Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Computer-Aided Analysis of Video Comments for Requirements Analysis

Bachelorarbeit

im Studiengang Informatik

von

Eklekta Kristo

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Ann-Christin Klünder
Betreuer: M. Sc. Oliver Karras

Hannover, 12.02.2021

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 12.02.2021

_____

Eklekta Kristo

Zusammenfassung

Computergestützte Analyse von Video Kommentaren für die Anforderungsanalyse

In dieser Arbeit werden Anforderungen für die Anforderungsanalyse aus den Youtube Kommentaren von *vision videos* extrahiert. Der Prozess der Erstellung und Vorbereitung eines Datensatzes wird beschrieben und die Güte von verschiedenen automatisierten Ansätzen wird evaluiert. Die YouTube API wird benutzt um Kommentare zu extrahieren, diese werden dann in *Spam* bzw. *Ham* kategorisiert. Die manuelle Klassifikation ist nötig um die Ergebnisse der automatischen zu verifizieren. Um Einsichten in die relevanten Kommentar zu erhalten und spezifischere Kategorien zu finden werden *word clouds* benutzt. Die gefundenen Kategorien sind *Feature Request*, *Flaw Report*, *Safety Related*, *Efficiency Related* und manchmal *Questions*. Für die automatische Klassifikation in die Kategorien *Spam* / *Ham* werden die Algorithmen *Random Forest*, *Support Vector Machine*, *Linear Regression Classifier*, *Naive Bayes* und ein *Voting Classifier* welcher die ersten drei kombiniert benutzt. Für die Klassifizierung in spezifische Kategorien wird ebenfalls der *Voting Classifier* verwendet. Für die Analyse der Stimmung werden *TextBlob* und *SentiStrength*, und um die relevanten Kommentare zusammenzufassen wird *SumBasic* benutzt.

Abstract

Computer-Aided Analysis of Video Comments for Requirements Analysis

In this thesis requirements suitable for requirements engineering are extracted from comments below vision videos on the platform YouTube. The process of creating and preparing a dataset is described and the performance of different automated approaches is evaluated. The YouTube API is used to extract the comments, that are then classified into the categories *Spam* / *Ham* according to their content and sentiment. The manual classification is necessary to evaluate the results of the automated one. *Word clouds* are used to get an insight into the content of the relevant comments and decide on more specific categories to classify them according to their content. More specifically the categories *Feature Request*, *Flaw Report*, *Safety Related*, *Efficiency Related* and sometimes *Questions* are found. For the automated classification into the categories *Spam* / *Ham* the algorithms *Random Forest*, *Support Vector Machine*, *Linear Regression Classifier*, *Naive Bayes*, and a *Voting Classifier* that combines the first three are used. To classify comments according to their sentiment *TextBlob* and *SentiStrength* are used. For the classification into specific categories, the *Voting Classifier* is used again. The *SumBasic* algorithm is used to summarize the relevant comments.

Keywords: Requirements Engineering, Vision Videos, Youtube, Sentiment Analysis, Automated Classification

Contents

1 Introduction **1**
- 1.1 Problem Formulation . . . 1
- 1.2 Solution Approach . . . 2
- 1.3 Structure of Thesis . . . 3

2 Principles **5**
- 2.1 Requirement Engineering . . . 5
  - 2.1.1 Vision Videos . . . 5
  - 2.1.2 Crowd-based requirements engineering (CrowdRE) . . . 6
- 2.2 Natural Language Processing Algorithms . . . 6
  - 2.2.1 Binary Classification . . . 6
  - 2.2.2 Sentiment Analysis . . . 7
  - 2.2.3 Measurements for Algorithmic Performance . . . 8

3 Constructing a Dataset **11**
- 3.1 The Foundation of a Dataset . . . 11
  - 3.1.1 Choosing a Vision Video . . . 11
  - 3.1.2 Retrieving the Comments of a Vision Video . . . 13
  - 3.1.3 Dataset Cleaning . . . 14
- 3.2 Complementing the Dataset using Manual Classification . . . 16
  - 3.2.1 Detecting Relevant Comments for the Requirement Analysis . . . 17
  - 3.2.2 Applying Sentiment Analysis on the Comments . . . 20
  - 3.2.3 Further Knowledge Extraction . . . 26

4 (Semi-) automatic Categorization of Video Comments **31**
- 4.1 Spam / Ham Categorization . . . 32
  - 4.1.1 (Semi-) automatic Approach . . . 32
  - 4.1.2 Naive Approach . . . 34
- 4.2 Summarizing Comments classified as Ham . . . 35
- 4.3 Sentiment Analysis . . . 37
- 4.4 Content Related Classification . . . 37

5 Evaluation of Further Datasets **41**
5.1 Constructing datasets 42
5.1.1 The Land Rover Transparent Bonnet dataset 42
5.1.2 The Hyperloop Dataset 44

6 Related Works **51**

7 Summary and Outlook **53**
7.1 Summary 53
7.2 Outlook 54

A Complementary Details to Datasets **57**
A.1 Tunnels Dataset 57
A.2 Land Rover Dataset 57
A.2.1 Land Rover Dataset Comments' Source 57

Chapter 1

Introduction

1.1 Problem Formulation

Both in traditional as well as in the agile development of new systems and technologies, involving commercial as well as private stakeholders to elicit their requirements to the system plays a vital role. However, not only does the developer team and stakeholders have diverse competencies, but also the stakeholders themselves are a rather heterogeneous group. They come from different backgrounds, age groups, education levels, have different experiences, previous knowledge, and intentions [26]. Although there is usually a tremendous amount of stakeholders for a product, only a few participate in the development process since the requirement engineers can not ask each person individually but can only involve a representative group.

A reason that only a few persons participate in the development process is that the terminology and complexity used to present the new technology are unappealing or even daunting to several persons, so establishing a shared understanding is a crucial task. *CrowdRE* is an approach used to increase the number of persons participating in requirement engineering. With the help of *crowdRE* we can reach more participants through other channels like social media. However, this approach brings some challenges with it. For example, it is crucial to convey the information so that despite the persons having different backgrounds, and the terminology's complexity, they can understand the information. One technique that can be used for this purpose is to produce vision videos and show them to the possible stakeholders. In Karras et al. [15] vision videos are defined as: "Videos that represent a vision or parts of it for the purpose of achieving shared understanding among all parties involved by disclosing, discussing, and aligning their mental models of the future".

Another obstacle is to reach and identify as many stakeholders as possible. Usually, only a few persons actively participate in discussions and give feedback because of their character or because of their educational level relating to the discussed

subject. Furthermore, inviting stakeholders to interviews, user studies, or other similar events is expensive, difficult to plan, and time-consuming [26]. Therefore, taking into consideration the rise of popularity and widespread usage of web 2.0 platforms as well as the viral effects of user sharing [31], it would be interesting to observe the effect of sharing vision videos on platforms, like YouTube, where users can write their feedback in the form of comments. This way, a broad range of stakeholders can be reached, and the costs of organizing the meetings in person are omitted and can be redistributed into further development. Furthermore, it is possible to reach persons who were previously overlooked while identifying the stakeholders.

Some companies like "The Boring Company" are already taking advantage of web 2.0 platforms' popularity to post vision videos online. For example, their vision video named "Tunnels", was watched over 6.9 million times during the last 3 years and has gathered 6776 comments until October 18th 2020. Now that we can quickly gather feedback from stakeholders using online platforms, it is necessary to find efficient ways to analyze it. The works Guzman et al. [12], and Guzman and Maalej [10] use approaches to classify and extract relevant information from Tweets or app reviews.

This thesis aims to use the same methods or similar ones to examine if we can extract relevant information from comments related to vision videos using automated or (semi-) automated approaches. In case there is relevant feedback in such comments, we could try to classify them further and extract more precise requirements. Such methods clearly should be able to cope with unstructured data like natural language. They should have the ability to work with elements like links to webpages or emojis and orthographic mistakes because these elements are often present in comments on social media. Additionally, it would be beneficial if our approach could also be applied to other vision videos without many changes and be automated entirely because of tremendous amounts of comments.

1.2 Solution Approach

My solution approach will be to build datasets containing YouTube comments related to vision videos. Based on related work, I will choose some approaches to analyze these data firstly manually. The manual classification is an essential step undertaken in many similar studies, for example, to analyze coding videos on YouTube using user comments [20]. Then I will evaluate the insights of the manual approach. In case, there is indeed relevant feedback in comments under vision videos that can be used for the requirement engineering, I will try to find ways to extract this feedback manually. Afterwards, I will try to find automated or (semi-) automated approaches that can also evaluate the comments and compare their results to my manual approach. Finally, if this approach works well on a dataset, I will evaluate if it works similarly on other datasets.

1.3 Structure of Thesis

This thesis consists of seven chapters which are structured in the following manner.

Chapter 1 serves as an introduction to the subject discussed in this thesis by explaining the motivation behind the topic. This chapter explains the problem this thesis intends to solve by outlining the current obstacles and defining the goals. In short, it introduces the idea of analyzing the numerous comments of vision videos posted on online platforms to determine if the information extracted from them can be relevant for requirement engineering.

In chapter 2, the algorithms and measurement methods for algorithmic performance and the essential terms used in this thesis are explained briefly.

Chapter 3 gives a detailed description of the process of creating a dataset out of comments related to vision videos and gives insights into the structure and contents of the comments related to a vision video. It starts by choosing a vision video and then extracting and preprocessing the comments to classify them manually.

In chapter 4, I use different algorithms to evaluate to what degree I can extract relevant information for the requirement engineering out of video comments by using automated approaches.

Chapter 5 is a bit similar to chapter 3, because, in this chapter, I have created two additional datasets in a similar way as before. The reason I did this was to evaluate the approaches introduced in the previous chapters using additional data.

In chapter 6 I compare my thesis to related works of other authors to determine the similarities and differences.

Chapter 7 briefly summarizes the discussed topics and provides an outlook for future work that could be motivated by this thesis.

Chapter 2

Principles

In this chapter, I will briefly explain the fundamental concepts of this thesis. We will start by explaining some basic requirement engineering concepts, because, in this thesis, we will analyze comments related to vision videos which are part of the requirement engineering. Then I will go on with the algorithms used in chapter 4 and the measurements needed to evaluate their performance.

2.1 Requirement Engineering

Requirements Engineering (RE) is a process of elicitating the services that a customer requires from a system and the constraints for its operation and development. The goal of RE is to create documents concerning the system requirements for knowledge sharing. In contrast, Agile Development (AD) has the same goal but focuses more on face-to-face communication between the agile teams and customers [17]. In other words, requirement engineering analyses the users' requirements towards a product and presents them so that the developer can understand them to build a product that fulfills the users' requirements. On the other hand, requirement engineering also consists of adapting ideas and questions of the developers in a way that the users can understand and be able to give their feedback. According to the agile manifesto[1], requirement engineering is about people, communication and functioning software.

2.1.1 Vision Videos

As stated in Schneider and Bertolli [25], *vision videos* show, among other things, products and how they are used, including the reactions of the persons using them and many more aspects that are difficult to express or comprehend by text.

[1] http://agilemanifesto.org/iso/en/manifesto.html

Vision videos are used during the early phases of a project complementary to textual representations [26]. They serve as a technique for indirect and efficient communication between the developer and client [7].

2.1.2 Crowd-based requirements engineering (CrowdRE)

Crowd-based requirements engineering (CrowdRE) means performing activities such as elicitation of user requirements by involving many stakeholders [9]. So the word crowd, in this case, is referring to a large group of stakeholders. It is stated in Groen et al. [9] that CrowdRE turns RE into a more participatory approach, leading to more precise requirements and improving software quality. This approach supports better requirement management, for example, by prioritizing and segmenting requirements, while gathering a continuous flow of user feedback.

2.2 Natural Language Processing Algorithms

2.2.1 Binary Classification

Classification is a procedure which assigns each entity to one of the predefined classes. In this case, an entity is a comment, and classes can be spam and not spam (ham) [28]. In this example, there are two possible classes for the classification, spam and ham. Therefore the process of entity classification into two possible classes is called *binary classification*.

Random Forest (RF)

This method is a variant of bagging methods. Like them, it constructs a decision tree for each of the bootstrap samples drawn from the data. The main difference to bagging is that random forest randomly selects a subset of predictors to determine the optimal splitting rule for each tree node [8].

Naive Bayes Classifier

Naive Bayes is a method that can be used for binary classification. The word Bayes stands for the Bayesian theorem on which this classifier is based. The term naive originates from the fact that although this algorithm uses Bayesian techniques, it does not consider dependencies that may exist [4]. So the events or conditions are assumed to be independent, for example, it is assumed that a word does not influence the other words in deciding if a message is spam or not [23].

Support Vector Machine (SVM)

A *support vector machine* (SVM) is a computer algorithm that learns by example to assign labels to objects [27]. After giving an SVM model sets of labelled training data for each category, the model can categorize new text. Support vector machines (SVM) usually perform better than Naive Bayes and are quite useful at text classification [29]. More detailed information on SVM can be found in Schölkopf and Smola [27].

Linear Regression Classifier

The goal of *linear regression classifiers* is to identify an object's category by making a classification decision based on a linear combination of the characteristics [32]. In other words, linear regression classifiers try to calculate a linear equation that best fits the data points. The python machine learning tool *scikit-learn*[2] has an implementation of this algorithm and additional information on its usage.

Voting Classifier

In some challenging classification tasks, single algorithms do not perform well. In that case, *voting classifiers* can come in handy. Before building a voting classifier, it is necessary to select an optimal set of classifiers and then combine them by a specific fusion method into a voting classifier [24]. The python machine learning tool *scikit-learn*[3] has an implementation of this algorithm.

2.2.2 Sentiment Analysis

Sentiment analysis is a field that combines approaches like natural language processing and text mining to analyze the opinions, viewpoints, conclusions, evaluations, attitudes, judgements, and emotions of a person towards a particular subject, product, service, organization, individual, event or activity and their attributes [23, 29]. Using Sentiment analysis, we can determine whether people's reactions are positive, negative or neutral [2]. In the case of vision videos, we can analyze a person's reaction and response after watching a particular video [16] by analyzing the sentiment of the comments regarding this video written by the user [11].

[2] https://scikit-learn.org/stable/modules/linear_model.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

TextBlob

TextBlob is a Python library for processing textual data and provides an API for diving into typical natural language processing (NLP) tasks such as sentiment analysis, classification, translation, and more. This and further information, including the installation, usage and documentation of TextBlob can be found on the following webpage: `https://textblob.readthedocs.io/en/dev/`.

SentiStrength

SentiStrength is a lexical sentiment extraction tool, specialized in dealing with short, low-quality texts. Previous research has shown good accuracy in analyzing short messages written in informal language on Twitter, in movie reviews, and GitHub commit messages to mention some of the scenarios [11, 12]. It estimates the strength of positive and negative sentiment in short texts, even for informal language. This makes SentiStrength a good candidate for YouTube video comments analysis. More information on this program, how to download and use it, can be found on the following webpage: `http://sentistrength.wlv.ac.uk/`

2.2.3 Measurements for Algorithmic Performance

Accuracy, precision, recall and F-Score are possible measurements to evaluate an algorithm's performance.

Accuracy

Accuracy describes the model's ability to predict unseen data correctly and can be considered as an excellent method to measure the performance of symmetric datasets where values of false positives and false negatives are almost equal [28]. If the dataset is not symmetric, accuracy can be misleading due to the unknown probability distribution, so it is necessary to consider additional measures, like displaying the confusion matrix, which can sometimes be beneficial [28].

Precision

Precision measures the exactness of a classifier. Precision is defined as the number of True Positives divided by True Positives and False Positives. We can derive from the formula that a low precision value means a high number of False Positives [28].

Recall

Recall (also called sensitivity or True Positive Rate) is the number of True Positives divided by the number of True Positives and False Negatives. Recall measures the completeness of a classifier. From the formula, we can conclude that a low recall value means many False Negatives [28].

F1-Score

The *F1-score* combines both the precision and recall using the harmonic mean, where an F1 score reaches its best value at 1 and worst score at 0 [14].

Chapter 3

Constructing a Dataset

To investigate the usefulness of comments related to a vision video for requirement analysis, a dataset containing such comments is needed.

The first step in creating the dataset will be to select a medium that contains vision videos. Afterwards, we will choose a random video, preferably containing numerous comments, fetch the comments, and manually classify them into different categories. Then we store the comments and the categories they belong to according to the manual classification in a table which we call dataset. A detailed description and explanation of the above-mentioned procedures and the steps carried out in them will be explained in the following sections of this chapter.

3.1 The Foundation of a Dataset

3.1.1 Choosing a Vision Video

The subject of this section will be to firstly select a platform to search for vision videos than randomly choose a certain video.

A popular platform that contains vision videos, among various other content, is YouTube. YouTube was founded on 14. February 2005, and has been a vital social media platform for video sharing ever since [1]. According to YouTube statistics[1], there are more than 2 billion users, which amounts to almost one-third of the internet.

