 2004). "Some remarks about the number of permutations one should consider to perform a permutation test". In: *Statistica* 1. DOI: 10.6092/issn.1973-2201/32 (cit. on p. 42).
Menon, Sachit et al. (2020). *PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models*. arXiv: 2003.03808 [cs.CV] (cit. on p. 14).
Miller, Frederic P., Agnes F. Vandome, and John McBrewster (2009). *Internet Movie Database*. Alpha Press. ISBN: 6130099681 (cit. on p. 30).
Miller, Tim (2017). *Explanation in Artificial Intelligence: Insights from the Social Sciences*. arXiv: 1706.07269 [cs.AI] (cit. on pp. 19, 26).
Mohseni, Sina and Eric D. Ragan (2018). *A Human-Grounded Evaluation Benchmark for Local Explanations of Machine Learning*. arXiv: 1801.05075 [cs.HC] (cit. on p. 22).
Narayanan, Vivek, Ishan Arora, and Arjun Bhatia (2013). "Fast and accurate sentiment classification using an enhanced Naive Bayes model". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, pp. 194–201 (cit. on p. 32).
Nguyen, Dong (2018). "Comparing automatic and human evaluation of local explanations for text classification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1069–1078 (cit. on p. 21).
Pang, Bo and Lillian Lee (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the ACL* (cit. on p. 30).
Papernot, Nicolas and Patrick McDaniel (2018). *Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning*. arXiv: 1803.04765 [cs.LG] (cit. on p. 23).
Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035 (cit. on p. 32).
Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on p. 33).
Plunkett, Kim and Jeffrey L Elman (1997). *Exercises in rethinking innateness: A handbook for connectionist simulations*. Mit Press (cit. on pp. 32, 42).
Preece, Alun (2018). "Asking 'Why' in AI: Explainability of intelligent systems–perspectives and challenges". In: *Intelligent Systems in Accounting, Finance and Management* 25.2, pp. 63–72 (cit. on pp. 19, 20, 26).
Raaijmakers, Stephan, Maya Sappelli, and Wessel Kraaij (2017). "Investigating the Interpretability of Hidden Layers in Deep Text Mining". In: *Proceedings of the 13th International Conference on Semantic Systems*. Semantics2017. Amsterdam, Netherlands: Association for Computing Machinery, pp. 177–180. ISBN: 9781450352963. DOI: 10.1145/3132218.3132240 (cit. on p. 25).
Reimers, Nils and Iryna Gurevych (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. arXiv: 1908.10084 [cs.CL] (cit. on p. 32).
Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). *Why Should I Trust You? Explaining the Predictions of Any Classifier*. arXiv: 1602.04938 [cs.LG] (cit. on pp. 19, 21, 22).
Robnik-Šikonja, Marko and Igor Kononenko (2008). "Explaining classifications for individual instances". In: *IEEE Transactions on Knowledge and Data Engineering* 20.5, pp. 589–600 (cit. on pp. 25, 39).
Ross, Andrew Slavin, Michael C. Hughes, and Finale Doshi-Velez (2017). "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations". In: *CoRR* abs/1703.03717. arXiv: 1703.03717 (cit. on pp. 19, 22, 24).
Rudin, Cynthia and Berk Ustun (2018). "Optimized scoring systems: toward trust in machine learning for healthcare and criminal justice". In: *Interfaces* 48.5, pp. 449–466 (cit. on p. 22).
Sap, Maarten et al. (2019). "The risk of racial bias in hate speech detection". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1668–1678 (cit. on p. 14).
Schmitz, G. P. J., C. Aldrich, and F. S. Gouws (Nov. 1999). "ANN-DT: an algorithm for extraction of decision trees from artificial neural networks". In: *IEEE Transactions on Neural Networks* 10.6, pp. 1392–1401. ISSN: 1941-0093. DOI: 10.1109/72.809084 (cit. on p. 24).
Timmaraju, Aditya and Vikesh Khanna (2015). "Sentiment analysis on movie reviews using recursive and recurrent neural network architectures". In: *Semantic Scholar* (cit. on p. 32).
Truong, Kevin (June 2020). *This Image of a White Barack Obama Is AI's Racial Bias Problem In a Nutshell*. (Visited on 07/02/2020) (cit. on p. 14).
Weitz, Katharina et al. (2019). ""Do You Trust Me?": Increasing User-Trust by Integrating Virtual Agents in Explainable AI Interaction Design". In: *Proceedings of the 19th ACM International Conference on Intelligent*

Virtual Agents. IVA '19. Paris, France: Association for Computing Machinery, pp. 7–9. ISBN: 9781450366724. DOI: 10.1145/3308532.3329441 (cit. on p. 22).

