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

Development of a Mood Feedback Tool for Developers in Meetings

Bachelorarbeit

im Studiengang Informatik

von

Kliti Nikollau

**Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Klünder
Betreuer: Martin Obaidi**

Hannover, 19.08.2022

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 19.08.2022

Kliti Nikollau

Abstract

With the increase in complexity and size of software projects, software development is seen mainly as a team effort. Meetings are seen as an essential part of software projects and as a main communication route for information exchange. Positive and productive meetings can directly influence the mood of the team members and since happy team members are essential for the success of a project, it is very important to better analyse and organise meetings.

Sentiment analysis tools have been mostly used to automatically evaluate the sentiment in a given text and also during meetings. The automatic tools though, still have some problems such as precision outside the context in which they were build, recognising irony or sarcasm and limited overall accuracy. Manual coding schemes of interactions during meetings have also been used to analyse meetings, but this approach brings the problem of subjectivity with it, since the evaluations are made from people outside the team.

Within the scope of this bachelor thesis, a tool is to be developed which allows the meeting participants to manually enter their feedback during a meeting. This feedback is than saved in files, which can be used to compare different meetings with one another. The user can also send the file manually to the project manager, so that they can compare the feedback from the whole team regarding the meetings progress. This can help the managers in realising problems that might arise in a team at an early stage and taking appropriate measures to avoid them. Furthermore, it can also help developers to compare their feedback during different meetings and gain a better overall perspective about their progression.

To evaluate this tool, 15 participants working in a software engineering context tested the tool in a meeting and answered a survey about their experience and feedback regarding the tool and what could be further improved.

Kurzfassung

Mit der zunehmenden Komplexität und Größe von Softwareprojekten wird die Softwareentwicklung hauptsächlich als Teamarbeit betrachtet. Meetings werden als wesentlicher Bestandteil von Softwareprojekten und als Hauptkommunikationsweg für den Informationsaustausch angesehen. Positive und produktive Meetings können sich direkt auf die Stimmung der Teammitglieder auswirken, und da zufriedene Teammitglieder für den Erfolg eines Projekts entscheidend sind, ist es sehr wichtig, Meetings besser zu analysieren und zu organisieren.

Tools zur Stimmungsanalyse wurden bisher hauptsächlich dazu verwendet, die Stimmung in einem bestimmten Text und auch während eines Meetings automatisch zu bewerten. Die automatischen Tools haben jedoch noch einige Probleme, wie z. B. die Präzision außerhalb des Kontexts, in dem sie erstellt wurden, die Erkennung von Ironie oder Sarkasmus und die begrenzte Gesamtgenauigkeit. Manuelle Kodierungsschemata von Interaktionen während Meetings wurden ebenfalls zur Analyse von Meetings verwendet, aber dieser Ansatz bringt das Problem der Subjektivität mit sich, da die Bewertungen von Personen außerhalb des Teams vorgenommen werden.

Im Rahmen dieser Bachelorarbeit soll ein Tool entwickelt werden, das es den Meetingteilnehmern ermöglicht, ihr Feedback während eines Meetings manuell einzugeben. Dieses Feedback wird dann in Dateien gespeichert, die zum Vergleich verschiedener Meetings untereinander genutzt werden können. Der Benutzer kann die Datei auch manuell an den Projektleiter senden, so dass dieser das Feedback des gesamten Teams zum Verlauf des Meetings vergleichen kann. Dies kann den Managern dabei helfen, Probleme, die in einem Team entstehen könnten, frühzeitig zu erkennen und geeignete Maßnahmen zu ergreifen, um sie zu vermeiden. Darüber hinaus können die Entwickler ihr Feedback in den verschiedenen Meetings vergleichen und sich einen besseren Überblick über den Verlauf ihrer Arbeit verschaffen.

Um dieses Tool zu evaluieren, haben 15 Teilnehmer, die in einem Software-Engineering-Kontext arbeiten, das Tool in einem Meeting getestet und eine Umfrage über ihre Erfahrungen und ihr Feedback bezüglich des Tools und darüber, was noch verbessert werden könnte, beantwortet.

Contents

1 Introduction **1**
- 1.1 Problem . . . 2
- 1.2 Solution Approach . . . 3
- 1.3 Objective of the thesis . . . 3
- 1.4 Structure of the thesis . . . 4

2 Foundations **5**
- 2.1 Sentiment Analysis . . . 5
  - 2.1.1 Applications . . . 7
  - 2.1.2 Example of usage . . . 8
- 2.2 Emotions . . . 10
- 2.3 Programming Language . . . 12
  - 2.3.1 Python . . . 12
  - 2.3.2 Framework and Libraries . . . 13

3 Related works **15**
- 3.1 Datasets . . . 15
- 3.2 Sentiment Analysis Tools in Software Engineering . . . 17
- 3.3 Meeting Analysis . . . 19

4 Implementation **23**
- 4.1 Concept . . . 23
  - 4.1.1 Stakeholders . . . 23
  - 4.1.2 Requirements . . . 23
  - 4.1.3 Use Cases . . . 25
- 4.2 Development . . . 25
  - 4.2.1 Start Meeting . . . 26
  - 4.2.2 Overview . . . 29
  - 4.2.3 Language and Help . . . 31

5 Evaluation **33**
- 5.1 Demographics . . . 33
- 5.2 Meetings and Sentiment Analysis . . . 34

5.3 Tool feedback 37

6 Discussion **41**
6.1 Answering the research question 41
6.2 Discussing the results 42
6.3 Threats to validity 43

7 Conclusion **45**
7.1 Summary 45
7.2 Outlook 46

Bibliography **47**

Chapter 1

Introduction

Sentiment analysis is a computational study of people's opinions, attitudes, and emotions toward an entity, which can be an individual, an event, or a topic [30]. In its basic usage scenario, sentiment analysis is used to classify written opinions as negative, neutral, or positive [27]. A recent systematic literature review about the development and application of sentiment analysis tools in software engineering demonstrated the frequent usage of these tools in a software engineering context [34]. These tools have been applied to a wide variety of data sources which include GitHub[1], Stack Overflow[2] and JIRA[3] [34].

Given the increasing complexity and size of software projects, software development is seen more as a team effort rather than as an one-person activity [24]. Meetings are an essential part of most software projects [46]. Due to the COVID-19 pandemic, many software developers were asked to switch their typical office-based working habits to a new working from home setting on short notice [45]. This meant that even the meetings would have to take place remotely through video calls. This makes it even harder to detect the mood of the participants in a meeting, since visual contact is therefore reduced. Much time and energy is devoted to meetings at work [52], aiming to accomplish goals such as information sharing, decision making, and problem solving [26]. If they are well organized and if the participants interact adequately, meetings are an efficient way to transport a lot of information in a short amount of time [37]. Schneider et al. [46] showed that proactive statements followed by supportive statements in a meeting, increase positive affect afterwards and since emotions are an inseparable part of human nature, which influence activities and interactions, they

[1] github.com
[2] www.stackoverflow.com
[3] https://www.atlassian.com/software/jira

can also affect task quality, productivity, creativity, group rapport and job satisfaction [8] [10]. For this reason, studying the interactions and the effects that meetings have on the team members, has an increased importance and can help in realising problems that may arise in their early stages.

1.1 Problem

The usage of sentiment analysis in meetings remains, to the best of knowledge, a relatively unexplored domain, despite some attempts being made [21] [13] [46]. The research from Herrmann and Klünder [13] confirms that sentiment analysis can be applied to meetings, but automatic sentiment analysis tools brings with them several problems [27] [20]. As previous research has shown, we can not expect 100% accuracy from sentiment analysis tools, since even humans are often not able to agree about the sentiment of given a sentence [27]. An "out-of-the-box" usage of sentiment analysis tools leads to poor accuracy when they are applied in a different context from the one which they have been designed and/or trained for [20]. Some of the datasets used for sentiment analysis tools in software engineering are created from data gathered from websites such as JIRA[36], GitHub[11] and Stack Overflow[33]. Although it was mentioned that sentiment analysis can be used in meetings, we can not expect that the usage of existing sentiment analysis tools in this context can give a fully correct result, since there are no tools trained with datasets from this domain. Other problems related to the usage of automatic sentiment analysis tools may be caused by the usage of irony [17] [31]. Irony is a figure of speech in which the intended meaning is the opposite of the literal meaning and can therefore play the role of a polarity reverser, with respect to the words used in the text unit [4]. The same can be said for sarcasm. Sarcastic sentences with or without sentiment words are hard to deal with, e.g., "What a great car! It stopped working in two days." [28]. If the tools are not trained to also recognize the usage of irony and sarcasm, this might lead to a false evaluation of the words. Constant interruption can also be a factor which leads to a non-optimal flow of the meeting and therefore can create a negative sentiment for the participant. This must also be taken in consideration by an automatic tool.

1.2 Solution Approach

As a solution to the problems discussed above, as part of this bachelor thesis, an application is to be developed, which will be named *Sentiment-Dashboard*, where the meeting participants can give their feedback manually about how the meeting is going and how they are feeling. The tool will also be consisting of a graphical user interface (GUI). The feedback is than saved and displayed through different graphics at the end of the meeting. The graphics will show an overview of the mood of the participant during the course of the meeting and also the range of emotions that the user inputed throughout this meeting. The feedback can also be saved in files stored locally, which can than be shared manually with other team members or project managers to compare and get an overall feeling about the sentiment of the participants in the meeting. The user can also use this feedback to compare their older meetings and to gain a better view of how the meetings progressed. Since the feedback will be given manually by the user, the risk of a general misinterpretation from the sentiment analysis tool will be none. The evaluation will be entirely as percepted from the users. This might differ from user to user, because not everyone can have the same evaluation for a statement, but this can be a positive aspect for the managers, since it offers different perspectives from different team members. The usage of this tool, also eliminates the problem that sentiment analysis tools have when applied in an unknown context as discussed above. The sentiment will be saved exactly as given by the user, and will not go into further editing which might change the original values.

1.3 Objective of the thesis

The objective of this thesis is to provide a tool, in which the participants of a meeting, in a software engineering context, can give their feedback about the mood of the meeting in real time while the meeting is taking place and also to evaluate this tool. After the meeting, each participant will give its feedback about the tool by completing a survey. Their answers will be used to make an evaluation about different aspects of the *Sentiment-Dashboard* and their feedback will also serve as a basis for further improvement of the tool. Following research questions will be answered:

1. Is there a need for the usage of the *Sentiment-Dashboard* in meetings?
2. Can the tool be used during meetings without disturbing the work flow and the concentration of the users?

1.4 Structure of the thesis

This thesis is structured as follows:
In Chapter 2, the fundamentals of sentiment analysis and emotions will be discussed, as well as the used tools to design the GUI.
In Chapter 3, works related to the usage of sentiment analysis in a software engineering context will be discussed.
In Chapter 4, the implementation of the *Sentiment-Dashboard* will be presented. This chapter is divided in two main parts: The concept section and the development section.
The evaluation part of this thesis will be covered in Chapter 5 where the questionary used to gather feedback from the participants relating their usage of the tool will be discussed and the results will be presented.
Chapter 6 covers the discussion part of the thesis where the results with be interpreted.
The conclusion of this thesis will be discussed in Chapter 7.

Chapter 2

Foundations

In this chapter, the fundamental knowledge needed for this bachelor thesis will be discussed. The reader of this work is expected to have at least intermediate knowledge in the software engineering field and to be familiar with basic programming concepts as well as with user interface terminology. This chapter will present definitions of sentiment analysis together with some works in this field to use as an example. Furthermore, an overview on emotions and their classification will be presented. Lastly, the programming tools which were used to develop the graphical user interface will be discussed shortly.

2.1 Sentiment Analysis

Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [28]. It includes several application fields. An overview of these fields from Pozzi et al. [41] is shown in Figure 2.1.

The most common application of sentiment analysis is in the area of reviews for consumer products and services [9]. Acquiring public and consumer opinions has long been a huge business itself for marketing, public relations, and political campaign companies [28]. Businesses and organizations conducted surveys, opinion polls and focus groups to find out their consumers opinions about their products and services, while with the explosive growth of social media, the aforementioned methods are no longer necessary since there is an abundance of such information publicly available [28]. The web provides a universal platform for information exchange, where people can show their personality and views, record and share their feelings, express their like/dislike on products,and so on [7]. Opinion mining helps in this case people to better use the available information and support their decision on diverse issues [7]. For example, when a person wants to buy a product online they will typically start by searching for reviews and opinions written by other people on the various offerings [9] in order to make a better decision regarding the product.