Unregistered users can view videos while registered users can also comment, like, dislike any video or upload their own content [22].

Although YouTube consists principally of content produced by the users and depends on sharing and spreading this content, the large number of users have made YouTube attractive for business entities and public figures to create their own page and upload

[1] https://www.youtube.com/intl/en-GB/about/press, accessed on 2020-12-18

their content [4].

A further advantage of using YouTube to build the dataset is that since it is a popular site, the participants taking part in the manual comment labelling would most likely be familiar with this site as end-users [1].

After searching YouTube for vision videos, we decided to pick the video "Tunnels" uploaded on April 28th 2017 by "The Boring Company"[2].

In approximately the first three years since the upload, this video has been viewed over 6.9 million times, has received 59,621 likes, 4,287 dislikes and 6,506 comments. The numerous comments were the main reason for selecting this video as a data source for a dataset.

Some of these comments contain replies that sometimes also contain further replies as well as likes and dislikes. For a graphical illustration of the structure of comments and replies on YouTube, we can have a look at figure 3.1.

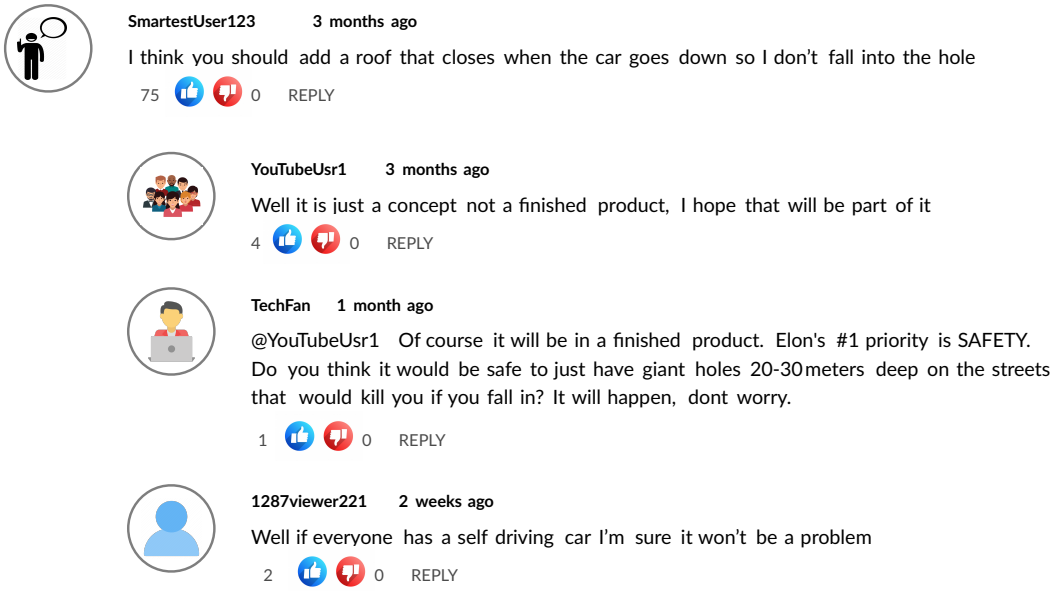

Figure 3.1: Example of YouTube comments and replies. The comment and replies are randomly selected from the video "Tunnels" on YouTube. All the other data like username, profile image, number of likes and dislikes, the publishing time and number of replies are fictional.

[2]https://www.youtube.com/watch?v=u5V_VzRrSBI, accessed on 2020-10-13

In this example, the comment of *SmartestUser123* has received three replies. If we have a closer look at the replies, we can see that one of the replies, namely the one written by *TechFan* is not directly a reply to the comment of *SmartestUser123* but rather to another reply written by *YouTubeUsr1*. This is denoted by the token *@YouTubeUsr1* written at the beginning of this reply to indicate the user this reply refers to. So we can say that the reply to the comment of *SmartestUser123* written by *YouTubeUsr1* does also have a reply, namely the one written by *TechFan*.

Now that we already have selected a video, the next step would be to store the comments in a dataset. However, typing the comments one at a time in the dataset table cells would take a lot of time considering their quantity. Therefore we will discuss an approach to achieve this automatically in the following section.

3.1.2 Retrieving the Comments of a Vision Video

A tool that can aid the automatical extraction of YouTube comments is the YouTube Data API. YouTube Data API is an *Application Programming Interface* that allows us to receive video statistic and channel-related data from YouTube [22]. In this thesis, we will use the version 3.0 of this API. This tool was also used in similar works like Abdullah et al. [1], Asghar et al. [3], and Obadimu et al. [19].

With this API's aid, we can extract the comments and replies and their corresponding metadata, like the author's username, comment identifier (ID), the number of likes and responses. The comment ID is necessary to match a comment to its replies since they all share the same ID. Then we store this data in two different tables, one for the comments and one for the replies. You can find a snippet of each of the tables to demonstrate their structure and an example of a comment's connection with its replies through the standard ID in the appendices A.1 and A.2.

In figure 3.2 we can see concrete values about the number of comments and replies extracted from the video "Tunnels" of "The Boring Company" on YouTube on October 13rd 2020. As we can see, 66.7% of all fetched comments are concerning the video while 33.3% are replies to comments.

By looking at the dataset's content, it appears that some comments are duplicates, which means they consist of the same characters. Moreover, some other comments consist only of one character like, for example, a question mark. In the first case, we do not need to keep duplicates since they do not present any previously unknown information; keeping only the first occurrence would be sufficient. In the second case, we can discard the entry because a single character can not give any crucial feedback for the requirement analysis process.

This observation makes it necessary to preprocess the data before going on with labelling comments into different categories. In the next section, we will set up a list with conditions that the comments should fulfill to be removed from the dataset and

see how preprocessing the data according to these requirements affects the dataset.

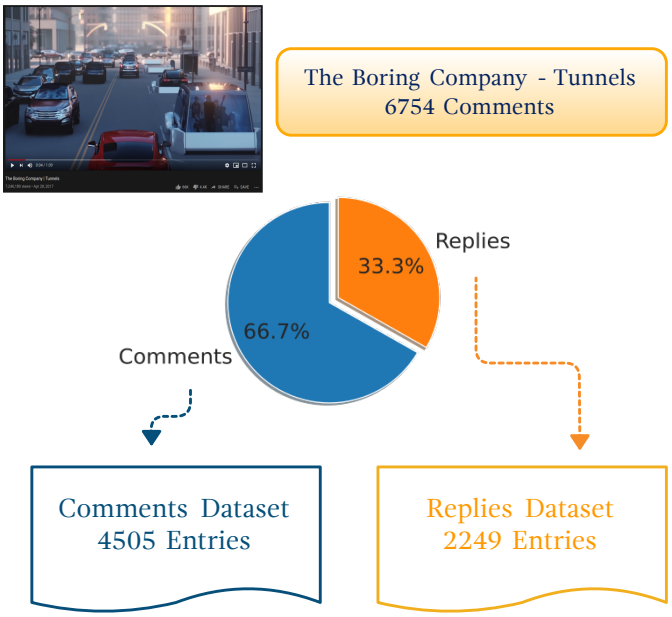

Figure 3.2: Number of comments and replies in the dataset "Tunnels".

3.1.3 Dataset Cleaning

This section will dive into more details about the characteristics of the comments that we will remove during the cleaning process. Besides duplicates and very short comments, some comments contain words or sentences in languages other than English. Therefore, before starting with the actual dataset cleaning, we manually translate each of these comments to English using Google Translate.

Next, we remove all comments from the dataset that satisfy at least one of the conditions listed below.

- The comment could not be translated into English by Google Translate. These comments often contain wrong spelling or are written in mixed languages. The comments that get only partly translated, which means one or more words stay in the original language, should also be removed.

- The comment is a duplicate, which means the same content already occurred in the dataset.

- The comment is an empty string or contains only one character, like for example *?*, *9*, *k*. Emojis also fall into this category, while emoticons do not. The reason for this is because Emojis consist only of one symbol like ☺ or ☹, while emoticons consist of more than one character like in *:-)* or *:-(*.

- The comment contains more than one character, but it does not contain any words. In this case, the comments consist of Emojis, emoticons, special characters or punctuation marks.

We could continue to refine the list above, but we intentionally keep it plain so that it is easily possible to verify all the listed conditions automatically.

The removal of comments that meet the previously listed criteria leads to a reduction of the dataset's size. In figure 3.3 we can see concrete values regarding this reduction.

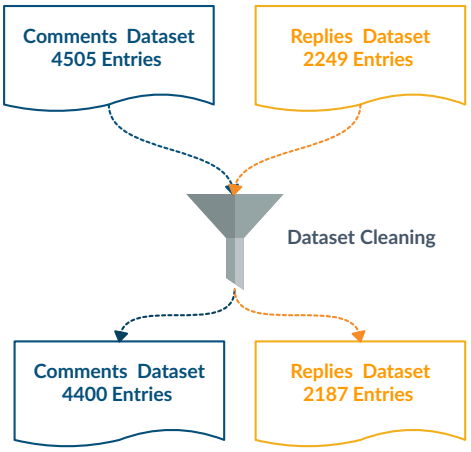

Figure 3.3: The reduction of comments and replies in the dataset "Tunnels" after the cleaning process.

Now that we completed the dataset's construction, we can go on with the manual classification. In the next section, we will introduce a method for performing the manual classification and decide which categories to use for labelling the comments and replies.

3.2 Complementing the Dataset using Manual Classification

As we mentioned in section 3.1.2, we stored the comments and replies into two different tables. If we have a closer look at the replies dataset's content, we can see that they are rarely answers or feedback regarding the video, but instead regarding other comments. For example, they disagree, support or complement other comments, further specific examples of replies and what they express can be found in the appendix A.3. Therefore, we will shift our focus on the comments, and leave the replies in the background.

So, we will classify the comments according to their relevance in section 3.2.1, according to the sentiment they express in section 3.2.2, and further other categories related to their content like a problem report or feature request in 3.2.3.

Furthermore, we will let three persons manually classify the comments, and also use them in the (semi-)automatic classification discussed in chapter 4. We involve more than one person in the classification process of comments to reduce bias, which, unfortunately, we can not eliminate. However, similar studies by Maalej and Nabil [18] and by Guzman et al. [12] also use the approach of manual classification by multiple persons while noting the possibility of still having some bias. We could further reduce bias by involving more raters. However, this will stay as something that we could analyse in later studies because of a lack of resources.

In contrast, we classify the replies only by one person (the author) because although we are not further interested in them, we still want to get an idea of the amount of relevant information they contain.

To better understand the manual classification process done by one and three raters, we can look at figure 3.4.

We can see one person that classifies a dataset containing *n* entries on the left. Hereafter the classification process is completed, but it is not very objective since the decisions are made only according to one person's opinions.

On the right, we can see the slightly more complicated manual classification process that involves three persons. This process starts with two persons classifying *n* comments independently. Afterwards, we find the differences in their classification. According to this example, there are *x* comments in which both persons agree, so for the remaining *n* - *x* comments the participants do not agree on the comment's category. To make a final decision, these persons meet with an additional person to discuss and decide the suitable category. By increasing the number of persons participating in this process, we can reduce bias since the final decision does not depend only on one person.

After introducing the manual classification of comments by one or more persons, we will apply these methods in the following sections to classify our dataset.

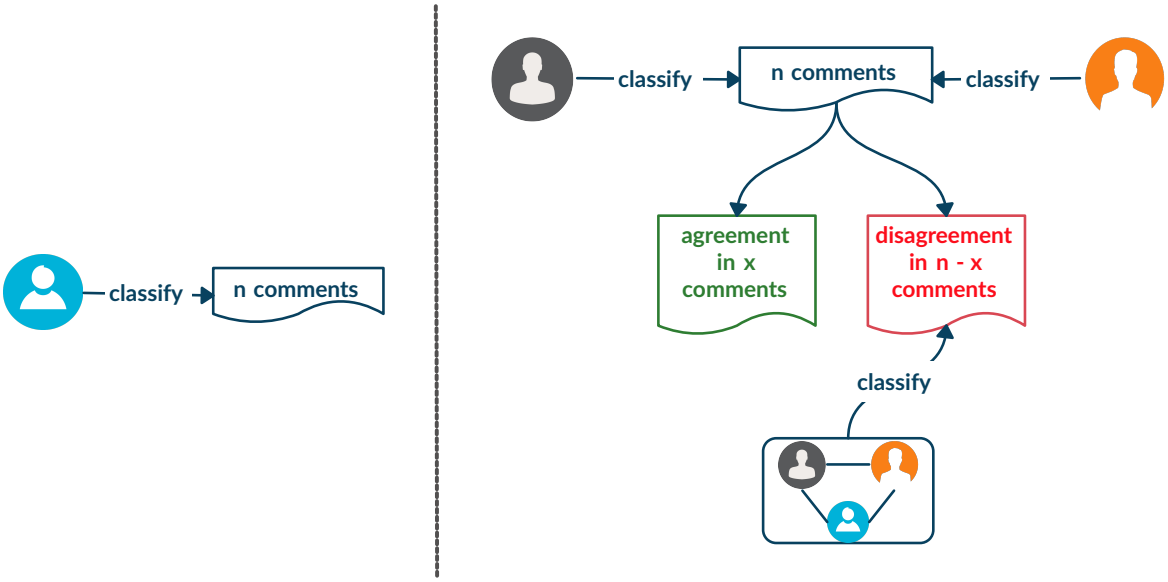

Figure 3.4: On the left, we can see a depiction of one person's manual classification process. On the right, we can see an illustration of the manual classification where three persons are involved.

3.2.1 Detecting Relevant Comments for the Requirement Analysis

Through the comments section in social media, like the one on YouTube, users can distribute spam comments with unrelated or abusive content, and URLs for advertisement and redirection to other sites [5].

Some people can ignore spam while others, especially new internet users, are unaware of it [5]. Since spam is prevalent in day-to-day life and can even be harmful, Das et al. [5] propose an approach that can detect comment spam. Abdullah et al. [1] also studied some standard filtering techniques to filter malicious content in YouTube comments. However, both approaches mentioned above depend on the video viewer's perception. In this thesis, the classification conditions will be slightly different since we classify the comments according to the video uploader's perspective.

To manually isolate the spam buried in the comments is challenging since there are usually numerous comments under each video [5]. For example, in our previously built datasets we have 4400 comments and 2187 replies as illustrated in figure 3.3. To simplify this process, we could use the built-in tool for spam control on YouTube, allowing us to choose which comments we want to see. We can choose to display all comments, only comments approved by YouTube or all comments except potentially inappropriate comments [1]. However, this tool is not sufficient for combating malicious

content, or spam content in comments [1], so taking further measures to detect spam is necessary.

It is important to note that the meaning of comment spam is goal-related. For example, the goal in the Das et al. [5], and Abdullah et al. [1] was to filter malicious or spam comments. However, our goal is to gather useful information for the requirement engineering process out of the comments. Therefore, spam comments in our case will be comments that do not contain any value for the requirement engineering process. In contrast, comments classified as ham contain the commentator's opinion about the video's content, which can be about a particular idea, or object [16] so it is relevant content.

Before continuing with the manual classification, we have to prepare a list of manual annotation guidelines based on our goal. This way, we can filter comments according to the relevance for requirement engineering, so that irrelevant comments with low-quality information or undesired content get classified as spam [28] and the others as ham.

The following guidelines apply to our manual classification regarding the relevance of the comments:

1. Comments that fulfil one or many of the following conditions are irrelevant, so they fall in the category spam.

    - Comments that consist only of emojis, emoticons, numbers, punctuation marks or just a single word, because these terms do not provide sufficient information to be regarded as relevant. *For example: 1948205871, :D, XD, :P, ., $, ??, awesome, boring*
    - Comments that contain URLs or do advertisement of any kind. *For example: "Please visit my channel and watch my newest video", "Check this out: https://www.youtube.com/watch?v= u5V_VzRrSBI"*
    - Comments that do not give any feedback[3] about the introduced product or comments that express the author's sentiment regarding the product (be it positive, negative or neutral) but do not justify it.
      For example: "This defeats the definition of cars", "This is awesome cannot wait to see this", "I do not see the benefits in this."

2. Comments that explain why the author has a negative or positive opinion on a feature or anything related to the product or comments that point out possible flaws or problems are relevant, so they fall in the category ham.
   For example: "Man this is an amazing idea, but there are some flaws like 1) what

[3] According to the Cambridge Dictionary feedback is defined as information or statements of opinion about something, such as a new product, that can tell if it is successful or liked. (https://dictionary.cambridge.org/dictionary/english/feedback, accessed on 2020-10-21)

if the power goes out 2) what if the tunnel collapses 3) earthquake 4) what if someone jumps out of there car or manages to get their car off the thing 5) can it flood.", "It would make traffic way worse because the cars behind would have to wait.", "What if someone fell into the hole when the Tesla went down?"