Wiegreffe, Sarah and Yuval Pinter (2019). *Attention is not not Explanation*. arXiv: 1908.04626 [cs.CL] (cit. on p. 24).

Xu, Kelvin et al. (2015). "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*, pp. 2048–2057 (cit. on p. 24).

Yessenalina, Ainur, Yejin Choi, and Claire Cardie (2010). "Automatically generating annotator rationales to improve sentiment classification". In: *Proceedings of the ACL 2010 Conference Short Papers*, pp. 336–341 (cit. on p. 25).

Zaidan, Omar, Jason Eisner, and Christine Piatko (Apr. 2007). "Using "Annotator Rationales" to Improve Machine Learning for Text Categorization". In: *NAACL HLT 2007; Proceedings of the Main Conference*, pp. 260–267 (cit. on pp. 15, 22–25, 30, 32, 42, 48, 56, 58, 62, 68).

Zaidan, Omar F. and Jason Eisner (Oct. 2008). "Modeling Annotators: A Generative Approach to Learning from Annotator Rationales". In: *Proceedings of EMNLP 2008*, pp. 31–40 (cit. on p. 25).

Zhang, Quanshi, Ying Nian Wu, and Song-Chun Zhu (2017). *Interpretable Convolutional Neural Networks*. arXiv: 1710.00935 [cs.CV] (cit. on p. 24).

Zhong, Ruiqi, Steven Shao, and Kathleen McKeown (2019). "Fine-grained sentiment analysis with faithful attention". In: *arXiv preprint arXiv:1908.06870* (cit. on p. 24).

Zou, James and Londa Schiebinger (2018). *AI can be sexist and racist—it's time to make it fair* (cit. on p. 14).

A Appendix

A.1 Padding

The longest document in the dataset contains 116 sentences and Sentence-BERT converts every sentence to a vector of length 768. Therefore every document is stored as a 116×768 multidimensional array. The padding used is an 768-length vector of zeros. The padding consists of a vector of only zeros with similar dimensions as one Sentence-BERT embedded sentence, and should represent an empty sentence. This means that no information is stored in the padding.

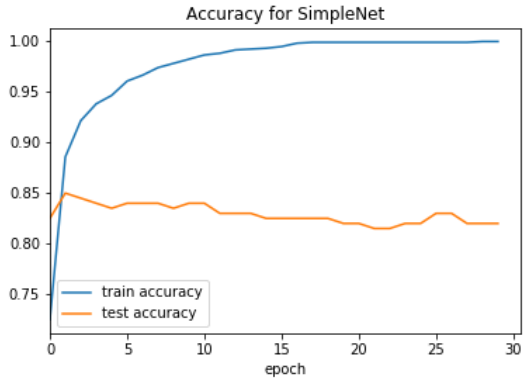
A.2 Significance testing

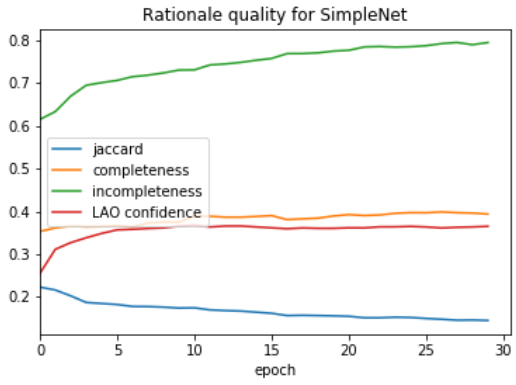
In this study, I used permutation testing to determine significant difference in outputs. For rationale quality, the significance in differences added insight. Comparing model classifications resulted in some unexpected values, especially for the rounded predictions. Permutation testing should not be applied to data consisting of small sets of categorical values (like 0 or 1) when measuring significant differences, because randomly permuting these kinds of datasets does not give similar results to more continuous datasets. Another approach for comparing classification behaviour of ML models for the same task and dataset should be taken.

