

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

Entwicklung einer grafischen
Benutzeroberfläche zur Analyse
von Stimmungen in
Softwareprojektmeetings

Development of a graphical user interface for sentiment
analysis in software project meetings

Bachelorarbeit

im Studiengang Informatik

von

Juri Linnemann

Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Dr. Jil Ann-Christin Klünder
Betreuer: Alexander Specht

Hannover, 10. Mai 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 10. Mai 2022

Juri Linnemann

Zusammenfassung

In Softwareprojekten sind Meetings ein unerlässlicher Teil der Arbeit. Die Qualität der Meetings kann sich auch maßgeblich auf die Qualität der Software selbst auswirken. Daher ist es für eine ideale Erfüllung der Aufgabe zielführend, wenn sich entweder die Teilnehmer selbst oder die Teamleitung mit der Stimmung, auch Sentiments, innerhalb der Meetings auskennen. Um die Stimmung eines solchen Meetings zu bestimmen wurde ein Tool, der SEnti-Analyzer, erstellt welcher mittels Sentiment Analyse die Stimmung ermittelt.

Der SEnti-Analyzer ist bisher aber nur auf einer Kommandokonsole benutzbar, was es für die zu nutzenden Personen schwer macht oder sie gar ganz abschreckt. Ziel dieser Arbeit ist die Entwicklung einer grafischen Oberfläche um die Benutzung des SEnti-Analyzers schnell und einfach für viele Personen zu ermöglichen. Um dieses Ziel zu erreichen wurde eine Anforderungsanalyse durchgeführt. Teil davon waren Interviews mit Mitarbeiter*innen des Fachgebiets Software Engineering. Neben der erstellten grafischen Oberfläche wurde in dieser Arbeit auch evaluiert, wie die Oberfläche und der SEnti-Analyzer selbst noch weiter verbessert werden können.

Abstract

Development of a graphical user interface for sentiment analysis in software project meetings.

In software projects, meetings are an indispensable part of the work. The quality of the meetings can also have a significant impact on the quality of the software itself. Therefore, for an ideal fulfillment of the task, it is purposeful if either the participants themselves or the team management are familiar with the mood, also sentiments, within the meetings. In order to determine the mood of such a meeting, a tool, the SEnti- Analyzer, was created, to determine the mood of such a meeting by means of sentiment analysis.

So far, the SEnti-Analyzer can only be used on a Terminal, which makes it difficult for the people to use it or even discourages them completely. The goal of this work is the development of a graphical user interface to make the use of the SEnti-Analyzer fast and easily for many people. To achieve this, a requirements analysis was conducted. Part of it were interviews with employees of the department of Software Engineering. Besides the created graphical user interface, it was also evaluated in this work, how the user interface and the SEnti-Analyzer itself can be further improved.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	2
1.3	Lösungsansatz	3
1.4	Ergebnisse der Arbeit	3
1.5	Struktur der Arbeit	3
2	Grundlagen	5
2.1	Stimmungsanalyse	5
2.1.1	Stimmungsanalyse in Softwareprojekten	7
2.1.2	Level der Stimmungsanalyse	8
2.2	Technische Grundlagen	10
2.2.1	Grafische Oberflächen	10
2.2.2	GUI Programmierung	11
2.2.3	Audiodateien	11
2.2.4	Textdateien	12
2.2.5	Die Einzelnen Tools	13
2.3	Der SEnti-Analyzer vor der GUI	14
2.3.1	Die Funktionen des SEnti-Analyzers	15
3	Verwandte Arbeiten	17
4	Anforderungsanalyse	19
4.1	Im Vorfeld gesammelte Ideen	19
4.1.1	Hauptmenü	19
4.1.2	Prozessbildschirm	20
4.2	Interviews	21
4.3	Anpassung der gesammelten Ideen	25
5	Entwicklungsprozess der GUI	27
5.1	Programmstruktur	27
5.2	Nicht hinzugefügte Funktionen	36

5.3	Verbesserungen	37
6	Funktionstests des SEnti-Analysers	39
7	Limitierungen dieser Arbeit	43
8	Zusammenfassung und Ausblick	47
8.1	Zusammenfassung	47
8.2	Ausblick	47

Kapitel 1

Einleitung

Ob ein Softwareprojekt den gewünschten Erfolg hat, hängt von vielen Faktoren ab.[4] Zumal haben unterschiedliche Personen in unterschiedlichen Positionen ein unterschiedliches Verständnis von einem erfolgreichen Projekt. Ein Projektleiter/Eine Projektleiterin möchte beispielsweise, dass die Software den im Vorfeld vereinbarten Funktionsumfang erfüllt und die pünktlich fertig und nicht zu teuer ist. Für die Entwickler*innen selbst könnte in erster Linie wichtig sein, dass die Software so funktioniert, wie sie soll. In diesem Fall ist es annehmbar, dass die Wahrnehmung des Erfolgs des Projektleiter*innen direkt von der Wahrnehmung der Entwickler*innen abhängt. Wenn die Entwickler*innen gute Arbeit leisten und das vom Projektleiter*innen gewünschte Ergebnis liefern, dann sind alle beteiligten Personen zufrieden. Neben Aspekten wie der Projektplanung, von Zeit- und Geldressourcen, die unmittelbar mit dem Erfolg eines Softwareprojekts zusammenhängen, stellt sich die Frage in wie fern das Umfeld der Entwickler*innen bei dem Erfolg eines Softwareprojekts eine Rolle spielt. Diese Frage wird im folgenden beantwortet.

1.1 Motivation

De Barros Sampaio et. al. beschreiben, dass ein Softwareprojekt von einer Vielzahl an Faktoren abhängt. [4] Einer dieser Faktoren ist die Motivation der Entwickler*innen, welche in den letzten Jahren immer wichtiger als Forschungspunkt geworden ist. [4] Dass zufriedene Entwickler*innen bessere Arbeitsergebnisse erzielen, wurde bereits gezeigt. [3] Sie arbeiten produktiver und die Erfolgchancen steigen. Dies ist ein wünschenswerter Zustand. Da Entwickler*innen

vermehrt nicht alleine arbeiten, muss ein möglichst angenehmes Arbeitsklima herrschen, um die Motivation der Entwickler*innen nicht zu senken oder im Idealfall sogar zu steigern. Ein Aspekt des Arbeitsklimas ist die Stimmung innerhalb eines Entwicklerteams. Um ein ideales Arbeitsklima zu erzeugen, ist es daher unter anderem von Relevanz die Stimmung innerhalb des Teams zu bestimmen. Daher ist es von Vorteil, wenn Mittel genutzt werden, welche Stimmung innerhalb des Teams feststellen können. Eine Gelegenheit dafür bietet sich in Meetings. In Meetings ist meist das gesamte Team versammelt und bei beispielsweise einem SCRUM kommt jede*r zu Wort.

Jedoch ist es schwer und auch nicht die Aufgabe der Entwickler*innen ihre eigene Stimmung oder gar die der Kollegen zu analysieren. Vor allem, da in den letzten Jahren durch die Coronapandemie viele Meetings auch von Zuhause aus stattfinden. Dies macht es schwieriger die Stimmung grundlegend einzuordnen. Zwar lässt sich die Stimmung einer Person auch an der Mimik und Gestik erkennen, aber auch das was gesagt wird ist ein wichtiger Ansatzpunkt um Stimmung festzustellen. Zu diesem Zweck existieren mehrere Tools die diese Analyse durchführen. Eines davon ist der SEnti-Analyzer[6]. Damit dieses Tool bestmöglich funktioniert, muss es, unter anderem, einfach nutzbar sein.

1.2 Problemstellung

Das Tool, der SEnti-Analyzer, für die Bestimmung der Stimmung innerhalb von Meetings existiert bereits. Doch bisher ist der SEnti-Analyzer nur auf der Kommandokonsole benutzbar, es gibt keine grafische Oberfläche für diesen. Orientiert wird sich dabei an den Qualitätsaspekten der ISO 25010. Die Usability des SEnti-Analyzers ist im Punkt der Operability nicht gegeben. Für Personen, welche sich nicht damit auskennen, ist es nicht intuitiv damit zu arbeiten und schreckt im schlimmsten Fall sogar komplett von der Benutzung ab. Der SEnti-Analyzer ist nicht Accessible genug. Der SEnti-Analyzer bietet zurzeit ein umfangreiches Funktionsrepertoire, jedoch kann die Erweiterung der Functional Completeness den Nutzer*innen gegenüber durch eine grafische Oberfläche vereinfacht werden. Für die Nutzer*innen ist es einfacher mit einer grafischen Oberfläche zu interagieren als mit der Kommandokonsole. Somit ist es von großem Vorteil, wenn der SEnti-Analyzer um eine solche grafische Oberfläche erweitert wird.

1.3 Lösungsansatz

Um eine möglichst passende grafische Oberfläche zu erstellen, wurde zuerst eine Literaturrecherche zu den Grundlagen der Sentimentanalyse und der Programmierung von grafischen Oberflächen eingeleitet. Bevor die eigentliche Oberfläche programmiert wurde, fand eine Anforderungsanalyse statt, um festzustellen wie die Oberfläche idealerweise gestaltet werden kann. Dazu wurden Interviews mit Mitarbeiter*innen des Fachgebiets Software Engineering, welche näher mit dem SEnti-Analyzer arbeiten, durchgeführt. Der erstellte Programmcode der grafischen Oberfläche orientiert sich dabei am bereits existierenden Code vom SEnti-Analyzer. Nachdem die Oberfläche fertig programmiert wurde, wurde sie mit dem SEnti-Analyzer verknüpft und anschließend im Rahmen einer kleinen Feldstudie auf den Aspekt der Usability getestet.

1.4 Ergebnisse der Arbeit

Das Ergebnis dieser Arbeit ist die grafische Oberfläche für den SEnti-Analyzer. Zusätzlich sind neben der grafischen Oberfläche auch Funktionen hinzugekommen, wie das Speichern der Dateien unter bestimmten Namen, das Navigieren ins Hauptmenü, die Darstellung einer vereinfachten Tabelle und die grafische Darstellung der Analyseergebnisse. Darüber hinaus bietet diese Arbeit auch einen umfangreichen Ausblick für weitere mögliche Verbesserungen für den SEnti-Analyzer.

1.5 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen der Arbeit beschrieben. Dies umfasst Grundlagen zur Sentimentanalyse, sowie zur Programmierung von grafischen Oberflächen, zu dem SEnti-Analyzer selbst und weitere allgemeinen technischen Grundlagen, die für das Verständnis der Arbeit und des SEnti-Analyzers notwendig sind. In Kapitel 3 werden verwandte Arbeiten vorgestellt auf denen diese Arbeit aufbaut. In Kapitel 4 wird die durchgeführte Anforderungsanalyse und deren Ergebnisse dargestellt. In Kapitel 5 wird der Entwicklungsprozess der grafischen Oberfläche beschrieben und erklärt. In Kapitel 6 wird die Durchführung und deren Ergebnisse für Tests der grafischen Oberfläche für den SEnti-Analyzers beschrieben. In

Kapitel 7 werden die Limitierungen die dieser Arbeit unterliegen dargestellt. In Kapitel 8 wird eine kurze Zusammenfassung der Arbeit geboten und ein Ausblick in die Zukunft geworfen.

Kapitel 2

Grundlagen

2.1 Stimmungsanalyse

Stimmungsanalyse, auch Sentiment Analyse, manchmal Opinion Mining, ist ein breites Feld mit vielen Anwendungsgebieten. Vor allem kommt es im Bereich der Produktanalyse zum Tragen, aber generell lässt sie sich in jeder Art von Text dazu benutzen, um von diesem die Stimmung festzustellen. Die Bezeichnungen meinen im Grunde alle das gleiche, sind jedoch unterschiedlich konnotiert. So ist etwa der englische Begriff Opinion Mining, im deutschen Meinungsforschung, eher darauf angelegt, dass es um Meinungen geht. Wohingegen Sentiment Analyse vom französischen *Sentiment*, zu deutsch Gefühl, abstammt und eher auf die Analyse von Gefühlen und Empfindungen zielt. In dieser Arbeit werden die Begriffe Stimmung und Meinung, genau wie auch Stimmungsanalyse, Sentiment Analyse und Opinion Mining als Synonyme zueinander genutzt, hauptsächlich werden dabei die Begriffe Stimmungsanalyse und Stimmung genutzt, da es in dieser Arbeit um eben diese Stimmung oder Gefühle bzw. Empfindungen geht und nicht um Meinungen. Im Kern geht es bei der Sentiment Analyse darum, dass Informationen aus Text herausgezogen werden. Diese Informationen sind wie oben erwähnt die Meinungen bzw. Stimmungen im Bezug auf ein bestimmtes Ziel. Ein großes Anwendungsgebiet der Sentiment Analyse ist die Produktindustrie, dort wird sie vor allem oft verwendet um die Meinung von Kunden zu bestimmten Produkten herauszufinden. Im Zeitalter des Internets kann mit einem Computer oder vom Handy aus unterwegs vieles bewertet werden. Von einzelnen Produkten von großen Online-Händlern, Restaurants auf Google Maps, Apps für mobile Endgeräte aus dem Google

PlayStore oder dem AppStore bis hin zu Bahnhaltstellen, alles lässt sich bewerten. Dabei werden zwar teilweise unterschiedliche Bewertungsmöglichkeiten gegeben - etwa eine Bewertung in Sternen, oft reichend von 1 bis 5, eine Bewertung auf einer Punkteskala, reichend von 1 bis 5 oder 10, oder ein binäres Ja/Nein, Daumen hoch/Daumen runter System - doch es gibt oft eine Möglichkeit nicht nur die eigentliche Bewertung abzugeben, sondern auch einen Kommentar dazu zu verfassen. Kommentare werden aus vielerlei Gründen geschrieben. Unabhängig davon, ob die eigene Meinung begründet oder sogar gerechtfertigt, Kritik oder Lob verfasst oder einfach noch ein bisschen weiter ausgeholt werden soll, manchmal ist die eigene Meinung oder Empfindung ja auch nicht in einer Bewertung ausdrückbar. So kann zum Beispiel der Besuch eines Restaurants grundsätzlich positiv aufgenommen worden sein, aber eine bestimmte Servicekraft war vielleicht etwas unfreundlich. Ob dem Restaurant dann bei einer Bewertung insgesamt eine niedrigere Bewertung gegeben wird oder sich die Zeit genommen wird einen Kommentar zu verfassen, die Möglichkeit ist da und diese wird oft benutzt, bei einer Vielzahl von Produkten und Serviceleistungen. Und dies ist auch oft der erste Blick von einem selbst, wenn sich eine Person nicht sicher ist, ob sie vielleicht etwas kaufen oder einen Service in Anspruch nehmen möchte. Was denken andere? Was haben andere für Erfahrungen gemacht? Wo früher die Meinung der Eltern, Großeltern oder Freunden eingeholt wurde, so werden heutzutage die Meinungen von wildfremden Menschen im Internet genutzt, um sich zu beraten. Wenn nun der/die Verkäufer*in eines bestimmten Produktes oder der/die Anbieter*in von einer bestimmten Serviceleistung möglichst erfolgreich sein möchte, dann möchte sich diese*r über die Meinung der Kunden über das eigene Produkt oder die eigene Serviceleistung im Klaren sein. Denn nur mit dem Wissen womit Leute unzufrieden sind, kann etwas verändert und verbessert werden, um so mehr Leute mit dem eigenen Produkt oder der eigenen Leistung zufriedenzustellen.