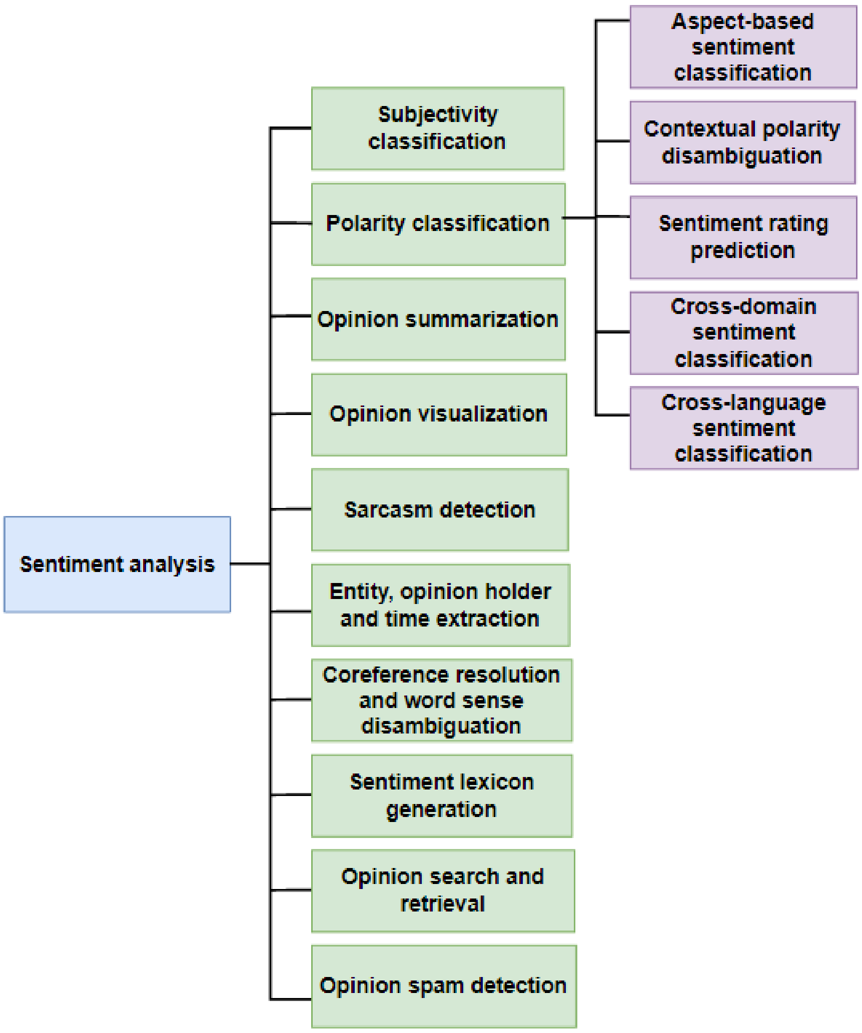

Figure 2.1: An overview of Sentiment Analysis tasks by Pozzi et al. [41]

The most important indicators of sentiments are sentiment words, also called opinion words [28]. These are words that are commonly used to express positive or negative sentiments like for example, *good*, *wonderful*, or *amazing* are positive sentiment words, and *bad*, *poor*, or *terrible* are negative sentiment words and a list of such words and phrases is called a sentiment lexicon [28]. A sentiment word may have opposite evaluation in different application domains [28] and this is a difficulty that some sentiment analysis tools have, as shown by Obaidi and Klünder [34]. For example, the word "kill" has a very negative sentiment in the real world domain, but in a software engineering context, "to kill a process" is a phrase with a neutral sentiment [32]. This is why several researchers have prepared software engineering specific datasets in order to evaluate words and phrases correctly in this context [36] [2] [33].

Different tools have been created to perform sentiment analysis on different sources. The most important approaches taken by these tools are the machine-learning approaches and dictionary based approaches. Dictionary/lexicon-based sentiment analysis is typically based on lists of words with predetermined emotional weight [38]. Examples of such dictionaries include the General Inquirer (GI) [53] and the "Linguistic Inquiry and Word Count" (LIWC) software [40]. Both lexicons are built with the aid of experts that classify tokens in terms of their affective content (e.g., positive or negative) [38].

The machine learning approach is used for predicting the polarity of sentiments based on trained as well as test data sets [3]. It applies the ML algorithms and uses linguistic features [3]. Machine learning algorithms typically predict sentiment based upon occurrences of individual words, word pairs, and word triples in documents [50]. The main advantage of this method is the ability to adapt and create trained models for specific purposes and contexts. Its main disadvantage on the other hand, is the low applicability of the method on new data since the availability of labeled data that could be costly or even prohibitive [3]. This approach may also perform poorly on informal text because of spelling problems and creativity in sentiment expression, even if a large training corpus is available [50].

2.1.1 Applications

Sentiment analysis has been used both inside the software engineering context and also outside of it. Its applications outside software engineering, have spread to almost every possible domain, from consumer products, services, healthcare, and financial services to social events and political elections [28]. Chen et al.[7] used sentiment analysis to study political standpoints. Liu et al. [29] proposed a sentiment model to predict sales performance. Data gathered from social media has been used in different sentiment analysis projects such as for example, Tumasjan et al. [51] used Twitter sentiment to predict election results. O'Conner et al. linked Twitter sentiment to public opinion polls [35]. Thelwall et al. [50] developed *SentiStrength* which can be used to evaluate the sentiment on short text based on data comments gathered from MySpace.

In a software engineering context, some of the applications are for example *SentiCR* from Ahmed et al. [2] which was used to evaluate code reviews. Calefato et al. [6] proposed *Senti4SD*, a sentiment analysis classifier which was used to classify posts from Stack Overflow. Herrmann et al. [13] [14] developed *SEnti-Analyzer* which was used to detect the sentiment in a meeting by recording the meeting, transcribing it and than applying sentiment analysis to the resulting text. Islam and Zibran [17] proposed *SentiStrength-SE* which was based on the original *SentiStrength* from Thelwall et al. [50] but adapted it to the software engineering context by training the tool on GitHub commit messages. These tools will be better described in Chapter 3.

2.1.2 Example of usage

In order to better demonstrate the usage of a sentiment analysis tools, an example of such a tool will be described. For this example, *SentiStrength* from Thelwall et al. [50] is chosen, since it is the sentiment analysis tool with the most application in software engineering papers when compared to similar tools, as proved by Obaidi and Klünder [34]. This tool also served as a base for the creation of *SentiStrength-SE* from Islam and Zibran [17] which adapts *SentiStrength* to software engineering. Also it served as a base of comparison for other sentiment analysis tool in software engineering such as *Senti4SD* by Calefato et al. [6]. *SentiStrength* was presented for the first time in 2010 and it uses a dictionary of sentiment words with associated strength measures and exploits a range of recognized nonstandard spellings and other common textual methods of expressing sentiment [50]. It was developed through an initial set of 2,600 human-classified MySpace comments, and evaluated on a further random sample of 1,041 MySpace comments, that were different from the comments used in the development phase and were classified by three people. The selected comments were judged on a 5-point scale as follows for both positive and negative sentiment: (no positive emotion) 1–2–3–4–5 (very strong positive emotion), and (no negative emotion) 1–2–3–4–5 (very strong negative emotion) [50]. Some examples of texts and their respective evaluation from each of the coders are listed below:

- omg my son has the same b-day as you lol (scores: positive: 4, 3, 1; negative: 1, 1, 1)
- What's up with that boy Carson? (scores: positive: 1, 1, 1; negative: 3, 2, 1)

The text "omg my son has the same b-day as you lol" has received a positive rating of respectively 4, 3 and 1 from each of the 3 coders and a negative rating of 1 from each of them. This positive score would be the rounded mean of the positive scores, in this case 3. The negative score would be 1. The same is done with every other statement.

The core of the algorithm is the sentiment word strength list, which is a collection of 298 positive terms and 465 negative terms classified for either positive or negative sentiment strength with a value from 2 to 5 [50]. Negative sentiment was predicted with an accuracy of 72.8% when compared to the labelled dataset. *SentiStrength* reached a level of accuracy of 60.6% for positive sentiment, which is a moderate level of accuracy and is similar to the degree of agreement between the human coders [50]. The main reason for *SentiStrength*'s relative success seems to be procedures for decoding nonstandard spellings and methods for boosting the strength of words, which accounted for much of its performance [50], such as the aforementioned word strength list.
The authors later also presented an improved version of the algorithm, *Senti-Strength 2* [49], which had an increased accuracy.

Since 2014, *Senti-Strength* is also available as a web application [1]. The user can enter a string with a maximum of 120 characters and it receives as an output the sentiment of the given string together with an explanation which shows which word influenced the decision. An example of usage of this tool is shown in Figure 2.2. The inputed text was "I really liked the movie but the cinema was very dirty.".

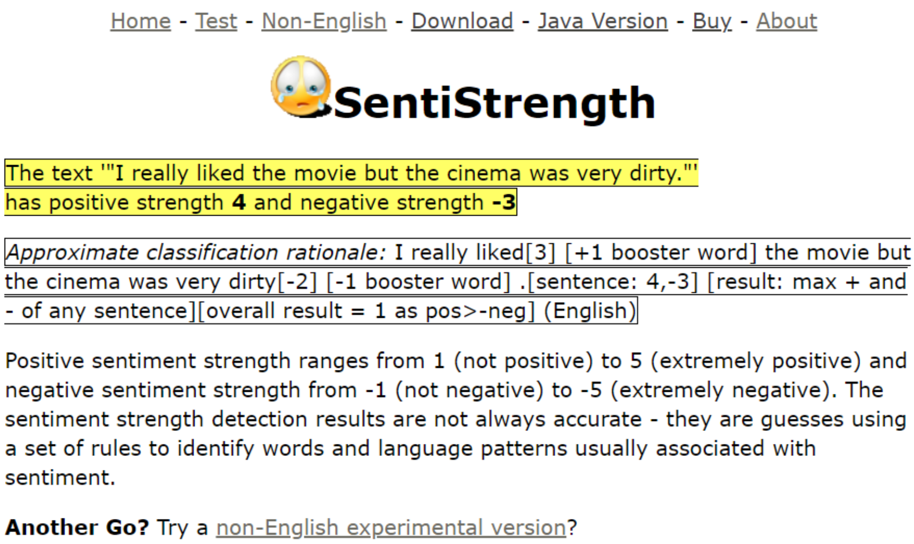

Figure 2.2: Example output of Senti-Strength

[1] http://sentistrength.wlv.ac.uk/

2.2 Emotions

Emotions have been studied in multiple fields, e.g., psychology, philosophy, and sociology [28]. Viewing emotion knowledge from a prototype perspective suggests why it has been difficult for psychologists to agree on a number of fundamental issues concerning emotion [47]. The traditional view of an emotional episode is too nebulous and implicit to characterize precisely, and psychologists have developed different versions [44]. Liu [28] defined emotions as our subjective feelings and thoughts. Russell [44] captured assumptions from James [18] and Rachman [43] about the nature of the emotion, in a graph which depicts a causal chain centred on the emotion. The event causes the emotion, which causes all its various "manifestation" [44]. This chart is shown in Figure 2.3.

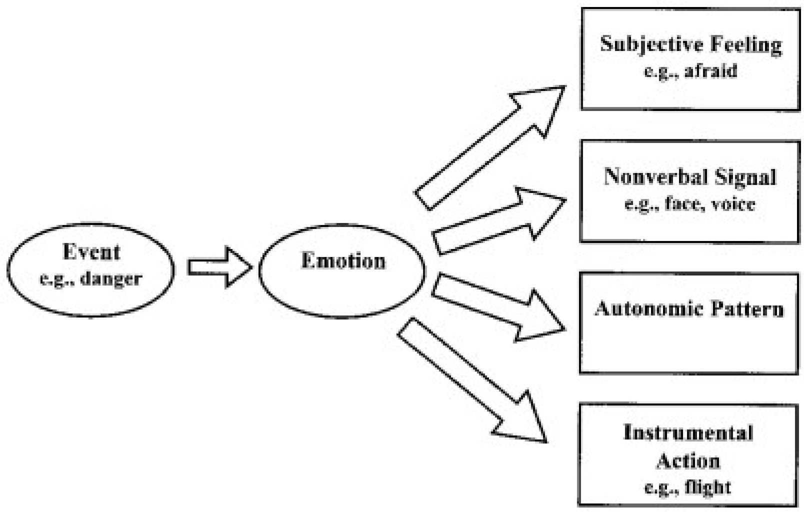

Figure 2.3: The traditional view in which emotion is an event that mediates between an antecedent and its various manifestations, by Russell [44]

Opinions are multidimensional semantic artifacts [5]. When people are exposed to information regarding a topic or entity, they normally respond by developing a personal point of view or orientation which reveals how the opinion holder is polarized by the entity [5]. Most opinion mining algorithms attempt to identify the polarity of sentiment in text: positive, negative, or neutral [50]. In the context of this bachelor thesis, when the user is asked how they are feeling, they can choose an option from a 5-Point Scala ranging from *Very bad* to *Very good*. These are equivalent to the polarities very negative and very positive which allow the user to further emphasize the strength of their opinion. Furthermore, the user can select from a range of emotions which will be discussed below. Bravo-Marquez et al. [5] proposed an approach for boosting Twitter sentiment analysis by combining aspects such as opinion strength, emotion and polarity indicators. Their results show that the composition of these features achieves significant improvements over single approaches [5]. Calefato et al. [6] also used a similar approach in combining emotions with polarity when labelling their Stack Overflow dataset by explicitly requested the coders to provide a polarity label, according to the specific emotion detected.