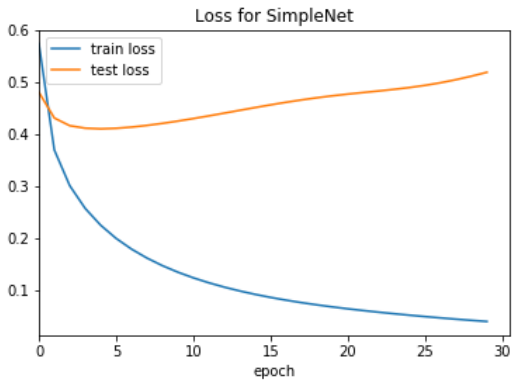
A.3 Classification models

A.3.1 SimpleNet

Figure 26: Training statistics for the SimpleNet model with 1 linear layer





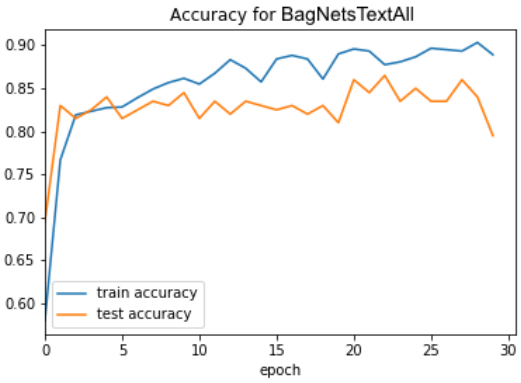


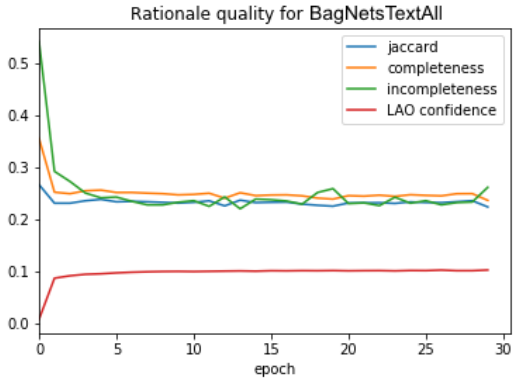
Model performance metrics				
accuracy	loss	prec.	recall	f1-score
0.800	0.354	0.861	0.715	0.781
	Rationale quality metrics			
	LAO-conf.	Jaccard	Compl.	Incompl.
	0.351	0.192	0.370	0.302

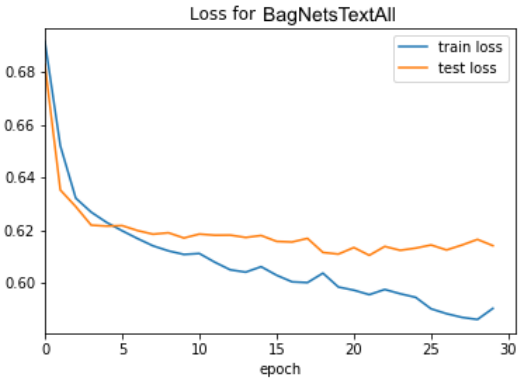
Table 20: SimpleNet Test Metrics

A.3.2 BagNetsTextAll

Figure 27: Training statistics for the BagNetsTextAll model





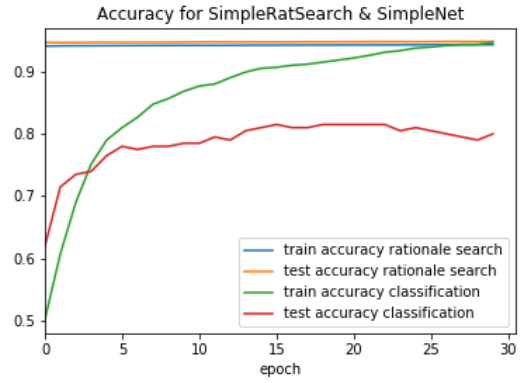


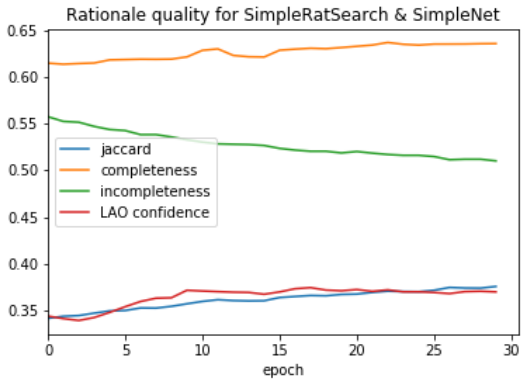
Model performance metrics				
accuracy	loss	prec.	recall	f1-score
0.830	0.622	0.787	0.905	0.842
Rationale quality metrics				
	LAO-conf.	Jaccard	Compl.	Incompl.
	0.096	0.245	0.262	0.763

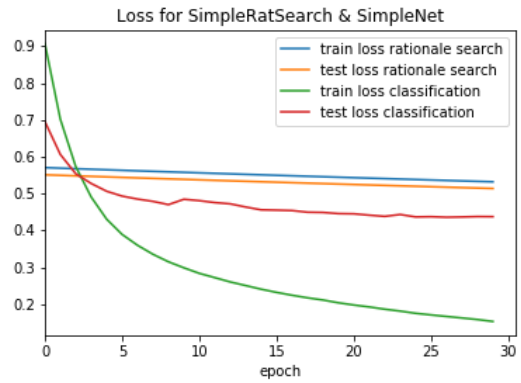
Table 21: BagNetsTextAll Test Metrics

A.3.3 RationaleSearch

Figure 28: Training statistics for the RationaleSearch model







Model performance metrics				
accuracy	loss	prec.	recall	f1-score
0.777	0.457	0.776	0.780	0.778
Rationale quality metrics				
	LAO-conf.	Jaccard	Compl.	Incompl.
	0.352	0.328	0.586	0.578

Table 22: RationaleSearch Test Metrics

A.3.4 Deteriorated models

For the user evaluation (described in Section 7) **deteriorated model**s were used to find out how interpretable explanations are to users. To deteriorate a model, it was trained on an adjusted version of the training dataset, where all class labels were randomly shuffled. Testing and validating was done using the same sets used in the normal training process.