We use this list as a guideline for all the persons who participate in the manual classification process to help them have a similar perception about what spam in our case means.

After obtaining the raters' manually classified comments, we can observe the amount of spam and ham in the comments and replies.

As we can see in figure 3.5 both datasets consist primarily of spam comments, but the comments dataset contains less spam than the replies dataset, respectively 82.6% and 94.3%.

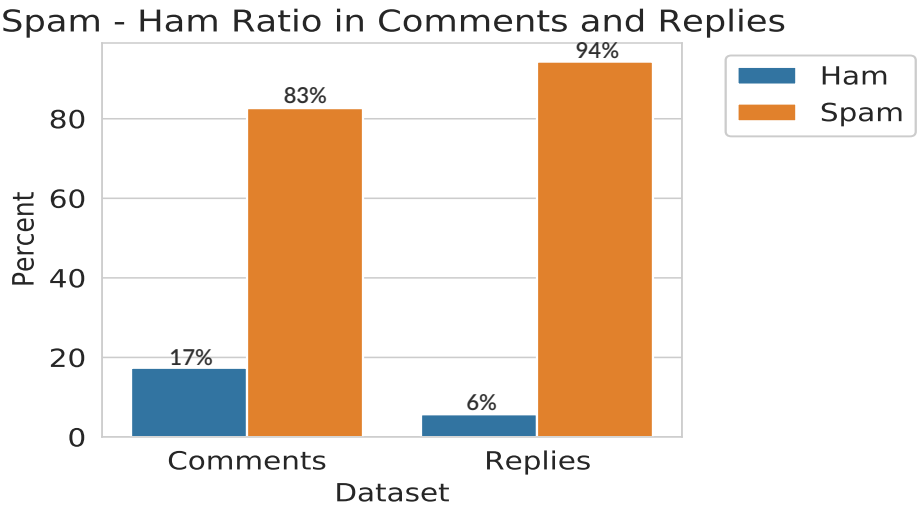

Figure 3.5: The ratio of spam and ham in the comments dataset and the replies dataset, based on the manual classification.

During the manual classification by three persons, the agreement between the two persons that classify the comments independently at the beginning of this process plays a vital role because it decides if an additional comparison involving a third person is necessary or not.

According to Viera and Garrett [30], studies that use methods where the agreement between two or more observers are relevant should include a statistic that considers that the observers will sometimes agree or disagree by chance. The most commonly used statistic for this purpose is the kappa statistic (or kappa coefficient) [30]. Based on Viera and Garrett [30] we can interpret the kappa statistic as a perfect agreement

when the value it yields is one or close to one, whereas a kappa of zero indicates random agreement.

Thus we measure the agreement of the two independent manual classifications, and we get the value 0.35 which according to figure Viera and Garrett [30] means fair agreement. Therefore making a final decision by having a discussion between these two persons and a third person (the author), was necessary.

Afterwards, we will go on to analyzing the comments according to their sentiment in the next section.

3.2.2 Applying Sentiment Analysis on the Comments

A related study that analyzes the sentiment of a YouTube video's comments is Khan et al. [16]. Their focus was on examining comments associated with a YouTube Video that compared two operating system types: iOS and Android. These comments contain comparative content where the users compare both products and share their preferences with or without justification [16].

This research's main difference to our sentiment analysis is that the video they studied compares two technologies that already exist. However, our video presents a vision of possible future technology. Hence, they calculate the sentiment for both operating systems separately, as well as the overall sentiment. In contrast, we calculate the sentiment of comments classified as ham, the sentiment of the comments classified as spam and the overall sentiment.

The process of manually classifying the comments into neutral, negative or positive sentiment by three persons, is similar to the comment classification into spam or ham introduced in 3.2.1. Therefore, we have written down a guideline for the manual sentiment analysis participants. A similar list as in Al-Tamimi et al. [2] was used as an orientation to create this guide.

The following rules were given the annotators as a guide for the manual sentiment analysis:

- We regard the comments that agree with the video content or support the video's creator as positive comments.
- We classify comments that oppose the video content or the video's creator as comments expressing negative sentiment.
- Any comment containing both positive and negative opinions in equal or almost equal ratios is considered neutral. Furthermore, we consider spam comments that are not related to the video as neutral even though they sometimes hold a particular sentiment. For example, the comment *"Click the link to check my awesome channel, you will not be disappointed..."*, has according to

SentiStrength a positive sentiment. Still, since it is not related to the video, it does not express any positive or negative sentiment regarding the video. Hence it is considered as neutral.

After both annotators independently classify the comments, we measure the kappa statistic. We get the value 0.14 which according to Viera and Garrett [30] means a slight agreement. Because of the low kappa value, to take a final decision, we discuss the comments' sentiment were the raters did not agree, with the two raters and a third person (the author).

So after manually classifying the comments, we can say that the neutral comments make up the most of the dataset, followed by negative and then positive comments as depicted in figure 3.6.

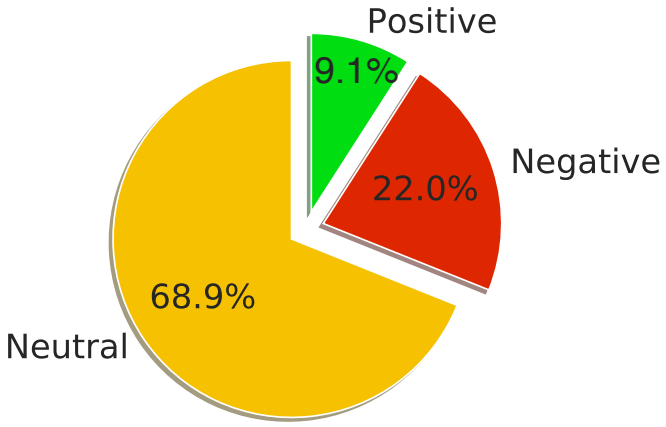

Figure 3.6: The distribution of the sentiment in the comments of the "Tunnels" dataset.

However, just knowing the sentiment of all comments is not sufficient. According to the guideline for manual sentiment analysis stated previously in this section, we mentioned that the spam comments that do not regard the vision video in any way, for example, comments promoting other YouTube channels, would be classified as neutral. However, they may express a positive or negative sentiment. Classifying these comments as neutral could be a reason that we have mostly neutral comments. Therefore, we have to see a more detailed distribution of the sentiment.

For example, we can look into the sentiment of spam and ham comments separately, as depicted in figure 3.7.

Based on this figure, we can say that both spam and ham comments have mostly neutral, followed by negative and at the last positive sentiment. However, there are much more positive comments in spam than in ham. In figure 3.8 we can additionally

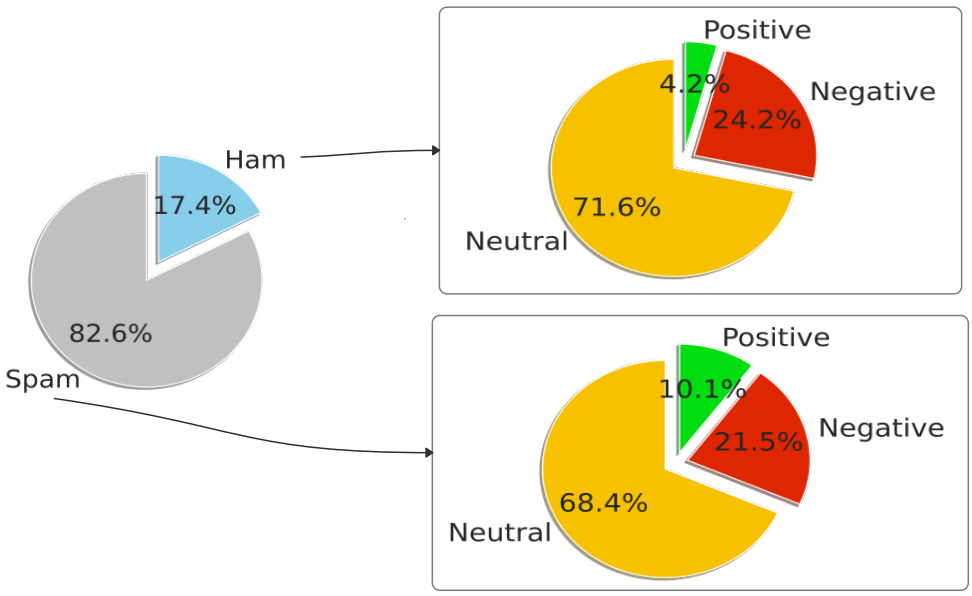

Figure 3.7: This figure illustrates the sentiment in the ham and spam comments of the dataset "Tunnels" separately.

observe that the comments with a positive sentiment are up to 92% spam. On the other hand, the negative and neutral ratios in ham and spam are similar.

So studying the content of the comments in more detail could help to find explanations for these observations.

Main Themes of the Comments

In figure 3.9 we have illustrated some examples of comments for each classification category and each theme. The themes serve as a title for groups of comments to summarize the subject discussed in these groups.

In the *Spam-Positive* group, we have comments that praise the idea or company. However, some comments praise the idea because the vision video's product is mistaken for a video game. Some other comments in this group show positive sentiment about the future because of how the technology introduced in the vision video would change the future.

Comments in the *Spam-Neutral* group do not concern the video's content in any way. For example, they contain advertisements for other YouTube Channels, ask other users

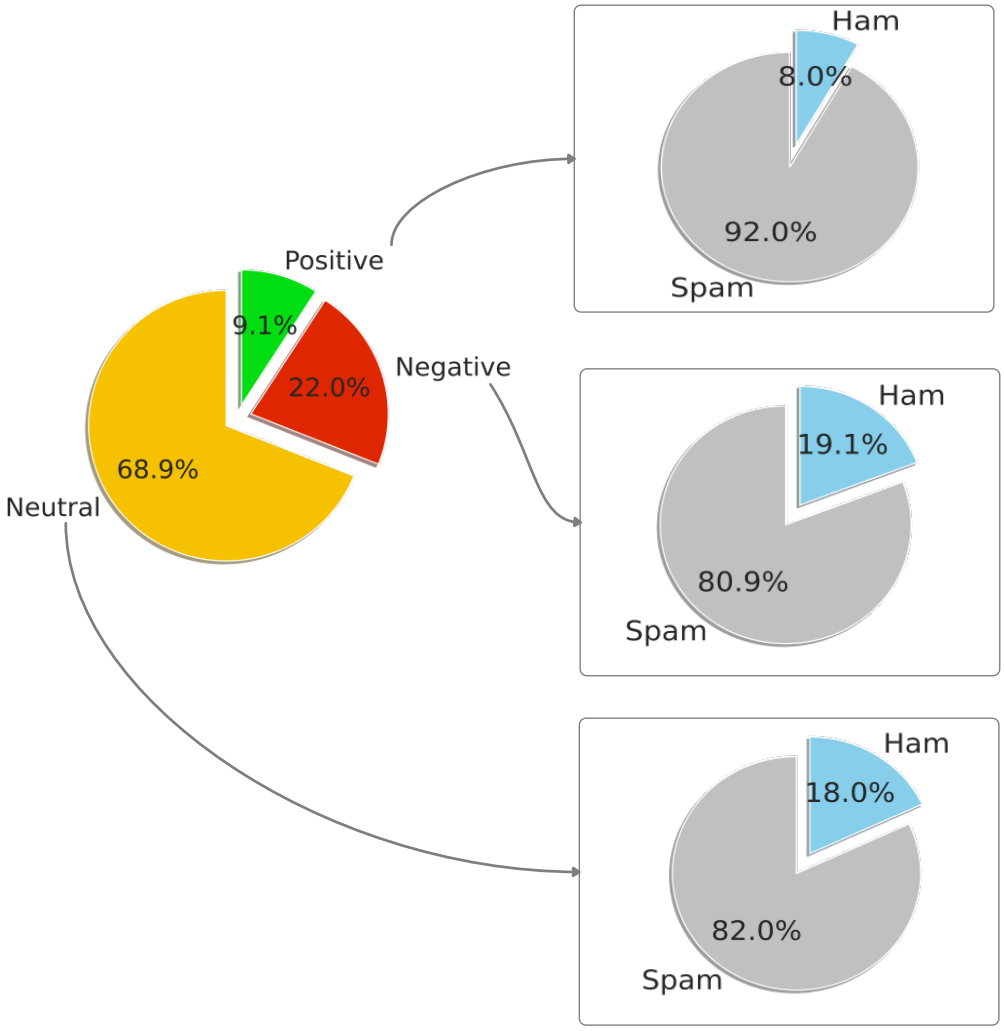

Figure 3.8: This figure illustrates the ham spam ratio in the comments classified as positive, negative and neutral of the dataset "Tunnels" separately.

about the title of the song used in the video (which is video related but not content-related), or complain about YouTube recommending this video to them.

Furthermore, some comments in this group are jokes about the idea or company; for example, the vision video gets compared to the technology used in fantasy films or video games.

Comments in the *Spam-Negative* group, contain complaints unrelated to the video's

content, similarly to the *Spam-Neutral* group, for example, a complaint that a person has to learn the content of this video for a school project but finds it boring. The difference to the complaints in the *Spam-Negative* group is that this group's comments contain words with a negative sentiment like *boring*.

Furthermore, some commentators dislike or make fun of the idea or the company without justification or are pessimistic about the idea or the future in general. For example, they state that there are already existing ideas better than this one or that the project will take forever to complete because of its complexity.

An interesting theme in this group is the one that expresses hate towards the problem this idea solves. Such comments fall into this category because they do not provide feedback, and they contain words with a negative sentiment like *hate*. However, hatting the problem, this idea will solve, means that the person likes the idea, so the sentiment to the idea is positive. For example, one user expressed hate towards traffic and thinks this technology will solve traffic problems. Such comments are, therefore, a challenge for sentiment analysis.

Despite the various themes addressed in the spam comments, they all have in common that they give no feedback which we could use in the requirement analysis process.

In the *Ham-Positive* group, the comments express support for the idea accompanied by rationales, improvement suggestions, requests for additional features, or questions about some details that are not obvious by watching the video. These questions are requests for more information or asking if a specific feature will be part of the introduced system or not. For example, some users ask if the Global Positioning System (GPS) would work in these tunnels.

An interesting case where sentiment analysis again meets its limits is when comments point out possible flaws through irony. For example, comments like "Great, now the traffic will not be just above ground but underground too.", show a positive sentiment because of the word *great*. However, the author of the comment seems not impressed by the idea presented in the vision video, so the actual sentiment, in this case, would be neutral or negative but not positive.

Comments in the *Ham-Neutral* group contain questions about possible scenarios and their consequences to the system, for example, in the case of meteorological phenomena like earthquakes and heavy snow.

Other users ask about additional information, for example, if there will be a cover for the hole left on the road while a car is lowering down to prevent others from falling, or how the emergency cars can come inside of the tunnel in case of an accident.

The comments in this category also point out possible problems. In such cases, they include praise or avoid being negative, so the comment's overall sentiment stays neutral, for example, "Awesome idea, but dangerous to drive during earthquakes".

A fair amount of comments in this group make suggestions, request changes, or

additional system features. For example, some users suggest letting only self-driving cars use this system, while others request a possibility to charge electric cars while riding the tunnel.

The last category in figure 3.9 is the *Ham-Negative* group. Some of the group's comments are short and lack the rationale when they express disliking towards the idea. Also, when pointing out a possible flaw or problem of the system, there is sometimes no justification about why this may be a problem. For example, some comments mention the high construction costs, earthquakes, and a weak earth crust resulting from the tunnels' construction. Furthermore, they do not clarify how earthquakes could affect the system and the people who will use them or give suggestions on how to prevent their impact. Nevertheless, such comments can encourage reflection about the themes they mention, to analyze their importance in the system's conception. For example, it can encourage the system's creators to reason about an earthquake's scenario and reflect about the tunnels being sturdy enough to withstand an earthquake.

However, most comments in this group use a more precise way of pointing out a problem. For example, the comment "If the hole on the road does not close when the platform lowers down, many people will fall and die.", points out a problem and a possible effect, that is why is it considered ham and has a negative sentiment because of the word choice.

By analysing the comments manually, we could find some of the most mentioned themes in the comments. However, this approach takes much time, especially when we face a large number of comments. Therefore, we will try to extract the most occurring themes in the comments using word clouds.

Exploring the Comments through Word Clouds

In Heimerl et al. [13], the authors describe word clouds as a visually appealing text analysis method to provide an overview of the words with the highest frequency in a piece of text.

In figure 3.10 we can see the word clouds for each of the categories introduced previously.

Similarly to the findings of manual analysis, it appears that the *Spam-Positive* group contains mostly praise, the *Spam-Neutral* group is pessimistic about the idea, and the *Spam-Negative* group expresses high objection. All three categories lack feedback.