Shaver et al. [47] argued that English and many other languages contain hundreds of terms that seem to refer to emotions and it is obvious that some of the emotional states referred to are closely related (e.g., anger, annoyance, hatred, and rage), whereas others (e.g., contentment and despair) are quite distinct. To support this claim, they conducted a study to explore the hierarchical organization of the emotion domain. 135 terms were selected to be grouped according to their similarity with one another. 100 students in introductory psychology courses (50 men and 50 women) participated in the similarity-sorting phase of the study [47].

For each subject, a 135x135 co-occurrence matrix was constructed, with 1 indicating that two terms were placed in the same category and 0 indicating that they were not [47]. These matrices were added across the 100 subjects to form a single 135x135 matrix in which cell entries could range from 0 to 100, representing the number of subjects who placed a particular pair of words in the same category. Among the few highest scoring words were love, joy, surprise, anger, sadness, and fear [47]. The authors had reservations about the surprise cluster since it was smaller and less differentiated in comparison to the others and therefore did not examine it further in their article [47]. It seems possible, given the results, that all of the terms in the emotion lexicon refer in one way or another to a mere handful of basic-level emotions [47].

Parrott [39] further developed this approach by dividing emotions in primary, secondary and tertiary emotions. The primary emotions from Parrott [39] are identical with the basic emotions from the classification from Shaver et al. [47] except from the inclusion of the term surprise in the primary emotions. A simplified graphic demonstrating the emotion model from Parrott [39] is shown in Figure 2.4 [2]. In the scope of this bachelor thesis, the primary emotions defined from Parrott will be used as basis for the user to choose as input. The only difference is that the emotion "Surprise" will be divided into two categories: "Negative Surprise" and "Positive Surprise". This is done to better define this emotion and to allow the users to better express themselves.

[2] www.researchgate.net Last visited 11.06.2022

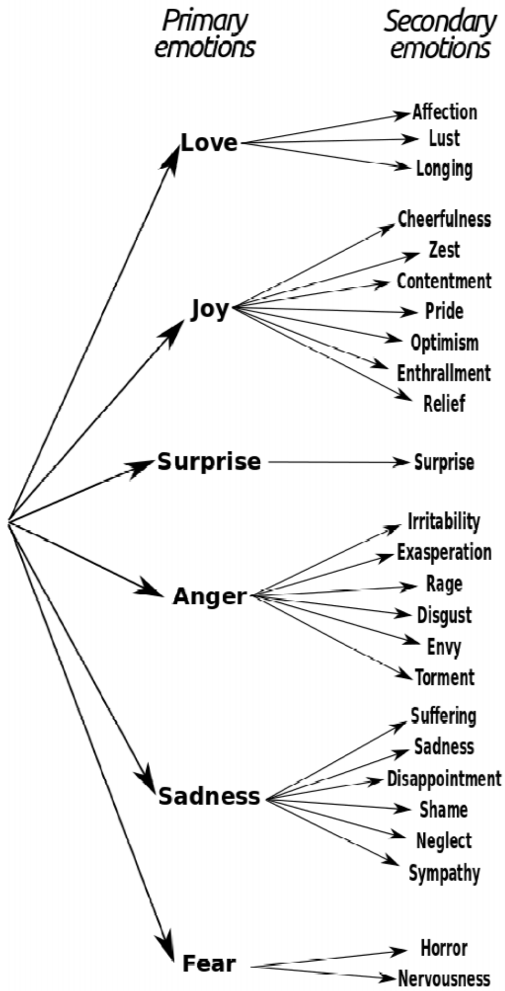

Figure 2.4: First two layers of Parrot's emotion classification [39]

2.3 Programming Language

2.3.1 Python

For the programming of the GUI in this bachelor thesis, the *Python* programming language was used. *Python* [3] is an interpreted, high level language that was released in 1991 from Guido van Rossum as a successor to the *ABC* programming language. It supports most of programming main paradigms such as object-oriented-programming, structured-programming etc. It has a very compact style and its syntax resembles the English language which makes it very easy to learn even for novice programmers. This, along

[3] www.python.org/about/

side with the support of a huge amount of frameworks and libraries in different domains, makes it one of the most used programming languages at the moment. According to GitHut 2.0 [4], *Python* was the language with the most pull requests on GitHub in 2021.

2.3.2 Framework and Libraries

Tkinter [5] is the standard *Python* interface to the Tk GUI toolkit . It is the only framework that is built into the standard library in Python. It offers support for a number of widgets such as buttons, labels, text fields etc. Widgets can be organized inside of frames and windows to allow the creation of specific program-flows according to the needs of its user.

Matplotlib [6] is a *Python* library which allows the creation of different plots such as pie chars, bar charts, line charts etc. It has a compact syntax which allows the user to plot a graphic with the provided data with just a few lines of code. It can be also integrated in a *Tkinter*-built GUI using for example the Figure widget to allow the display of the charts inside the graphical user interface.

CSV [7] helps the user in dealing with .csv files by providing methods to simplify the writing and reading of .csv files inside a Python program.

Datetime [8] is another *Python* library which allows the user to access and use the actual date and time. This library was used in combination with the *csv* library to save the .csv files containing the meeting data with the current date and time as file name, to allow for a better ordering of the files.

NumPy [9] (Numerical Python) is an open source Python library that's used in almost every field of science and engineering. It is the universal standard for working with numerical data in Python. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

[4] www.madnight.github.io
[5] www.docs.python.org/3/library/tkinter.html
[6] www.matplotlib.org/
[7] www.docs.python.org/3/library/csv.html
[8] www.docs.python.org/3/library/datetime.html
[9] www.numpy.org/doc/stable/

Chapter 3

Related works

This chapter will discuss some works related to this bachelor thesis. This includes sentiment analysis tools used in software engineering. Furthermore, some datasets used to train these tools will be presented. Finally, some works related to meeting analysis, together with their methods and results will also be discussed in this chapter.

3.1 Datasets

Research has shown that using sentiment analysis tools out of the context they were created for, leads to unreliable results [20]. In order to achieve better results in exploring sentiment analysis in a software engineering context, several dataset have been created by researchers by manually labeling context specific data related to software engineering.

Ortu et al. [36] have contributed to the research of affects on software artifacts by providing a labeling of emotions present on issue comments on JIRA. They labeled manually 2000 issue comments and 4000 sentences written by developers with emotions based on Shaver's emotions model [47] such as love, joy, surprise, anger, sadness and fear. The objective of their paper was to address the lack of data in affects associated to software artifacts, by providing manual labeling of emotions present within issue comments [36]. This data is highly valuable for investigating the impact of sentiment on software development and also for training tools for sentiment detection. In conclusion, since this dataset hosts data like story points, sprints etc., related to Agile development, it can be used also for research related to Agile practices [36].

Ahmed et al. [2] created a training dataset by using 2000 code review comments from the repositories of 20 popular Open Source projects which they manually labelled. They than used this dataset on existing sentiment analysis tools, which delivered a poor performance [2]. This motivated the authors to create ***SentiCR*** [2], a supervised learning based sentiment analysis tool. To create the training dataset, code reviews from 20 popular Open Source Projects were mined [2]. Afterwards the authors inspected the comments manually to remove comments made from bots. 2000 review comments were than randomly selected, each with at least 50 characters. The randomness ensured that all types of comments could be included in this dataset. Three of the authors labelled each of the comments individually as "positive", "neutral" or "negative" based on their own perception [2]. The raters had a consensus on 1239 comments and the rest was than discussed until a final decision could be taken. Undersampling was than used to remove 400 neutral comments since the dataset was highly unbalanced. The remaining 1600 labeled comments serve as a valid dataset since it also satisfies Thewall's recommendation of minimum 1000 labeled text for sentiment training [48].

Novielli et al. released a dataset containing 4800 posts from Stack OverFlow in the form of question, answers and comments [33]. For each post in the dataset, they distribute both the set of individual annotations provided by the raters and also the gold label obtained by applying majority of voting [47]. Their dataset contributes to the building of a shared corpus of annotated resources to support research on emotion awareness in software development [47]. The annotation guidelines were based on the framework by Shaver et al. [47]. The dataset from Novielli et al. complements the effort made by Ortu et al. [36], towards the construction of a gold standard dataset to support the study of emotions in software engineering [47].

What separates the aforementioned works from the ***Sentiment-Dashboard***, is the fact that each of them uses data gathered from a specific source of information and manually labels the statements to a specific polarity. From here, the problem of subjectivity of manual labeling arises. This is a problem which is mentioned in many papers, as Obaidi and Klünder [34] have shown. For the tool developed in this thesis, the ***Sentiment-Dashboard***, the dataset is the input which is given directly by the user, since they input their own sentiment and emotions directly to the tool. In this way, the subjectivity from manual labeling from an external rater is avoided.

3.2 Sentiment Analysis Tools in Software Engineering

In this section, sentiment analysis tools which were using in a software engineering context will be presented along with their respective approaches and results.

Islam and Zibran [17] created *SentiStrength-SE* which was the first sentiment analysis tool developed especially for the software engineering domain. This tool was build on top of the existing tool *SentiStrength* from Thelwall et al. [50] which was discussed previously on Section 2.1.2. The authors were also the first to expose the challenges that sentiment analysis tools face in software engineering [17]. They did this by using *SentiStrength* on the dataset from Ortu et al. [36]. Out of the 392 issue comments that were selected, 151 comments were evaluated as incorrect. Islam and Zibran analysed the causes of the misinterpretation for these comments and it was noticed that domain-specific meaning of words was the most frequent cause of difficulties for the *SentiStrength* [17]. The authors tackled this problem by developing a domain dictionary for software engineering texts. To do this, a large dataset which was studied in the work of Islam and Zibran [16] consisting of 490 000 GitHub commit messages was used. When compared to the original *SentiStrength*, *SentiStrength-SE* achieved a better precision score (73% vs. 61.69%) [17]. Nevertheless, even in this case, 100% accuracy can not be expected [17].

After creating the training dataset from the code review comments, Ahmed et al. [2], evaluated seven sentiment analysis tools using their dataset. The tools had a poor performance, especially in recognising negative comments [2]. This motivated the authors to implement *SentiCR*, a supervised learning based sentiment analysis tool. Before applying sentiment analysis on it, the data first underwent an 8-step pre-processing [2]. Contractions (shortened form of one or two words) were expanded to avoid misinterpretation. URLs, Code Snippets and Stop-words were removed since they do not provide any sentiment in them. Emoticons were substituted by words since they have an influence in expressing sentiment. Negation words (e.g. not, never, nothing) were preproccesed in order to avoid misclassifications for comments including them. Word stemming was also applied to parse text into a list of words and finally, a feature vector was generated. For this, the authors computed TF-IDF (Term Frequency - Inverse Document Frequency) to extract the features of the classification and than used this feature vector to train their classifiers. Eight supervised algorithms used for sentiment analysis, including Gradient Boosting Tree, Random Forest and Naive Bayes, were than evaluated. Gradient Boosting Tree reached the highest accuracy with 83% [2].

Calefato et al. [6] proposed another sentiment analysis classifier named

Senti4SD. To train and test *Senti4SD*, the authors built a gold standard dataset of 4423 posts mined from Stack Overflow. *Senti4SD* was trained using Support Vector Machines [6]. To further evaluate the tool, its performance was also compared with *SentiStrength* [50], which is a similar approach with the one Islam and Zibran took when developing *Senti-Strength-SE* [17] as discussed above. Looking at the performance of *Senti4SD*, a 19% improvement in precision for the negative class and a 25% improvement in recall for the neutral class was observed when compared with *SentiStrength*. What is noteworthy is that *SentiStrength* actually outperformed *Senti-Strength-SE* when used to classify the polarity of the posts in the above used Stack Overflow dataset [6]. This comes to show that even software engineering based sentiment analysis tools, might have contradicting results between one another, as research has also shown [15] [19].