Deteriorating the models showed different behaviour for models. In Table 23 the classification performance for the deteriorated models is shown. The accuracy lies around 0.5, with a significantly lower accuracy for the RationaleSearch model on both the tuning set and the test set. The LinearSVC model did not converge during training. See Table 24 for an overview of the significant differences in predictions. The loss for the deteriorated BagNetsTextRats is greater than 1.

The RationaleSearch model was deteriorated the same way as the other models, meaning that only the classification sub-model is deteriorated, and not the rationale search sub-model. This is why the rationale search sub-model still shows high accuracy scores. Deteriorating the rationale search sub-model resulted in an extremely low accuracy (0.01) and empty machine-annotated rationale sets. Since empty machine-annotated rationale set cannot be used as explanation, deteriorating the rationale search sub-model was not useful. Only deteriorating the classification sub-model had the effect that some documents were classified differently on exactly the same set of found rationales.

	LinearSVC	SimpleNet	BagNetsTextAll	BagNetsTextRats	RationaleSearch	
					search	classification
Train						
epochs		5	30	30	30	
accuracy	0.48	0.540	0.510	0.525	0.946	0.460
loss		0.742	0.690	1.20	0.553	0.838
Test						
accuracy	0.533	0.460	0.502	0.535	0.947	0.482
loss		0.852	0.690	1.050	0.560	0.865

Table 23: Training accuracy and loss for deteriorated models.

	LinearSVC	SimpleNet	BagNetsTextAll	BagNetsTextRats	RationaleSearch
LinearSVC		**0.410**	0.001	0.001	**0.493**
SimpleNet	0.004		0.001	0.001	**0.115**
BagNetsTextAll	0.001	0.010		0.002	0.001
BagNetsTextRats	0.009	**0.152**	**0.066**		0.001
RationaleSearch	**0.261**	**0.259**	0.020	**0.869**	

Table 24: Significant differences in deteriorated model predictions on the test set. Left bottom half: non-rounded predictions. Right top half: rounded predictions. Non-significant values are in bold.

A.4 Rationale Extraction methods

A.4.1 Custom Feature Extraction from documents

Wordlist 1: Positive words (custom method):

jackie, war, perfect, beautiful, throughout, disney, extremely, wonderful, dark, strong, sometimes, fiction, deal, science, excellent, son, political, voice, enjoy, despite, tale, effective, definitely, viewer, single, others, impressive, reality, simple, towards, due, husband, animated, similar, violence, personal, among, sets, heart, future, told, aliens, feature, truman, history, leads, brilliant, emotional, oscar, force, tom, success, tells, important, surprisingly, private, amazing, stories, nature, certain, powerful, eventually, change, score, robert, call, easily, easy, hilarious, perfectly, uses, somewhat, happy, genre, lee, brings, created, release, america, create, using, latest, cameron, greatest, memorable, near, words, form, wars, visual, released, crime, child, george, working

Wordlist 2: Negative words (custom method):

worst, supposed, attempt, save, stupid, space, boring, kill, obvious, saw, worse, material, poor, seemed, van, late, wild, jokes, earth, harry, sequences, career, attempts, guess, killed, filmmakers, fails, mars, middle, waste, basically, thriller, talent, figure, guys, talk, happen, sequel, apparently, mission, bunch, mess, dr, giant, murder, none, went, eddie, planet, difficult, interest, laugh, annoying, predictable, murphy, potential, elements, ridiculous, decent, batman, girls, suspense, add, taking, begin, thinks, move, peter, somehow, absolutely, feeling, theater, wasted, plan, paul, hand, terrible, laughs, mostly, experience, crew, previous, talking, brother, brothers, business, piece, writer, obviously, alone, cool, rock, huge, appear, writing

A.4.2 Feature Extraction from model

Wordlist 3: Positive words (extracted from model):

great, quite, excellent, hilarious, fun, seen, overall, people, job, frank, life, different, change, enjoyable, takes, pace, memorable, enjoyed, mel, pulp, today, american, hollywood, comic, perfectly, force, horror, unlike, terrific, station, fine, surprise, enjoy, performances, sweet, true, animation, fantastic, matrix, mulan, times, performance, laughs, including, political, mamet, mike, leave, special, gas, single, works, follows, clean, bit, wife, titanic, gibson, entertaining, joe, seeing, casablanca, store, tucker, class, right, hackman, run, allows, fiction, witty, view, aliens, family, amazing, italian, personal, town, karen, seat, head, attention, dragon, kid, makes, davis, war, gay, playing, violence, chronicles, movies, school, ash, slightly