Ein weiteres Anwendungsgebiet ist im Social Media Bereich, in dem, ähnlich wie in der Produktindustrie, auf z.B. Tweets auf Twitter Informationen zu ausgewählten Themengebieten gesammelt und analysiert werden können.

Dabei hilft die Sentimentanalyse. Es gibt einige Tools die für die Sentiment Analyse entwickelt worden sind. Namentlich sind das zum Beispiel SentiStrength[1], SentiStrengthSE, SentiCR[2] und Senti4SD[5]. Solche Tools nehmen Textinput ein und Verarbeiten

diese über verschiedenen Arten.

Die meisten Stimmungsanalyse-Tools arbeiten entweder lexikonbasiert, mithilfe von machine learning, oder einer Kombination aus den beiden Ansätzen. Dabei sind sowohl die Tools gemeint, welche vom SEnti-Analyzer genutzt werden, als auch etwaige andere Tools.

Bei lexikonbasierten Tools existiert eine große Sammlung von vorher definierten Schlüsselwörtern oder -phrasen, die mit dem erhaltenen Input verglichen werden. Diese Sammlung von Vergleichswerten existiert im Vorhinein und bestimmt damit die Wertigkeit des erhaltenen Inputs. In dieser Sammlung steht für die Vergleichswerte welche Auswirkung bestimmte Wörter oder Phrasen haben, anhand der Häufigkeit mit der erkannte Wörter oder Phrasen im Input vertreten sind wird dann eine Bewertung über den Input gegeben.

Bei Tools die auf machine learning basieren kommt eine trainierte künstliche Intelligenz zum Tragen. Ähnlich wie bei lexikonbasierten Tools existiert eine Grundsammlung zur Bewertung. Dieser Grundwortschatz kann jedoch jederzeit mithilfe von Training verbessert werden. So können z.B. manuell eingestufte Daten dazu genutzt werden, um zukünftige Ergebnisse zu verbessern. Umso mehr Daten dabei für die künstliche Intelligenz zur Verfügung stehen, desto bessere Ergebnisse kann diese liefern. Die Wahrscheinlichkeit, dass dem Input das korrekte Sentiment zugewiesen wird steigt dabei dann an.

Stimmungsanalyse-Tools klassifizieren den Input. Üblich sind dabei unter anderem Einordnungen auf einer bipolaren Skala positiv, negativ oder auf einer tripolaren Skala, bei der noch neutral als eine dritte Wertung hinzukommt. Aber auch die Bewertung anhand eines Zahlenwerts wird oft genutzt, dabei wird meistens die Einordnung im Bereich -1, für negativ, bis +1, für positiv, vorgenommen, aber auch andere Bereiche werden genutzt.

Vorangegangen war nun eine Erklärung die erst einmal nicht direkt etwas mit Softwareentwicklung im allgemeinen oder Softwareprojekten im genauen zu tun haben. Im folgenden Abschnitt wird erklärt, welcher Zusammenhang zwischen Stimmungsanalyse und Softwareprojekten besteht.

2.1.1 Stimmungsanalyse in Softwareprojekten

Wie vorangegangen erklärt wird bei der Stimmungsanalyse Stimmung analysiert. Nun finden sich im Gebiet der Softwareentwicklung oder im genaueren bei Softwareprojekten keine Produkte, welche

bewertet werden sollen. Vielmehr kommt die Stimmungsanalyse hier als unterstützender Faktor in der Lenkung des Arbeitsalltages vor.

Eine wichtige Anmerkung ist dabei, dass es einen technischen Unterschied zwischen Meinungen und Stimmung gibt. Wie oben angemerkt ist bei dieser Arbeit eigentlich nur die Bezeichnung Stimmung korrekt, weswegen von hier an nur noch den Begriff Stimmung benutzt wird.

Ein wichtiger Faktor der zum Erfolg eines Softwareprojekts beiträgt, ist die Motivation der Entwickler*innen. Neben privaten oder äußeren Umständen, die auf die Motivation einer Person Einfluss nehmen, ist die Kommunikation innerhalb eines Teams bei einem Softwareprojekt von großer Bedeutung, was die Motivation der Entwickler*innen angeht. Und genau hier kommt die Stimmungsanalyse zum Tragen. Mithilfe von Stimmungsanalyse lässt sich die Stimmung innerhalb eines Meetings feststellen. Wenn dabei dann festgestellt wird, dass die Stimmung innerhalb des Teams oder auch von einzelnen Teammitgliedern negativ ist, können Maßnahmen ergriffen werden, um die Stimmung wieder zu heben und so die Produktivität zu steigern. Im Gegensatz dazu lässt sich so auch feststellen wenn die Stimmung positiv ist, wobei dann die Möglichkeit besteht Maßnahmen aufrechtzuerhalten, die im Moment getroffen werden. Insgesamt hat es große Vorteile, wenn die Stimmung innerhalb des Teams bekannt ist, damit die Arbeitsweise darauf angepasst werden kann.

Es existieren verschiedene Ebenen der Stimmungsanalyse. Beschrieben von Bing Liu et. al. [8] Warum diese Ebenen für den SEnti-Analyzer von Bedeutung sind, wird im folgenden zusammen mit den einzelnen Ebenen erklärt.

2.1.2 Level der Stimmungsanalyse

Entity and Aspect Level

Das Entity and Aspect Level ist das feinste Level der Analyse. Hier wird nicht nur die Stimmung bzw. Meinung selbst analysiert, sondern auch das Ziel der jeweiligen. Anstatt auf Muster wie Satzbau, Phrasen, Absätze etc. zu schauen wird hier feiner analysiert. Denn eine Stimmung bzw. Meinung ist nur dann auch etwas Wert, wenn das Ziel dieser bekannt ist. So ist der Satz „Das Essen war lecker.“ laut z.B. SentiStrength positiv, bezieht sich aber nur auf das Essen. Wenn es zum Beispiel um eine Hotelbewertung geht, dann sagt diese Aussage nichts über den Rest des Hotels aus. Wenn der

Informationsgehalt als [Sentiment, Ziel] beschrieben wird, wäre das [positiv, Essen]. Bei dem Beispiel des Hotels, wird es schwieriger bei einem Satz der Art „Insgesamt ein schöner Aufenthalt, aber das Essen war nicht lecker.“ denn dieser Satz enthält mehrere Informationen. SentiStrength bewertet den ersten Teil des Satzes „Insgesamt ein schöner Aufenthalt[...]“ als positiv und den zweiten Teil „[...]aber das Essen war nicht lecker“ als negativ. Der gesamte Satz wird jedoch positiv bewertet. Das bedeutet zwar, dass der Satz mehr positiv als negativ eingestuft wird, jedoch gehen Informationen verloren, wenn der Satz im gesamten betrachtet wird, wie es die nächste Ebene tut.

Sentence Level

Dieser Level der Analyse geht, wie der Name schon verrät, in die einzelnen Sätze hinein. Jedem einzelnen Satz lässt sich eine Bewertung zuordnen, welche sich aus der Gesamtheit der Entity and Aspect Level Bewertungen zusammensetzt. Hierbei lässt sich so herausfinden ob ein Satz ein subjektiver Satz ist der eine bestimmte Meinung oder eine Stimmung widerspiegelt oder ob es ein objektiver Satz ist, der einfach nur Fakten wiedergibt. Zu beachten ist dabei, dass es bei positiv und negativ eingestuften Sätzen klar ist welche Bewertung ausgedrückt wird. Bei neutral bewerteten Sätzen wird davon ausgegangen, dass diese keinerlei Meinung oder Stimmung wiedergeben. Möglich ist es potentiell jedoch, dass ein Satz gleichermaßen positiv wie negativ bewertet wird.

Document Level

Ähnlich wie der Sentence Level lässt auch der Name des Document Level schon vermuten worum es sich hierbei handelt. Auf der nächsthöheren Ebene werden sich alle Bewertungen des Sentence Level oder des Entity and Aspect Level angeschaut und zusammengefasst zu einem größeren Document Level. Ein kleiner Stolperstein ist im Namen jedoch zu finden, so muss es sich auf dieser Ebene keineswegs um ein tatsächliches Dokument handeln, welches analysiert wird. Eine Menge an Produktreviews kann ebenso gut widerspiegelt werden wie ein tatsächliches Textdokument. Beides sind in ihrer Gesamtheit auf dem Document Level. Anzumerken ist hierbei, dass es sich nur um ein einzelnes Subjekt, oben Ziel, handeln kann. So kann aus der Document Level Analyse von Reviews eines Produktes z.B. etwa keine Information zum Versand herausgezogen werden,

auch wenn es Anmerkungen zu diesem in einer Vielzahl von Reviews gab. Dazu wäre eine zweite Document Level Analyse des Produktes notwendig, die sich nur auf das Subjekt des Versands bezieht.

Der SEnti-Analyszer, welcher in dieser Arbeit beschrieben wird, analysiert die Stimmung grundlegend auf dem Sentence Level und abstrahiert sie dann auf das Document Level. Hierbei ist ein „Dokument“ ein Meeting, welches analysiert wird. Aufgrund der einzelnen Aussagen, die während einem Meeting getätigt werden, wird somit auf die Gesamtstimmung innerhalb des Teams geschlossen.

2.2 Technische Grundlagen

Zuerst wird in diesem Abschnitt der Begriff einer grafischen Oberfläche definiert, gefolgt von einer Erklärung welches Framework für die vorliegende Arbeit genutzt wurde. Für das Verstehen wie der SEnti-Analyszer funktioniert ist unter anderem die Kenntnis von zwei bestimmten Dateiformaten notwendig, weiterhin arbeitet der SEnti-Analyszer mit verschiedenen Tools zur Sentimentanalyse. Eine kurze Beschreibung der einzelnen Tools, wie diese arbeiten und Sentiments klassifizieren folgt später innerhalb des Abschnittes.

2.2.1 Grafische Oberflächen

Es gilt erst einmal die Frage zu klären, was eine grafische Oberfläche, englisch graphical user interface, kurz GUI, genau ist. Eine grafische Oberfläche ist eine Verbindung, eine Schnittstelle, zwischen dem Menschen und der Maschine. Der Zweck einer grafischen Oberfläche ist es die alltägliche Arbeit mit einem Endgerät oder einer Anwendung möglichst benutzerfreundlich zu gestalten. Besonders wichtig sind dabei Symbole und Bilder, damit eine textunabhängige Anwendung erreicht werden kann. Dies sorgt dafür, dass Elemente universell erkennbar sind, so ist beispielsweise ein Wi-Fi Symbol weitgehend von vielen Personen erkannt, auch wenn ein textueller Ausdruck dafür in verschiedenen Sprachen komplett unterschiedlich sein kann. Eine grafische Oberfläche vereinfacht die Kommunikation zwischen Mensch und Maschine, normalerweise indem es die Interaktion mit Maus und Tastatur ermöglicht. Die Maus wird durch die Nutzer*innen bewegt und der Zeiger auf dem Bildschirm bewegt sich dementsprechend. Ein anderes Wichtiges Element sind Programmbuttons oder Symbole. Eine grafische Oberfläche erlaubt

es den Nutzer*innen Funktionen des Programms aufzurufen ohne diese in die Kommandokonsole eingeben zu müssen.

2.2.2 GUI Programmierung

An dieser Stelle werden weniger die theoretischen Grundlagen, was genau alles zu beachten ist, wenn eine grafische Oberfläche programmiert wird, erklärt. Dafür ist es zu viel und es gibt kein richtiges Grundrezept für eine „gute“ grafische Oberfläche. Allgemein trifft hier der Satz „Es kommt darauf an“ in großem Maße zu, da jede grafische Oberfläche andere Anforderungen hat. Vielmehr wird hier erklärt, warum das Framework gewählt wurde, das gewählt wurde; und zwar PySimpleGui. Der Größte technische Teil liegt darin, dass der SEnti-Analyzer komplett in Python geschrieben ist. Aus Kompatibilitätsgründen ergibt es daher Sinn ein Framework zu nutzen, welches ebenfalls auf Python basiert.

Die nicht-technischen Gründe sind jedoch genauso wichtig zu erwähnen. Der Name verrät schon einen Punkt, es ist ziemlich simpel gehalten. Es lässt sich schnell in PySimpleGui einfinden und zurecht kommen. Weiterhin bietet es viele Schnittstellen zu anderen GUI Frameworks wie TKInter oder MathPlotLib. Der mit Abstand wichtigste Grund ist jedoch, dass keine finale Version der grafischen Oberfläche für den SEnti-Analyzer erstellt werden kann. Das wäre auch nicht möglich gewesen. Genau wie der SEnti-Analyzer sich selbst weiter entwickelt, so muss sich auch die grafische Oberfläche weiterentwickeln. Da dann im Zweifelsfall andere Personen an der grafischen Oberfläche arbeiten, ist ein Punkt sehr wichtig: PySimpleGui ist wirklich gut dokumentiert. Für alles was gebraucht wird, ist eine Antwort schnell findbar. Neben einer Definitionsseite, auf der alle Objekte, deren Parameter und Funktionen nachgeschaut werden kann, existiert es auch noch ein sogenanntes „Cookbook“, ein Kochbuch, von PySimpleGui. Dieses bietet einzelne Beispiele, wie die einzelnen Objekte verwendet werden könnten. Das hilft enorm beim Verstehen von PySimpleGui und daher auch beim Verstehen der grafischen Oberfläche vom SEnti-Analyzer.