What the above mentioned tools have in common is that they use an existing dataset to train the tool and than apply the tool to a different use case. In all these tools, the evaluation of the sentiment is done automatically from the tool based on the used algorithms and the dataset that was used to train each tool. This approach has of course benefits, such as requiring little effort from the user, but also downsides such as the tools not agreeing with each other [19] [15] or the limited overall accuracy, as we discussed previously. As mentioned above, the *Sentiment-Dashboard* allows the users to input their feedback directly, avoiding the aforementioned problems that may arise from the usage of automatic tools.

3.3 Meeting Analysis

In this Section, some works which are related meeting analysis will be presented. This includes the *act4teams* [21] and *act4teams-SHORT* [22] coding schemes, a method to measure meeting succes by Prenner et al. [42] and a sentiment analysis tool specially developed for analysing meetings by Herrmann and Klünder [13].

Kauffeld and Lehmann-Willenbrock [21] analysed a total of 92 team meetings using the *act4teams* coding scheme. The *act4teams* coding scheme was designed for analysing real team meetings in organizations [21]. It distinguishes four types of team interaction: problem-focused, procedural, sozio-emotional, and action-oriented communication. Problem-focused communication is directly related to understanding the issue, finding appropriate solutions, and evaluating those solutions. Positive procedural communication concerns statements that are aimed at structuring and organizing the discussion. Sozio-emotional statements capture the relational interaction that occurs in teams. Action-oriented statements describe a teams' willingness to take action to improve their work. Each of these interaction types is subdivided into several divisions. In turn, each division is subdivided into a set of categories culminating in a total of 44 observation categories [21]. A total of 92 teams from 20 medium-sized organizations were examined. Interaction data were collected during regular team meetings. Any particular statement from these meetings was assigned manually to exactly one *act4teams* category. Some interesting findings from this evaluation were that teams that showed more functional interaction, such as problem-solving interaction and action planning, were significantly more satisfied with their meetings [21]. Also better meetings were associated with higher team productivity [21].

Scheider et al. [46] also analyzed the first team meeting of 32 student development teams using the *act4teams* [21] coding scheme. Each team consisted of 3 to 5 students, leading to a total number of 155 participants. Results of this study [46] showed that the rate of proactive statements within a meeting does not significantly influence positive group affect after the meeting. Also, positive group affective tone is triggered by proactive statements via support [46]. Finally, the probability of a supportive statement is significantly higher when a proactive statement has been made [46].

Klünder [22] proposed a shorter versioned of act4teams called act4teams-SHORT. It is based on two main principles: selective and discrete coding of the data which means that not the whole meeting will be coded and also, only important statements will be coded [22]. This scheme has only

9 categories (as of Version 4.0) which are better separated from each other [22]. This reduces cognitive effort for distinguishing the categories from one another [22]. These categories are: problems, solutions, connections and networking, destructive behaviour, proactive behaviour, collegial behaviour, methodical-structured behaviour, information and knowledge transfer and others [22].

Prenner et al. [42] developed a feedback method to measure the success of a meeting and a tool to apply this method. They defined meeting success based on three aspects: effectiveness, efficiency, and satisfaction. To gather feedback from the meeting participants regarding their perception about these aspects, the authors defined questions for each of the aspects. This method was tested on two meetings. The participants answered the questions at the end of their meeting. The results were compared with those from the *act4teamsLight* method [12] and both methods showed similar tendencies [42].

Herrmann and Klünder created a tool which can transcribe the audio file of a recorded meeting and than it applies sentiment analysis to the resulting text [13] [14]. The goal is to analyze the mood of the team in real-time and at the end of the meeting, provide an overview of the overall mood which can than be useful to project leaders, for example, to gain a direct feedback about the meeting [13]. In order to achieve this goal, the authors combined the approaches of sentiment analysis tools with automatic speech recognition in order to analyse the mood of the team in real-time based on their verbal communication. The approach taken consists of two main steps: Firstly the meeting is transcribed from audio to text and secondly, the application of the sentiment analysis tool SEnti-Analyzer [23] to the text. The processing steps from the raw audio to the sentiment prediction is shown in Figure 3.1

The transcript is generated using the Mozilla Deepspeech framework from Agarwal and Zesch [1]. The text is then fed to the *SEnti-Analyzer* which interprets the results and presents them to the user. The *SEnti-Analyzer* tool was tested on a student software project meeting at the Leibniz University Hannover. Their recorded meeting was transcribed and classified manually in order to get training data for the *SEnti-Analyzer*. Together with another pre-recorded meeting from another iteration of the software project, which underwent the same procedure, made up the training set for this tool, which consisted of 712 manually transcribed and labelled statements.

The trained model performs and generalizes well and has a more divergent distribution of the sentiment classes in comparison to the training set [13]. The meeting was classified as neutral to positive which corresponds with the feedback given by the meeting participants after the meeting ended. To further validate the results, the researchers picked 50 from 140 statements and manually classified them. Interestingly, the tool resulted in a more

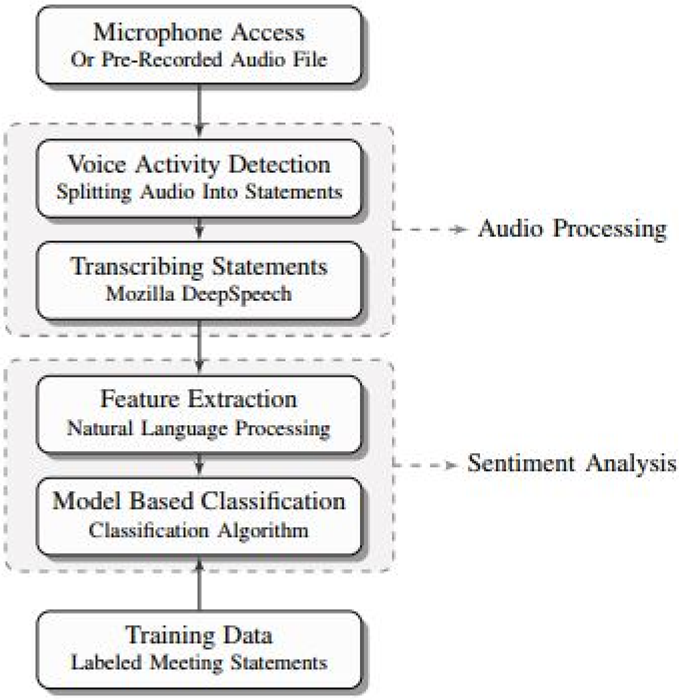

Figure 3.1: Simplified processing pipeline of the SEnti-Analyzer [13]

diversified classification than the human tester. The total calculated Fleiss' κ -value is 0.56. This is considered as an upper-moderate agreement according to the scale provided by Landis and Koch [25]. This research provided some interesting insights since it showed that automatic speech recognition system could be applied successfully to already existing sentiment analysis tools. Also the tool provided a moderate agreement according to Fleiss' κ-value between the automatically produced result and the manual classification of the human observer [13]. Also it showed that sentiment analysis could be applied to meetings and laid the ground for future research in this area.

The tool presented from Herrmann and Klünder [13], although it uses a different approach for the inputed data by using an audio file, its similar to the tools mentioned in Section 3.2 3.2. Therefore, it has the same limitations as the other tools in terms of accuracy and subjectivity along with differentiating from different speakers and dealing with interruptions, sarcasm and irony. The *act4teams* coding scheme uses a different method from the ones mentioned before, since it does not automatically process the input and gives a result about the sentiment, but rather allows the raters to code each valuable statement into a given category of interactions. This approach can also be vulnerable to subjectivity problems since the raters may not agree on all statements. Also this a very lengthy process, which also takes a lot of effort to train the raters and to then apply for a meeting. The *Sentiment-Dashboard* on the other hand, requires little to no effort to learn and can be easily used in any number of meetings with ease. The method presented by Prenner et al. [42] focuses on measuring the meeting success, while the tool presented in this thesis, focuses on gathering feedback about the mood of the participants during the meeting.

Chapter 4

Implementation

In this chapter the concept and the implementation of the *Sentiment-Dashboard* will be presented. In the "Concept" section, the stakeholders, the main requirements for the tool and the two main use cases will be discussed. In the "Implementation" section, the methods used to implement the tool will be presented along with the features of the tool and some examples.

4.1 Concept

4.1.1 Stakeholders

The stakeholders for this project are the meeting participants, which can be developers, and the project managers. The developers will use this tool the most during meetings to express their emotions about the meeting. The project managers will mostly use this tool to check the overviews of the meetings by gathering the feedback from the developers. They can also use it themselves during their meetings to gather their own feedback and possibly compare it with the developers and other team members. The project managers are the most important stakeholders, since they want to gather the feedback from their team and from there consider what measures can be taken to improve the mood of the team or the meeting interactions, if the feedback from the team has been mainly negative.

4.1.2 Requirements

The main context in which the tool is expected to be used are meetings. The *Sentiment-Dashboard* should offer a possibility for its users to input their emotions during the meeting and than also allow the users to compare the results from different meetings. This can help in gathering important feedback and early realising problems that may arise. Meetings nowadays can take place in person but also remote, via video calls. In the development of this tool, both scenarios should be taken in consideration. Since during a meeting, the users are usually engaged in conversation, it is necessary for this tool to be easy to use in order to not be distracting for the users and make them lose important discussion points. Also, this tool should be easy to learn so that the users do not lose much time before the meetings to learn it. In the case in which the meeting is taking place remotely through a video call, the user normally has to see the other meeting participants in the screen and often also search for information or documents during the meeting or even present something. By keeping this scenario in mind, it was important that the tool should not occupy the whole screen and therefore block important interactions. Nevertheless, every important information inside the tool should be clearly visible. The user should also be able change the language of the tool between English and German, in order for it to be more accessible.

Through the *Sentiment-Dashboard*, the users should be able to input both their mood polarity as well as their emotions. The user should be asked for their input before, during and after the meeting. This way, a comparison of the moods before and after the meeting can be made, in order to identify the effect that the meeting might have had on the user. During the meeting, the amount of times that the user can input their feedback is not limited. The input should be easy for the user and the options should be distinguishable from one another, so that user error can be minimized. After the user has entered their feedback and ended the meeting, an overview of their inputed polarities and emotions should be displayed. This overview should be displayed through charts which contain the most important information. The user should have the possibility to save each chart, for later comparison between meetings.

Apart from the usage during a meeting, the tool should also offer the user the possibility to compare different past meetings. This can be the case for project managers comparing the mood from different team members, or for the team members themselves, comparing their mood in different meetings they had. Therefore, the meeting information should be stored locally to allow the user to import it again inside the tool for the comparison. This comparison should be made through charts displaying the most important information from the meeting, such as the mood and emotions. The information from the different meeting files should be displayed in a common graph in order to better acknowledge the differences between meetings. Each chart should be savable.

4.1.3 Use Cases

The main use case of the *Sentiment-Dashboard* is when the user takes part in a meeting and records their emotion. The actor in this case is the meeting participant. The user opens the application and in the home screen presses the button to start the meeting. After that, the user is asked to choose the actual mood, before the meeting starts. After a mood is selected, the user is taken to the meeting frame in which the user can continuously input their emotions and their mood. This happens until the user presses the button to end the meeting. After the meeting has ended, the user is asked about their mood again. The input methods are the same as when the user was asked before starting the meeting. Following the input of the mood after the meeting, the user is presented to a graphic which shows the moods of the user during the meeting, together with the timestamp for each input. The user presses the button to go to the next graph, which shows a comparison of the mood before, during and after the meeting. The mood during the meeting is the numerical average of all inputs during the meeting. The user presses again the button to go to the next graph. A graph is displayed which shows all emotions that the user has entered during the meeting. The user selects to either save the meeting data or not and than exits the app.

The other important use case is when the user wants to see an overview of past meetings. The actor in this case is a previous meeting participant. The user selects the corresponding option to open the overview menu. There, the user can upload the files they want to inspect. After selecting the needed files, the user presses the button to go to the overview of the files. Afterwards, a graph is shown displaying the mood during each selected meeting in a shared graph. The user presses the button to go to the next graph and there, a graph displaying an overview about the selected emotions from each meeting is displayed. The user presses the button to go to the next chart. An overview is shown of the mood before, during and after the meetings, for each meeting. From here the user can acknowledge if the meeting had an impact on the mood afterwards. The user exits the app.