Wordlist 4: Negative words (extracted from model):

completely, women, crap, don, cheap, spawn, having, jason, seagal, screenplay, tired, batman, beast, style, lions, jesse, throw, space, godzilla, grace, writer, scott, loses, transported, julie, flat, tedious, looked, apparently, tries, series, recently, left, house, harry, brothers, wonder, make, self, old, ridiculous, mediocre, girls, acting, idea, giant, eve, career, mess, tv, brother, maybe, thriller, saved, jennifer, fails, effort, wasted, adam, worse, annoying, headed, attempt, grade, carpenter, jakob, plot, mario, talent, unfunny, given, lame, pointless, dull, 90, better, looks, material, joke, director, awful, stupid, poor, script, reason, boring, words, supposed, going, unfortunately, bad, worst, extraordinarily, horrendous, waste

A.5 Rationales in documents

Document	Tag	# sentences
negR_868.txt	1.0 (negative)	10

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	3		1.000				
	LOO	2	True	0.683	0.138	0.000	0.000	1.000
	FE	7	True	0.559	0.013	0.250	0.286	0.333
	BNAll	2	True	0.514	0.008	0.667	1.000	0.333
	BNRats	2	True	0.985	0.244	0.667	1.000	0.333
	RS	2	True	0.704	0.285	0.667	1.000	0.333

Table 25: Metrics for negative document negR_868.txt.

Aspiring broadway composer Robert (Aaron Williams) secretly carries a torch for his best friend, struggling actor Marc (Michael Shawn Lucas). The problem is, Marc only has eyes for "perfect 10s," which the geeky, insecure Robert certainly is not. Meanwhile, Marc's spoiled (hetero) female roommate, Cynthia (Mara Hobel), spends her days lying about their apartment and harrassing magazine editor Tina brown. Writer-director Victor Mignatti's "very romantic comedy"

(as the ad campaign states) is supposed to be (pardon the pun) a gay ol' romp, but it's hard to have much fun with these annoying, self-absorbed characters and their shallow personal problems : Marc and Cynthia have sitcom-level domestic "crises" (such as trying to kill bugs–how hilarious); Robert and Marc go to acting class (how riveting); the zaftig Cynthia goes on eating binges (how original). But more than anything else, the three whine. Constantly. Marc whines about his turbulent romance with an apparent "10," David (Hugh Panaro), the hunky musician from across the way; Robert whines about not being able to find the right guy; Cynthia whines about having to find a job (horrors). The terrible trio whine their way to a happy ending that is wholly undeserved. Add in overly broad performances and some laughable lipsynching by Panaro, and you're left with one astonishing piece of cinematic damage.

Figure 29: Rationales for negative document negR_868.txt.

Document	Tag	# sentences
negR_875.txt	1.0 (negative)	12

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	7		1.000				
	LOO	5	True	0.824	0.402	0.714	1.000	0.286
	FE	7	False	0.338	0.247	0.400	0.571	0.429
	BNAll	4	True	0.513	0.016	0.571	1.000	0.429
	BNRats	3	True	0.959	0.329	0.429	1.000	0.571
	RS	5	True	0.791	0.199	0.333	0.600	0.571

Table 26: Metrics for negative document negR_875.txt.

Everything in the phantom you have seen many times before and there is nothing new presented here. Wincer displays absolutely no skill in setting up an exciting action sequence. Billy Zane is wooden as the hero. Kristy Swanson is given very little to do, and does very little with it. Treat Williams, looking like Rhett Butler but sounding like Mickey Mouse, is one of the worst villains i have ever seen in a movie. Only Catherine zeta Jones, as one of Williams cohorts turns in a good performance. She has energy and spunk, which the movie needed much more of. Oh yeah, the phantom also has a secret identity but this is so poorly played out you won't even care. About the only things i can recommend are a good performance by Jones, and some colorful scenery. However, if you're looking for a fun family movie, go watch the underrated flipper. This

is not a good movie.

Figure 30: Rationales for negative document negR_875.txt.

Document	Tag	# sentences
posR_774.txt	0.0 (positive)	12

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	2		0.000				
	LOO	6	False	0.543	0.210	0.000	0.000	1.000
	FE	7	True	0.112	0.396	0.286	0.286	0.000
	BNAll	1	True	0.499	0.004	0.500	1.000	0.500
	BNRats	2	False	0.658	0.651	0.333	0.500	0.500
	RS	2	True	0.222	0.777	1.000	1.000	0.000

Table 27: Metrics for positive document posR_774.txt.