There is also praise in the *Ham-Positive* group but accompanied by remarks about possible problems, such as earthquakes or the project's cost. The groups *Ham-Neutral* and *Ham-Negative*, contain mostly notes about the system's flaws or problems that it could face in the future. For example, the holes on the ground, traffic jam, earthquakes, high cost, getting stuck in the tunnel or using individual cars for the transport. The

content of these two groups is similar, according to the word clouds. However, in the *Ham-Negative* group, more words are expressing a negative sentiment, like "toxic driver" or "people falling".

If we have a closer look at the figure 3.10, it appears that the terms in the word cloud are pairs of words. I choose to use pairs of words instead of single words because single words deliver incomplete information. For example, the word *earthquakes* in a world cloud could mean that the author of the comment worries about earthquakes occurring while driving the tunnel, but it could also mean that the author thinks the tunnel is safe to drive during earthquakes.

Based on the most common themes discussed in the comments as seen in 3.9, the most frequent word pairs according to the word clouds in 3.10 and the categories used by Maalej and Nabil [18] and Guzman and Maalej [10] we will classify the relevant comments in the next section into content-related categories.

3.2.3 Further Knowledge Extraction

The categories we will use are *Feature Request*, *Flaw Report*, *Safety Related*, and *Efficiency Related*. In *Feature Request* users ask for missing functionality, which other products may already provide, and share ideas on improving the product by adding or changing features [18]. Comments that mention flaws of the current system design or describe possible problems resulting from these flaws fall in the category *Flaw Report*. The requested features and detected flaws contain many comments related to safety, such as earthquakes or people falling into the holes left in the road while another car is lowering down into the tunnel. Furthermore, some comments concern the construction of the tunnels costing a lot or taking too long. Therefore we decided to use the categories *Safety Related* and *Efficiency Related* in addition to *Feature Request* and *Flaw Report*. Then we let three persons manually classify the comments into these categories. I also mentioned to these raters that the comments do not necessarily fall into any of these categories, while others fulfill one, two, three, or four categories. As we can see in table 3.1, Cohen's Kappa Value is low for all four categories. Therefore making a final decision by having a discussion between the two raters and a third person (the author), was again necessary.

Table 3.1: Cohen's Kappa value for the manual classification done by three persons for all four categories.

Category	**Feature Request**	**Flaw Report**	**Safety Related**	**Efficiency Related**
Cohen's Kappa Value	0.15	-0.01	0.28	0.26
Interpretation	slight agreement	no agreement	fair agreement	fair agreement

In figure 3.11, we can see the occurrence of the themes *Feature Request*, *Flaw*

Report, *Safety Related* and *Efficiency Related* in the comments classified as ham. It appears that the themes *Flaw Report* and *Safety Related* occur the most. However, we can not say if the *Flaw Report* comments include complains about safety, efficiency, none or both them so it would be interesting to analyse these themes as pairs. If we look at figure 3.12, it becomes clear that the *Flaw Report* - *Safety Related* comments occur the most, which means that the frequently mentioned problems concern the safety of the persons using this system. Additionally, we have 100 comments that fall into none of these categories. After looking at their content in detail, these comments almost only consist of questions. For example, the viewers ask about additional information because it is not apparent just by watching the vision video if the system already includes certain features or not. Although these comments mention features, it is unclear if the viewer wants to have this feature as part of the system or simply wants to know if the company plans on implementing it. Therefore these comments do not fall into the category *Feature Request*.

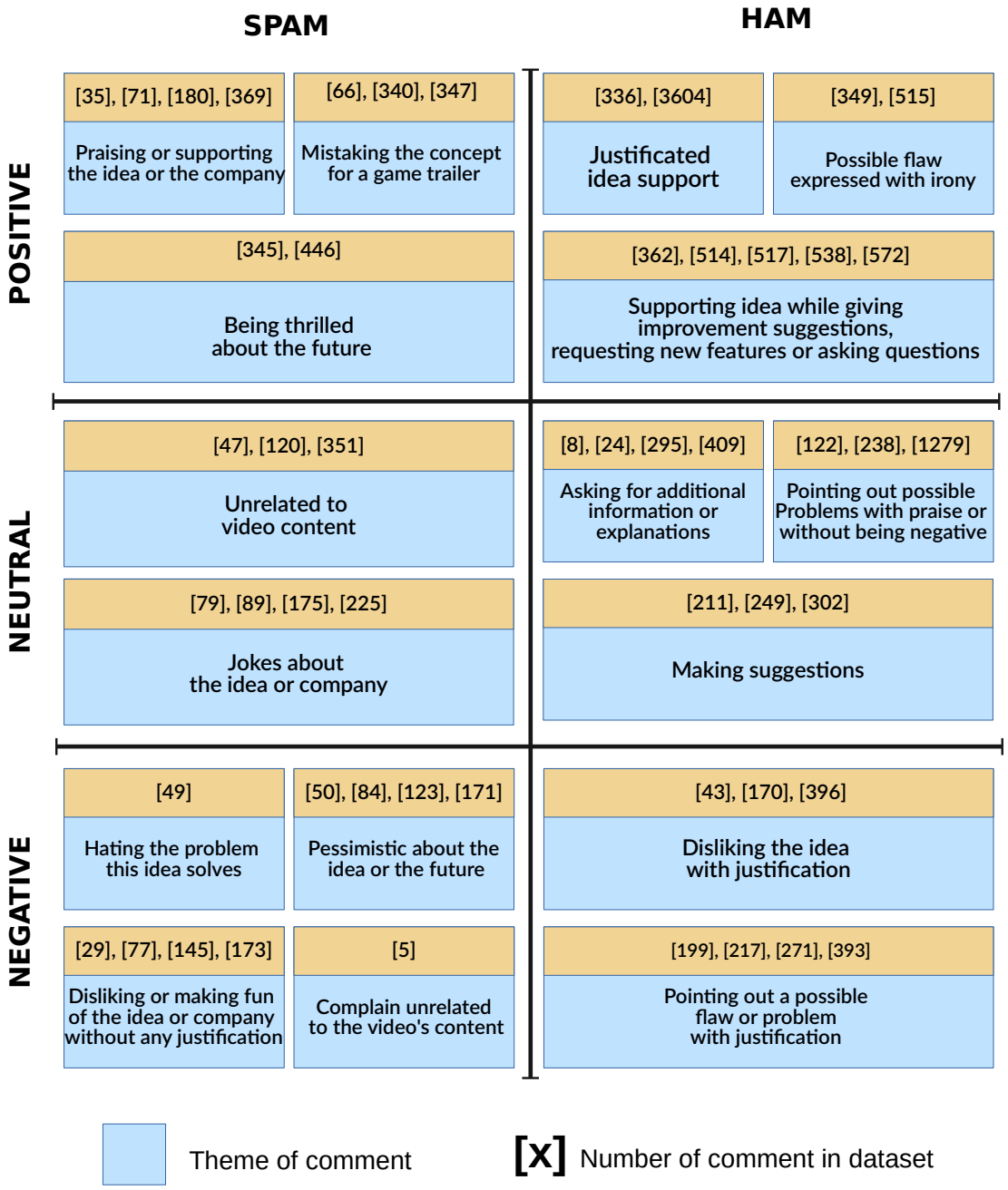

Figure 3.9: This figure is based on figure 4 of Vistisen and Poulsen [31], but we use different categories to classify the comments. The mapping of the numbers used in this diagram and the comments serving as examples for the themes can be found in the dataset file attached to this thesis as a compact disc.

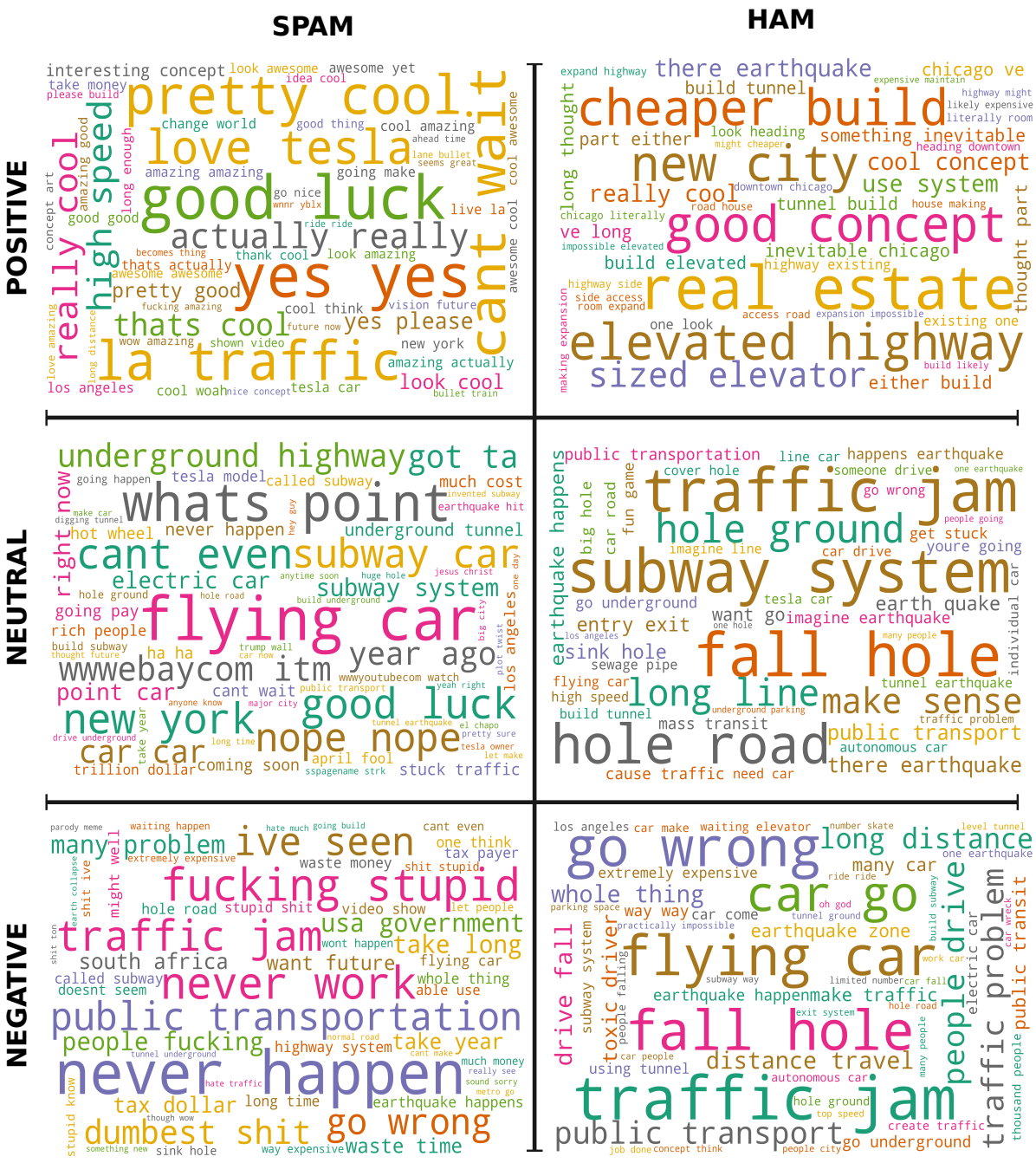

Figure 3.10: The most frequent words in the same categories as in figure 3.9 displayed through word clouds.

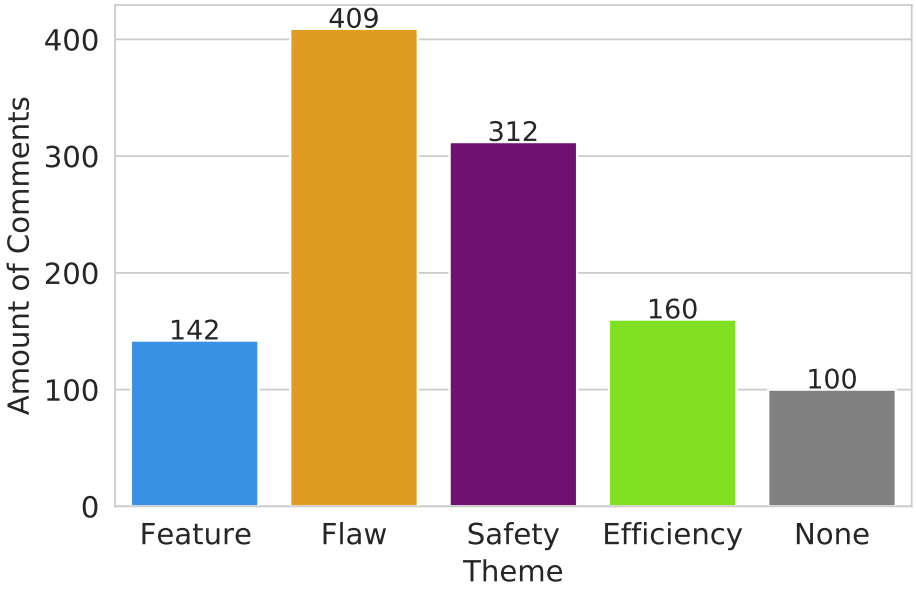

Figure 3.11: The number of comments classified as ham where the themes *Feature Request*, *Flaw Report*, *Safety Related*, *Efficiency Related*, or none of the above occur.

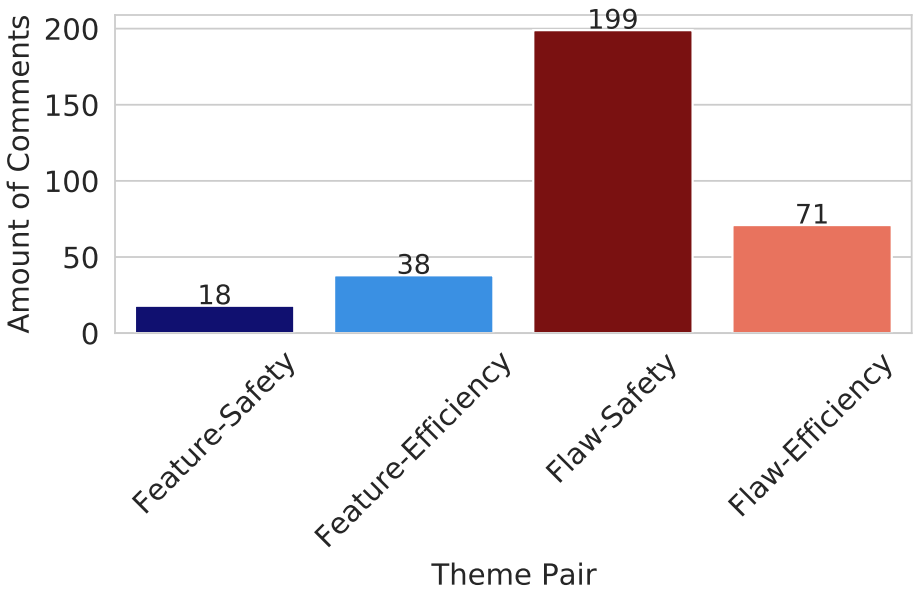

Figure 3.12: The number of comments classified as ham where the stated theme pairs occur.

Chapter 4

(Semi-) automatic Categorization of Video Comments

This chapter aims to classify the comments into the same categories as in the manual classification of chapter 3 but this time in an automated fashion where we benchmark different algorithms against each other with the goal to find the best ones. I will compare the algorithmic classification results to the manual classification of chapter 3 to analyze the performance of the algorithmic approach. This is also the reason for the word *semi* in the title. This approach is not entirely automated because I use manually annotated data to evaluate it. The algorithms I decided to apply in this chapter are used in other similar studies or have performed well in them. I did not choose only algorithms that yielded the best results on similar studies, because if an algorithm has a high accuracy for a dataset, it does not necessarily mean that it will have a similarly high accuracy for other datasets. For example, we can have a look at table 4.1 which contains the accuracy of the *Support Vector Machine (SVM)* algorithm in classifying the comments of five different YouTube music videos into spam and ham. It appears that the *SVM* algorithm has the worst accuracy for the "Katy Perry" dataset and the best accuracy for the "LMFAO" dataset, even though both datasets contain comments extracted from YouTube and both videos these comments are related to are music videos.

Table 4.1: Data snippet from table 3 on Sharmin and Zaman [28].

Algorithm	**Katy Perry Dataset**	**Shakira Dataset**	**Psy Dataset**	**Eminem Dataset**	**LMFAO Dataset**
SVM	57.43%	70.27%	93.71%	92.05%	91.09%

4.1 Spam / Ham Categorization

Similarly as in chapter 3, the first step in the classification process will be to distinguish the relevant from irrelevant comments.