4.2 Development

The Sentiment-Dashboard was developed using Python and the TKinter framework [1], which offer a range of useful tools for the development of a graphical user interface, as discussed in Section 2.3. The home screen of the Sentiment-Dashboard has 5 possible options to choose from, as displayed in Figure 4.1.

Each of these functions will be discussed in more detail below. For the logo

[1] www.docs.python.org/3/library/tkinter.html

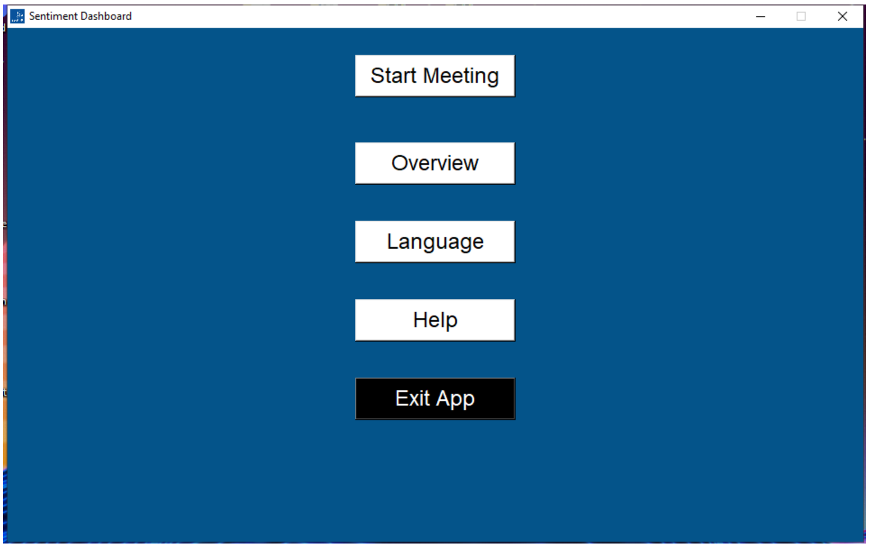

Figure 4.1: Home menu of the Sentiment-Dashboard

of the application, the logo of the Leibniz University of Hanover was used. The background colour was also selected to be similar to the blue used in the logo. The dimensions of the window (950 x 570 pixels) were selected so that the application would not take up the whole screen and therefore possibly limiting the users productivity during a meeting.

4.2.1 Start Meeting

In the Start Meeting section, the user can start recording their input about a meeting. They are first asked how they feel before starting the meeting. They can choose from 5 options: "Very Bad", "Bad", "Neutral", "Good" or "Very good". Each of this options has a different colour, with the negative polarities having nuances of the red colour and the positive polarities having nuances of the green colour. The neutral button has a yellow color which is to be seen as something "in-between" the red and the green.

After pressing one of the buttons to input their polarity before the meeting, this input is saved and the user is than directed to the meeting frame, as shown in Figure 4.2.

Here the user can input their sentiment about the meeting, selecting from the same Scala as mentioned above and also select from emotions that they are feeling. The emotions are "Anger", "Sadness", "Fear", "Negative Surprise", "Positive Surprise", "Joy", "Love". These emotions are the primary emotions from Parrott's emotion classification [39]. The only difference here is the subdivision of the "Surprise" emotion into "Positive surprise" and "Negative surprise". This was made in order to give the user

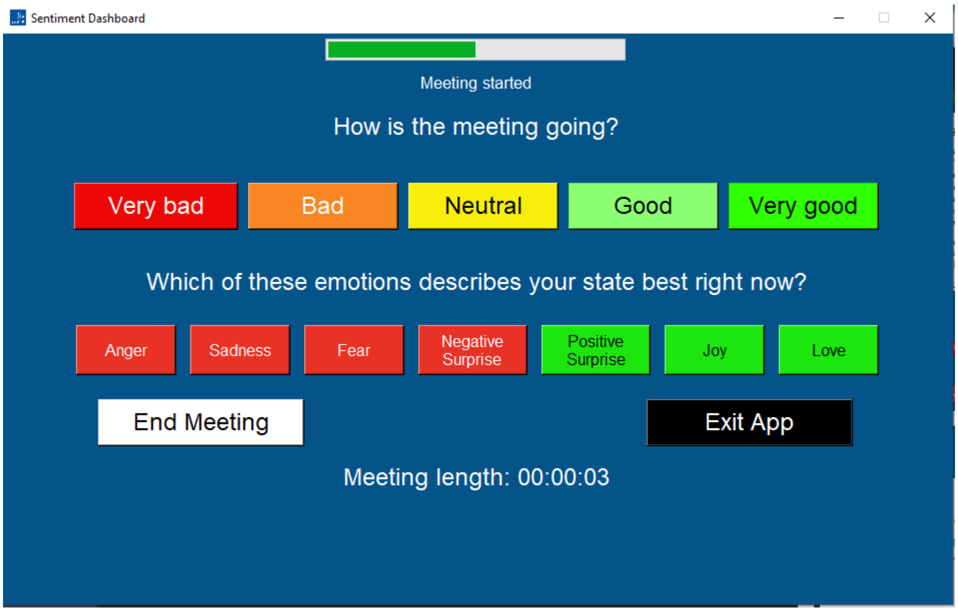

Figure 4.2: Meeting started screen

a more clearer choice of input. Here, same as by the polarities, the negative emotions have a red coloured button, while the positive ones have a green coloured button. On top of the screen, a text label and a progress bar show the user at which point they currently are. The options are "Before the meeting", "Meeting started" and "After the meeting". This are displayed to help the users find their way during the usage of the tool. On the bottom of the window, a meeting timer is also displayed, which shows the meeting duration. This timestamp is also saved for each feedback that the users give, in order to use it later in the overviews.

After the user presses "End Meeting", they are than asked for their mood after the meeting, similar to the question before starting the meeting. The format has been kept the same in order to preserve the familiarity that the user has won with the tool. Subsequently of inputting their mood after the meeting, the user is shown a line chart. This chart shows an overview of the whole meeting. Every polarity inputted from the user is displayed, together with the timestamp from each evaluation. This way the user can better notice the mood changes during the meeting and connect them with the respective timestamp. An example of this graph with some fictive values is shown in Figure 4.3. The user has the option to save each graph, which will be stored as a .png file inside the "Graphics" folder.

When pressing "Next graphic", another line chart is shown. This chart

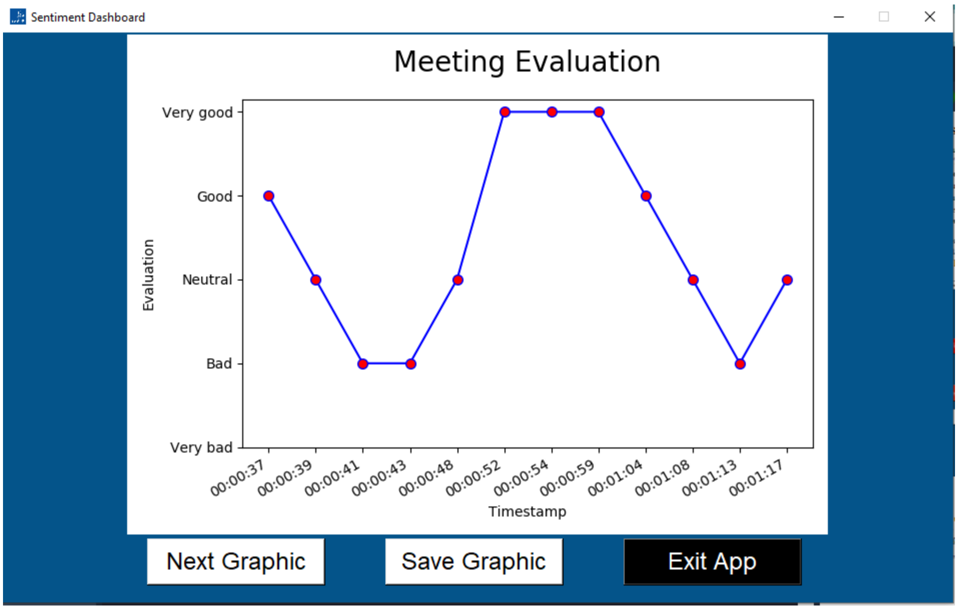

Figure 4.3: Example of meeting evaluation chart

displays only 3 points. The evaluation before the meeting, the average of all evaluations during the meeting and the evaluation inputed from the user after the meeting. To calculate the average of the evaluations during the meeting, the following numerical coding for the evaluations was used:

{"Very Bad" : -2, "Bad" : -1, "Neutral" : 0, "Good" : 1, "Very good" : 2}.

The average A was than calculated using the following formula:

$$A = \frac{1}{n} \sum_{i=1}^{n} a_i$$

where n is the number of evaluations given and a_i is the numerical conversion of the i-th evaluation. This chart can help users better notice the effect that the meeting had on their mood by comparing the mood before and after the meeting together with the average mood during it.

After pressing the button "Next graphic", the user is shown the last graph, which is a pie chart displaying an overview of the emotions that the user inputted during the meeting, as displayed by an example in Figure 4.4.

Each emotion is also shown with the respective percentage of the inputed times in comparison to the total emotions selected. This can help the user to overview their emotions during the meeting and better understand how they

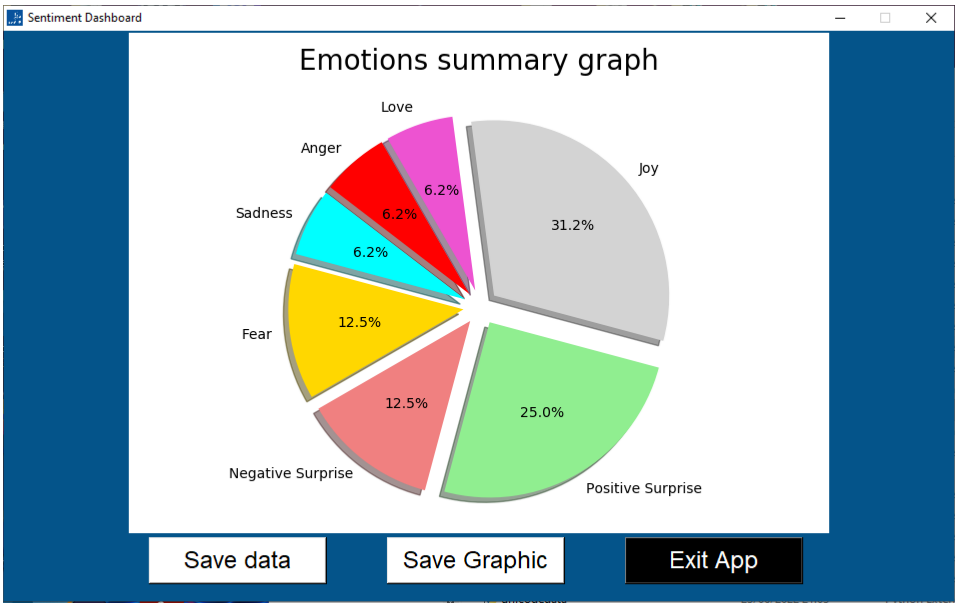

Figure 4.4: Example of the emotions summary graph

generally felt. The user can than save the data gathered from this meeting by pressing "Save data". This data is saved locally in a .csv file which can be found in the "Meetings" folder. This file can than be reused later to compare different meetings data with another. This will be discussed in the following subsection. The user can also save the chart by pressing "Save Graphic" just like in the other cases above. This would save the graph as a .png file in the "Graphics" folder.

4.2.2 Overview

In the Overview section, the user has the possibility to upload older meeting files and get a comparison between the uploaded meetings. When pressing the "Upload files" button, a Windows Explorer dialog box is displayed allowing the user to select from saved meeting files. The older meeting files are saved in the "Meetings" folder. Only .csv files starting with "Meeting" in their name can be selected. This is done to prevent the upload of incorrect files. After uploading the desired files, the user can press "Go to overview" to display the overview charts. Firstly, a line chart is displayed containing each evaluation from each meeting, together with the respective timestamps. The timestamps are gathered in a common x-axis. Each meeting is displayed through one line in the chart. This helps the users better see the differences between the moods in the different meetings at different points in them. An example with fictional values is shown in Figure 4.5.