Almost a full decade before Steven Spielberg's saving private Ryan asked whether a film could be both "anti war" and "pro-soldier", john Irvin's hamburger hill proved it could. Lost in the inundation of critical acclaim that greeted Oliver stone's platoon, this excellent film was dismissed as "too militaristic". It's hard to understand exactly why—unless Irvin, in assembling his motley collection of young men who for predictable (and often naive) reasons "chose to show up" for the Vietnam debacle, —has refused to present us with the stone killer, drug-stoked psycho and ruthless opportunist who have become to Vietnam war epics what "the polack, the hillbilly and the kid from Brooklyn" became to WWII movies. Hamburger hill, based on a true story, is not an easy film to watch. There is a scene that will have graying anti-war activists squirming in their seats, or moved to genuine tears. And the climactic final assault on the "hill" in question is visually confusing. Gristly realities are presented in brief flashes, as if the brain dared not acknowledged what it had encountered. And in the mud and smoke officer and enlistee, veteran and "newbie", black soldier and white, become almost indistinguishable from each other, as they do in the chaos of actual combat. The acting throughout is solid with an absolutely stellar performance rendered by Courtney b. Vance as doc–in a role that will have many flatly disbelieving that this is same actor they cheered as "seaman Jones" in McKernan's red October If you've seen private Ryan, you owe it to yourself to see hamburger hill–if only to determine that the all the valour and horror of Spielberg's

vision was as present in the ashau valley as it was at Omaha beach.

Table 28: Rationales for positive document posR_774.txt.

Document	Tag	# sentences
negR_702.txt	1.0 (negative)	16

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	4		1.000				
	LOO	5	True	0.701	0.308	0.500	0.600	0.250
	FE	7	True	0.622	0.058	0.100	0.143	0.750
	BNAll	3	True	0.514	0.013	0.400	0.667	0.500
	BNRats	3	True	0.958	0.309	0.400	0.667	0.500
	RS	4	True	0.846	0.152	0.333	0.500	0.500

Table 29: Metrics for negative document negR_702.txt.

In 1990, the surprise success an unheralded little movie called ghost instantly rescued the moribund careers of its trio of above-the-title stars, Patrick Swayze, demi Moore, and whoopi Goldberg Eight years later, Moore and Goldberg's careers aren't exactly thriving, but they have had their share of screen successes since; the same can't be said of Swayze, who has just added yet another turkey to his resume with the aptly named black dog. Forget the mortal kombat movies–this trucksploitation flick is the closest the movies has come to video games. Good truck driver Jack Crews (Swayze) must drive a cargo of illegal firearms from Atlanta to new jersey. Along the way, Jack and his crew of three run into a number of obstacles–such as a highway weigh station, evil truckers, and deadly uzi-firing motorcyclists. Every so often, like at the end of a video game "level" or "stage," the main baddie pops up : red (meat loaf, fresh from the triumph of spice world), who wants to steal the cache of guns. Just in case you forget his name or have trouble keeping track of who's driving what, all of red's vehicles, be it a pickup or a big rig, are painted–you guessed it–red. I could go into more of the plot specifics (such as Jack's dream of having a nice home with his family, the past trauma that sent him to prison and cost him his trucking license, the FBI/atf crew tracking the cargo), but they are of little importance. All that matters to director Kevin hooks and writers William Mickelberry and Dan Vining are the obstacles Jack confronts in his drive from point a to point b. But they fail at even this modest goal, for none of the highway chaos, as credibly staged as

it is, is terribly interesting, let alone exciting. Once you've seen a couple of trucks bang against each other or a big rig explode the first time, you've seen it every time. As dreary as black dog is as an entertainment, the saddest part about the film has nothing to do with what shows up onscreen; it's that Swayze has to reduce himself to such work. While far from the best of actors, he is certainly not horrible, and he is a charismatic presence. I don't know if it's his judgment or the dearth of quality job offers that leads him to involve himself with bombs such as black dog. Regardless, if he continues on this career track, could a tv series be far behind?

Figure 31: Rationales for negative document negR_702.txt.

Document	Tag	# sentences
posR_760.txt	0.0 (positive)	13

	Method	# MaRs	correct	prediction	LAO-conf.	Jaccard	Compl.	Incompl.
	AR	6		0.000				
	LOO	3	False	0.537	0.200	0.000	0.000	1.000
	FE	7	False	0.530	0.041	0.300	0.429	0.500
	BNAll	5	True	0.485	0.020	0.571	0.800	0.333
	BNRats	3	True	0.049	0.857	0.500	1.000	0.500
	RS	1	True	0.207	0.349	0.167	1.000	0.833

Table 30: Metrics for positive document posR_760.txt.