4.1.1 (Semi-) automatic Approach

The first two algorithms I applied were *Random Forest (RF)* and *Support Vector Machine (SVM)*. The results can be found in table 4.2. From the table, I can see that the accuracy if high for both *RF* and *SVM* algorithms, but does this mean they do well on this dataset? To be able to answer this question we should have a look on figures 4.1 and 4.2 where the confusion matrices are depicted. It appears that almost all comments were classified as spam by *RM* and all comments are labelled as spam by *SVM*. However, we know that this can not be right because as we can see in figure 3.5 the "Tunnels" dataset has a significant number of ham comments as well. In other words, the algorithm simply classifies all or almost all comments as spam, and since there is a high amount of spam in this dataset, the accuracy is high. So to answer the question stated above, this classification does quite badly. This is because the dataset is strongly imbalanced, which means that both categories used for the classification are not equally represented in the dataset. In other words, there is much more spam than ham in the "Tunnels" dataset. There are different approaches to solve the problem of an unbalanced dataset. The work Gao et al. [8] introduces some of them. I have decided to solve this problem by keeping as many spam comments as ham in the dataset.

Table 4.2: The scores of the first two algorithms I used to classify the unbalanced dataset of the YouTube video "Tunnels" into spam and ham. "H" in the table means ham while "S" means spam.

Algorithm	Accuracy	Precision	Recall	F1-Score
Random Forest (RF)	83%	H: 0.84 S: 0.83	H: 0.13 S: 0.99	H: 0.23 S: 0.91
Support Vector Machine (SVM)	82%	H: 0.00 S: 0.82	H: 0.00 S: 1.00	H: 0.00 S: 0.90

I then test the now balanced dataset again on the two introduced algorithms above and some additional ones. The results of this step can be observed in table 4.3. Based on this table and on the confusion matrices, I can say the best performing classifier for the *Spam / Ham* categorization, and the dataset "Tunnels" is the *Voting Classifier*

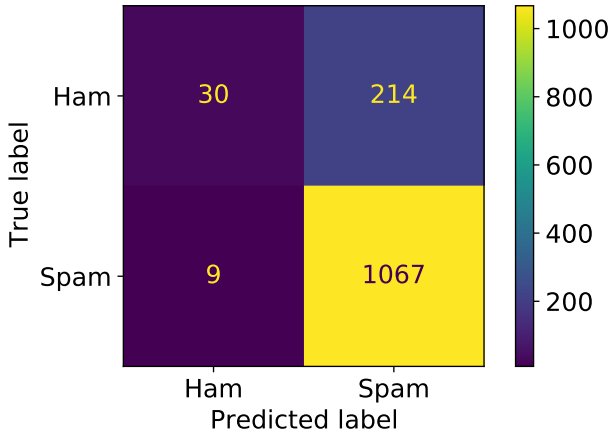

Figure 4.1: Confusion matrix of the results of the *Random Forest* algorithm on the unbalanced "Tunnels" dataset.

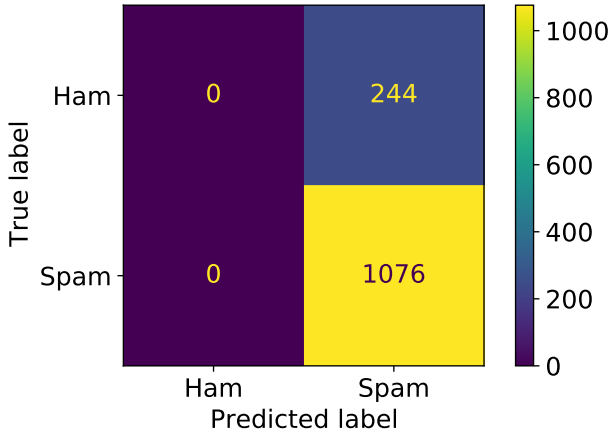

Figure 4.2: Confusion matrix of the results of the *SVM* algorithm on the unbalanced "Tunnels" dataset.

that combines the results of *RF*, *SVM* and *LR*. To compare the confusion matrices for *RF* and *SVM* before and after balancing the "Tunnels" dataset, the matrices with the data after balancing the dataset can be found in the appendices A.1 and A.2. The remaining matrices for the other algorithms can be found in the compact disc attached to this work.

Table 4.3: The scores of the algorithms I used to classify the comments of the "Tunnels" dataset into spam and ham. "H" in the table means ham while "S" means spam.

Algorithm	Accuracy	Precision	Recall	F1-Score
Random Forest (RF)	80%	H: 0.76 S: 0.86	H: 0.86 S: 0.76	H: 0.81 S: 0.81
Support Vector Machine (SVM)	79%	H: 0.74 S: 0.85	H: 0.85 S: 0.74	H: 0.79 S: 0.79
Naive Bayes	69%	H: 0.70 S: 0.69	H: 0.61 S: 0.77	H: 0.65 S: 0.73
Linear Regression (LR)	78%	H: 0.75 S: 0.81	H: 0.80 S: 0.77	H: 0.77 S: 0.79
Voting Classifier (RF, SVM, LR)	81%	H: 0.76 S: 0.87	H: 0.87 S: 0.76	H: 0.81 S: 0.81

4.1.2 Naive Approach

In Rahim et al. [22] it was suggested that, besides the comments, other data like the number of views, likes, or dislikes can be used to predict the gross income of movies while mining trailers data from YouTube comments. This statement motivated me to analyze additional data to the comments like the number of likes and replies to explore if I can derive any information from this data about the comments' relevance.

In this section, we will have a look at some naive and straightforward approaches to distinguish spam and ham comments. I hypothesise that the comments with the most likes and replies should be relevant since they draw the attention of the users who react to them. Furthermore, the comments containing the most characters and punctuation marks should also be relevant because this would mean that the viewers have explained their ideas in great detail. However, by looking at 4.3, I can see that the length of a comment, the number of punctuation marks it contains, and the number of likes and replies it has received does not influence the relevance of the comment, so I have to reject my previous hypothesis. For a quantitative explanation, we can have a look at table 4.4. If I were to choose the comments with at least one like while hoping to receive the relevant comments, I would have obtained only 10.73% of all actual ham comments. Similarly, by obtaining comments with at least one reply, I would get only 10.99% of the overall ham. By selecting comments with at least one like and at least one reply, I would receive an even smaller amount of ham, just 6.02%. Concluding, I can say that this naive approach is not beneficial for distinguishing relevant comments.

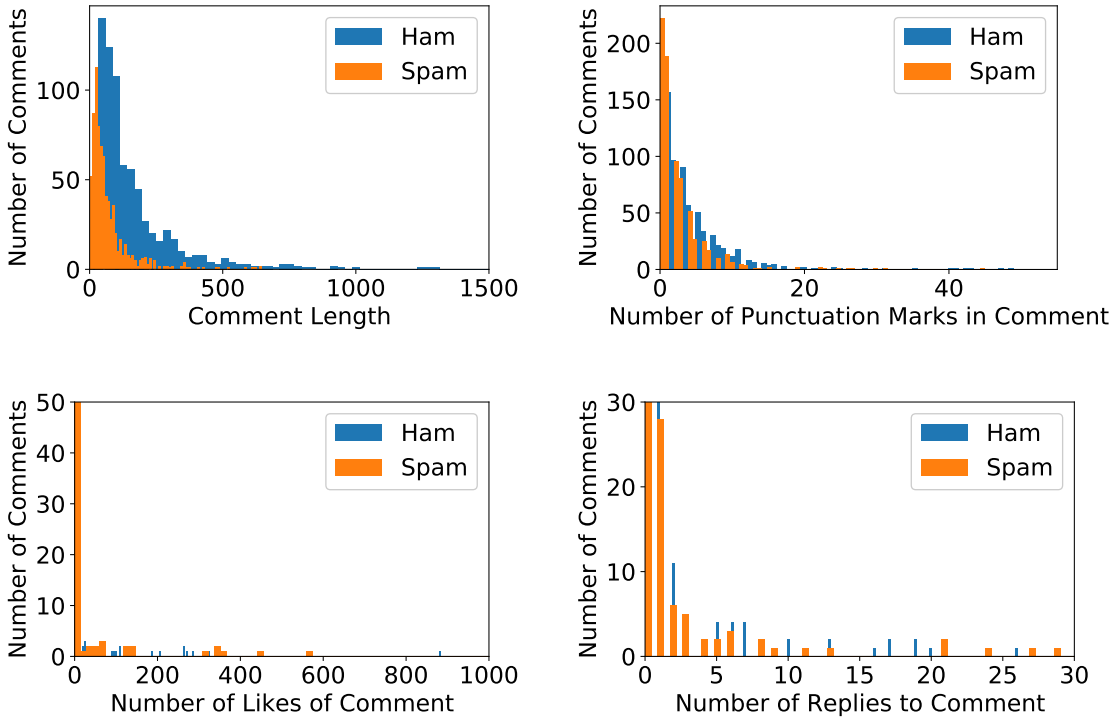

Figure 4.3: The influence of the length, punctuation marks, likes and replies to the comment's category.

Table 4.4: The ratio of comments that fulfill the condition on the left to all comments classified as ham.

Condition	Rate of Comment with Condition on the left to Overall Ham Comments
Comments with at least 1 Like	10.73%
Comments with at least 1 Reply	10.99%
Comments with at least 1 Like and 1 Reply	6.02%

4.2 Summarizing Comments classified as Ham

Using the *Spam/Ham* categorization, I was able to pick the relevant comments out of a vast amount of comments. However, there are still many relevant comments, so reading them all would take long. Furthermore, while reading a similar work

by Poché et al. [20], I got the idea to summarize comments. The work Poché et al. [20] analyzed YouTube user comments regarding coding tutorial videos. They described and investigated three different text summarization methods called *Term Frequency (TF)*, *Hybrid TF.IDF* and *SumBasic* for YouTube comments summarization. Hence, they concluded that *SumBasic* as a frequency-based summarization technique could sufficiently capture the main user's concerns expressed in YouTube comments. Therefore, I decided to try *SumBasic* on comments previously classified as ham. After using *SumBasic* on comments classified as ham, I counted the number of characters in these comments before and after summarization. As depicted in figure 4.4, the number of characters decreased drastically after the summarization. Subsequently, I looked at the content of the summarized comments, and similarly as Poché et al. [20] I observed that *SumBasic* could sufficiently capture the primary user's concerns in the comments.

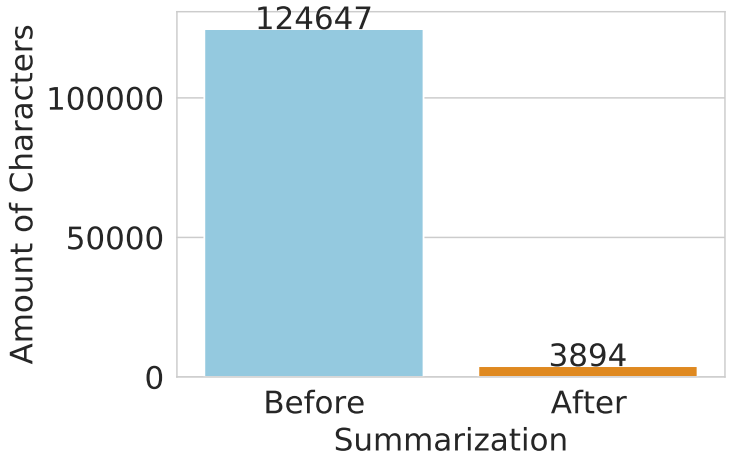

Figure 4.4: The number of characters of all comments classified as ham before and after summarization.

Additionally, I randomly selected seven comments out of all comments classified as ham and summarized them too. It appears that the summarized version again captures the user's concerns. However, the number of characters decreased just slightly, as depicted in figure 4.5. As far as I can see, this could be because the summarization process of *SumBasic* is based on the frequency of the terms. In vast quantities of comments, there are usually many recurrent ideas which can be summarized. On the other hand, there are most likely few to no repetitions in seven comments, so there is not much a summarizer can do. Hence, I can say that using a summarizer like *SumBasic* can come in handy when dealing with a considerable amount of comments.

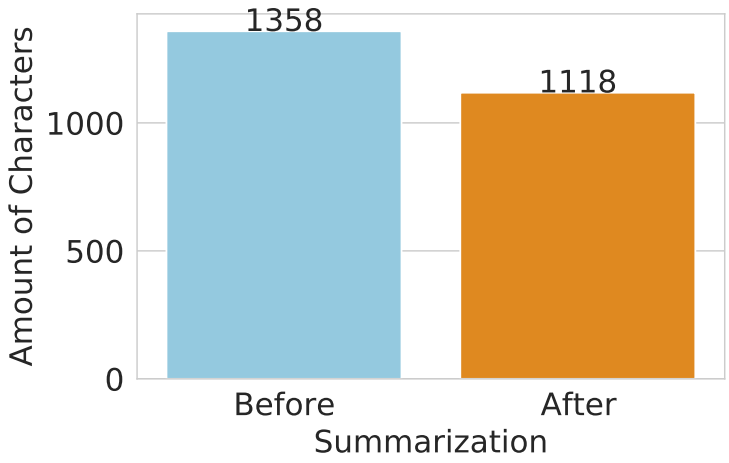

Figure 4.5: The number of characters of seven randomly selected comments out of all ham comments before and after summarization.

4.3 Sentiment Analysis

To perform the computer-based sentiment analysis, I used *TextBlob* and *SentiStrength* and subsequently the manual classification of chapter 3 to evaluate their performance on the dataset "Tunnels". The best result yielded *SentiStrength* as depicted in 4.5. By looking at the results in table 4.5, I can say that both *TextBlob* and *SentiStrength* work best to detect neutral comments. Detecting negative comments seems to be a more difficult task, while identifying positive comments seems to be the most challenging task. As far as I can see, this could be because the positive sentiment prevails in most comments classified as positive. However, they also contain negative or neutral sentiment. On the other hand, negative comments are often entirely negative (without a positive or neutral sentiment), making it easier to detect their sentiment as negative.

4.4 Content Related Classification

According to Maalej and Nabil [18], four multiple binary classifiers, one for each category type, perform significantly better than a single multiclass classifier in all cases. Therefore, I will use the binary *Voting Classifier (RF, SVM, LR)* which performed best in classifying comments into ham or spam, on each of the four classes *Feature Request*, *Flaw Report*, *Safety Related* and *Efficiency Related*. We can observe the result of this classification in 4.6. The *Voting Classifier (RF, SVM, LR)* does well in identifying comments related to safety and efficiency out of comments that do not concern safety

Table 4.5: The scores of the two algorithms I used to classify the balanced dataset of the YouTube video "Tunnels" into neutral, negative and positive.

Algorithm	Accuracy	Precision	Recall	F1-Score
TextBlob	67%	Positive: 0.32 Neutral: 0.64 Negative: 0.76	Positive: 0.57 Neutral: 0.33 Negative: 0.80	Positive: 0.41 Neutral: 0.43 Negative: 0.78
SentiStrength	73%	Positive: 0.49 Neutral: 0.72 Negative: 0.77	Positive: 0.59 Neutral: 0.34 Negative: 0.88	Positive: 0.54 Neutral: 0.46 Negative: 0.83

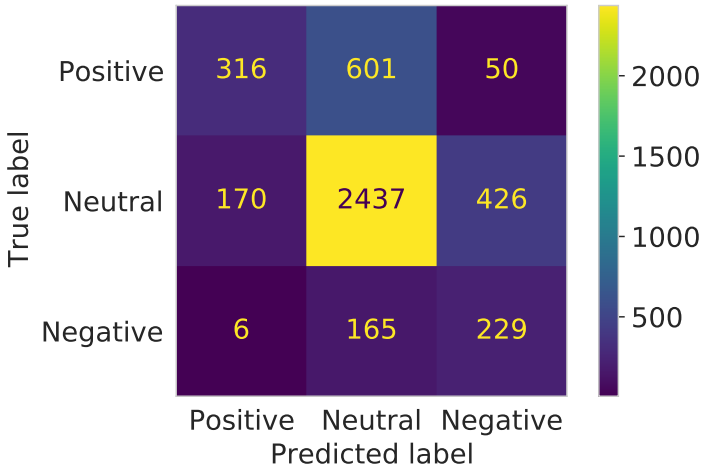

Figure 4.6: "Tunnels" Dataset classified using TextBlob.

and efficiency, respectively. For comments related to feature request and flaw report, this algorithm yields lower accuracy values, which means that these categories are more challenging for the classifier.

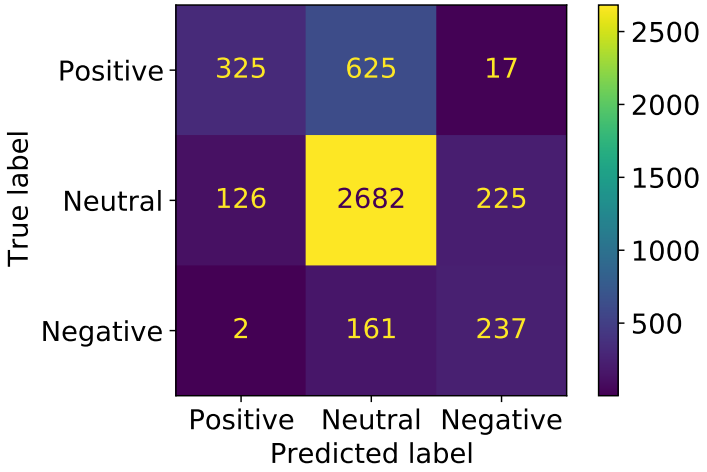

Figure 4.7: "Tunnels" Dataset classified using SentiStrength.