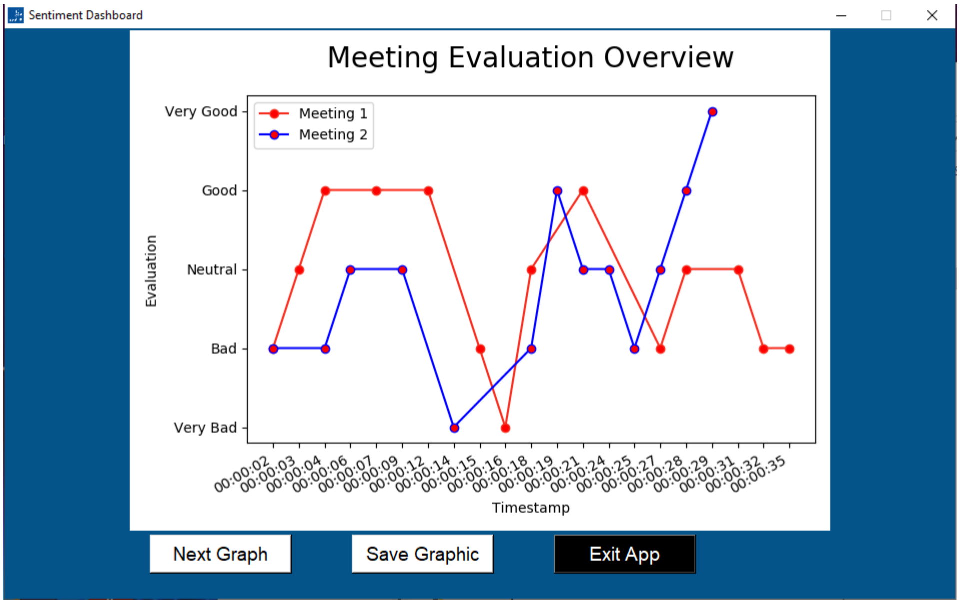

Figure 4.5: Example of a meeting evaluations overview

After pressing "Next graphic", the user is shown a bar chart, which serves as an overview for the emotions in every meeting. Each emotion from every meeting is shown through a bar. The sum of each emotion through all the meetings, is shown in the background through a wider bar. This way the user can easily differentiate between each meeting and emotion, in order to detect how the general mood was and which emotion was the most dominant. An example with some fictional values is shown in Figure 4.6.

As the last part of the overview, a table is shown, containing the evaluations from before and after the meeting, together with the average of the evaluations during the meeting. The same approach as discussed above was used, in converting the polarities into numbers. In this case though, it was not used a chart, since the user can upload many meetings, but the number of points remains 3. Therefore it is very possible that many meetings have similar values and that the lines would be drawn over one another and then, it would not be possible to receive a clear view of the wanted graphical representation. For this reason, a table containing these values was chosen as an overview item. Under the table, a small textual summary is shown, which says in how many from the selected meetings, the mood after the meeting got better, got worse or remained the same, when compared to the mood before the meeting. This can help the user interpret the effect that the meetings had on the participants. From here, the user can exit the app.

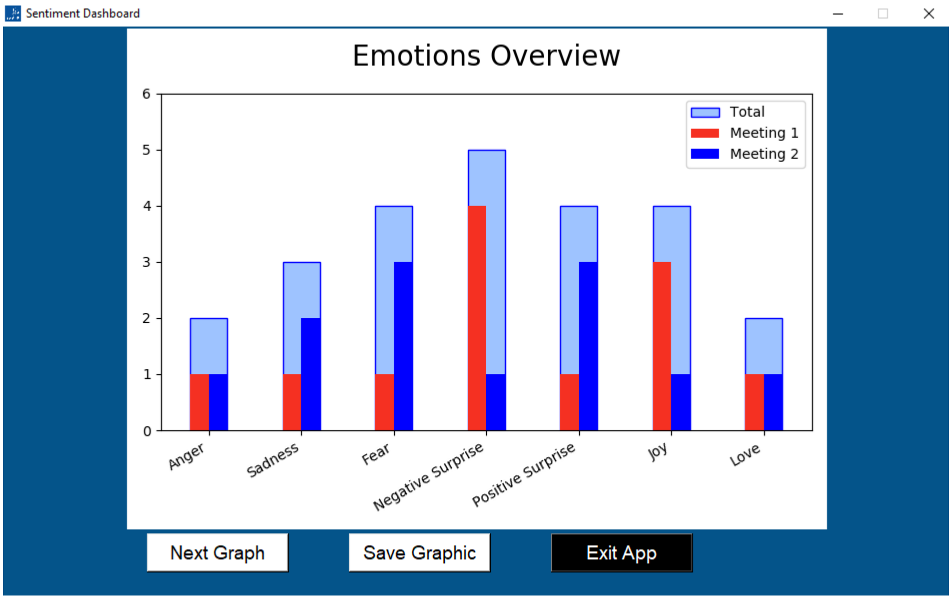

Figure 4.6: Example of an emotions overview

4.2.3 Language and Help

In the "Language" menu, the user can change the language between German and English. English is selected as default when opening the application. To do this, the user can press the desired language and than press "Apply". This changes the language in the whole application, including the polarities, emotions and the graphical labels and axes. In the "Overview" menu, emotions and polarities in the charts, will be displayed in the currently selected language, even if they were saved in another language in the original file.

In the "Help" menu, a short instruction text about the usage of the application is shown. This text's goal is to explain the users the most important functions in the application.

Chapter 5

Evaluation

In order to evaluate the tool, a survey was conducted. The tool was tested by 15 individuals, who work in the software engineering branch. The participants were asked to test the tool during a meeting. The participants only tested the feature "Start Meeting" and did not test the "Overview" function of the tool, which would require the usage of the tool in several meetings. This could prove difficult to test properly and the risk of having uneven experimental conditions for the survey participants could arise. Therefore, the participants were asked to only test the tool in one meeting. After the meeting, the participant filled out a survey asking for their opinion regarding the usage of the *Sentiment-Dashboard* during it. The survey was divided in three sections. In the first section, the participants were asked about their demographics and their experience in the software engineering branch. In the second section, the participants were asked about their experience and opinions regarding meetings and sentiment analysis tools. At the final section, the participants were asked about their experience with the *Sentiment-Dashboard*. This included closed questions, in which the users could select one option to evaluate a feature, and also open questions in which the users could express their feedback freely by text.

5.1 Demographics

The tool was tested by 15 individuals, all working in the software engineering branch. 9 of them were males and 6 females. They were between 20 and 35 years old. Their work experience in software engineering ranged between 1 and 7 years. The participants all worked as either IT Project Managers, Developers or Researchers at the Leibniz University of Hanover(LUH). The distribution is shown in Figure 5.1. 9 of the participants are still students and work part-time as working students, while the other 6 have already concluded their studies. Except the 4 Researchers at the LUH, all 11 other participants work at external companies.

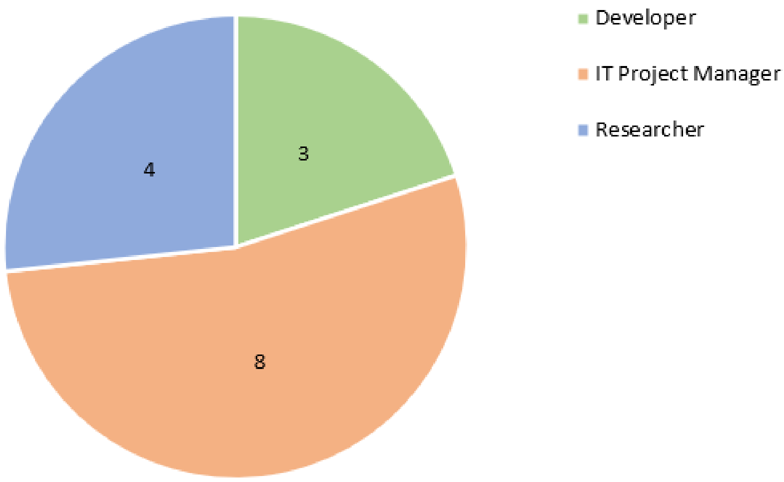

Figure 5.1: Distribution of the participants profession

5.2 Meetings and Sentiment Analysis

The participants were also asked about their experience and opinions regarding meetings and sentiment analysis tools. The users were asked how many meetings per week do they normally have. Most of the users had 1-3 meetings per week. The full distribution is shown in Figure 5.2.

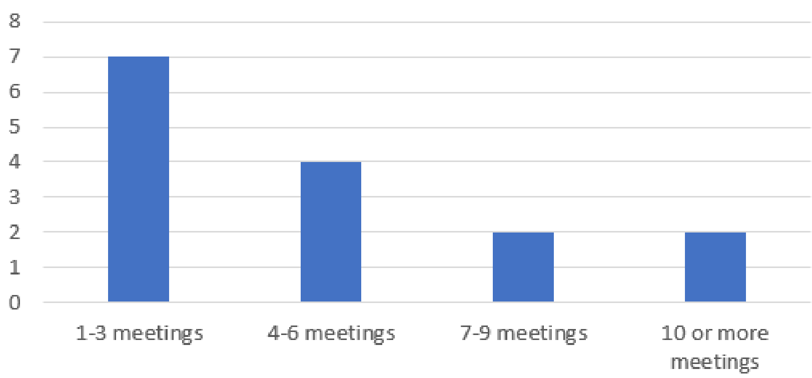

Figure 5.2: Meetings per week distribution

The participants were asked how much they agreed with the statement: "Meetings are a main source of communication and information exchange at work." , see Figure 5.3, and also with the statement: "For a meeting to be successful, every participant needs to feel good during the meeting." , see Figure 5.4 using the scale "1: Strongly disagree, 5: Strongly agree". The users mostly agreed with both sentences. The exact results are shown below.

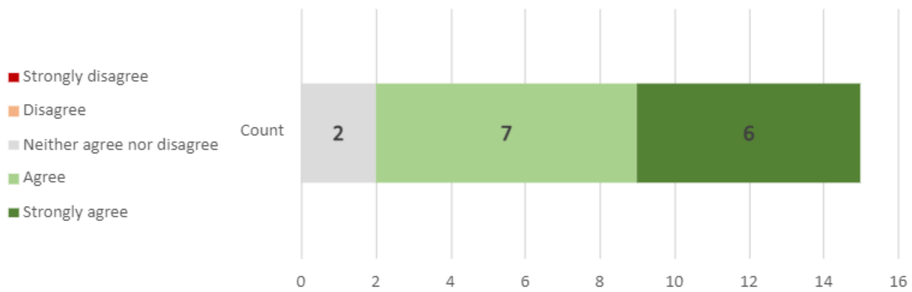

Figure 5.3: Agreement with: "Meetings are a main source of communication and information exchange at work."

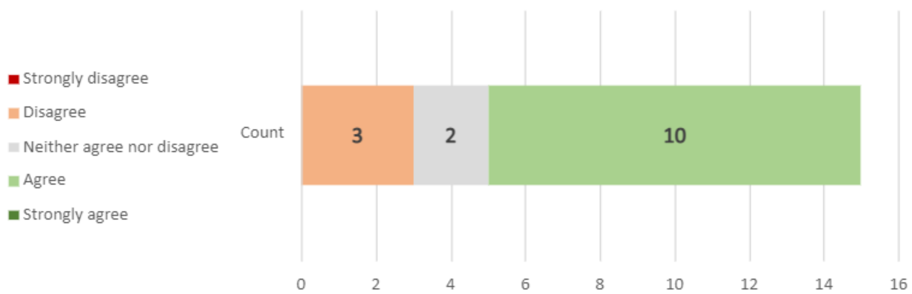

Figure 5.4: Agreement with: "For a meeting to be successful, every participant needs to feel good during the meeting."

The participants were than asked regarding their knowledge and opinion regarding sentiment analysis tools. Most of the participants were not very familiar with these tools, as Figure 5.5 shows. When asked about their preference of usage between a manual and an automatic sentiment analysis tools during meetings, 10 of them would rather choose a manual tool, while only 5 would chose an automatic tool to perform sentiment analysis during a meeting. An important reason for this would be trust, since the participants do not have a high level of trust in automatic tools, as shown in Figure 5.6.