The "Italian Hitchcock" and acknowledged master of the Giallo murder-mystery Dario Argento again offers us a fascinating turn on the formula in pheneomena. This time the twist comes in the form of Jennifer Corvino (Jennifer Connelly), a bright teenager with gift for telepathically communicating with insects. Sent to a girls boarding school in Switzerland, she soon learns of a series of bizarre disappearances and at least one murder that has the school's population terrified. A chance meeting with a brilliant entomologist (Donald pleasance) leads the two of them to team up and solve the mystery with the aid of her remarkable gift. Phenomena is an imaginative, original thriller. Writer/director Argento creates several sequences of surreal, haunting beauty here, including a masterfully shot sleepwalking episode and a striking scene when a swarm of flying insects descends on the school at Jennifer's beckoning. The plot takes some wonderfully bizarre turns and the killer's identity is genuinely shocking and surprising.

The director took a big gamble with a soundtrack that mixes elements as diverse as heavy metal band iron maiden, ex-rolling stone bill Wyman, and Argento's favourite gothic/electronic outfit goblin. But it gels surprisingly well. The film's opening music reccurs several times, an eerie and evocative score that perfectly sets the overall tone. Argento fans beware: the film was released outside Europe in a terribly butchered form re-titled as creepers. This deleted nearly half an hour of footage, mainly of key dialogue scenes.

Figure 32: Rationales for positive document posR_760.txt.

A.6 Re-annotated documents

filename	Jaccard	Completeness	Incompleteness	# original rationales	# custom rationales
posR_701.txt	0.250	0.400	0.400	5	5
posR_719.txt	0.000	0.000	0.000	2	1
posR_723.txt	0.091	0.167	0.167	6	6
posR_743.txt	0.500	0.667	0.667	9	9
posR_753.txt	0.333	0.500	0.500	2	2
posR_772.txt	0.333	0.400	0.667	3	5
posR_789.txt	0.000	0.000	0.000	3	2
posR_796.txt	0.562	0.692	0.750	12	13
posR_799.txt	0.111	0.250	0.167	6	4
posR_808.txt	0.571	1.000	0.571	7	4
posR_826.txt	0.286	0.400	0.500	4	5
posR_850.txt	0.364	0.500	0.571	7	8
posR_869.txt	0.333	0.500	0.500	6	6
posR_893.txt	0.250	0.429	0.375	8	7
posR_899.txt	0.333	0.444	0.571	7	9
negR_709.txt	0.000	0.000	0.000	7	7
negR_713.txt	0.133	0.250	0.222	9	8
negR_725.txt	0.125	0.200	0.250	4	5
negR_744.txt	0.421	0.571	0.615	13	14
negR_759.txt	0.250	0.375	0.429	7	8
negR_762.txt	0.231	0.333	0.429	7	9
negR_771.txt	0.429	0.643	0.562	16	14
negR_780.txt	0.500	0.750	0.600	5	4
negR_802.txt	0.167	0.250	0.333	3	4
negR_847.txt	0.375	1.000	0.375	8	3
negR_852.txt	0.500	1.000	0.500	6	3
negR_866.txt	0.400	0.500	0.667	6	8
negR_869.txt	0.105	0.333	0.133	15	6
negR_885.txt	0.182	0.333	0.286	7	6
negR_892.txt	0.200	0.400	0.286	14	10
average	0.278	0.443	0.403	7.1	6.5

Table 31: Custom re-annotated documents compared to original annotated documents.

Incorrect classifications

Correct classifications				
model	incorrect	correct	incomplete	total
AR	0	37 (92.50%)	3 (7.50%)	40
BNRats	3 (8.11%)	29 (78.37%)	5 (13.51%)	37
FE	1 (3.22%)	25 (80.65%)	5 (16.12%)	31
LOO	3 (7.69%)	30 (76.92%)	6 (15.38%)	39
RS	0	37 (97.36%)	1 (2.63%)	38

Table 32: Results from the blind study from the user evaluation.

	All		
model	incorrect	correct	total
BNRats	69 (52.27)	63 (47.72%)	132
FE	81 (61.36%)	51 (38.63%)	132
LOO	60 (45.45%)	72 (54.54%)	132
RS	81 (750%)	27 (250%)	108

Table 33: Results from the model quality task from the user evaluation.