Table 4.6: Results of the evaluation for each class using the *Voting Classifier (RF, SVM, LR)*.

Category	Accuracy	Confusion Matrix	Precision	Recall	F1-Score	Label
Feature Request	72%	$\begin{bmatrix} 23 & 4 \\ 12 & 18 \end{bmatrix}$	0.66	0.85	0.74	Not Feature Request
			0.82	0.60	0.69	Feature Request
Flaw Report	68%	$\begin{bmatrix} 56 & 10 \\ 36 & 40 \end{bmatrix}$	0.61	0.85	0.71	Not Flaw Report
			0.80	0.53	0.63	Flaw Report
Safety Related	78%	$\begin{bmatrix} 51 & 13 \\ 15 & 46 \end{bmatrix}$	0.77	0.80	0.78	Not Safety Related
			0.78	0.75	0.77	Safety Related
Efficiency Related	78%	$\begin{bmatrix} 29 & 8 \\ 6 & 21 \end{bmatrix}$	0.83	0.78	0.81	Not Efficiency Related
			0.72	0.78	0.75	Efficiency Related

Chapter 5

Evaluation of Further Datasets

This chapter evaluates the same computer-aided approach as used in chapter 4 to classify comments according to their relevance, sentiment, and topics they cover, on another dataset. The reason behind this is to test the performance of the approach introduced in chapter 4 on additional previously unseen data. So, I will be using here the same source for the data, namely YouTube comments. Furthermore, the video the comments are referring too will again be a vision video. The categories used to label the comments will not change; solely, products and technologies presented in the vision video will differ, to keep the datasets as similar as possible. Testing the approach on more diverse datasets is left to future work.

The initial plan for this chapter was to use a freely available dataset constructed by other authors. Unfortunately, I could not find a similar dataset to the one used in chapter 4. In table 5.1, we can see some of the datasets I could find by browsing online platforms like `zenodo.org` or `datasetsearch.research.google.com`. As we can see from the table 5.1, there are enough datasets available that contain YouTube comments. However, none of them contains comments related to vision videos but related to movie trailers, cooking or music videos. Furthermore, the categories applied to label the comments do not match those used in this thesis. For example, the dataset "YouTube Spam Collection" labels the comments of the top ten YouTube Videos at the time (2020-03-15) into spam and ham but not into the other categories used in this thesis. So the datasets in table 5.1 only partly fulfill the requirements. Therefore I decided to create a new dataset for testing purposes, that fulfills all the conditions, as mentioned in table 5.1.

Table 5.1: Found datasets and their characteristics.

Does the data originate from YouTube? (yes/no)	Is Data related to a Vision Video? (yes/no)	Contains Spam/Ham Classification? (yes/no)	Contains Sentiment Dependent Classification? (yes/no)	Contains Topic Dependent Classification? (yes/no)
Sentiment Self-driving Cars [1]				
no	no	yes	yes	no
Trending YouTube Video Statistics and Comments [2]				
yes	no	no	no	no
YouTube Spam Collection [3]				
yes	no	yes	no	no
YouTube comments on Oscar-nominated movie trailers [4]				
yes	no	no	no	no
YouToxic English [5]				
yes	no	no	no	yes

5.1 Constructing datasets

5.1.1 The Land Rover Transparent Bonnet dataset

To create the dataset for testing purposes, I decided to use the comments of the same videos used by Vistisen and Poulsen [31]. The authors mentioned above performed a manual classification of YouTube comments of some vision videos using different categories. Hence, it would allow me to compare the results of another manual classification approach to mine. So the first step would be to classify the same comments as Vistisen and Poulsen [31] manually. The dataset used is unfortunately not available. However, the authors have listed in a table in their appendix 1 all the

[1] `data.world/crowdflower/sentiment-self-driving-cars`, accessed on 2021-01-19

[2] `kaggle.com/datasnaek/youtube`, accessed on 2021-01-19

[3] `kaggle.com/prashant111/youtube-spam-collection`, accessed on 2021-01-19

[4] `data.world/promptcloud/youtube-comments-on-oscar-nominated-movie-trailers`, accessed on 2021-01-19

[5] `zenodo.org/record/2586669#.YCRCL-oxmWg`, accessed on 2021-01-19

YouTube videos serving as sources for creating the dataset, so it is possible to recreate it. Nevertheless, the table's data originates from December 2nd 2016, so while trying to reconstruct their dataset, I noticed that some of the data is no longer up to date. For example, the number of views and likes of the videos has changed. More critical was that some of the videos were not available at the same URL as in 2016 on YouTube. So, I decided to construct a new similar table with the latest data for the available videos and note the video's unavailability otherwise. All videos I considered are taken from appendix 1 in Vistisen and Poulsen [31] except for the last four. Since several of these YouTube videos are no longer available, I tried to gather comments from other videos, which I found using the same terms as in Vistisen and Poulsen [31]. These terms were: "land rover transparent car", "land rover transparent hood", "land rover transparent", "land rover transparent bonnet", "land rover invisible hood", "land rover invisible bonnet", and "land rover invisible car". The data from these other videos is the source of the last four rows of the table A.4. The table A.4 contains the video ID, number of views, likes, unlikes, and comments for each video used to build this dataset. A video ID on YouTube is a distinct string located in the URL part that comes after *https://www.youtube.com/watch?v=* and is used to identify a video.

After preprocessing the dataset with the same steps used in 3.1.3, only 106 entries remained. Because of the low amount of comments in this dataset, I decided to use this dataset only to compare the manual classification results done by me with the one done by Vistisen and Poulsen [31]. For testing purposes, I choose to construct another dataset, as described in section 5.1.2. Due to lack of resources, all datasets in this chapter are manually classified only by one person (the author).

Comparing two manual Classification Methods

Another challenge was that not all results of the manual classification by Vistisen and Poulsen [31] were available. Solely, the manual classification results of just 21 comments used as examples in [31] were accessible. So I proceeded with the manual classification of these 21 comments using the same categories as in chapter 3.

In figure 5.1, we can see the groups *unserious-constructive*, *unserious-unconstructive*, *serious-unconstructive* and *constructive-serious* as created by Vistisen and Poulsen [31] in association to the groups of my manual classification (in red and blue). Futhermore, we can see that the groups *unserious-constructive*, *unserious-unconstructive* and *serious-unconstructive* contain only comments classified as spam, while the last group *constructive-serious* contain only ham comments. This match of categories is an exciting insight because Vistisen and Poulsen [31] state that "the constructive-serious block represents what a participatory design process would see as the core stakeholders, this group contains only ham comments, so my approach's results correspond to the results of the manual classification done in [31] as far as it

concerns identifying relevant comments.

On the other hand, sentiment detection seems to be challenging. According to the figure 5.1, only comments in the *unserious-unconstructive* group have a specific negative sentiment. In other groups, there is more than one sentiment present. We can receive further insights by looking at the dataset. For example, the comments in the category "Techno Pessimism" have negative sentiment, and the comments in the category "New Idea, Same Use" have a neutral sentiment. We have already seen a category similar to "Techno Pessimism", namely "Pessimistic about the idea or the future" in figure 3.9. Mutual for both datasets, this category matches the *spam-negative* group. Still, we have to keep in mind that these insights could change if there were more data from the manual classification done by Vistisen and Poulsen [31] available.

In this section, an obstacle was the lack of datasets that contain comments of vision videos on YouTube. Therefore, I decided to upload the dataset "Tunnels" built in chapter 3, to `zenodo.org` [6]. I choose this dataset because it contains the most entries out of all datasets created for this thesis.

5.1.2 The Hyperloop Dataset

To build this dataset, I extracted the comments of the "Hyperloop"[6] YouTube video using the same approach to preprocess and classify the comments as in chapter 3.

Analysing the Composition of the Dataset

According to the figure 5.2, the video "Hyperloop" has received almost as many replies as comments, in contrast to the video "Tunnels" which as mentioned in chapter 3 has received only 33.3% replies. In this section, I will analyse only the comments while the replies will be left as future work. To compare the spam-ham ratio of these datasets, we can have a look at figure 5.3 and 3.7 of the chapter 3. It appears that the "Hyperloop" dataset contains more ham comments (25%) than the "Tunnels" dataset (17%).

Additionally, there are more negative comments in the dataset "Hyperloop" (38.3%) than in the dataset "Tunnels" (22.0%).

Another difference between the two datasets is the most frequent theme in the ham comments. In the "Tunnels" dataset the theme that occurred the most was "Flaw Report" followed by "Safety Related" as we can see in figure 3.11. At the same time, in the "Hyperloop" dataset the most discussed subject is "Efficiency" followed by "Safety". However, the most common theme pair is for both datasets the category "Flaw Report - Safety Related".

As we can see in figure 5.5, for the dataset "Hyperloop", I have added another category called "Question" which contains user requests for additional information. I

[6] https://www.youtube.com/watch?v=S5fOWB6SNqs, accessed on 2020-01-24

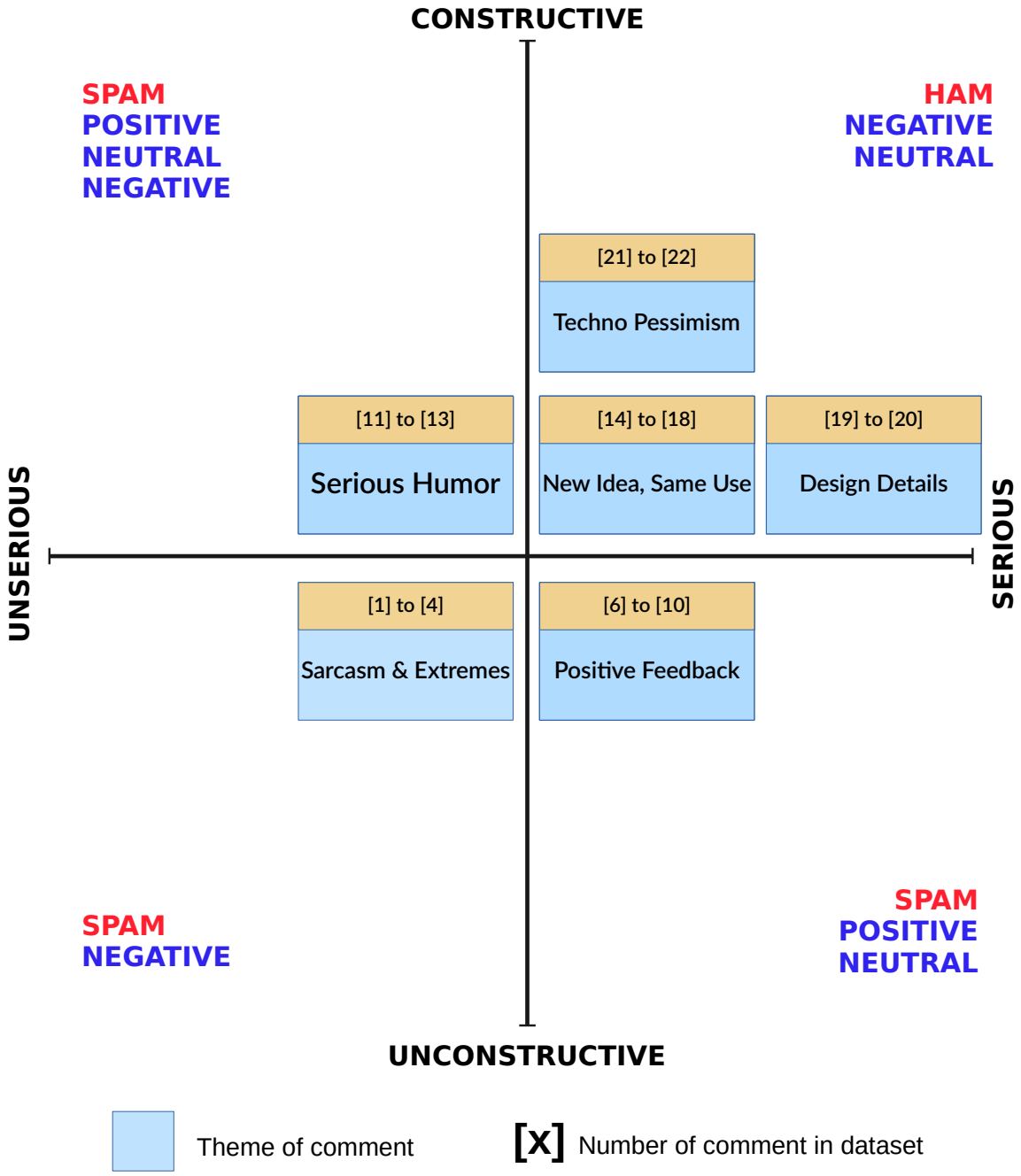

Figure 5.1: Comments used as examples in Vistisen and Poulsen [31] and their respective manual classification according to [31] and according to my categories.

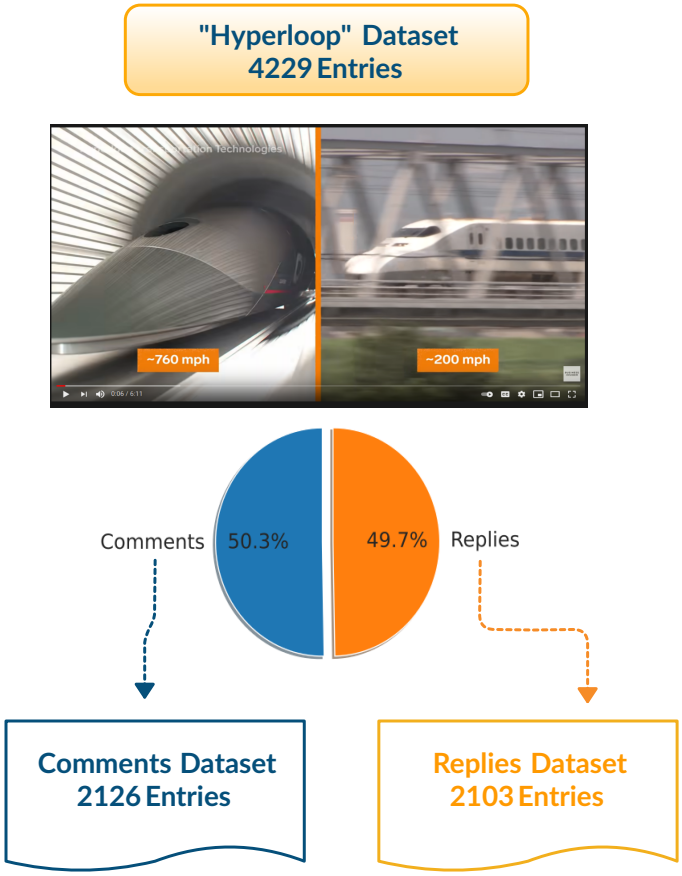

Figure 5.2: Structure of the "Hyperloop" dataset.

prepared this category because, in the dataset "Tunnels", most comments that did not fall in the four previously defined categories contained such questions. Despite adding an extra category, there are still 41 comments not related to any of the categories in figure 5.5. So I decided to have a closer look at their content. It appears that these comments mention mostly that this technology is not new and has been built before in other countries or by other companies. Therefore, an appropriate theme for this group would be, for example, "Same Idea, new Use" which was also one of the themes in Vistisen and Poulsen [31].

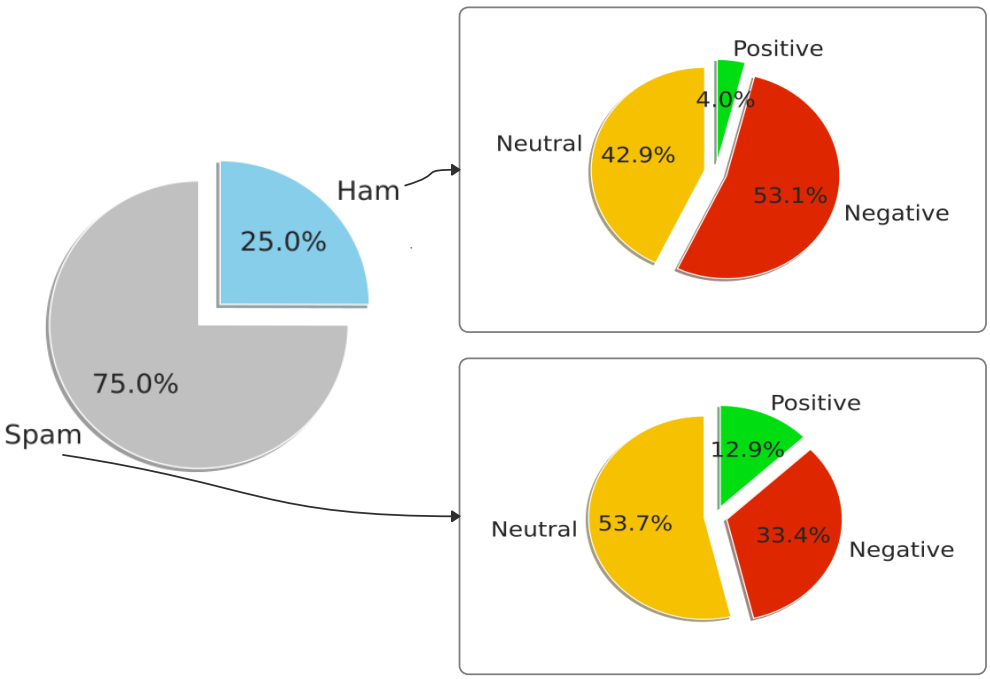

Figure 5.3: Sentiment analysis of ham and spam comments for the dataset "Hyperloop".