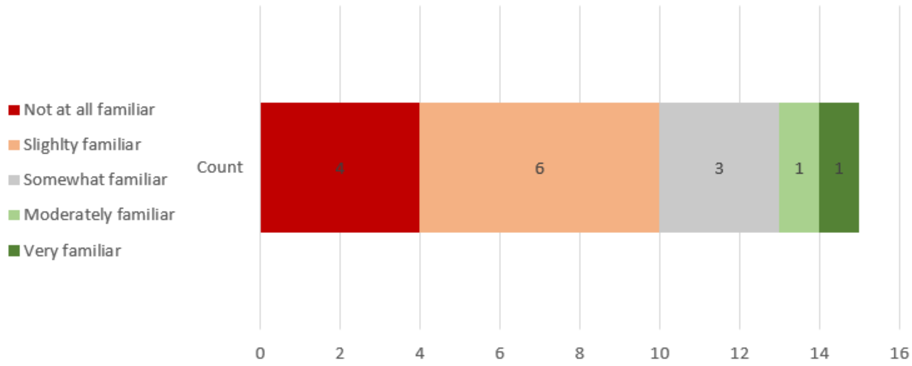

Figure 5.5: Familiarity with Sentiment Analysis

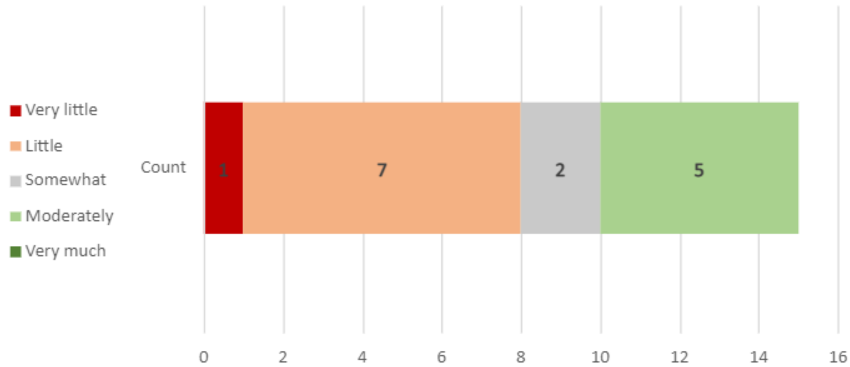

Figure 5.6: How much the participants would trust an automatic sentiment analysis tool used in meetings

As per the manual tools, when asked how much they would agree with the statement: "The usage of a manual sentiment analysis tool, which detects the mood of the participants or the meeting in general by gathering feedback from the participants directly, would help in realizing problems during meetings.", most of them agreed, as shown in Figure 5.7.

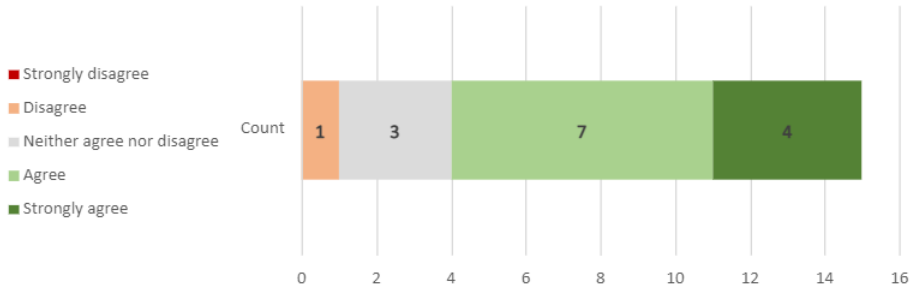

Figure 5.7: The usage of a manual sentiment analysis tool would help in realizing problems during meetings.

5.3 Tool feedback

Finally, the participants were asked about their opinion regarding their usage of the *Sentiment-Dashboard*. They were asked to evaluate aspects such as usability and design of the tool, along with how good they felt they could express their emotions during the meeting using the *Sentiment-Dashboard*. Most of the users liked how they could express their emotions. Most of them also rated the usability of the tool as good and very good. The design on the other hand, got a fairly neutral evaluation. The exact results are shown in Figure 5.8.

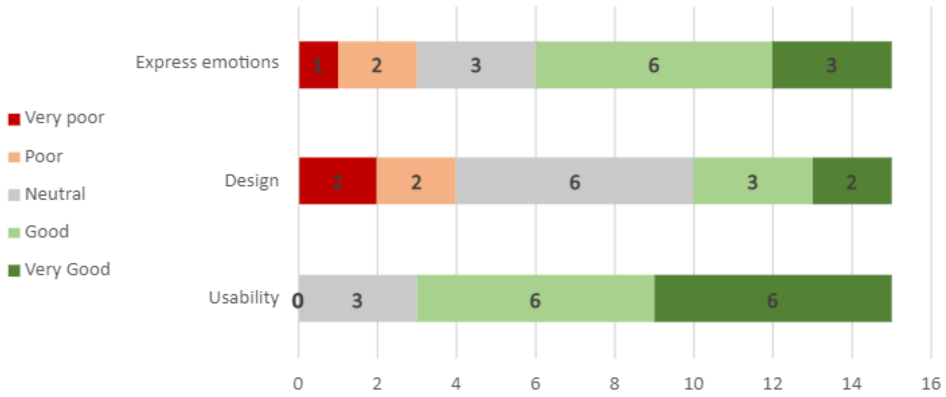

Figure 5.8: Evaluation for usability, design and expressing emotions

After the meeting ended, to the users of the tools was shown an overview of the feedback they gave during the meeting. Most of the users found this feature helpful and very helpful. Also, since the tool is to be used during meetings, the users were asked how much of a distraction the usage of the tool was for them during a meeting. Most of them found it only slightly distracting, as seen in Figure 5.9. When asked whether they agree that there is a need for the usage of the *Sentiment-Dashboard* in meetings, most of the participants agreed, as shown in Figure 5.10.

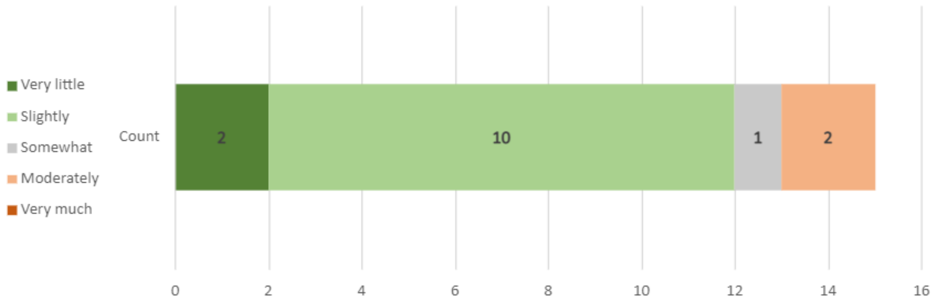

Figure 5.9: How much of a distraction was the tool during the meeting

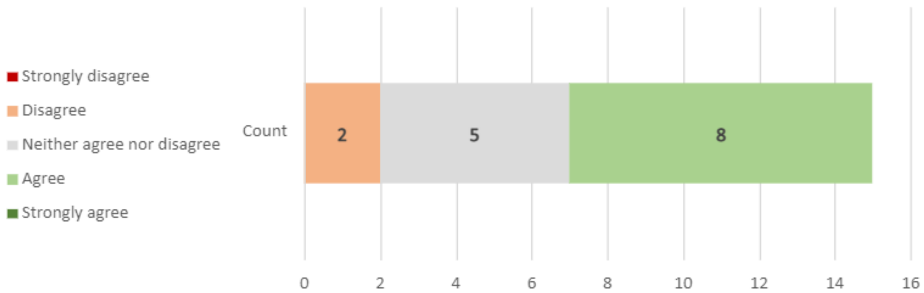

Figure 5.10: There is a need of usage for the *Sentiment-Dashboard* in meetings

Another important use case for the tool is the overview of the feedback from older meetings. The team members can send their data manually to their project managers who than can check how the overall mood of the team was during the meeting. Since the tool does not include an option to send the data automatically to others, the users were asked in this survey to which group of colleagues they would rather send their meeting feedback data(see Figure 5.11) and whether they would like to remain anonymous when sending this data or not. Out of 15 participants, 12 of them would prefer to remain anonymous when sharing their meeting feedback data. This feedback can be useful for a future expansion of the *Sentiment-Dashboard*.

Finally, the participants also had the possibility to give an open opinion regarding what they liked most about the tool, what they did not like, what other features they would like the tool to have and also whether they had any other comment regarding the tool or the survey. The users liked the overview graphics at the end of the meeting and also found the tool easy to use, user

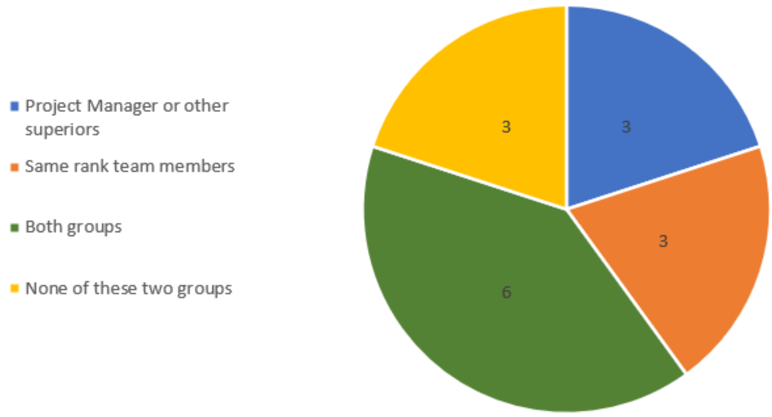

Figure 5.11: With whom would you rather share the meeting data?

friendly and interesting. 7 users felt that one of the aspects that could have been improved was the design of tool, while 4 others felt that the emotions, from which the users could select as input, needed better explaining and more options to chose from. Some of the features that the participants would like the tool to have, were the automatic sharing of results, more personalisation options regarding their feedback, a back button in the graphics overview or a reminder for inputting feedback during the meeting. In the free feedback question regarding the tool or the survey, some participants noticed that the tool needs more explanation for its users while others noticed that they found the general idea of the tool interesting and useful.

Chapter 6

Discussion

In this chapter, the research questions will be answered and the results will be discussed and interpreted. Furthermore, the threats to validity will be presented.

6.1 Answering the research question

In chapter 1, following research questions were presented. Both of these questions were asked in the survey to the participants.

RQ1 Is there a need for the usage of the *Sentiment-Dashboard* in meetings?

RQ2 Can the tool be used during meetings without disturbing the work flow and the concentration of the users?

Regarding RQ1, most of the participants (8 out of 15) agreed that there is a need for the usage of the Sentiment-Dashboard in meetings. It is to be noticed although that 5 participants neither agreed or disagreed, see Figure 5.10. This might have to do with the fact that the participants in this survey were not very familiar with sentiment analysis tools and that their usage and benefits might not be entirely clear to them yet. Also, the users noticed that they are some points that can be yet improved in the tool which might influence their evaluation. Finally, the fact that the survey participants only tested the tool during a meeting and did not test the "Overview" feature which compares meetings with one another, might also have an effect on their feedback since the full evaluation of the tool and its potential could not be tested. All in all, this is nevertheless a promising result and serves as a basis for further developing the tool.

Regarding RQ2, since the tool will be used during meetings, it is important that the user does not get distracted while using the tool and miss important discussion points in the meeting. Measures were taken when developing the tool to help in this direction, such as keeping the application in a small window and not overfilling the tool with information. Most of the survey participants (12 out of 15) felt that the tool was very little or only just slightly a distraction during the meeting, see Figure 5.9. Also 12 out of 15 participants found the usability of the tool good and very good and in the free feedback questions regarding the tool, participants answered that the tool was easy to use. These are very promising results since they show that the tool can be easily used during meetings, without disturbing the meeting flow and concentration of its users.

6.2 Discussing the results

As mentioned in chapter 5, a survey was conducted to evaluate the *Sentiment-Dashboard* tool after using it. The objective was to gather feedback regarding the tool after using it in a meeting, which is the main use case of the application. The tool received mostly positive feedback from the survey participants. It was tested by 15 participants, all with a background in software engineering and all currently working in a software engineering context. The participants worked either as developers, researchers at the Leibniz University or as project managers. This helped in gaining different perspectives when using and evaluating the tool. All of the users take part in meetings during their week and also agreed that they see meetings as a main source of communication and information exchange during work. This makes their feedback relevant, because they are already familiar with a meeting context. Most of the users though, were not familiar with sentiment analysis. This might have an effect on the evaluations that the participants have given regarding their preferences about sentiment analysis tools. When asked whether they would rather prefer to use an automatic or a manual sentiment analysis tool during the meeting, most of the users choose the manual tool. A main reason behind this might be the lack of trust that the participants might have toward an automatic evaluation of the sentiment, as Figure 5.6 showed. The participants would rather prefer having the possibility to input their feedback personally than to receive an evaluation from an automatic tool.