A.7 User evaluation results

A.7.1 Blind study

Document	Tag	# sentences
neg_939.txt	1.0 (negative)	19

	Method	# MaRs
	LOO	7
	FE	7
	BNRats	2
	RS	1

Post-chasing Amy, a slew of love-triangle movies: this month we have kissing a fool, co-starring Amy's own lee, and April brings us the object of my affection, which may as well be titled Chasing Allan, for it is the story of a woman who falls in love with her gay roommate. (To be absolutely six degrees of Kevin bacon about it, that film stars schwimmer's friend Jennifer Aniston.) If only Kevin smith could write them all. Schwimmer stars as womanizing Chicago sportscaster max, who falls in love with his best friend jay (lee) 's book editor Samantha (Avital) a mere twenty-four hours after meeting her. They are soon engaged, and max, because of his own raging libido, grows suspicious of Samantha's fidelity. He convinces jay to flirt with Samantha during the development of his book, to "test her". The trouble is, jay might be secretly in love with her. To stretch this flat, sitcom premise to feature length, the plot

is framed by a climactic wedding, at which bonnie hunt recounts the triangular tale–the events leading up to the nuptials–to an annoying fat man and his silly girlfriend. Hunt has the best comic timing of anyone in the film; Schwimmer can spin bad dialogue into mildly humorous dialogue; and lee, poor lee, is miscast. So hysterically funny in Chasing Amy, here he is forced to repress his comic instincts : to swear, to yell, to talk about oral sex. The script's idea of a character trait is to stress that jay is a "sensitive man", and then show him drinking pepto bismol when he's stewing over his girl trouble. As for Avital, an Israeli actress, she is warm and sweet, but we don't know anything about her character other than that it takes her an incredibly long time to realize the most obvious things. She also too closely resembles the stunningly beautiful Kari Wuhrer, who plays Schwimmer's assistant and personal temptress, turning that particular subplot into an unintentional riff on vertigo. There are a handful, a smattering, of good scenes in kissing a fool. I enjoyed a moment in a comedy club, during which jay gets up and asks "has anyone here ever hated their girlfriend so much you wanted to kill her?" Over and over until he's booted off stage. There are also a few obviously improvised lines that are fresher than anything that's on the page. Kissing a fool is never as clever as the thursday night joke-machine friends that spawned Schwimmer's movie career, so save yourself eight dollars and watch three episodes of that series back to back.

Figure 33: Document that was marked as incomplete for BNRats, LOO, and FE MaRs.

Document	Tag	# sentences
neg_939.txt	1.0 (negative)	19

	Method	# MaRs
	LOO	6
	FE	7
	BNRats	2
	RS	3

Originally titled 'don't lose your head', this parody of the scarlet pimpernel story was the first carry on to be produced by rank film productions. Two English fops, the 'powdered, be-wigged, be-ribboned' sir Rodney Ffing (Sidney James) and his counterpart lord Darcy pew (Jim dale) decide to travel to revolutionary France in an attempt to rescue their fellow french

royalists and aristocrats from losing their heads by the guillotine. Due to a series of machinations and disguises, they are largely successful. Ffing becomes known as 'the black fingernail' because he leaves a calling card behind which shows two fingers sticking up, one with a black fingernail. After the fingernail rescues a prominent royalist the duc de pommfrit (Charles Hawtrey), citizen Robespierre (Peter Gilmore) orders the head of the secret police citizen 'the big cheese' Camembert (Kenneth Williams) and citizen bidet (peter butterworth) to follow the fingernail to England and do away with him. (In fact, Darcy and Ffing are their coachmen!) Once at Calais, the fingernail meets Jacqueline (Dany Robin) and they fall in love instantly. He tells her his identity and gives her his locket. When Camembert realises that the fingernail is nearby, he searches the inn at Calais and captures Jacqueline, thinking that she is wearing a diguise and is really the fingernail! Jacqueline is imprisoned in the bastille and Camembert, his love Desiree Dubarry (Joan Sims), and bidet all travel to London in pursuit of the fingernail. They pretend to be of noble stock, calling themselves the duc and duchesse de la plume de ma tante (with bidet their assistant) and are invited by darcy to a ball held by Ffing. Desiree finds out that Ffing is the fingernail by wearing the locket around her neck, but she ends up falling in love with him. Ffing attempts to stall Camembert so that he can return to the bastille to rescue Jacqueline, Camembert has her moved to the 'chateau neuve', and a climactic sword-fight decides who will lose their heads at the end of the film! A more complex story than most carry ons, this film enjoys good production values (sets, costumes) and an on-form cast. Sid James is excellent as the English fop and black fingernail, Kenneth Williams excels as the evil Camembert, and peter Butterworth expertly plays the substantial part of Camembert's thick-witted crony. Other acting honours go to Joan Sims who is perfect as Desiree and Charles Hawtrey who is excellent as the French aristo pommfrit. Although it suffers from a disasterously over-long sword fight at the end of the film, it is largely successful due to the performances of the main stars, its slick and professional production, and its better-than-usual script. Definitely one of the best of the series and a joy to watch.

Figure 34: Document that was correctly classified by all users using MaRs from all methods.