2.2.3 Audiodateien

Der SEnti-Analyzer kann unter anderem Audiodateien einlesen und produzieren. Dies sind in beiden Fällen .wav Dateien. Dabei handelt es sich um Audiodateien die im Wellenformat (WAVEform)

speichern. Dies bedeutet, es handelt sich bei den Dateien um Rohdaten, welche viel größer als z.B. das bekannte mp3 Format sind. Dies bedeutet auch, dass die originale Audioqualität beibehalten werden kann. Ein besonderes Programm zur Öffnung von .wav Dateien wird nicht benötigt.

2.2.4 Textdateien

Neben Audiodateien arbeitet der SEnti-Analyzer auch mit Textdateien. Wie bei Audiodateien kann der SEnti-Analyzer diese Dateien sowohl einlesen als auch produzieren. Der verwendete Dateityp ist hierbei .csv, was für "comma-separated-values" oder auch selten für "character-separated-values", steht. .csv Dateien werden benutzt, um einfache Datensätze zu speichern und folgen einem bestimmten Schema im Aufbau: Ein Zeichen wird festgelegt, um die Datensätze voneinander zu trennen, dies ist meist ein Zeilenumbruch. Ein Zeichen wird festgelegt, um Datenfelder voneinander zu trennen, dies kann grundsätzlich jedes beliebige Zeichen sein, klassischerweise wird jedoch ein Komma, Semikolon, Doppelpunkt, Tabulatorzeichen oder Leerzeichen verwendet. Im Falle des SEnti-Analyzers können dabei für das Einlesen der Datei sowohl Semikolons oder Kommata verwendet werden, jedoch muss darauf geachtet werden, dass nicht beides innerhalb derselben Datei verwendet wird. Die vom SEnti-Analyzer produzierten .csv Dateien nutzen immer Semikolons

Außerdem besteht die Möglichkeit die erste Zeile als Kopfzeile zu verwenden. Dies ist sowohl beim Einlesen als auch bei den produzierten Dateien vom SEnti-Analyzer der Fall. Wenn eine .csv Datei eingelesen werden soll, so muss diese eine Spalte haben die mit „Text“ beschriftet ist, ansonsten funktioniert der SEnti-Analyzer nicht. Alle anderen Spalten werden dabei ignoriert, sowohl bei Dateien die der SEnti-Analyzer selbst produziert hat als auch bei externen Dateien. Ein Beispiel wie eine Datei aufgebaut sein muss folgt:

[...]	Text	[...]
[...]	Dies ist ein Beispiel	[...]
[...]	Welches Format eine	[...]
[...]	einzelnde Datei haben	[...]
[...]	müsste um analysiert	[...]
[...]	werden zu können	[...]

Die einzelnen Spalten müssen dabei, wenn mehr als eine Spalte vorhanden ist, von Semikolons getrennt werden. Ein besonderes

Programm zur direkten Bearbeitung einer .csv Datei ist ebenfalls nicht nötig.

2.2.5 Die Einzelnen Tools

Alle Sentimentanalyse-Tools arbeiten entweder lexikonbasiert oder mithilfe von machine learning, oder einer Kombination aus den beiden Ansätzen. Dabei sind sowohl die Tools gemeint, welche vom Senti-Analyzer genutzt werden, als auch andere etwaige Tools.

Bei lexikonbasierten Tools existiert eine große Sammlung von vorher definierten Schlüsselwörtern oder -phrasen, die mit dem erhaltenen Input verglichen werden. Diese Sammlung von Vergleichswerten existiert im Vorhinein und bestimmt damit die Wertigkeit des erhaltenen Inputs. In dieser Sammlung steht für die Vergleichswerte, welche Auswirkung bestimmte Wörter oder Phrasen haben. Anhand der Häufigkeit mit der die erkannte Wörter oder Phrasen im Input vertreten sind, wird dann eine Bewertung über den Input gegeben.

Bei Tools die auf machine learning basieren kommt eine trainierte künstliche Intelligenz zum Tragen. Ähnlich wie bei lexikonbasierten Tools existiert eine Grundsammlung zur Bewertung. Dieser Grundwortschatz kann jedoch jederzeit mithilfe von Training verbessert werden. So können z.B. manuell eingestufte Daten dazu genutzt werden, um zukünftige Ergebnisse zu verbessern. Umso mehr Daten dabei für die künstliche Intelligenz zur Verfügung stehen, desto bessere Ergebnisse kann diese liefern. Die Wahrscheinlichkeit, dass dem Input das korrekte Sentiment zugewiesen wird, steigt dabei dann an.

Der Senti-Analyzer benutzt für die Klassifizierung der Sentiments verschiedene Tools: Insgesamt sind es 4 Tools bei deutscher Sprachanalyse: GerVADER[7], BertDE, SentiStrengthDE und TextBlobDE; und 5 bei englischer: Senti4SD[5], SentiStrength[1], SentiStrengthSE, SentiCR[2] und TextBlob[9].

GerVADER ist eine deutsche Adaption von VADER und ist damit ein regel- und lexikonbasiertes Tool welches den Input mithilfe dieser Verarbeitet. Dabei wird eine Einordnung in positiv, neutral, negativ vorgenommen. Der Senti-Analyzer stellt diese Einordnung als Zahl zwischen 1.0, am meisten positiv und -1.0, am meisten negativ, dar. Zwar ist VADER, und damit auch GerVADER, auf social media fokussiert, findet hier aber trotzdem Anwendung.

BertDE ist, ebenfalls, eine deutsche Adaption, von Bert. Bert steht dabei für „Bidirectional Encoder Representations from Trans-

formers“ und arbeitet mit einem machine-learning-Modell. Die Einordnung der Sentiments erfolgt ebenfalls in positiv, neutral, negativ, wobei dies auch vom SEnti-Analyzer als solches weitergegeben wird.

SentiStrengtDE ist die deutsche Adaption von SentiStrength. SentiStrength hat eine Website zum Selbsttesten der Analyse, [“http://sentistrength.wlv.ac.uk/“](http://sentistrength.wlv.ac.uk/), und ist in insgesamt 16 Sprachen, darunter Englisch und Deutsch, verfügbar. SentiStrength weist einem Satz zwei Werte zu, einen positiven und einen negativen. Diese gehen im positiven von 1 (nicht positiv) bis 5 (extrem positiv) und im negativen von -1 (nicht negativ) bis -5 (extrem negativ). Um die Stimmung des gesamten Satzes zu bestimmen werden diese beiden Werte miteinander aufaddiert, so ergibt ein Satz mit positivem Wert 3 und negativem Wert -2 einen Gesamtwert von 1 und ist damit insgesamt leicht positiv. Der SEnti-Analyzer stellt dies wieder als Zahl zwischen 1.0 und -1.0 dar.

Darauf aufbauend existiert SentiStrengthSE, dies ist eine Erweiterung von SentiStrength. Dabei wurde das zugrundeliegende Lexikon so verändert, dass die Sentimentzuweisung spezialisiert auf den Bereich des Software Engineerings angepasst ist.

TextBlobDE ist die deutsche Erweiterung zu TextBlob. TextBlob arbeitet ebenfalls mit einem Lexikon, um Textinput zu verarbeiten. Die Klassifizierung fällt wieder in positiv, neutral, negativ und wird wieder als Zahl zwischen 1.0 und -1.0 vom SEnti-Analyzer dargestellt.

Senti4SD ist ein Analysetool, welches mithilfe von machine learning speziell darauf trainiert wurde Sentimentanalyse bei der Kommunikation zwischen Softwareentwickler*innen durchzuführen.

SentiCR [2] ist ein Analysetool, welches ebenfalls mithilfe von machine learning trainiert wurde. Spezialisiert ist SentiCR auf das Analysieren von Kommentaren bei Code Reviews. Zusammen mit SentiStrengthSE und Senti4SD gehört es damit zu den Analysetools die speziell darauf abzielen im Bereich der Softwareentwicklung tätig zu werden.

2.3 Der SEnti-Analyzer vor der GUI

Der SEnti-Analyzer ist ein von Marc Herrmann und Jil Klünder entworfenes Tool [6] für die Analyse von Sentiments in Softwareprojekt-Meetings.

Der Hintergedanke des Tools ist es, dass es von Vorteil ist, wenn Sentiment Analyse bei Live Audio durchgeführt werden kann, z.B.

bei Softwareprojekt-Meetings. Da es schon viele Tools gibt die dies für Text machen, aber noch nicht viele die dies auch für Live Audio oder Audiodateien können wurde der SEnti-Analyzer geschaffen.

Der SEnti-Analyzer ist in Python geschrieben, zur Nutzung des Tools wird die Kommandokonsole gestartet und die Parameter mit denen der SEnti-Analyzer arbeiten soll eingegeben. Dann wird noch das Python Script gestartet.

2.3.1 Die Funktionen des SEnti-Analyzers

Der SEnti-Analyzer wird grundsätzlich mit dem Befehl „python senti_analyzer.py“ gestartet. Mit diesem Befehl startet der SEnti-Analyzer ohne weitere Optionen und auf Englisch.

Neben dem eigentlichen Befehl existiert die Möglichkeit folgende Parameter mit zu übergeben: `-w`, `-f FILE`, `-g` und `-csv FILE`. Alle diese Parameter sind optional und greifen untereinander ein.

Der Parameter `-w` oder auch `-savewav` bestimmt, dass der SEnti-Analyzer die aufgenommenen Audiodateien als einzelne `.wav` Dateien abspeichert.

Der Parameter `-f FILE` oder auch `-file FILE` wird dazu benutzt, um, statt eine Audioaufnahme zu starten, eine schon vorhandene `.wav` Datei einzulesen. Das Programm behandelt die Datei dabei so, dass die `.wav` Datei als Mikrofoneingabe gewertet wird. Dabei ist `-f` beziehungsweise `-file` der Parameter selbst und `FILE` der Pfad der zu analysierenden Datei.

Der Parameter `-g` oder auch `-german` legt fest auf welcher Sprache das Tool die Analyse durchführt. Ohne diesen Parameter ist die gewählte Sprache Englisch und mit dieser ist die Sprache Deutsch.

Der Parameter `-csv FILE` oder `-csvfile FILE` wird dazu benutzt, um, statt einer Audioaufnahme zu starten, eine schon vorhandene `.csv` Datei einzulesen. Der Unterschied im Programmverhalten zum Einlesen einer `.wav` Datei liegt dabei darin, dass der Prozess einer Mikrofonaufnahme komplett übersprungen wird. Dabei ist wie `-f FILE` oder `-file FILE` `-csv` oder `-csvfile` der Parameter und `FILE` der Pfad zu der gewählten Datei.

Wenn nun das Tool gestartet ist und eine Audioaufnahme beginnt, wird einem auf der Konsole angezeigt, dass gerade eine Aufnahme stattfindet. Wenn nun gesprochene Sprache erkannt wird, wird diese auf der Konsole ausgegeben und eine vorübergehende Einschätzung des Sentiments wird angezeigt. Im Verlauf dieser Arbeit wird dies als eine Aussage bezeichnet. Die Aufnahme findet

so lange statt bis der Satz „Aufnahme Beenden“ im Deutschen oder „Stop recording“ im Englischen erkannt wird. Andernfalls lässt sich die Aufnahme auch mit der Tastenkombination „STRG + C“ beenden.

Wenn die Audioaufnahme beendet wurde, geht das Tool in die Nachbearbeitung über. Dort wird unter anderem ein Spellchecker benutzt um das Aufgenommene zu korrigieren. Danach findet eine umfassende Sentimentanalyse durch alle Tools statt. Wenn der SEnti-Analyzer auf Deutsch läuft, sind das die Tools GerVADER, BertDE, SentiStrengthDE und TextBlobDE. Im Englischen sind es die Tools Senti4SD, SentiStrength, SentiStrengthSE, SentiCR und TextBlob. In beiden Fällen wird ein Durchschnitt aller Tools gebildet, um eine Gesamteinschätzung zu finden. Diese kann teilweise auch von der vorübergehenden Einschätzung abweichen, die während der Audioaufnahme getroffen wird, da in der vorübergehenden Einschätzung nur die Einschätzung von TextBlobDE im deutschen und TextBlob im Englischen benutzt wird.

Dazu wird tabellarisch dargestellt, was die Ergebnisse sind. In der ersten Spalte eine Nummerierung der Aussagen, in der zweiten Spalte der jeweilige Satz oder Teilsatz. In den darauf folgenden Spalten befinden sich die Bewertung der einzelnen Tools und in der letzten das errechnete Mittel.

Weiterhin wird eine Übersicht über die Gesamtheit gegeben, eine kleine Übersicht darüber wie viele Aussagen jeweils positiv, negativ oder neutral waren. Dies wird einmal als absolute und einmal als prozentuale Zahl dargestellt, wobei von 0.0 für 0% bis 1.0 für 100% gerechnet wird.

Über die Analyse der Eingabe wird auch eine .csv Datei namens „predictions“ erzeugt, in der die Ergebnisse gespeichert werden. Diese Datei lässt sich auch vom SEnti-Analyzer wieder einlesen.

Kapitel 3

Verwandte Arbeiten

Diese Arbeit basiert auf der Arbeit von Marc Hermann und Jil Klünder[6]. In dieser wird der SEnti-Analyzer beschrieben. Diese Bachelorarbeit basiert auf dem SEnti-Analyzer auf, da für diesen eine grafische Oberfläche erstellt wird. Neben dem SEnti-Analyzer existieren auch andere Tools, welche eine grafische Oberfläche besitzen. Zum Beispiel die Website von SentiStrength[1]. Der Unterschied dabei ist, dass diese Arbeit eine grafische Oberfläche als eine Anwendung produziert.

Kapitel 4

Anforderungsanalyse

Um eine möglichst passende grafische Oberfläche zu entwickeln, wurde, begleitend zum Programmieren, eine Anforderungsanalyse durchgeführt. Ein Teil davon sind subjektive Vorstellungen davon, wie die grafische Oberfläche im Resultat auszusehen hat und ein Teil davon sind Interviews, die mit drei Mitarbeiter*innen des Fachgebiets Software Engineering, welche näher mit dem Senti-Analyzer zu tun haben, durchgeführt wurden.