(Semi-) automatic Content Related Classification of the Comments

For the content related classification, I decided to train and evaluate the algorithm *Voting Classifier (RF, SVM, LR)* used in chapter 4 on the "Hyperloop" dataset because there is one extra category in this dataset not contained in the "Tunnels" dataset, namely the category "Question". By doing so, I will also have the opportunity to compare this algorithm's classification results on two different datasets. Similarly to the results of the classification of the "Tunnels" dataset, we can observe in 5.2, that the *Voting Classifier (RF, SVM, LR)* does well in identifying comments containing questions as well as comments related to safety and efficiency. However, for comments related to feature request and flaw report, this algorithm yields lower accuracy values, which means that these categories are more challenging for the classifier. The most challenging category is "Feature Request" with 61% accuracy.

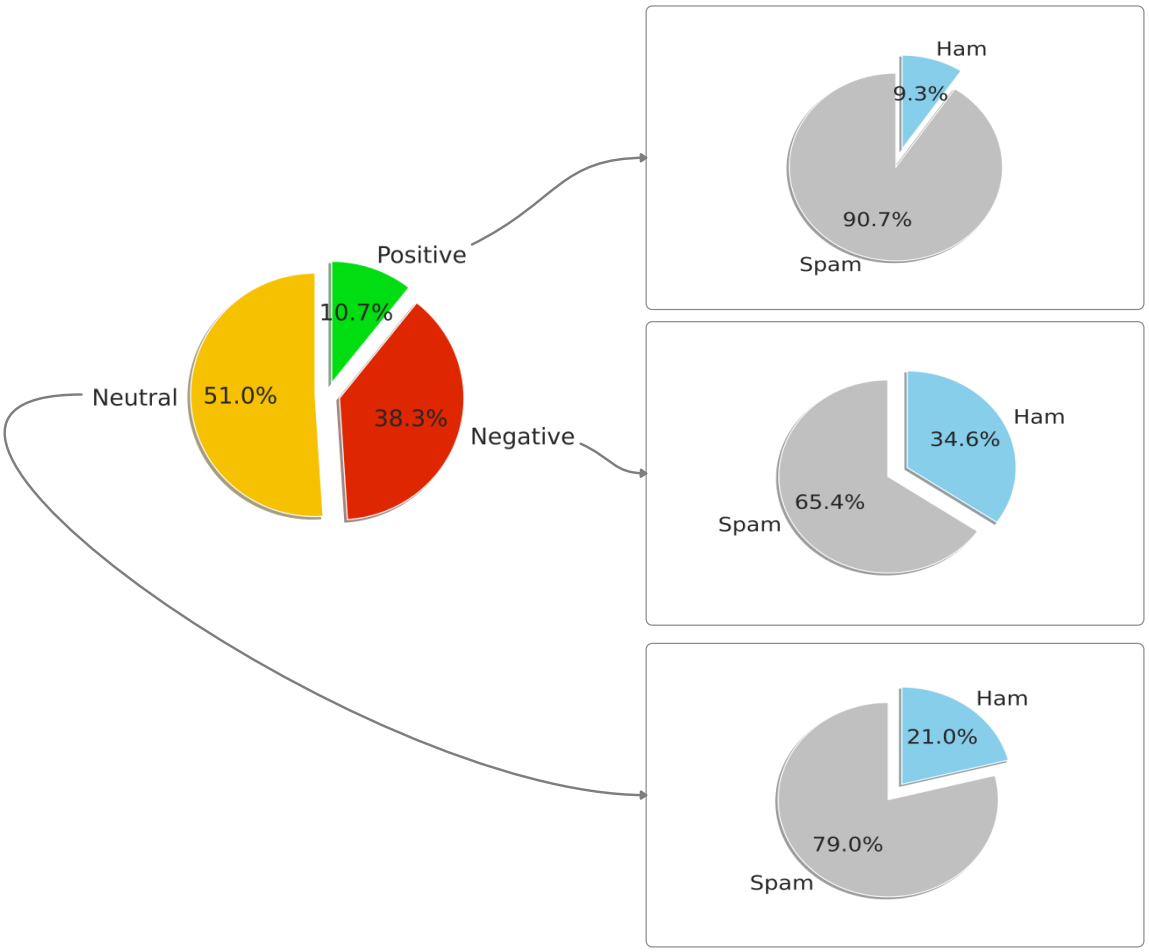

Figure 5.4: The spam and ham ratio in the comments classified as positive, negative and neutral of the dataset "Hyperloop".

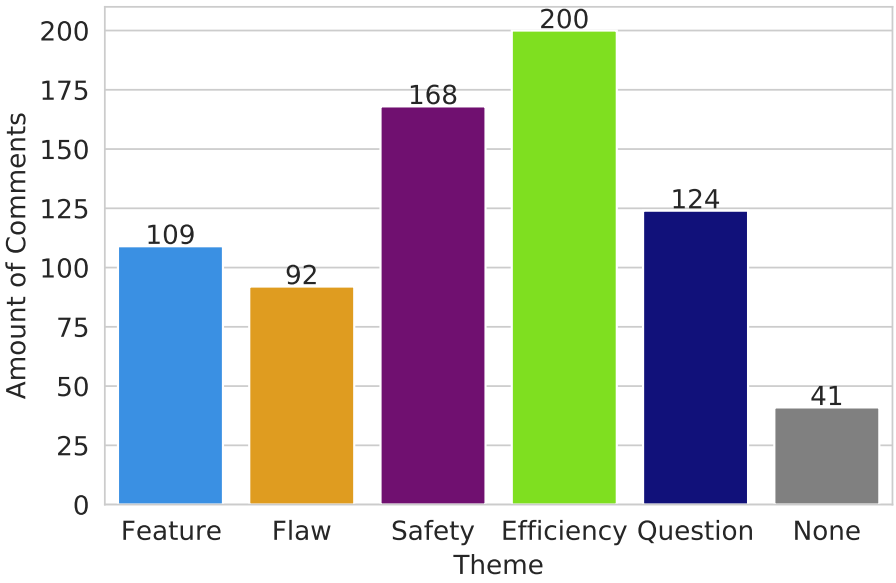

Figure 5.5: The number of comments of the "Hyperloop" Dataset classified as ham where the themes "Feature Request", "Flaw Report", "Safety Related", "Efficiency Related", "Asking for more Information" or none of the above occur.

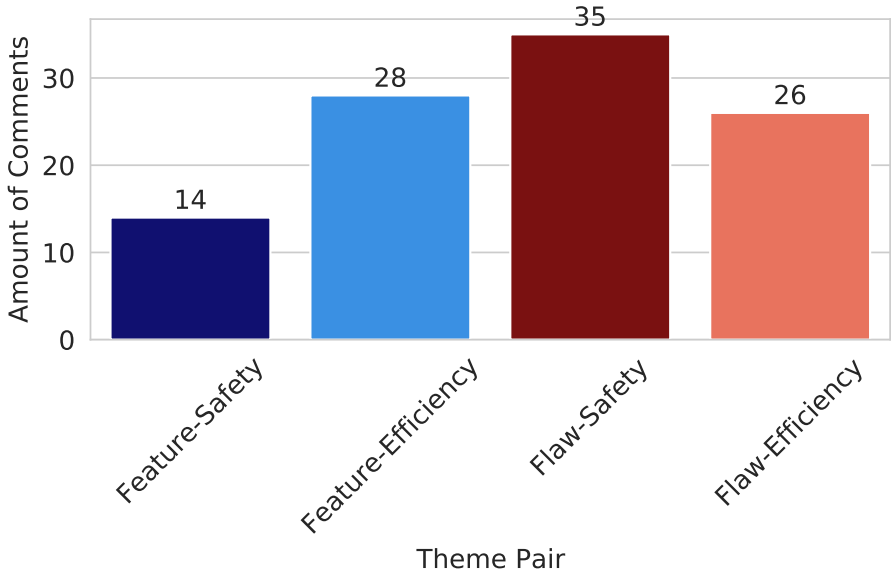

Figure 5.6: The number of comments classified as ham in the dataset "Hyperloop" where the stated theme pairs occur.

Table 5.2: Results of the evaluation for each class using the *Voting Classifier (RF, SVM, LR)*.

Category	Accuracy	Confusion Matrix	Precision	Recall	F1-Score	Label
Feature Request	61%	$\begin{bmatrix} 15 & 2 \\ 15 & 12 \end{bmatrix}$	0.50	0.88	0.64	Not Feature Request
			0.86	0.44	0.59	Feature Request
Flaw Report	70%	$\begin{bmatrix} 12 & 3 \\ 8 & 14 \end{bmatrix}$	0.60	0.80	0.69	Not Flaw Report
			0.82	0.64	0.72	Flaw Report
Safety Related	79%	$\begin{bmatrix} 26 & 11 \\ 3 & 28 \end{bmatrix}$	0.90	0.70	0.79	Not Safety Related
			0.72	0.90	0.80	Safety Related
Efficiency Related	75%	$\begin{bmatrix} 32 & 14 \\ 6 & 28 \end{bmatrix}$	0.84	0.70	0.76	Not Efficiency Related
			0.67	0.82	0.74	Efficiency Related
Questions	78%	$\begin{bmatrix} 21 & 2 \\ 9 & 18 \end{bmatrix}$	0.70	0.91	0.79	Not Questions
			0.90	0.67	0.77	Questions

Chapter 6

Related Works

This chapter gives a brief overview of related works and their similarities to this thesis.

The writing that motivated this thesis the most was Vistisen and Poulsen [31], where the authors manually classify comments related to vision videos posted on YouTube. This process's outcome was a blackboard covered in post-it notes, which signifies how complicated and confusing such manual classification can be. Therefore, my supervisor and I searched for related work where the authors do automated classifications. For example, the authors of Maalej and Nabil [18] and Guzman and Maalej [10] have automatically analyzed app reviews and Guzman et al. [12] have analyzed and prioritized posts on Tweeter to extract user feedback. Automated approaches were also used by Guzman et al. [11] to determine the sentiment of commits on GitHub, although this is not directly related to requirement engineering.

After selecting YouTube as the platform to search for vision videos, I searched for other works that automatically analyze YouTube comments. For example, Sharmin and Zaman [28], Abdullah et al. [1] and Das et al. [5] classify YouTube comments into ham or spam. However, the videos these authors analyzed are music videos. In R. Benkhelifa and F. Z. Laallam [21] and Poché et al. [20] YouTube video comments related to cooking recipes and coding tutorial videos were analyzed, respectively. There were also some other categories than ham and spam used. For example, Rahim et al. [22] investigated YouTube comments related to movie trailers to predict the movie's income, Asghar et al. [3] studied the sentiment of YouTube comments, Obadimu et al. [19] classified the comments according to their toxicity which was similar to sentiment analysis but toxicity, in this case, means how harmful the comments could be to other users, and Khan et al. [16] classified comments into multiple categories according to their content.

So as we see from the examples above, none of the related works I found classified vision videos' comments using automated approaches. However, the platform YouTube was often used in similar work as well as the categories ham and spam and sentiment analysis.

Chapter 7

Summary and Outlook

7.1 Summary

In this thesis, we successfully automated the process of extracting requirements out of YouTube vision video comments. The first step was to create two datasets out of YouTube video comments. I named these datasets "Tunnels" and "Hyperloop" respectively. Then I manually classified the comments of these datasets into the categories *Spam*/*Ham*, analyzed their sentiment and classified them according to their content. An exciting insight into these datasets' structure is that the first dataset "Tunnels", consists mainly of spam comments. In contrast, in the dataset "Hyperloop" the amount of ham and spam comments were almost equal. For all datasets in this thesis, it applies that some of the comments were written in other languages and could not be translated or comments consisting only of emojis. Since we do not receive feedback for the requirement engineering from such comments, we have to remove them from the dataset, so a dataset cleaning step is inevitable when dealing with YouTube comments. Additionally, both datasets consist mainly of comments with a neutral sentiment. To select what categories to use in the content related classification, I looked at related works like Maalej and Nabil [18] and built word clouds with the most frequent word pairs that occur in the comments. As a result, I decided to use the categories *Feature Request*, *Flaw Report*, *Safety Related* and *Efficiency Related*. In the dataset "Tunnels", the most frequent themes were *Flaw Report* and *Safety Related*. While in the other dataset "Hyperloop", the most frequent themes were *Safety Related* and *Efficiency Related*. I realised that these four categories were present in both datasets even though I created these categories based on the word clouds of the dataset "Tunnels". However, in both datasets, some comments did not fall into any of these categories. They contained, for example, questions asking about more details related to the design. Since the manual classification of Vistisen and Poulsen [31] was a huge motivation for this work, I rebuilt the same dataset as

the authors, named it the "Land Rover Dataset", manually classified it and compared the results of my manual and the manual classification of the authors. An important finding was that comments contained in the category "Constructive-Serious" which, according to the authors, contains all the relevant comments, correspond with the content of my category ham which also contains the relevant comments. In addition to the manual classification, I tried to extract the relevant comments using naive approaches like selecting comments according to their number of likes, replies, or characters. This naive approach was not successful since I could extract only around 10% of the relevant comments. However, the *Spam*/*Ham* classification was successful. The algorithm *Voting Classifier* combining the results of *Random Forest*, *Support Vector Machine* and *Linear Regression* achieved an 81% accuracy. For sentiment analysis *SentiStrength* yielded the best result (73% accuracy). The comments could also be classified quite well for the classes I created according to the comments' content. The accuracy of the categories *Safety Related* and *Efficiency Related* was the most remarkable (78%) compared to the other categories. The content-related classification worked similarly well for the dataset "Hyperloop" too. Although I could extract the relevant comments with the approaches mentioned above, there were still a large number of them. I tried to summarize the relevant comments using the *SumBasic* algorithm to reduce them, which also worked pretty well. The number of character in the relevant comments was drastically reduced, but the relevant information remained.

7.2 Outlook

To do the manual classification process in this thesis, I used spreadsheet programs which lead to some problems like missing cells in the dataset because the raters overlooked some entries in the table. Correcting such mistakes takes much time, and massive data administration in spreadsheets is error-prone. Therefore a program with a user interface is needed to prevent mistakes while building datasets. Such programs already exist like the one used by Al-Tamimi et al. [2] and Maalej and Nabil [18], but they are too specific. For example they are suitable to classify comments according to their sentiment, but I can not classify the comments according to their content. So a UI where the user can manage the labels used to classify the comments would be helpful.

The comments analyzed in this thesis are all extracted from vision videos posted on the online platform YouTube. It would be interesting to analyze datasets consisting of vision video comments posted on social media platforms other than YouTube. This way, we could see if the comments' quality changes with the social media platform and if the approaches used in this thesis would yield similar or other results.

After collecting the viewer's feedback to the vision videos, I analyzed the comments, amongst other categories, according to their sentiment. It appeared to me that the video creators might decide to make some adjustments to their design, for example,

by changing, adding or removing features as a response to the feedback collected from the comments. In such a case, they may update the vision video to gather new feedback regarding this development. Afterwards, sentiment analysis could be used again to compare the users' sentiment before and after the change and determine the effect of the change on the viewers' attitude towards the product presented in the vision video.

YouTube offers two possibilities to sort the comments of a video. The users can sort them by "Top comments" or by "Newest first". The first method displays only some of the comments that YouTube algorithms select as the most relevant comments. The second option will display all the comments sorted by the time they were posted starting from the newest on top and ending with the oldest one. A possible future work would be first to analyze the criteria used by YouTube's "Top comments" sorting algorithm (if available). Then the relevance of the "Top comments" could be examined, for example, by calculating the ratio of spam to ham in the top comments and comparing it to the ration of spam to ham in all comments. Additionally, the number of relevant comments that would be left out using the "Top comments" function should be investigated.