The tool itself got positive evaluations from the survey participants and especially the usability of the tool was rated mostly good and very good. Most of them were also satisfied with the given possibilities to express their emotions during the meeting. It is to be noticed although, that one participant rated this possibility as very poor, 2 as poor and 3 as neutral. The emotions from which the user can choose from were based on the Emotion Models from Shaver [47] and Parrott [39]. Adding more emotions to the 7 currently implemented in the tool, was deemed to have a negative effect on the usability of the tool and that it would fill the screen with too much information for the user to deal with. Considering that during a meeting the user has to pay constant attention to the matter being discussed, it is important for the tool to not be a distraction to them and that the user should choose their input and feedback easily. The design of the tool got a fairly neutral evaluation. It can be improved by applying a more modern overall look. Selecting another framework for the implementation could aid in this perspective since TKinter[1] offers a rather limited offer of tools regarding design. Both of these evaluations are consistent with the feedback the participants left in the free text questions regarding what aspects should be improved and what other features should be included, as design and more options from which to choose the emotions from were mentioned as possible improvements.

Another important discussion point is regarding the fact that 80% of the participants would rather remain anonymous,if they were to share their meeting feedback data with their colleagues. One reason can be that the participants would rather avoid sharing their emotions directly with their colleagues. Their feedback might be interpreted as a personal evaluation toward other colleagues which can than be cause for debate. This should be kept in consideration for future extensions of the application.

All in all, one can say that this thesis reached its objective in exploring a different approach for the usage of sentiment analysis in meetings and providing a tool which helps in this direction. Through this tool, the problems regarding the usage of automatic sentiment analysis tools in meetings are avoided and a more concrete and accurate evaluation of a meeting can be reached. The tool was also generally well received by its testers. The results of the survey were mainly promising and can lay the ground for further research and development of this approach.

6.3 Threats to validity

In order to discuss the ***Validity Threats*** of this bachelor thesis, the ***Threats*** presented by Wohlin et al. [54] will be used as basis.

The number of participants (15) that tested the tool and than answered the survey regarding it, is relatively small. This could influence the validity of the results when trying to generalise them. This results in a ***Threat to Conclusion Validity***. By testing the tool with a bigger number of participants, more valuable results could be gathered, which would than remove this threat.

The evaluation from the survey participants might also have been influenced from external factors such as their current mood when testing the tool or from the results of the meeting in which they took part. If the meeting

[1] www.docs.python.org/3/library/tkinter.html

was not productive or the participant had mainly negative reactions to it, than the evaluation about the tool might also have been influenced from it. The participants were instructed to answer the survey right after testing the tool, but it is possible that due to time reasons, some of them might have not had the opportunity to do so directly. Instead, they could have answered the survey later and therefore their evaluation might be affected from this. These scenarios result in a ***Interval Validity Threat***.

The results of the survey could have also been influenced by the fact that most of the users were not very familiar with sentiment analysis and might have needed a better introduction than the one given, regarding the usage of the tool and the concept behind it. Furthermore, since the users only tested one feature, the usage of the tool during a meeting, and not the "Overview" feature, which compares several meetings with one another, this might also have influenced the final evaluation about the tool. Both of these factors result in a ***Construct Validity Threat***.

Chapter 7

Conclusion

In this chapter, a summary of the overall work in this thesis will be presented, along with an outlook regarding the possible further development the tool.

7.1 Summary

Meetings are a main communication route for information exchange in software engineering projects which are becoming more complex and with more people involved. This is why it can be very important to analyse the interactions in a meeting along with the effects that the meeting had on its participants. Efforts have been made in this direction by categorising interactions in meeting and also using sentiment analysis during it. The usage of automatic sentiment analysis tools brings problems with it, such as reduced accuracy, inability to recognise irony and sarcasm and dependency on the dataset the tool was trained on. To tackle these problems, a tool was developed in this bachelor thesis, which allows the users to manually input their feedback regarding their current mood during a meeting. This way the aforementioned risks can be avoided. The tool saves this feedback in files and displays an overview at the end of the meeting, so that the user can get a better perspective of the progression of the meeting. The files from older meetings can be compared with one another in the "Overview" menu in order to gain a general overview of past meetings. This feature can also be used by project managers to compare the feedback from their team members regarding a meeting, provided that they manually receive the meeting files from their team members. This way the managers can directly evaluate the general mood of the team during the meeting and notice problems that may arise at an early stage. The tool was tested by 15 participants in a meeting, which than took part in a survey regarding their experience and feedback about the usage of the tool. The feedback was mainly positive, especially regarding the usability of the tool during the meeting, and it showed that the tool has potential to be useful. It also showed that there is a need for the usage of the tool during meetings to evaluate them and also that the usage of the tool does not distract the users and does not disturb the flow of the meeting. This feedback can also be used to further develop the tool. Through this survey, data was collected, not only for the purpose of evaluating the *Sentiment-Dashboard*, but also to gather information regarding topics such as sentiment analysis and meetings, which can be used in further research.

7.2 Outlook

The *Sentiment-Dashboard* can of course be further developed and improved. As the users noticed, improvements in its design and the ways that the user can input their emotions can be taken. The design can be modernised, with the usage of other frameworks, like *Kivy*[1] or *PyQt* [2], to build the graphical user interface. More emotions can be added as possible inputs for the user, in order to assure more variety in expressing their feelings. For example, instead of using just the primary emotions from Parrott´s Emotion Model [39], some of the secondary emotions from the same model can be also be used to expand the range of choice. Another idea is a text field which can be added in order for the users to input a textual feedback regarding events in the meeting, which might not be expressed with the current input methods. More questions regarding the mood and emotions of the user might be integrated before and after the meeting, in order to get a better picture of their emotional state and the effects that the meeting might have had on them.

Another important feature that can be added is the possibility for the users to automatically send their meeting data to their project manager after the meeting has ended. This way the manager can compare the data using the "Overview" feature and directly get a view of how the meeting was perceived from the perspective of the other team members. If the meeting participants were mostly on a bad mood for example, measures have to be taken in order to improve this. Also, if the mood of the team members got worse after the meeting when compared to the mood before the meeting, than it is possible that the meeting had a negative impact on them and this should also be taken in consideration. It is important to mention here though, that through the survey it was noticed that most of the participants would rather remain anonymous if they were to share they data, so this is something to keep in consideration if this feature is to be added. One possible solution to this would be uploading the data into the company server without having the possibility to trace which team member uploaded them. This way, a general overview of the mood of the team can be evaluated.

[1] www.kivy.org/
[2] www.riverbankcomputing.com/software/pyqt/intro

Bibliography

[1] Aashish Agarwal and Torsten Zesch. German end-to-end speech recognition based on deepspeech. In *KONVENS*, 2019.

[2] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. Senticr: a customized sentiment analysis tool for code review interactions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 106–111. IEEE, 2017.

[3] D Alessia, Fernando Ferri, Patrizia Grifoni, and Tiziana Guzzo. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3), 2015.

[4] Cristina Bosco, Viviana Patti, and Andrea Bolioli. Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE intelligent systems*, 28(2):55–63, 2013.

[5] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proceedings of the second international workshop on issues of sentiment discovery and opinion mining*, pages 1–9, 2013.

[6] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment polarity detection for software development. *Empirical Software Engineering*, 23(3):1352–1382, 2018.

[7] Bi Chen, Leilei Zhu, Daniel Kifer, and Dongwon Lee. What is an opinion about? exploring political standpoints using opinion scoring model. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[8] Munmun De Choudhury and Scott Counts. Understanding affect in the workplace via social media. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 303–316, 2013.

[9] Ronen Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.

[10] Robert Feldt, Lefteris Angelis, Richard Torkar, and Maria Samuelsson. Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, 52(6):611–624, 2010.

[11] Emitza Guzman, David Azócar, and Yang Li. Sentiment analysis of commit comments in github: an empirical study. In *Proceedings of the 11th working conference on mining software repositories*, pages 352–355, 2014.

[12] Ann-Christin Haak. Bachelor thesis: Analysis of interactions during software project meetings. 2017.

[13] Marc Herrmann and Jil Klünder. From textual to verbal communication: Towards applying sentiment analysis to a software project meeting. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 371–376. IEEE, 2021.

[14] Marc Herrmann, Martin Obaidi, and Jil Klünder. Senti-analyzer: Joint sentiment analysis for text-based and verbal communication in software projects. *arXiv preprint arXiv:2206.10993*, 2022.

[15] Nasif Imtiaz, Justin Middleton, Peter Girouard, and Emerson Murphy-Hill. Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people. In *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, pages 55–61. IEEE, 2018.

[16] Md Rakibul Islam and Minhaz F Zibran. Towards understanding and exploiting developers' emotional variations in software engineering. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 185–192. IEEE, 2016.

[17] Md Rakibul Islam and Minhaz F Zibran. Leveraging automated sentiment analysis in software engineering. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 203–214. IEEE, 2017.

[18] William James. What is an emotion? Mind, os-ix, 188-205, 1884.

[19] Robbert Jongeling, Subhajit Datta, and Alexander Serebrenik. Choosing your weapons: On sentiment analysis tools for software engineering research. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 531–535. IEEE, 2015.

[20] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools

for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, 2017.

[21] Simone Kauffeld and Nale Lehmann-Willenbrock. Meetings matter: Effects of team meetings on team and organizational success. *Small group research*, 43(2):130–158, 2012.

[22] Jil Kluender. *Analyse der Zusammenarbeit in Softwareprojekten mittels Informationsfluesen und Interaktionen in Meetings*. Logos Verlag Berlin GmbH, 2019.

[23] Jil Klünder, Julian Horstmann, and Oliver Karras. Identifying the mood of a software development team by analyzing text-based communication in chats with machine learning. In *International Conference on Human-Centred Software Engineering*, pages 133–151. Springer, 2020.

[24] Robert. E. Kraut and Lynn. A. Streeter. Coordination in software development. In *Communications of the ACM, vol. 38, no. 3*, pages 69–82, 1995.

[25] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[26] Desmond J Leach, Steven G Rogelberg, Peter B Warr, and Jennifer L Burnfield. Perceived meeting effectiveness: The role of design characteristics. *Journal of Business and Psychology*, 24(1):65–76, 2009.

[27] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment analysis for software engineering: How far can we go? In *Proceedings of the 40th international conference on software engineering*, pages 94–104, 2018.

[28] Bing Liu. Sentiment analysis and opinion mining. 2012.

[29] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Arsa: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 607–614, 2007.

[30] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[31] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. Can we use se-specific sentiment analysis tools in a cross-platform setting? In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 158–168, 2020.

[32] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. The challenges of sentiment detection in the social programmer ecosystem. In *Proceedings*

of the 7th International Workshop on Social Software Engineering, pages 33–40, 2015.

[33] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. A gold standard for emotion annotation in stack overflow. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pages 14–17. IEEE, 2018.

[34] Martin Obaidi and Jil Klünder. Development and application of sentiment analysis tools in software engineering: A systematic literature review. *Evaluation and Assessment in Software Engineering*, pages 80–89, 2021.

[35] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Fourth international AAAI conference on weblogs and social media*, 2010.

[36] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. The emotional side of software developers in jira. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, pages 480–483. IEEE, 2016.

[37] Ilan Oshri, Julia Kotlarsky, and Leslie P Willcocks. Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects. *The Journal of Strategic Information Systems*, 16(1):25–49, 2007.

[38] Georgios Paltoglou and Mike Thelwall. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):1–19, 2012.

[39] W Gerrod Parrott. *Emotions in social psychology: Essential readings*. Psychology Press, 2001.

[40] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[41] Federico Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu. *Sentiment analysis in social networks*. Morgan Kaufmann, 2016.

[42] Nils Prenner, Jil Klünder, and Kurt Schneider. Making meeting success measurable by participants' feedback. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, pages 25–31, 2018.

[43] Stanley Rachman. The overprediction of fear: A review. *Behaviour research and therapy*, 32(7):683–690, 1994.

[44] James A Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003.

[45] Daniel Russo, Paul HP Hanel, Seraphina Altnickel, and Niels Van Berkel. The daily life of software engineers during the covid-19 pandemic. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 364–373. IEEE, 2021.

[46] Kurt Schneider, Jil Klünder, Fabian Kortum, Lisa Handke, Julia Straube, and Simone Kauffeld. Positive affect through interactions in meetings: The role of proactive and supportive statements. *Journal of Systems and Software*, 143:59–70, 2018.

[47] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061, 1987.

[48] Mike Thelwall. The heart and soul of the web? sentiment strength detection in the social web with sentistrength. In *Cyberemotions*, pages 119–134. Springer, 2017.

[49] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[50] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558, 2010.

[51] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4, pages 178–185, 2010.

[52] Roger J Volkema and Fred Niederman. Organizational meetings: Formats and information requirements. *Small group research*, 26(1):3–24, 1995.

[53] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35, 2005.

[54] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.