4.1 Im Vorfeld gesammelte Ideen

Um die Interviews vorzubereiten wurden im Vorfeld subjektive Ideen für die grafische Oberfläche gesammelt. Grundsätzlich ist die Idee die grafische Oberfläche in zwei Teile aufzuteilen. Ein Hauptmenü, welches den bisherigen Kommandokonsolen-Input ersetzt, und einen Analysebildschirm, welcher das darstellt, was bisher während der Programmnutzung auf der Kommandokonsole ausgegeben wurde.

4.1.1 Hauptmenü

Das Hauptmenü sollte alle schon genannten Funktionen erfüllen: das Einstellen der Sprache, das Einlesen von sowohl .csv Text- als auch .wav Audiodateien und die Option einzelne erkannte Aussagen als .wav Dateien zu speichern. Da die Sprachauswahl sich auf Deutsch und Englisch beschränkt braucht es hier auch keinem komplizierten Menü, sodass die Sprachauswahl als zwei Buttons dargestellt werden kann, wovon jeweils nur einer Aktiv sein darf. Weiterhin sollte es einen Button für den Start der Audioaufnahme geben und die dazugehörige Option diese als .wav Dateien zu speichern.

Das Einlesen von Dateien findet ebenfalls über einen Button statt, welcher den Dateibrowser öffnet.

Zuletzt ist auch noch ein Button für das Beenden des Programms geplant, um das Beenden der Software zu vereinfachen.

Ein erster Mockup wurde erstellt:

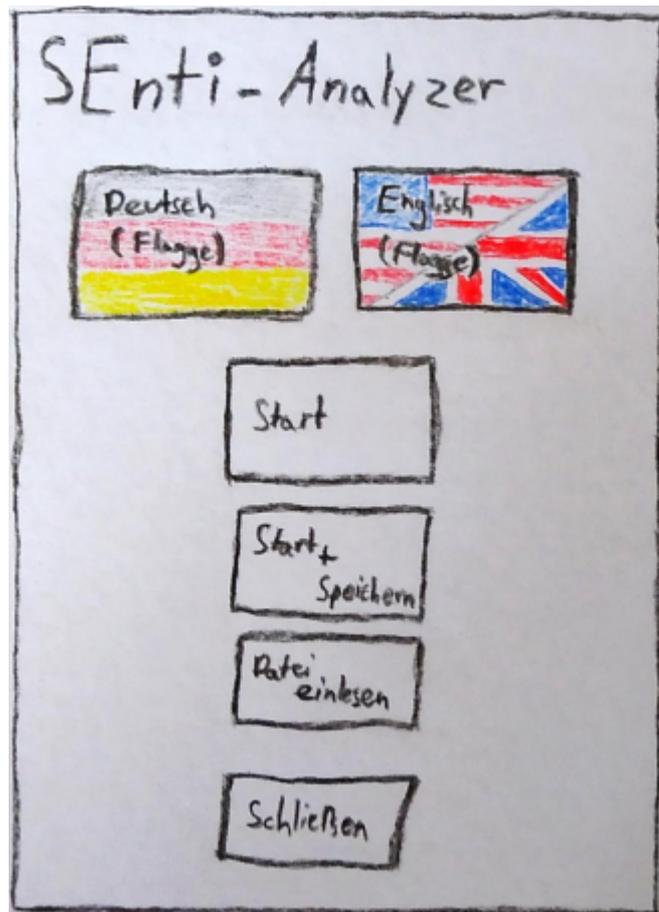


Abbildung 4.1: Mockup für das Hauptmenü

4.1.2 Prozessbildschirm

Der Prozessbildschirm sollte auch alle schon benannten Funktionen bieten. Das, was vorher auf der Kommandokonsole ausgegeben wurde, sollte dargestellt werden. Einmal die erkannten Aussagen mit der Vorauswertung während einer Audioaufnahme und nachdem eine Aufnahme beendet wurde, eine Art tabellarische Darstellung der genauen Ergebnisse, mit einer Übersicht von absoluten und relativen Werten, von erkannten positiven, neutralen und negativen

Aussagen. Die Gesamtübersicht wurde auf der Konsole schriftlich ausgegeben und die Idee ist es dies in ein Balkendiagramm zu überführen, damit die Ergebnisse auf den ersten Blick einschätzbar sind. Weiterhin ist eine grafische Darstellung eines gesamten Meetings geplant, wobei im Vorfeld unsicher war, was die beste Möglichkeit ist dies darzustellen. Im Vorfeld gab es Ideen zu zwei Möglichkeiten ein gesamtes Meeting darzustellen. Einmal als Zeitverlauf der einzelnen erkannten Sentiments und einmal als Zeitverlauf der Gesamtstimmung des Meetings.

Insgesamt war es so geplant, dass der Bildschirm zweigeteilt ist; auf der linken Bildschirmhälfte das, was während der Audioaufnahme erkannt wird und rechts die ausführliche Auswertung, wenn die Audioaufnahme beendet wurde.

Ebenfalls ein Mockup wurde erstellt:

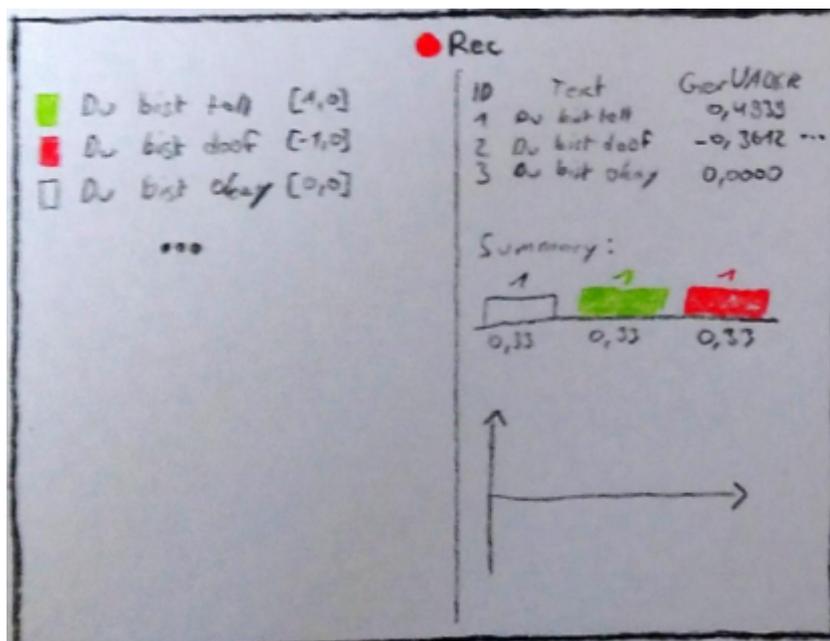


Abbildung 4.2: Mockup für den Prozessbildschirm

4.2 Interviews

Um weitere Meinungen und Sichtpunkte zu bekommen, wurden, wie erwähnt, Interviews durchgeführt. Ziel war es festzustellen, ob die im Vorfeld gesammelten Ideen genügen, oder, ob es Änderung bedarf. Die Gespräche wurden mit Einwilligung der Interviewten

aufgezeichnet.

Angefangen wurde damit, dass nachgefragt wurde, ob sie mit eigenen Worten kurz beschreiben können, was der SEnti-Analyzer ist, gefolgt von der Frage, ob sie wissen und wiedergeben können, was die Features des SEnti-Analyzers sind. Getan wurde dies, um eine gemeinsame Basis für das Interview herzustellen. Da vorher nicht gewusst werden konnte, inwiefern die Interviewten mit dem SEnti-Analyzer vertraut sind, musste sichergestellt werden, dass keine Missverständnisse auftreten was Features des SEnti-Analyzers angeht.

Als dies sichergestellt war, konnte zu der Frage übergegangen werden, was nach der Meinung der Interviewten wichtig für die grafische Oberfläche ist. Alle Interviewten waren sich Grundsätzlich einig, alle zuvor beschriebenen Funktionen, die auf der Kommando-konsole verfügbar waren, sollten auch in der grafischen Oberfläche verfügbar sein. Dabei wurde von einer Person der Fokus vor allem auf die Funktion vom Einlesen von Audio- und Textdateien gelegt.

Neben dem, was die grafische Oberfläche können muss, war auch von Interesse, ob die Interviewten andere Qualitätsvorstellungen haben, die sie sich für die grafische Oberfläche wünschen würden. Dabei wurde nur Intuitivität im Hinblick auf Usability erwähnt.

Nach diesen sehr offenen Fragen sollte es etwas konkreter werden. Den Interviewten wurden die Vorschläge von oben gezeigt. Beginnend mit dem Hauptmenü wurde bei Bedarf kurz erklärt, was die Gedanken waren und danach konnten die Interviewten ihre eigenen Ideen oder Änderungen hervorbringen und in die Vorschläge einzeichnen, wenn sie das denn wollten.

Das Konzept des Hauptmenüs war allen Interviewten direkt klar. Eine Sprachauswahl mit dazugehörigen Buttons und die einzelnen Funktionen werden durch einzelne Buttons angesteuert. Dabei kam allerdings einige Verwirrung auf. Den Interviewten war nicht sofort klar was mit „Start“ und „Start und Speichern“ im obigen Mockup4.1 gemeint war. Eine Klarifizierung welcher Button welche Funktion auslöst, wurde angesprochen. Weiterhin wurde erwähnt, dass es eine Verdeutlichung der gewählten Sprache geschehen sollte. Darüber hinaus kam die Nachfrage danach, was genau von der Sprachauswahl geändert wird. Ob dabei nur die Systemsprache oder auch die Analysesprache geändert wird, war nicht klar, dies sollte ebenfalls angezeigt werden. Ein Wunsch für das Hauptmenü war auch, dass angezeigt wird welche Tools in welcher Sprache arbeiten. Dazu käme die Funktion einzelne Tools an- und abzuwählen, die für

die Analyse benutzt werden sollen. Für das Einlesen der Datei wurde sich eine Sicherheitsstufe gewünscht, damit nicht aus versehen eine falsche Datei analysiert wird, die gar nicht analysiert werden soll und die möglicherweise viel Zeit und Rechenleistung benötigt. Zuletzt kam noch eine Äußerung dazu, dass das „SE“ vom Schriftzug „Senti-Analyzer“ durch das Logo des Fachgebiets Software Engineering ersetzen könne, das sei allerdings „[...]eine absolute Spielerei.“ .

Beim Analysefenster gab es etwas mehr Erklärungsbedarf. Nach einer Erklärung der Idee kam die Rückmeldung der Interviewten. Zum linken Teil des Analysebildschirms kam nur die Rückmeldung, dass die getroffene Vorauswahl verdeutlicht werden sollte. Was vorher auf der Kommandokonsole nur ein kleines einfarbiges Kästchen in grün für positiv, weiß für neutral und rot für negativ war, muss ebenfalls dargestellt werden. Das Problem an dieser Darstellung ist, dass kleine einfarbige Kästchen teilweise schwer zu erkennen sind. Besonders kompliziert wird es dabei, wenn die zu unterscheidenden Farben Rot und Grün sind. Da etwa 9% der männlichen und etwa 0,8% der weiblichen Bevölkerung an einer Protanopie, einer Rotblindheit, Protanomalie, einer Rotschwäche, Deuteranopie, einer Grünblindheit, oder Deuteranomalie, einer Grünschwäche leiden bedarf es hier einer Klarifizierung der einzelnen Icons.

Bei der Tabellarischen Darstellung wurden mehrere Sachen angemerkt. Zum einen wurde hier wieder der Wunsch nach einer Manipulationsmöglichkeit der Ergebnisse geäußert, also, dass im Nachhinein noch auswählbar ist, welche Tools gerade für die Analyse benutzt werden. Ebenfalls für die tabellarische Darstellung kam der Einwand, dass es simplifiziert werden könnte. Einmal die Rede von einem „Nutzermodus“ und einem „Entwicklermodus“ oder einfach nur das Ein- und Ausblenden der Details. Für die Nutzer*innen, die das Tool in Meetings verwenden, um die Stimmung festzustellen, ist egal welches der Tools welches genaue Ergebnis errechnet hat. Für diese Art von Nutzer*in ist nur wichtig, was das Gesamtergebnis, also der Median, ist und diese Art von Nutzer*in kann auch nichts mit irgendwelchen Zahlen anfangen. Das Einzige, was in diesem Fall wichtig ist, ist die Einordnung in positiv, negativ und neutral. Für die Graphen kam positive Rückmeldung für beide Arten und es kam die Frage auf, ob es Möglich ist mit dem Graphen zu interagieren, wie etwa auf einen Punkt im Graphen zu klicken um dann die dazugehörige Aussage angezeigt zu bekommen. Neben weiteren Vorschlägen für eine Funktion ein laufendes Meeting zu pausieren, angelehnt an etwas wie eine „Mute-“ oder „Stummschalten-“ Funktion und

einer Idee dafür, die Endergebnisse zu exportieren, zum Beispiel als Bild, haben sich die Interviewten was einen Aspekt der Analyse angeht sogar komplett widersprochen. Das was auf der linken Bildschirmhälfte ausgegeben wird, passiert zurzeit in Echtzeit. Dabei kam die Frage auf, ob es möglich wäre, auch den rechten Teil komplett in Echtzeit ablaufen zu lassen. Sowohl den Graphen, das Balkendiagramm als auch die Tabelle in Echtzeit zu bekommen war auf der einen Seite zwar als positiv aufgefasst, auf der anderen Seite kam allerdings auch die komplett endgegengesetzte Meinung. Wenn sich während einem Meeting etwas auf dem Bildschirm bewegt und das an 5 verschiedenen Stellen, könnte das unnötig vom Geschehen des eigentlichen Meetings ablenken. Es wurde sich also auch ausdrücklich dagegen ausgesprochen die Analyse komplett in Echtzeit ablaufen zu lassen.

Um den Rahmen für zukünftige Arbeiten am SEnti-Analyzer zu legen wurde abschließend gefragt, was sich die Interviewten in Zukunft generell für den SEnti-Analyzer wünschen würden, auch über eine grafische Oberfläche hinaus. Etwas was sich wohl immer weiter verbessern lässt ist die Spracherkennung und das zuverlässige Treffen der Sentimentzuweisung.