The authors in Al-Tamimi et al. [2] suggest that using only two classes (positive and negative) in sentiment analysis leads to better results because annotating comments with contrasting opinions as neutral represents a tricky task for classifiers. Therefore, we could evaluate the same dataset as future work, using only two labels (positive and negative) for the sentiment analysis. Then we could compare the performances of the same classifiers in the task of sentiment analysis using two and three labels and decide if using two labels leads to better results.

Even though after selecting the relevant comments and allocating them to different categories, there can still be many comments to process. A next step to take in this case would be to prioritize the comments according to their relevance. This would help the persons involved in software development to decide which comments they should react to first. This means that highly relevant comments should be handled as soon as possible. On the other hand, less relevant comments can be dealt with later. The authors in Guzman et al. [12] proposed an approach to prioritize tweets. Based on the insights of this paper, a similar process could be done using YouTube comments. The first step in the process described in Guzman et al. [12] was to do a survey requiring 84 participants involved in software engineering to prioritize tweets manually. Since this would go beyond this thesis's scope, this is left as a suggestion for future work.

All the datasets constructed in this thesis consist only of comments written in English. Still, in online comment sections, not all comments are written in this language. In my opinion it would be interesting to research if the discussed approaches in this thesis work similarly well for comments in other languages. If this is not the case, then other approaches or improvements of the approaches discussed here should be taken. Different languages bring additional challenges along. For example, in

Arabic, each word has many derivatives with different meanings and polarities for each derivative and different regional dialect [2]. The authors in Al-Tamimi et al. [2] conducted a sentiment analysis of YouTube Comments written in the Arabic language. Another study that analyses comments written in a language other than English is Gao et al. [8]. The authors examine YouTube comments written in Cantonese, which mixes traditional Chinese with some characters borrowed from the English language to represent spoken terms.

Appendix A

Complementary Details to Datasets

This appendix contains tables referenced in the previous chapters. The data in these tables is not essential to comprehend this thesis, but it plays a complementary role.

A.1 Tunnels Dataset

Table A.1: This table contains three randomly selected comments of the video "Tunnels" by "The Boring Company" on YouTube to demonstrate the dataset's structure where the comments and other data related to them like the author of the comment, or the number of likes are stored. The dots on the last row symbolize that the dataset contains more than three rows.

ID	Author	Comment	Likes	Replies
Ugz3BfY8u985NHg87054AaABAg	L F	Just wait until there's a derail in one of these tunnels, cars would smash into each other causing pileups, and would make traffic even worse than ever. Not to mention the fact that there is no emergency stop/braking system on the pad. What if there's a fire, an explosion and a pile of smoke coming from the tunnels, earthquakes? Etc. I'd just prefer to be stuck in traffic.	0	2
UgyAa_59LOosvrKYhKx4AaABAg	harsha sutapalli	Isn't it underground train	6	0
UgwHm4C8WsVuoMAaLzp4AaABAg	Ghost Umer	So how will people access gps	0	0
...	...	...	...	...

A.2 Land Rover Dataset

A.2.1 Land Rover Dataset Comments' Source

Table A.2: This table contains two entries of the replies dataset of the video "Tunnels" by "The Boring Company" on YouTube. This example shows that the first comment in the table A.1 has the same ID as the two replies in this table, signifying that these replies belong to this comment.

ID	Author	Comment	Likes
Ugz3BfY8u985NHg87054AaABAg	Cameron Norton	This was made 3 years ago, a prototype of something that could be. They have probably thought about the things you are discussing already and have fixed them with a large amount of testing	0
Ugz3BfY8u985NHg87054AaABAg	L F	@Cameron Norton Well, disasters can happen in anywhere inside a place, even in tunnels. If there's one clog problem in these tunnels then you're screwed, if there's a flood your car will just malfunction. Notice how in this video, once the car enters the underground, the hole doesn't close once its on the rail right? If a drunk driver or idiot accidentally swipes his car around the road and falls into the hole then I would feel sorry for you. If you accidentally press the reverse or drive gear then you are dead. We don't need so many 2055 tech in our world.	0
...	...	...	...

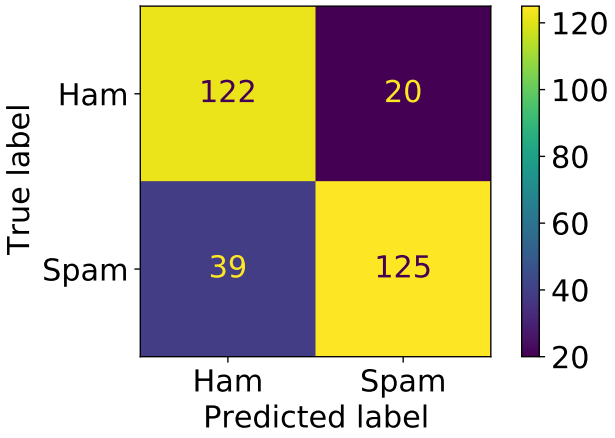

Figure A.1: Confusion matrix of the results of the Random Forest algorithm on the balanced "Tunnels" dataset.

Table A.3: This table contains some of the replies dataset entries, the comments they are related to, and a short description/category for each reply. Note that the categories listed in this table are not all possible categories in this dataset but more abundant ones.

Comment and Reply	**Reply Content Identifier / Relation to Comment**
Comment: Imagine the damage that an earthquake could cause to all this. Reply: Unless you are building tunnels between tectonic plates the tunnel has no way to rip off from between. If anything it moves with the ground like, say a submarine underwater.	Disagreement
Comment:Imagine one accident, causes a tunnel shutdown for the entire day. The cars will need to be AI driven on an automated track, human error causes too many issues for this to be realistic. Reply: The cars don't drive, they are on a platform.	Disagreement and/or giving additional/new information
Comment: What if it gets a flat tire or the tire explodes Reply: All teslas in the network will be notified, and will engage 'Warning lights', they will come to a stop, and notify the passengers that there has been a delay.	Giving a solution to a problem introduced in the comment
Comment: I don't think they actually plan on putting the elevators on roads. They'll probably put them in parking lots, parking garages, buildings, and empty space. There might be some in the street, but I'm sure if this becomes highly integrated which In cities, which I definitely see happening, then they'll think of the most efficient locations. Also, tunnels can be designed to withstand earthquakes. Reply: Imagine one accident, causes a tunnel shutdown for the entire day. The cars will need to be AI driven on an automated track, human error causes too many issues for this to be realistic.	Problem report
Comment: Only compatible with Teslas. Unless you own a flamethrower. Reply: Why only electric cars?	Asking for addition information

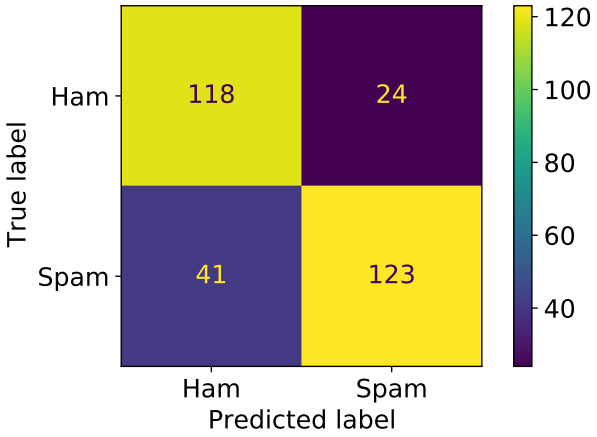

Figure A.2: Confusion matrix of the results of the SVM algorithm on the balanced "Tunnels" dataset.

Table A.4: Data fetched from YouTube on 23 January 2021.

YouTube Channel (number of subscribers in parentheses) and Video ID	**Views**	**Likes**	**Unlikes**	**Comments**	**Video Availability**
Land Rover USA (-) L7j1daOk72c	-	-	-	-	Video unavailable. This video is private.
Land Rover (-) 6TRGwLDLRp8	-	-	-	-	Video unavailable. This video is private.
GeoBeats News (-) OIfAiQ1Dmes	-	-	-	-	Video unavailable. This video is private.
E Birmingham (-) LrqeNbN0nKc	-	-	-	-	Video unavailable. This video is private.
TestDriven (-) 7j3-y-FqjBM	-	-	-	-	Video unavailable. This video is private.
CARWP (-) he5PxTPQTDg	-	-	-	-	Video unavailable. This video is private.
Land Rover Russia (-) iEI25YBcDZU	-	-	-	-	Video unavailable. This video is private.
World Insiders (-) 1AoefqGJBtI	-	-	-	-	This video is no longer available because the YouTube account associated with this video has been terminated.
CNET KOREA (-) -	-	-	-	-	Wrong Video URL
Skiddmark (-) OZu2x9wRGuE	-	-	-	-	Video unavailable.
Land Rover UK (99.1K) 1OlqditIsoM	705,941	1,2K	48	79	Available
MOTOR1 (216K) Xwfc3Bad9d4	123902	125	15	8	Available

Table A.4 continued from previous page

On Demand News (958K) vgOsPXobl7M	41232	150	6	16	Available
NEWCARNET (0) K8gPPuryqy4	504	5	0	1	Available
Autofácil (71.3K) iMw0sSLw0FI	1638	2	2	0	Available
Vrum (329K) zn5AGvDd0UU	4709	121	2	0	Available
Bloomberg Quicktake (2.47M) tIhkHiWMkSo	14086	19	2	2	Available
Official HD Mega Trailers (3) Df1dr4mcbKo	596	4	0	1	Available
Autoline Network (78.5K) -VjJmZGnmyM	3687	39	0	2	Available
RedditNewsNow (336) 1aoFSmg0cFg	618	0	2	0	Available
Jaguar Land Rover HK Official (941) 5YsPFfytNp4	435	3	0	0	Available
Jaguar Land Rover (4.3K) CDfoaAYHyl8	1247	10	0	1	Available
YsfAlgz (32) upOCl091bAQ	1344	3	1	0	Available
Land Rover Journal (1.62K) a3x__ND2mlo	309	3	1	0	Available
Adevraul Auto (36) 5f28H0aabAk	3865	0	0	0	Available
AutoConceptionTV (24.3K) mLFSNMBGyII	718	4	1	1	Available
YOUCAR (2.12M) a34jqLA7_Sc	20414	240	16	10	Available

Table A.4 continued from previous page

Autogefühl (517K) tN6dijNuMhw	6827	22	2	1	Available
SlashGear (76.7K) vo2tEwDSJ_Y	48313	239	9	13	Available

Bibliography

[1] A. O. Abdullah, M. A. Ali, M. Karabatak, and A. Sengur. A comparative analysis of common youtube comment spam filtering techniques. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–5, 2018. doi: 10.1109/ISDFS.2018.8355315.

[2] A. Al-Tamimi, A. Shatnawi, and E. Bani Issa. Arabic sentiment analysis of youtube comments. pages 1–6, 10 2017. doi: 10.1109/AEECT.2017.8257766.

[3] M. Z. Asghar, S. Ahmad, A. Marwat, and F. M. Kundi. Sentiment analysis on youtube: A brief survey. *arXiv preprint arXiv:1511.09142*, 2015.

[4] R. Chowdury, M. N. Monsur Adnan, G. A. N. Mahmud, and R. M. Rahman. A data mining based spam detection system for youtube. In *Eighth International Conference on Digital Information Management (ICDIM 2013)*, pages 373–378, 2013. doi: 10.1109/ICDIM.2013.6694038.

[5] R. K. Das, S. S. Dash, K. Das, and M. Panda. Detection of spam in youtube comments using different classifiers. In *Advanced Computing and Intelligent Engineering*, pages 201–214, Singapore, 2020. Springer Singapore. ISBN 978-981-15-1081-6.

[6] K. Eklekta. Relevance and Sentiment of YouTube Comments of a Vision Video, February 2021. URL `https://doi.org/10.5281/zenodo.4533302`.

[7] S. Fricker, K. Schneider, F. Fotrousi, and C. Thuemmler. Workshop videos for requirements communication. *Requirements Engineering*, 21, 06 2015. doi: 10.1007/s00766-015-0231-5.

[8] J. Gao, Q. Cheng, and P. L. H. Yu. Detecting comments showing risk for suicide in youtube. In *Proceedings of the Future Technologies Conference (FTC) 2018*, pages 385–400, Cham, 2019. Springer International Publishing. ISBN 978-3-030-02686-8.

[9] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, and M. Stade. The crowd in requirements engineering: The landscape and challenges. *IEEE Software*, 34(2):44–52, 2017. doi: 10.1109/MS.2017.33.

[10] E. Guzman and W. Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162, 2014. doi: 10.1109/RE.2014.6912257.

[11] E. Guzman, D. Azócar, and Y. Li. Sentiment analysis of commit comments in github: An empirical study. MSR 2014, page 352–355, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328630. doi: 10.1145/2597073.2597118. URL https://doi.org/10.1145/2597073.2597118.

[12] E. Guzman, M. Ibrahim, and M. Glinz. Prioritizing user feedback from twitter: A survey report. In *2017 IEEE/ACM 4th International Workshop on CrowdSourcing in Software Engineering (CSI-SE)*, pages 21–24, 2017. doi: 10.1109/CSI-SE.2017.4.

[13] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl. Word cloud explorer: Text analytics based on word clouds. In *2014 47th Hawaii International Conference on System Sciences*, pages 1833–1842, 2014. doi: 10.1109/HICSS.2014.231.

[14] H. Huang, H. Xu, X. Wang, and W. Silamu. Maximum f1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):787–797, 2015.

[15] O. Karras, K. Schneider, and S. Fricker. Representing software project vision by means of video: A quality model for vision videos. *Journal of Systems and Software*, 162:110479, 11 2019. doi: 10.1016/j.jss.2019.110479.

[16] A. Khan, M. Khan, and M. Khan. Naive multi-label classification of youtube comments using comparative opinion mining. *Procedia Computer Science*, 82:57–64, 12 2016. doi: 10.1016/j.procs.2016.04.009.

[17] A. Lucia and A. Qusef. Requirements engineering in agile software development. *Journal of Emerging Technologies in Web Intelligence*, 2, 01 2003. doi: 10.4304/jetwi.2.3.212-220.

[18] W. Maalej and H. Nabil. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 116–125, 2015. doi: 10.1109/RE.2015.7320414.

[19] A. Obadimu, E. Mead, M. N. Hussain, and N. Agarwal. Identifying toxicity within youtube video comment. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 214–223. Springer, 2019.

[20] E. Poché, N. Jha, G. Williams, J. Staten, M. Vesper, and A. Mahmoud. Analyzing user comments on youtube coding tutorial videos. In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*, pages 196–206, 2017. doi: 10.1109/ICPC.2017.26.

[21] R. Benkhelifa and F. Z. Laallam. Opinion extraction and classification of real-time youtube cooking recipes comments. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 395–404, Cham, 2018. Springer International Publishing. ISBN 978-3-319-74690-6.

[22] M. S. Rahim, A. Z. M. E. Chowdhury, M. A. Islam, and M. R. Islam. Mining trailers data from youtube for predicting gross income of movies. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 551–554, 2017. doi: 10.1109/R10-HTC.2017.8289020.

[23] S. L. Ramdhani, R. Andreswari, and M. A. Hasibuan. Sentiment analysis of product reviews using naive bayes algorithm: A case study. In *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 123–127, 2018. doi: 10.1109/EIConCIT.2018.8878528.

[24] Dymitr Ruta and Bogdan Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1):63–81, 2005. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2004.04.008. URL https://www.sciencedirect.com/science/article/pii/S1566253504000417. Diversity in Multiple Classifier Systems.

[25] K. Schneider and L. M. Bertolli. Video variants for crowdre: How to create linear videos, vision videos, and interactive videos. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 186–192, 2019. doi: 10.1109/REW.2019.00039.

[26] K. Schneider, O. Karras, A. Finger, and B. Zibell. Reframing societal discourse as requirements negotiation: Vision statement. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 188–193, 2017. doi: 10.1109/REW.2017.17.

[27] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press,

June 2018. ISBN 978-0-262-25693-3. doi: 10.7551/mitpress/4175.001.0001. URL https://doi.org/10.7551/mitpress/4175.001.0001.

[28] S. Sharmin and Z. Zaman. Spam detection in social media employing machine learning tool for text mining. In *2017 13th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 137–142, 2017. doi: 10.1109/SITIS.2017.32.

[29] D. K. Tayal and S. K. Yadav. Analysis of sentiments polarity computation of opinions. In *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, pages 1–6, 2017. doi: 10.1109/TEL-NET.2017.8343586.

[30] A. Viera and J. Garrett. Understanding interobserver agreement: The kappa statistic. *Family medicine*, 37:360–3, 06 2005.

[31] P. Vistisen and S. B. Poulsen. Return of the vision video: Can corporate vision videos serve as setting for participation? *Nordes*, 7(1), 2017.

[32] G. Yuan, C. Ho, and C. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012. doi: 10.1109/JPROC.2012.2188013.