Für zum Beispiel größere Meetings wurde sich eine Möglichkeit gewünscht, mehrere Instanzen des SEnti-Analyzers miteinander zu verbinden. Also, dass mehrere Personen zum Beispiel mit ihren eigenen Laptops den SEnti-Analyzer als eine Gesamtinstanz verwenden können. Außerdem wäre eine Möglichkeit mehrere Meetings miteinander zu vergleichen schön, damit unter anderem auch festgestellt werden kann, ob es Fortschritt über mehrere Meetings gibt. Weitere Features die hinzugefügt werden könnten, wären die Möglichkeit der genauen Zeitdarstellung in der Gesamtanalyse und eine sogenannte „Sorgenfunktion“, also eine automatische Erkennung von zum Beispiel einer Kette an besonders schlimmen Aussagen oder Ähnlichem. Dies ließe sich auch im positiven Sinne einsetzen, um festzustellen, wenn sich eine Person besonders positiv verhält. Abschließend ist ein wesentlicher Punkt der in Zukunft geschehen sollte, dass ein ausführlicher praktischer Test des SEnti-Analyzers durchgeführt wird, um festzustellen, ob der SEnti-Analyzer auch das tut was er tun möchte, Teams in Meetings dabei helfen, festzustellen, wie die Stimmung ist. Idealerweise wird der SEnti-Analyzer dabei von Personen getestet, die die Kunden für den SEnti-Analyzer darstellen würden.

4.3 Anpassung der gesammelten Ideen

Neben den ursprünglichen Ideen existieren nun auch die Blickwinkel von drei weiteren Personen, sodass die originalen Ideen überarbeitet werden konnten. Was das Hauptmenü angeht, gab es einiges an Feedback, wovon das meiste ohne weitergehende Überlegung übernommen werden konnte. Klarifikation der Sprachauswahl, Umbenennung der Buttons zur Verdeutlichung, etwa soll „Start“ in „Starte Audioaufnahme“ geändert werden, und Umgestaltung des Layouts, sowie das Hinzufügen einer Sicherheitsstufe davor, dass die falsche Datei analysiert wird, sind alles Dinge, die Sinn ergeben. Die ausgewählte Sprache soll heller dargestellt werden als die nicht-ausgewählte, das Speichern von .wav Dateien passiert über eine Checkbox und nicht über einen Eigenen Button und wenn der Dateibrowser geöffnet und eine Datei ausgewählt wird, dann soll die ausgewählte Datei erstmal nur angezeigt und dann separat gestartet werden. Der Analysebildschirm hat von der allgemeinen Gestaltung wenig Feedback erhalten, sondern mehr Feedback dazu, was für Möglichkeiten gegeben sind. Die einzige designtechnische Änderung, die für den Analysebildschirm vorgenommen werden soll, ist die Klarifizierung der Darstellung der vorbestimmten Sentimentzuweisung, im Hinblick auf das Farbsehen von roten und grünen Farben. Was vorher einfach nur einfarbige Kästchen waren, muss sich außerhalb der Farbe sonst noch irgendwie unterscheiden. Nach Überlegung welche Differenzierung vorgenommen werden könnte, um die beiden Sentimente positiv und negativ voneinander zu unterscheiden, wurde entschieden, dass in die grüne Darstellung für positive Sentimente ein in einem helleren Grünton erscheinendes „+“ zu platzieren. In der roten Darstellung für negative Sentimente wird ein gelbes „X“ zu sehen sein. Gewählt sind die Symbole, da ein „+“ einheitlich für etwas Positives und ein „X“ einheitlich für etwas Negatives steht. Bei der Farbwahl war ein helleres Grün ausreichend um den Unterschied darzustellen, bei Rot und Orange hingegen ist der Unterschied möglicherweise zu klein, weswegen Gelb gewählt wurde. Die einfarbige weiße Darstellung für neutrale Sentiments bleibt unverändert.

In gewisser Weise sinnvoll sind die erwähnten Funktionen sicherlich alle, da die meisten den Nutzer*innen nur mehr Möglichkeiten geben. Solange der Senti-Analyzer bei der Benutzung nicht unnötig kompliziert oder unübersichtlich wird, hat es keine Nachteile, mehr Funktionen anzubieten. Es ist grundsätzlich schwer, eine Software

oder ein anderes Produkt so zu entwickeln, dass jede einzelne Person die diese Software oder dieses Produkt benutzt, damit vollständig zufrieden ist. Aus diesem Grund ist es von Vorteil, wenn die Möglichkeit zu mehreren Funktionen angeboten wird. Ob der Kunde/die Kundin sie dann im Endeffekt nutzt, ist ihr/ihm überlassen.

Welche der Funktionen im Analysebildschirm tatsächlich Anwendung finden hängt allerdings auch von den technischen Gegebenheiten ab. An dieser Stelle wurden die erwähnten Funktionen erst einmal nur aufgenommen, welche tatsächlich im SEnti-Analyzer zu finden sind, klärt sich im nächsten Abschnitt.

Kapitel 5

Entwicklungsprozess der GUI

An dieser Stelle wird dargestellt, wie der SEnti-Analyzer aufgebaut ist, nachdem die grafische Oberfläche fertiggestellt war. Weiterhin folgt eine Darlegung, welche Features eingebaut wurden und welche nicht.

5.1 Programmstruktur

Die Struktur des SEnti-Analyzers hat sich nicht verändert. Der Programmcode, welcher die eigentliche Analyse durchführt und von Marc Hermann geschrieben wurde, ist in seiner Grundfunktion erhalten.[6] Lediglich einige kleine Änderungen wurden vorgenommen, um die Kommunikation zwischen der grafischen Oberfläche und dem eigentlichen Analyseprozess zu vereinfachen oder zu ermöglichen. Wie in der Idee vorangehend erläutert, ist die grafische Oberfläche eine Sicht auf die Dinge, die der Analyseprozess im Hintergrund durchführt. Dabei wurden die ursprünglichen Funktionen beibehalten und durch die grafische Oberfläche werden diese nun dargestellt. Der Programmablauf wird dabei mithilfe von Threads realisiert. Die Oberfläche nimmt dabei die Rolle des Hauptthreades ein, da dies der Teil des SEnti-Analyzers ist, mit dem die Nutzer*innen interagieren. Zu diesem Zweck ist die eigentliche Funktion des SEnti-Analyzers in einen Thread verschoben worden. Genauer gesagt ist der Prozess ein sogenannter Daemon, welches ein Thread ist, dessen Laufzeit an den Hauptthread gebunden ist. Sobald die grafische Oberfläche geschlossen wird, schließt sich sofort der Hintergrundprozess, unabhängig davon, ob dieser fertig ist oder nicht.

Das Hauptmenü des SEnti-Analyzers ist so realisiert, wie in der

Idee beschrieben, es ersetzt den Kommandokonsolen-Input, der für das Starten des SEnti-Analyzers vorher erforderlich war. Solange aus dem Hauptmenü keine Funktion ausgewählt wird, welche das eigentliche Tool startet oder den SEnti-Analyzer beendet, bleibt das Hauptmenü offen und wartet auf die Eingabe oder die Auswahl einer Funktion durch die Benutzer*innen. Das Hauptmenü sieht endgültig wie folgt aus:



Abbildung 5.1: Hauptmenü

Dabei sind die meisten Funktionen ebenfalls so umgesetzt, wie in der Anforderungsanalyse beschrieben; durch das Klicken auf die Flaggen wird die Sprache ausgewählt, standardmäßig ist dabei die englische Sprache ausgewählt. Um dies zu ändern müsste der Programmcode geändert werden, zu diesem Zweck existiert in

der „gui.py“ Datei eine globale Variable, welche vorgibt ob die Startsprache Deutsch sein soll oder nicht.

Die Funktion eine Audioaufnahme zu starten existiert weiterhin, jedoch wurde im Gegenzug zum Aufruf durch die Kommandokonsole hinzugefügt, dass standardmäßig die aufgenommenen Teilsätze als .wav Dateien gespeichert werden. Beim Aufrufen durch die Kommandokonsole musste dies erst als zusätzliche Option ausgeführt werden. Zusätzlich wurde die Möglichkeit hinzugefügt, bei einer Audioaufnahme einen Namen zu wählen, unter dem das gesamte Meeting gespeichert werden soll, wenn die Aufnahme fertig ist. Zuvor wurden die Ergebnisse jedes Meetings in der „predictions.csv“ Datei gespeichert, wobei alte Ergebnisse dann überschrieben wurden. Bei keinem eingegebenen Namen wird „predictions“ als Name für die Datei gewählt.

Zusätzlich darunter der Button ist dafür zuständig, den Dateibrowser des aktuellen Betriebssystems zu öffnen, um dort eine Audio- oder Textdatei auszuwählen. Dabei ist der geöffnete Standardordner der „SEnti-Analyzer/output“ Ordner. Dieser wurde aus dem Grund gewählt, da die vom SEnti-Analyzer selbst erzeugten Dateien im „vSEnti-Analyzer/output“ oder im „SEnti-Analyzer/output/recordings“ abgespeichert werden. In ersterem landen die .csv Dateien, in letzterem werden die aufgenommenen .wav Dateien vom Programm abgespeichert.

Geblichen ist die Funktion, dass der SEnti-Analyzer nur .csv und .wav Dateien analysieren kann, aus diesem Grund werden einem nur standardmäßig nur die .csv Dateien im „SEnti-Analyzer/output“ Ordner angezeigt. Beim Wechsel in den „SEnti-Analyzer/output/recordings“ Ordner sind dort die .wav Dateien auswählbar. Es ist nicht möglich Dateien auszuwählen welche nicht dem .csv oder .wav Format angehören, jedoch existiert eine weitere Abfrage mit einem Popup, welches signalisiert, dass ein falscher Dateityp ausgewählt wurde, falls dies doch geschehen sollte. Für den Fall, dass extern erstellte Dateien analysiert werden sollen, muss sichergestellt sein, dass diese entweder dem .csv Format, mit einer Spalte namens „Text“ , oder dem .wav Format angehören. Externe Dateien lassen sich entweder manuell in den „SEnti-Analyzer/output“ Ordner verschieben um einfacher gefunden zu werden, aber auch die Manövrierung zu anderen Ordnern und Dateien die sich auf dem Computer befinden ist möglich.

Wenn eine Datei ausgewählt wird, wird diese jedoch nicht sofort analysiert, sondern erst im Textfeld rechts daneben angezeigt.

Dies soll verhindern, dass versehentlich falsche Dateien analysiert werden. Auch muss darauf geachtet werden, dass die richtige Sprache ausgewählt ist, wenn eine Datei analysiert werden soll, da keine automatische Erkennung stattfindet. Weiterhin lässt sich sowohl keine Datei analysieren, wenn zuvor keine ausgewählt ist, als auch die Audioaufnahme starten, obwohl eine Datei ausgewählt ist. Falls eine Audioaufnahme gestartet wird während eine Datei ausgewählt ist, wird die ausgewählte Datei ignoriert und der SEnti-Analyzer startet nur die Audioaufnahme. Die Auswahl der Dateien sieht folgendermaßen aus:

Name	Änderungsdatum	Typ	Größe
 recordings	09.05.2022 01:07	Dateiordner	
 Testmeeting_60_min	08.05.2022 21:31	CSV-Datei	27 KB
 Testmeeting_90_min	04.05.2022 19:20	CSV-Datei	82 KB

Abbildung 5.2: Ansicht beim Auswählen einer Datei

Zum Beenden des SEnti-Analyzers ist sowohl der programminterne Beenden-Button als auch das in jedem Programm existierende „X“ in der oberen rechten Ecke nutzbar.

Die Idee des Analysebildschirms bestand ursprünglich daraus, dass die erkannten Aussagen auf der linken Bildschirmhälfte angezeigt wird, während nach Beendigung der Aufnahme die Analyse auf der rechten Bildschirmhälfte sichtbar wird. Aus Platzgründen wurde dies jedoch aufgeteilt, sodass während einer Aufnahme der gesamte Bildschirm genutzt wird. Nachdem die Aufnahme beendet wird, wird die gesamte Anzeige unsichtbar und die Analyse startet im Hintergrund. In der oberen Bildschirmmitte wird angezeigt, was gerade vom SEnti-Analyzer getan wird. Genau wie auf der Kommandokonsole ausgegeben wurde, welches Analysetool gerade arbeitet, wird dies zur Nachvollziehbarkeit angezeigt. Während einer Audioaufnahme werden erkannte Aussagen in der grafischen Oberfläche angezeigt, nacheinander werden mehr Aussagen sichtbar und untereinander angezeigt. Jeweils mit angezeigt wird die vorläufig getroffene Sentimentzuweisung von TextBlob oder TextBlobDE, je nachdem, ob der SEnti-Analyzer auf Deutsch oder auf Englisch verwendet wird. Wie folgend und später näher erläutert, birgt dies einige Schwierigkeiten. Zuerst bietet der Analysebildschirm nur begrenzt Platz in der Höhe, was vorläufig erweiterbar ist, indem die Bildschirmgröße angepasst wird, jedoch ist auch das

keine endgültige Lösung. Das kompliziertere Problem liegt jedoch darin, dass sich Texte in PySimpleGui zwar dynamisch erweitern lassen, indem die neue erkannte Aussage an die vorherigen Sätze angehängt wird, diese Möglichkeit besteht jedoch bei den dargestellten Sentimentzuweisungen nicht, da diese einzelne Bilder sind, welche je nach zugeordnetem Sentiment geladen werden. Hier lässt sich keine dynamisch erweiterbare Lösung für die Darstellung der Bilder realisieren. Aufgrund dessen ist die Anzahl der auf dem Analysebildschirm angezeigten Aussagen auf die jeweils 20 letzten Aussagen begrenzt. Nachdem 20 Aussagen erkannt wurden, werden bei einer neuen Aussage die vorherigen Aussagen nach oben aufgeschoben.

Jedoch ist während einer Audioaufnahme darauf verzichtbar, dass alle erkannten Aussagen einsehbar sind, da dies ist nach dem Beenden einer Audioaufnahme möglich ist, wenn die endgültige Analyse fertiggestellt wurde.

Weiterhin existieren während der Audioaufnahme die Möglichkeiten dazu, die Aufnahme per Knopfdruck zu beenden, die Anzeige zu verstecken und die Audioaufnahme abubrechen. Bei letzterem sind einzelne Optionen verfügbar um entweder zum Hauptmenü zurückzukehren oder den SEnti-Analyzer komplett zu beenden. Sinnvoll sind die Möglichkeiten zum einen damit ein zuverlässiger Weg neben den Sprachkommandos „Aufnahme Beenden“ oder „Stop recording“ existiert um eine Audioaufnahme zu beenden, dies ersetzt damit den Kommandokonsolen-Input von „STRG+C“. Außerdem lässt sich eine potentielle Ablenkung durch die Anzeige auf dem Bildschirm verhindern, indem die Möglichkeit zum Verstecken der Anzeige gegeben wird. Die Möglichkeit die Audioaufnahme abubrechen ist sinnvoll, falls etwas falsch eingestellt oder versehentlich gestartet wurde.

Ein Beispiel wie der SEnti-Analyzer während einer Audioaufnahme aussieht folgt, dies ist eine Vergrößerte Ansicht, aus der leerstehender Platz aufgrund der Lesbarkeit entfernt wurde:

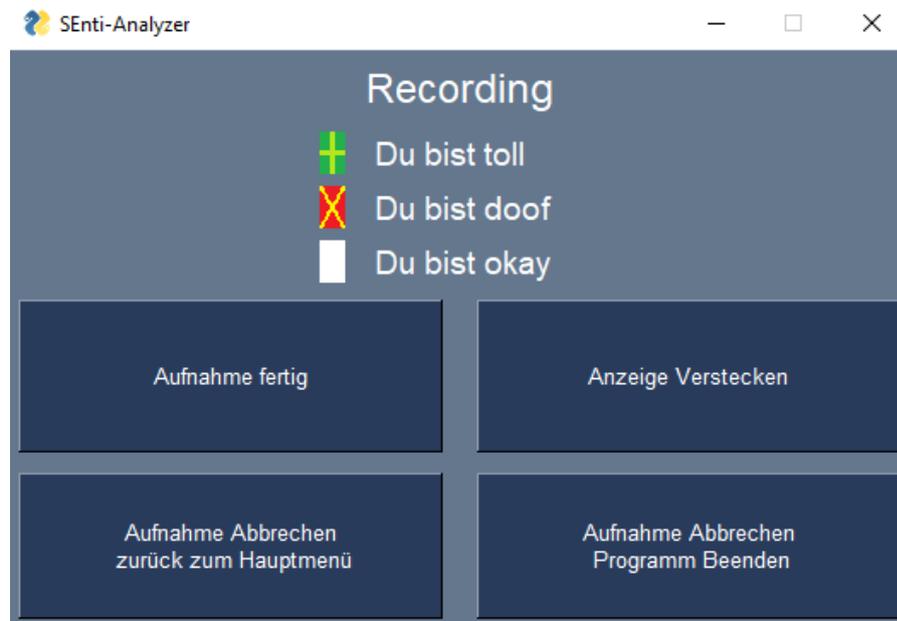


Abbildung 5.3: Der Bildschirm während einer Audioaufnahme

Nachdem eine Audioaufnahme beendet wird, wird eine ausführliche Analyse gestartet. Dies geschieht wie in den Funktionen des SEnti-Analyzers beschrieben 2.3.1. Dabei wird sowohl eine Tabelle angezeigt, welche die zuvor auf der Kommandokonsole ausgegebene Übersicht, die schon eine tabellarische Art annahm, widerspiegelt. Diese wird jedoch zuerst als vereinfachte Tabelle dargestellt, bei der nur der jeweils errechnete Median angezeigt wird. Dazu gibt es eine Option Details anzuzeigen, welches eine genaue Bewertung aller eingesetzten Analysetools einblendet. Ein Beispiel für eine Tabelle folgt:

ID	Text	GerVADER	BertDE	SentiStrengthD	TextBlobDE	Median
1	Du Bist toll.	0.4939	positive	0.0	1.0	positive
2	Du Bist doof.	-0.3612	negative	0.0	-1.0	negative
3	Du Bist Okay.	0.0	negative	0.25	0.0	neutral

Abbildung 5.4: Tabellarische Ansicht aller Aussagen

Zusätzlich wird die Übersicht über Gesamtzahlen mit relativen und totalen Anteilen in einer Erweiterung zur Kommandokonsoleausgabe als Balkendiagramm aufgezeigt. Ein Beispiel für ein Balkendiagramm folgt:

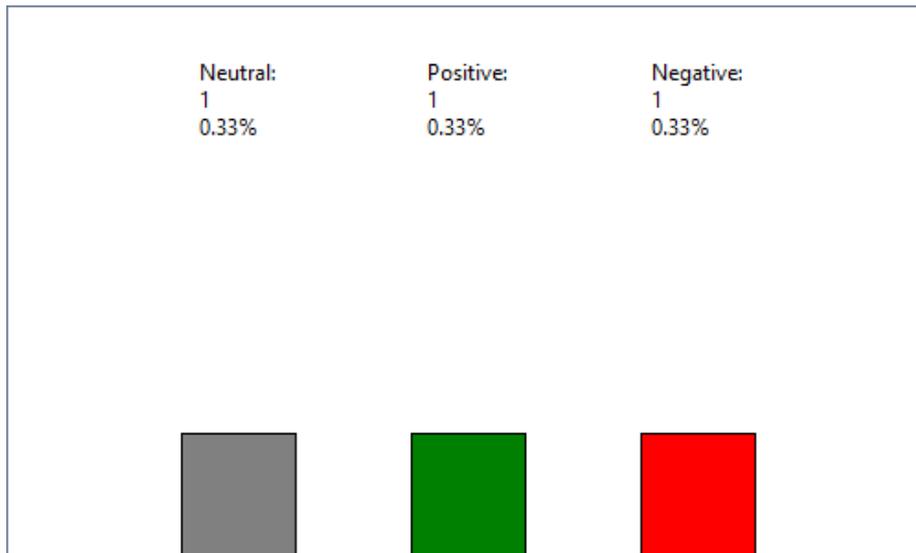


Abbildung 5.5: Balkendiagramm der Ergebnisse

Ebenfalls als Erweiterung zur Kommandokonsole werden nun zwei Graphen erstellt. Der Graph von PySimpleGui ist simpel zu benutzen, sodass für das Erzeugen dieser Graphen auf die Einbindung externer Frameworks für Graphen verzichtet wurde. Der Graph arbeitet mit einem eigenem, selbst definierbarem Koordinatensystem, welches einfach verändert werden kann. Zu diesem Zweck befinden sich ebenfalls globale Variablen für die Grenzen der Koordinaten der einzelnen Graphen.

Einer der Graphen stellt einen einfachen Zeitverlauf dar, wann welche Sentimenteinstufung geschehen ist. Dabei stellt die X-Achse den Zeitverlauf dar, wobei die einzelnen erkannten Aussagen die Zeitpunkte darstellen, da diese vorher durchnummeriert wurden. Auf der Y-Achse findet dabei eine Darstellung von positiv, neutral oder negativ statt. Ein Beispiel für einen Zeitverlaufsgraphen folgt:

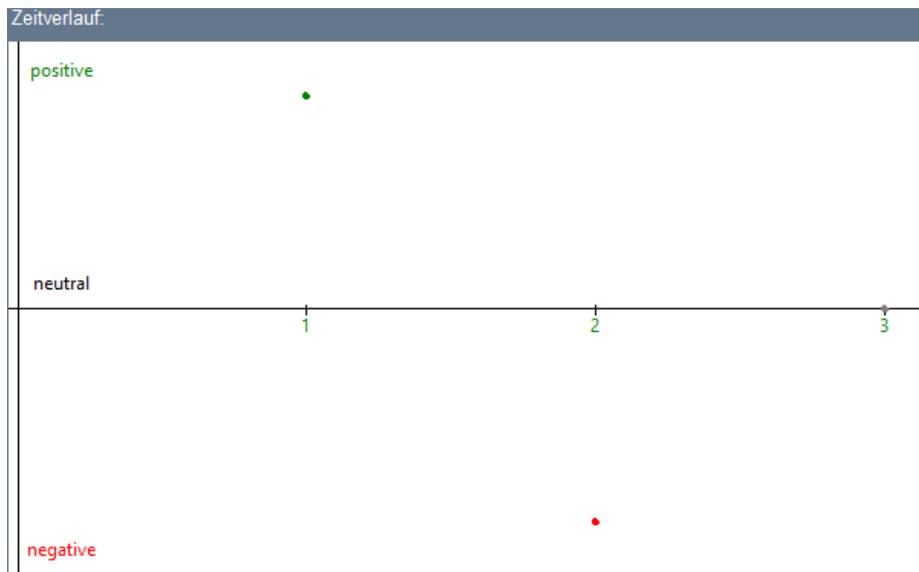


Abbildung 5.6: Zeitverlaufsgraph

Der zweite Graph stellt einen Gesamtstimmungsverlauf über das Meeting dar. Die Zeiteinstufungen auf der X-Achse sind dabei die gleichen wie bei dem einfachen Zeitverlaufsgraphen. Die Y-Achsenwerte sind dabei ganzzahlige Zahlenwerte. Der Graph verläuft dabei so, dass ein erkanntes positives Sentiment den Graphen um 1 in positive Richtung und ein erkanntes negatives Sentiment um 1 in negative Richtung verändert. Bei einem erkanntem neutralen Sentiment verändert sich der Wert auf der Y-Achse nicht. Im Vorhinein wird dabei errechnet welches die höchste positive und die niedrigste negative Stimmung innerhalb des Meetings war, um die Grenzen auf der Y-Achse festzulegen. Dabei wird von einem Wert von 0 beginnend von allen Aussagen der Median verrechnet. Bei positivem Median wird der Wert inkrementiert und bei negativem dekrementiert. Falls dabei ein neuer höchster oder niedrigster Wert erreicht wird, wird dieser gespeichert. Um dabei keinen verzerrten Graphen zu erzeugen wird am Ende der Betrag des niedrigsten Negativen mit dem größten Positiven verglichen. Der größere der beiden Werte wird dann sowohl im positiven als auch im negativen als Grenzen verwendet. Dies stellt sicher, dass die Y-Achse gleichermaßen eingeteilt ist.

Bei beispielsweise dem größten Positiven von 5 und dem niedrigsten Negativen von -3 wird die 5 mit dem Betrag von -3 verglichen.

Da die 5 größer ist wird der Wertebereich von -5 bis 5 auf der Y-Achse festgelegt. Bei werten im positiven von 3 und im negativen von -5 wird der Wertebereich also auch von -5 bis 5 festgelegt. Ein Beispiel für einen Gesamtstimmungsgraphen folgt:

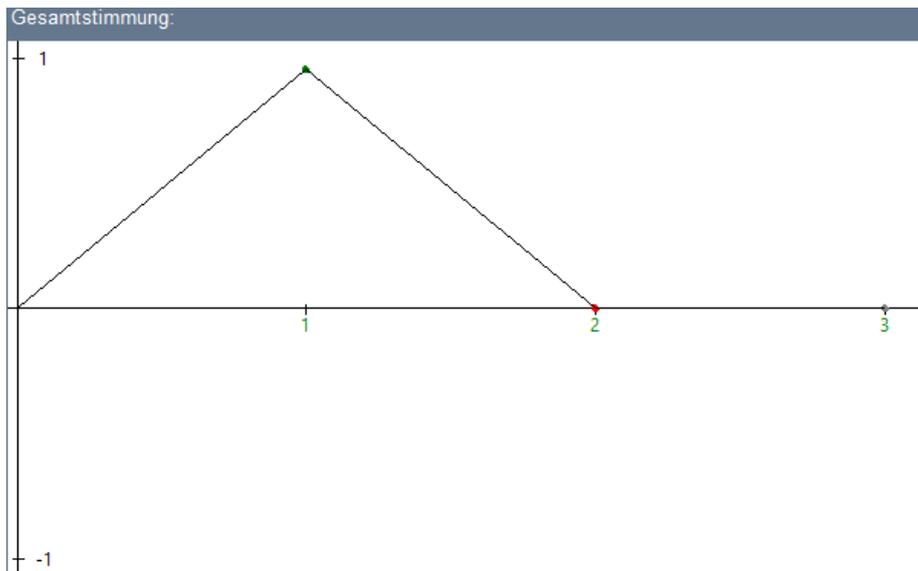


Abbildung 5.7: Gesamtstimmungsgraph

Beide Graphen nutzen die gleiche Einteilung an der X-Achse. Da diese in ihrer Gesamtheit alle einzelnen erkannten Aussagen darstellt, kann diese nicht so statisch eingeteilt werden, dass es für jede Gesamtanzahl an erkannten Aussagen angenehm eingeteilt ist. Beispielsweise lässt sich eine durch 10 Teilbare Gesamtanzahl gut in 10 Teile zerteilen, sodass jeder Schritt mit einer Ganzzahl versehen wird. Bei einer Zahl, die nicht glatt durch 10 teilbar ist, geht dies jedoch nicht. Eine komplett dynamische Einteilung der Achse ist jedoch auch nicht möglich, da theoretisch eine unbegrenzte Anzahl an Aussagen analysiert werden könnte.

Ein Kompromiss der beiden Ansätze liegt darin, dass eine Zahl gefunden wird, die die Gesamtanzahl an Aussagen glatt teilt und diese dann für die Einteilung der X-Achse zu verwenden. Dabei wird versucht, den größten Teiler kleiner als 11 zu finden und diesen als Einteilung zu nutzen. Dies verhindert eine Überschneidung von einzelnen Einteilungen und bietet trotzdem eine ungefähre Einteilung für die X-Achse.

Aufbauend auf den vorangehend benutzten Beispielen ist ein Beispiel für das Aussehen einer vollständigen Analyse wie folgt, dies ist eine Zusammensetzung der einzelnen oberen Teile. Die Beiden Buttons am unteren Bildschirmrand sind dabei für die Navigation ins Hauptmenü oder das komplette Beenden des SEnti-Analyzers.

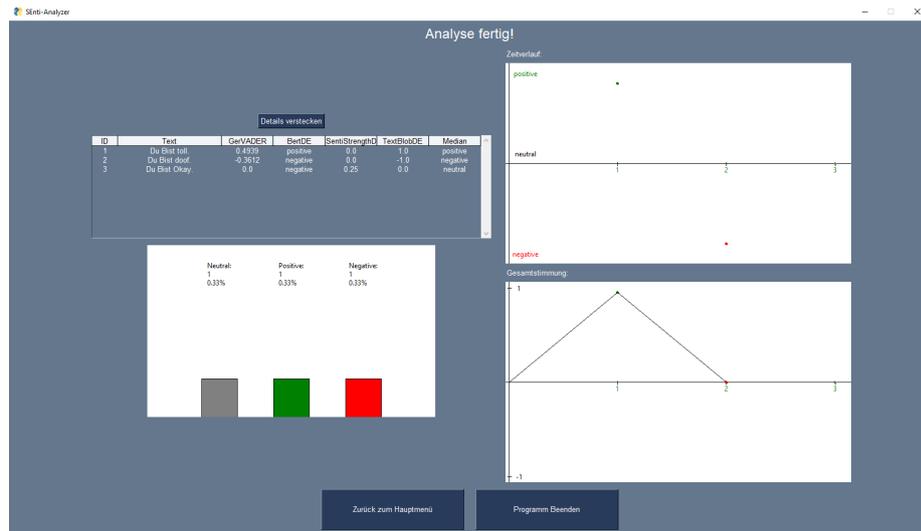


Abbildung 5.8: Der Bildschirm während einer fertigen Analyse

5.2 Nicht hinzugefügte Funktionen

einige Features konnten im Verlaufe dieser Arbeit leider nicht realisiert werden.

Unter anderem wurde von mehrere Interviewten Personen die Auswahl der einzelnen Analysetools angesprochen. Dies zu implementieren bedarf jedoch eine strukturelle Änderung des gesamten SEnti-Analyzers und würde den Rahmen dieser Arbeit überziehen. Aus diesem Grund wurde diese Funktion nicht eingebaut. Daraus folgt, dass weder Hauptmenü, noch bei dem Analysebildschirm eine Auswahl für die einzelnen Tools eingebaut wurde.

Ebenfalls wurde die „Mute“ Funktion nicht eingebaut, da auch dies eine strukturelle Änderung benötigen würde.

Leider konnte aus Zeitgründen nicht realisiert werden, dass mit den Graphen genauer interagiert werden kann.

Das exportieren der Ergebnisse in Bildform ist auch nicht realisiert, die Ergebnisse werden jedoch in einer .csv Datei gespeichert.

Diese lässt sich zu späteren Zeitpunkten erneut analysieren.

5.3 Verbesserungen

Bei den Verbesserungen im Vergleich zum SEnti-Analyzer ohne grafische Oberfläche wird sich auf die ISO Normen 25010 bezogen. Die generelle Functional Suitability des SEnti-Analyzers war vor der grafischen Oberfläche schon gegeben, der SEnti-Analyzer hat Audio- und Textinput verarbeitet und darauf eine Sentiment Analyse durchgeführt. Zwar kein direkter Einfluss der grafischen Oberfläche, jedoch trotzdem im Verlaufe dieser Arbeit eine hinzugefügte Funktion, ist die oben beschriebene Möglichkeit einen Namen für die am Ende resultierende Datei einzugeben. In den Interviews angesprochen wurde die Idee, dass mehrere Meetings miteinander verglichen werden. Das ist grundsätzlich nur möglich, wenn auch mehrere Ergebnisse der verschiedenen Meetings existieren und diese nicht jedes Mal überschrieben werden. Das Eingeben eines Namens für die erzeugte .csv Datei ermöglicht eben diese Speicherung von mehreren Ergebnissen, die auch zu einem späteren Zeitpunkt erneut analysiert werden können. Weiterhin existiert innerhalb der grafischen Oberfläche mehrere Optionen direkt zurück zum Hauptmenü zu navigieren, was vorher nur durch einen manuellen Neustart möglich war. Grundsätzlich hängt diese Änderung nicht mit der grafischen Oberfläche zusammen, jedoch macht es die Nutzung dieser einfacher, wie im Folgenden auch beschrieben wird.

Der größte Teil der sich verbessert hat liegt in der Usability. Ziel dieser Arbeit war es den SEnti-Analyzer besser nutzbar zu machen. Die Learnability hat sich erheblich verbessert, da die Benutzer*innen nicht schon vorher wissen müssen, welche Funktionen theoretisch möglich wären. Die grafische Oberfläche stellt alle Funktionen gegenüber den Nutzer*innen dar, sodass diese innerhalb des SEnti-Analyzers erkennen und lernen können welche Funktionen dieser bietet. Die Aspekte, die die Learnability verbessert haben, wirken sich auch auf die allgemeine Operability aus. Den SEnti-Analyzer zu starten und aus der grafischen Oberfläche heraus zu benutzen ist angenehmer als die Kommandokonsole zu öffnen, in den richtigen Ordner zu navigieren und dann von dort aus die Befehle auszuführen. Einige der hinzugefügten Features wirken sich positiv auf die User Error Protection und die Accesibility aus. So existiert nun die Möglichkeit einen laufenden Prozess ohne Umwege abzubrechen, ohne, dass darauf gewartet werden muss, dass dieser

Fertig gestellt wird. Dies wird besonders wichtig, nachdem die Ergebnisse des Funktionstests im späteren dargelegt werden. Die Accessibility wurde dahingehend verbessert, dass die vorangehend beschriebene Änderung der Sentiment Vorentscheidung besser zu erkennen ist, vor allem für Personen mit Beeinträchtigungen in der Rot-Grün-Wahrnehmung.

Weiterhin wurde während des Programmierens auf Maintainability viel Wert gesetzt, um die zukünftige Arbeit mit der grafischen Oberfläche des Senti-Analyzers möglichst einfach zu gestalten.

Kapitel 6

Funktionstests des SEnti-Analyzers

Um sowohl die Funktionsfähigkeit des Senti-Analyzers zu testen, als auch allgemeine Informationen zum SEnti-Analyzer zu sammeln, wurden einerseits während der Programmierung ständige Tests durchgeführt, andererseits wurden auch Tests durchgeführt nachdem die grafische Oberfläche komplett fertig programmiert war.

Die relevanten Tests nach der Fertigstellung der grafischen Oberfläche werden im Folgenden dargestellt, sowie deren Ergebnisse.

Um verschiedene Daten zu sammeln, wurden die Tests mit unterschiedlich vielen Personen und unterschiedlich lange durchgeführt. Bei beiden Meetings saßen die Probanden in einer Art Kreis um einen Tisch, auf dessen Mitte das aufnehmende Mikrofon platziert wurde, möglichst mittig zwischen allen Probanden.

Beim ersten Testmeeting waren 5 Personen beteiligt, des weiteren wurde in diesem Testmeeting eine Zeitbegrenzung von 90 Minuten gesetzt. Der SEnti-Analyzer hat während diesem Testmeeting insgesamt 1130 Aussagen erkannt und aufgezeichnet. Leider hat das beenden des Meetings nicht funktioniert, da der SEnti-Analyzer keine Sprache mehr erkannt hatte und auf keine Eingabe mehr reagierte. Da der SEnti-Analyzer zu diesem Zeitpunkt die erkannten Aussagen noch sowohl in der grafischen Oberfläche angezeigt, als auch auf der Kommandokonsole ausgegeben hatte, konnten die bis zu diesem Zeitpunkt erkannten 1130 Aussagen jedoch kopiert und gespeichert werden, sodass diese dann mithilfe der Dateianalysefunktion trotzdem analysiert werden konnten. Erkannt wurden dabei 35 positive, 36 negative und 1059 neutrale Aussagen. Dies entspricht einem Anteil von etwas mehr als 3% positiven und

40KAPITEL 6. FUNKTIONSTESTS DES SENTI-ANALYSZERS

negativen und dementsprechend etwas weniger als 94% neutralen Aussagen. Leider kann nicht genau gesagt werden, wann genau die Aufnahme des SEnti-Analyzers aufgehört hatte zu funktionieren, jedoch kann sich an der Anzahl der erkannten Aussagen trotzdem orientiert werden.

Im zweiten Testmeeting war eine Person weniger beteiligt und das Zeitlimit war auf 60 Minuten begrenzt. In dieser Zeit wurden 420 Aussagen erkannt. Davon waren 12 positive, 6 negative und 402 neutrale Aussagen, was einen Anteil von knapp 3% positiven, knapp 1.5% negativen und ca. 95.5% neutralen Aussagen ergibt.

Nachdem die Testmeetings beendet wurden, wurde die Analyse dieser gestartet, um die gerade beschriebenen Ergebnisse zu messen. Besonders interessant war dabei für diese Arbeit, wie lange die Analyse bei „echten“ Meetings dauert. Im Verlauf des Programmierens wurden der SEnti-Analyser zwar immer wieder getestet, jedoch nur mit Eingaben von maximal 20 Aussagen um bestimmte Funktionen zu testen. Für die tatsächliche Benutzung des SEnti-Analyzers ist es interessant zu wissen, wie lange die Analyse eines tatsächlichen Meetings dauert, eben zu diesem Zweck wurden die Testmeetings durchgeführt, um einen in etwa realistischen Rahmen für die Menge an erkannten Aussagen zu bieten.

Realisiert wurde diese Zeitmessung mithilfe von Ausgaben auf der Kommandokonsole, die als Programmbefehle jeweils vor und nach dem Analysestart der einzelnen Analysetools temporär hinzugefügt wurden.

Für die Analyse des 90-minütigen Testmeetings brauchte GerVADER ca. 1 Sekunde, TextBlobDE ca. 4 Sekunden, SentiStrengthDE knapp 4 Minuten und BertDE 2 Stunden und 6 Minuten. Insgesamt ergibt dies eine Dauer von ungefähr 2 Stunden und 10 Minuten. Bei der Analyse für das 60-Minütige Testmeeting brauchte GerVADER weniger als 1 Sekunde, TextBlobDE ca. 1 Sekunde, SentiStrengthDE ca. 90 Sekunden und BertDE knapp 50 Minuten. Dies resultiert insgesamt in einer Dauer von knapp 52 Minuten.

Es fällt auf, dass BertDE deutlich länger braucht als die anderen Analysetools und auch länger, als für den Kontext angemessen, da die Gesamtdauer fast ausschließlich von BertDE eingenommen wird. Aus Interesse für weitere Zeiteinschätzungen wurde die Analyse mit den englischen Analysetools wiederholt. Dies ergibt zwar keine aussagekräftige Einordnung der Sentimente, gibt jedoch trotzdem Auskunft über die Dauer, die für die Analyse benötigt wird.

Bei dem 90-minütigen Testmeeting hat TextBlob knapp 4 Se-

kunden, SentiCR knapp 24 Sekunden und Senti4SD ca. 2 Minuten gebraucht. Jeweils SentiStrength und SentiStregthSe haben knapp 8 Minuten gebraucht. Das ergibt eine Gesamtdauer von ca. 18 Minuten. Bei dem 60-minütigem Testmeeting hat TextBlob knapp 1 Sekunde, SentiCR knapp 25 Sekunden, Senti4SD ca. 50 Sekunden und SentiStrength und SentiStregthSE jeweils knapp 3 Minuten gebraucht. Insgesamt ergibt sich daraus eine Zeit von ca. 7 Minuten und 15 Sekunden. Besonders interessant dabei ist die Beobachtung, dass SentiCR bei dem 90-minütigen Testmeeting sogar 1 Sekunde schneller in der Analyse war als bei dem 60-minütigem Testmeeting.

Eine Gesamtzeit von 18 Minuten bzw. 7 Minuten und 15 Sekunden ist im Kontext betrachtet zwar auch zu lange, da nach einem Meeting nicht lange auf die Ergebnisse von diesem gewartet werden sollte. Im Vergleich zur deutschen Analysedauer ist dies jedoch immer noch um einen großen Faktor, knapp 7 beim 60-minütigem und sogar knapp 12 beim 90-minütigem Testmeeting, schneller.

Bei der Annahme, dass jede Person innerhalb eines Testmeetings einen gleichmäßig Verteilten Redeanteil besitzt, entspricht das beim 90-minütigem Testmeeting 226 Aussagen pro Person im Meeting, was ungefähr 2,5 Aussagen die Minute entsprechen würde. Im 60-minütigem Testmeeting sind 105 Aussagen pro Person zu verzeichnen, was 1,75 Aussagen pro Person pro Minute ergibt. Grundsätzlich lässt sich daraus schließen, dass in größeren Meetings mehr Redebedarf untereinander besteht. Interessant ist jedoch, dass der Anteil an positiven Aussagen über beide Testmeetings fast exakt gleich war, der Anteil der negativen Aussagen war dabei auch bei beiden Testmeetings ungefähr konstant und der mit Abstand größte Teil der Aussagen wurde als neutral klassifiziert.

Über beide Testmeetings gab es Beobachtungen, die interessant für die tatsächliche Benutzung des Senti-Analyzers sind. Zum einen war zu beobachten, dass das Erkennen von Aussagen sehr davon beeinträchtigt wird, wenn mehrere Leute gleichzeitig reden oder eine Person eine andere unterbricht. Der Senti-Analyzer erkennt nicht, dass zwei verschiedene Personen gleichzeitig reden und versucht zu erkennen was gesagt wird, woraus dann teilweise keine sinnvollen Aussagen erkannt werden können.

Es wurde auch beobachtet, dass teilweise generell nicht vollständig erkannt wurde was gesagt wird, was aber auch am genutzten Mikrophon liegen kann. Besonders auffällig war jedoch die Beobachtung dazu, dass der Senti-Analyzer nach nur kleinen Pausen

42KAPITEL 6. FUNKTIONSTESTS DES SENTI-ANALYSZERS

eine Eingabe als kompletten (Teil-)Satz annimmt. Eine Folge von einzelnen Wörtern die eine einzelne erkannte Aussage ergeben war oft wahrzunehmen, ein Beispiel wie diese aussehen ist in der folgenden Tabelle dargestellt:

ID	Text	[...]	Median
[...]	[...]	[...]	[...]
19	Ein	[...]	neutral
20	Bei	[...]	neutral
21	Wir	[...]	neutral
22	Am	[...]	neutral
23	Das	[...]	neutral
24	An	[...]	neutral
[...]	[...]	[...]	[...]

Weiterhin war zu beobachten, dass unter anderem englische Fachbegriffe bei einer deutschen Sprachauswahl nicht erkannt wurden. Im Kontext von Softwareprojektmeetings ist dies etwas, was der erfolgreichen Nutzung des SEnti-Analyzers durchaus im Weg stehen kann, da dieses Gebiet sehr durch Fachbegriffe geprägt ist.

Kapitel 7

Limitierungen dieser Arbeit

Für diese Arbeit wurde eine grafische Oberfläche für den SEnti-Analyzer entwickelt. Dabei sind unter anderem Probleme aufgetaucht und es wurden Grenzen erreicht.

Grundsätzlich sind die Ergebnisse dieser Arbeit teilweise mit Vorsicht zu betrachten. Während dieser Arbeit wurde nur an einem Computer gearbeitet und programmiert. Dies bedeutet, dass theoretisch jedes vorher und im folgenden beschriebene Problem ein Ergebnis von technischen Gegebenheiten, wie dem genutzten System oder der genutzten Software sein könnte. Es besteht die Möglichkeit, dass Probleme die bei dieser Arbeit aufgetreten sind keine Relevanz außerhalb dieser Arbeit haben.

Beginnend bei der Installation kann es sein, dass Umstände für den SEnti-Analyzer nicht so waren, wie sie hätten sein sollen, da für diese Arbeit der SEnti-Analyzer soweit installiert wurde, sodass dieser funktionstüchtig ist. Besonders bei der Installation der einzelnen benutzten Analysetools kann davon ausgegangen werden, dass etwas fehlt oder falsch installiert wurde, da eine Sentimentanalyse unter keinen Umständen so lange dauern sollte wie aus den Testmeetings resultierend. Jedoch muss für den Rest dieser Arbeit davon ausgegangen werden, dass dies nicht der Fall ist.

Die Programmierete grafische Oberfläche wurde ebenfalls nur auf einem System programmiert und dementsprechend getestet. Es wurde zwar darauf geachtet, dass diese kompatibel mit anderen Systemen ist, eine Garantie dafür existiert jedoch nicht.

Einige Teile der grafischen Oberfläche mussten aufgrund der dynamischen Möglichkeiten unschön programmiert werden. So existieren für die Darstellung während einer Audioaufnahme jeweils 20

eigene Objekte für Aussagen die erkannt wurden und Bilder, welche die vorläufigen Sentimenteinstufung darstellen. Ebenfalls muss sich ein Array die aktuell angezeigten Bilder merken, da ein Bild Objekt in PySimpleGui keine Möglichkeit bietet, die aktuelle Quelle des Bildes abzugreifen. Diese lässt sich zwar aktualisieren, jedoch nicht direkt abfragen, sodass sie anderweitig zwischengespeichert werden muss. Weiterhin ist die Länge der erkannten Aussagen innerhalb des Senti-Analyzers theoretisch nicht begrenzt. Wenn eine Person ohne Pause weiterredet wird dies vom Senti-Analyzer immer weiter als eine Aussage aufgenommen. Erst wenn eine Sprechpause eingelegt wird, wird diese eine Aussage als fertig erkannt und in der grafischen Oberfläche dargestellt. Dies kann theoretisch dazu führen, dass eine Aussage aus dem Bildschirm heraus ragt und nicht vollständig zu sehen ist. In der Praxis kommt dies jedoch nur äußerst selten vor.

Die Funktion, dass zurück zum Hauptmenü navigiert wird, wird ebenfalls von technischen Gegebenheiten und der Art wie PySimpleGui funktioniert, beeinflusst. PySimpleGui erlaubt es nicht eine geschlossenes Layout erneut zu öffnen. Aus diesem Grund musste die Möglichkeit des Navigierens zum Hauptmenü durch Neustart des Senti-Analyzers realisiert werden.

Ein Problem bei der aktuellen Art wie die Graphen erstellt werden ist, dass die Einteilung der X-Achse erst nach der Gesamtanalyse geschehen kann und dabei ein Teiler gesucht wird, der die Gesamtanzahl der erkannten Aussagen glatt teilt. Daraus folgt, dass bei einer Gesamtanzahl von erkannten Aussagen die durch keine Zahl kleiner 11 teilbar ist, also einer Primzahl, keine Einteilung der X-Achse geschieht. Auch nicht ideal bei den Graphen ist, dass für sämtliche Berechnungen innerhalb des Graphen der errechnete Median genutzt wird. Dies spiegelt grundsätzlich die tatsächlichen Ergebnisse dar, jedoch wirkt sich so eine maximal positive Aussage innerhalb der Graphen genau gleich auf den Graphenverlauf aus wie eine normal positive oder leicht positive Aussage. Gleiches gilt im negativem Bereich, wobei die Wechselwirkung zwischen positiven und negativen Aussagen auch beeinflusst wird. Maximal positive Aussagen werden von leicht negativen Aussagen neutralisiert und umgekehrt.

Es wurden zwei Funktionstests für den Senti-Analyzer durchgeführt, dies ist eine sehr geringe Stichprobenmenge. Jedoch wurden während diesen Testmeetings alle Funktionen der grafischen Oberfläche getestet. Etwaige Daten die über die Funktionsweise der grafischen Oberfläche hinaus gehen und aus den Testmeetings

heraus gewonnen wurden, sind aus diesem Grund nicht repräsentativ und im Falle des 90-minütigen Testmeetings unter Umständen sogar verfälscht. Beim 90-minütigen Funktionstests hörte der SEnti-Analyzer an einem Zeitpunkt auf Eingaben zu verarbeiten, dieser Zeitpunkt konnte nicht genauer bestimmt werden. Die daraus gewonnen Daten können jedoch trotzdem verarbeitet werden und diese Fehlfunktion wirkt sich nicht auf die Funktionsweise der grafischen Oberfläche selbst aus.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

In dieser Arbeit wurde der zuvor entwickelte SEnti-Analyzer erklärt und mithilfe einer grafischen Oberfläche verbessert. Diese grafische Oberfläche ist das Produkt einer Anforderungsanalyse, die im Verlaufe dieser Arbeit durchgeführt wurde, bestehend aus subjektiven Ideen, das Durchführen von Interviews und daraufhin Anpassung der Ideen. Die grafische Oberfläche ist ein wichtiger Schritt in Richtung Usability des SEnti-Analyzers, jedoch wird im anschließenden Ausblick erläutert wie der SEnti-Analyzer über diese Arbeit hinaus weiterhin verbessert werden kann.

8.2 Ausblick

Sofern der SEnti-Analyzer weiter entwickelt wird, so muss auch die grafische Oberfläche weiter entwickelt werden. Abgesehen von aus den Interviews hervorgegangenen gewünschten Features für die Zukunft des SEnti-Analyzers, wobei die Oberfläche daraufhin weiter angepasst werden muss, ist auch für die Oberfläche noch weiterer Raum für Verbesserungen. Vor allem die Anzeige der erkannten Aussagen könnte erweitert werden um eine Möglichkeit zu realisieren, dynamisch mehr als 20 Aussagen gleichzeitig anzuzeigen und die im Hauptteil genannten Probleme zu lösen. Eine Möglichkeit das Problem der zu langen Sätze zu lösen wäre es, innerhalb der erkannten Aussagen Zeilenumbrüche zu platzieren, wenn diese

zu lang werden. Dies würde jedoch dazu führen, dass sich das Layout des gesamten Bildschirms verschiebt und die angezeigten vorentschiedenen Sentimente möglicherweise nicht mehr zu den dazugehörigen Aussagen ausgerichtet sind. Die Funktion direkt zurück zum Hauptmenü zu navigieren lässt sich ebenfalls verbessern, stand dieser Arbeit öffnet sich eine Kommandokonsole, welche von selbst den SEnti-Analyzer neu startet. Dies erfordert zwar keinen Input von den Benutzer*innen, ist jedoch trotzdem unschön.

Weiterhin, falls die Funktion hinzugefügt wird, entweder vor dem Analysestart oder sogar nachdem die Analyse fertig gestellt ist, einzelne Analysetools aus- und abzuwählen und nur diese für die Analyse miteinzubeziehen, müssen die Tabellen angepasst werden. Eine Tabelle ist in PySimpleGui nicht dynamisch in der Größe veränderbar, genauso kann die Titelzeile im Nachhinein nicht verändert werden. Zum Zeitpunkt dieser Arbeit wäre die einzige Möglichkeit eine solche Funktion mittels Tabellen darzustellen, dass eine eigene Tabelle für jede verschiedene mögliche Permutation an Analysetools erstellt wird und die passende Tabelle sichtbar und alle anderen unsichtbar werden. Dies ist eine Möglichkeit die unschön ist, besonders falls irgendwann neue Analysetools hinzugefügt werden. Die Anzahl an benötigten Tabellen entspricht $2^x - 1$, wobei x der Anzahl der insgesamt genutzten Analysetools entspricht, wobei die Tabelle für keine benutzen Analysetools weggelassen werden kann. Im Moment existieren 3 Tabellen, eine für die Analyse auf englischer Sprache, eine für die Analyse auf deutscher Sprache und eine Tabelle für die simplifizierte Ansicht, welche nur eine Tabelle ist, da dort die einzelnen Analysetools nicht aufgeführt werden und der Inhalt der Tabelle sehr wohl im Nachhinein änderbar ist, sodass die detaillierte Tabelle runter gebrochen werden kann um nur den errechneten Median anzuzeigen. Für die Möglichkeit die Tools auswählbar zu machen wären im aktuellen Zustand für die insgesamt 4 deutschen und 5 englischen benutzten Tools $(2^4 - 1) + (2^5 - 1) = 46$ Tabellen nötig, die manuell erstellt und gewartet werden müssen. Wie oben erwähnt ist das zwar eine Möglichkeit, lässt den Code jedoch weiter unschön werden. Idealer wäre dort eine alternative Lösung.

Für die Graphen, die vom SEnti-Analyzer erstellt werden, besteht das vorangehend beschriebene Problem. Dies ließe sich verbessern, falls der errechnete Median nicht nur aus positiv, neutral oder negativ bestehen würde, sondern ein genauer dazugehöriger Wert berechnet wird.

Die Funktion mit den Graphen genauer zu interagieren könnte

hinzugefügt werden, wenn auf einen bestimmten Punkt innerhalb einem Graphen geklickt wird, ließe sich diese Aussage z.B. in der Tabelle hervorheben.

Aufgrund der Ergebnisse der Funktionstests wäre es ratsam BertDE durch ein anderes Analysetool zu ersetzen oder einfach wegzulassen, da BertDE zu einer erheblichen Verlängerung der Analysedauer führt

Über die erwähnten Features während der Interviews hinaus wäre eine differenzierte Spracherkennung von unterschiedlichen Personen sinnvoll. So können sowohl Unterbrechungen erkannt und die eigentlich gesagten Aussagen aufgezeichnet werden. Besonders die Feststellung von Unterbrechungen ist im Kontext von Meetings interessant, da diese die Ursache oder das Resultat einer negativen Stimmung im Team sein können.

Der größte Faktor der sich verbessern lässt, liegt zur Zeit jedoch in der Installierbarkeit des SEnti-Analyzers. Verfügbar ist der SEnti-Analyzer auf GitLab, zurzeit ist dieser jedoch nicht öffentlich. Wie auf GitLab üblich befindet sich dort eine README.md, welche alles Wichtige verrät. Nach einer kurzen Vorstellung des Tools, findet sich dort auch ein Abschnitt für die Installation des SEnti-Analyzers. Ebenfalls wie üblich muss das Projekt aus GitLab geklont werden, um genutzt zu werden. Nötig sind für den SEnti-Analyzer stand dieser Arbeit Java 8 und Python 3.9.5. Dann muss noch eine Vielzahl von Bibliotheken installiert werden, um die Funktionsfähigkeit des SEnti-Analyzers zu ermöglichen. Da das Tool in Python geschrieben ist, existiert dort eine requirements.txt, von der mit dem Befehl „pip install -r requirements.txt“ auf der Kommandokonsole alle darin aufgeführten Bibliotheken installiert werden können.

Mit dieser requirements.txt gab es im Verlauf dieser Arbeit einige Probleme. Insgesamt lassen sich die Bibliotheken nicht komplett mit diesem Befehl installieren, da zumindest auf Windows die Bibliothek von PyAudio Fehlermeldungen verursacht. Diese muss manuell installiert werden, durch die Befehle „pip install pipwin“ und „pipwin install pyaudio“ . Dies ist auch im GitLab dokumentiert. Neben PyAudio gibt es noch einige Bibliotheken die im Konflikt mit der geforderten Java oder Python Version standen und eine Dopplung war in der requirements.txt ebenfalls zu finden. Aufgeführt waren sowohl „spellchecker“ als auch „pyspellchecker“ welche im Programm selbst die gleiche Funktionalität erfüllen. Benötigt wird nur der „pyspellchecker“ und nach einem Austausch mit dem Entwickler des SEnti-Analyzers stellte sich heraus, dass der spellchecker über

den pyspellchecker installiert wurde. Dies sorgte dafür, dass die requirements erfüllt waren, dass beides installiert ist, aber, dass die Funktionalität der spellchecker Bibliothek nicht gegeben war, da dort Dateien vorhanden waren, die nicht vorhanden sein sollten.

Wobei der letzte Fehler im Verlaufe dieser Arbeit behoben wurde, bauen all diese beschriebenen Beispiele auf ein und dasselbe Problem. Der SEnti-Analyzer lässt sich nicht benutzerfreundlich installieren. Eine grafische Oberfläche ist ein sehr wichtiger Schritt in Richtung Usability des SEnti-Analyzers. Wenn der SEnti-Analyzer jedoch tatsächlich extern bei Kunden zum Einsatz kommen soll muss die Installation deutlich verbessert werden, da kein Kunde sich damit auseinandersetzen möchte.

Literaturverzeichnis

- [1] Sentistrength. <http://sentistrength.wlv.ac.uk/>. Letzter Zugriff: 09.05.2022.
- [2] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi. SentiCR: A Customized Sentiment Analysis Tool for Code Review Interactions. In *32nd IEEE/ACM International Conference on Automated Software Engineering (NIER track)*, ASE '17, 2017.
- [3] P. A. Daniel Graziotin, Xiaofeng Wang. Are happy developers more productive? the correlation of affective states of software developers and their self-assessed productivity. *Product-Focused Software Process Improvement*, 2:50–64, 2014.
- [4] S. C. de Barros Sampaio, E. A. Barros, G. S. de Aquino, M. J. C. e Silva, and S. R. de Lemos Meira. A review of productivity factors and strategies on software development. In *2010 Fifth International Conference on Software Engineering Advances*, pages 196–204, 2010.
- [5] C. F., L. F., M. F., and N. N. Sentiment polarity detection for software development, 2018.
- [6] M. Herrmann and J. 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, 09 2021.
- [7] P. P. Karsten Michael Tymann, Matthias Lutz and C. Gips. Gervader - a german adaptation of the vader sentiment analysis tool for social media texts. in proceedings of the conference "lernen, wissen, daten, analysen". 2019.
- [8] B. Liu. *Sentiment Analysis and Opinion Mining*. 2012.

- [9] S. Loria. textblob documentation. *Release 0.15*, 2, 2018.