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 Entwicklungsteams

Development of a Graphical User Interface for the Analysis of Sentiments in Development Teams

Bachelorarbeit

im Studiengang Informatik

von

Tjelvar Olsen

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

Hannover, 01.07.2022

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 01.07.2022

_____

Tjelvar Olsen

Kurzfassung

Entwicklung einer grafischen Benutzeroberfläche zur Analyse von Stimmungen in Entwicklungsteams

Die Stimmung innerhalb von Entwicklerteams hat einen maßgeblichen Einfluss auf deren Arbeit. Durch sie wird die Produktivität und die Qualität der Resultate beeinflusst. Zur Ermittlung der Stimmung dienen Stimmungsanalyse-Tools als möglichst objektive Herangehensweise, um beispielsweise versteckte Missstände aufzudecken. Auf diesem Wege wird der Nutzende bei der Untersuchung der Produktivität nicht durch die Inhalte der Arbeit abgelenkt oder in die Irre geführt. Diese Tools werden oftmals über die Kommandozeile gesteuert. Dies ist aber eventuell problematisch für potenzielle Anwendende von solcher Software, denn diese sind nicht zwangsläufig informationstechnisch bewanderte Entwickler, sondern es könnte sich um Manager, Projektleiter oder Kunden handeln. Für diese sind unhandliche Terminal Eingaben und spezifische Abhängigkeiten eine Barriere. Solch eine Barriere schreckt von der Nutzung der Software ab und verhindert somit, dass Stimmungsanalyse in praktischen Fällen zum Einsatz kommt.

Um diese Barriere zu verringern, soll im Rahmen dieser Arbeit eine einfach zu bedienende Benutzeroberfläche geschaffen werden, in welcher die von Nutzenden durchzuführenden Schritte auf ein verständliches Mindestmaß reduziert werden. Dadurch soll eine leichtere Anwendung gewährleistet werden. Dies kann den Zugang zu diesen Tools deutlich erleichtern.

Um zu bewerten, ob die geschaffene Software für diesen Zweck geeignet ist, wird eine Interviewstudie durchgeführt. In dieser wird die Software bedient und ein Feedback dazu aufgezeichnet.

Abstract

Development of a Graphical User Interface for the Analysis of Sentiments in Development Teams

The sentiment within development teams has a significant influence on their work. It influences productivity and the quality of results. To determine the sentiment, sentiment analysis tools serve as the most objective approach possible. They can be used to uncover hidden problems, for example. In this way, the user is not distracted or misled by the content of the work when examining productivity. These tools are often controlled using the command line. However, this may be problematic for potential users of such software. The Users may not necessarily be information-savvy developers, but could be managers, project leaders, or customers. For them, unwieldy terminal inputs and specific dependencies are a barrier. Such a barrier discourages the use of the software and therefore prevents sentiment analysis from being used in practical cases.

To reduce this barrier, this thesis aims to create an easy-to-use user interface in which the steps to be performed by users are reduced to an understandable minimum. This should ensure an easier application. This can significantly improve the access to these tools.

In order to evaluate whether the created software is suitable for this purpose, an interview study will be conducted. In this study, the software is operated and feedback is recorded.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Stimmungsanalyse bietet einen Einblick in die Stimmung innerhalb von Entwicklerteams. Die Stimmung in Entwicklerteams steht im Zusammenhang mit deren Produktivität[35]. Beispielsweise wurde beobachtet, dass Probleme, bei denen in den dazu geführten Konversationen eine positive Stimmung zu beobachten war, schneller gelöst werden als solche, in denen das Gegenteil der Fall war[35]. Die Stimmung lässt sich mit Stimmungsanalyse-Tools automatisiert und überwiegend objektiv analysieren. Ein Beispiel für ein solches Tool ist CoreNLP[33]. CoreNLP[33] ist eines der fünf meist verwendeten Stimmungsanalyse-Tools in wissenschaftlichen Arbeiten[34]. Dieses Tool lässt sich nur über die Kommandozeile bedienen und benötigt eine spezifische Eingabesequenz, um die Stimmung eines Textes zu analysieren. Zusätzlich gibt das Programm umfangreiche Daten zu seinen Ergebnissen aus. In diesen sind nicht nur die Stimmungen enthalten, sondern auch zusätzliche Metriken, welche für den Prozess der Stimmungsanalyse notwendig waren. Dadurch ist die Ergebnisausgabe schwer zu lesen. Solche Eigenarten haben zur Folge, dass dieses Tool eher nicht von Nutzenden ohne Vorerfahrung verwendet wird[26]. Viele der gebräuchlichen Stimmungsanalyse-Tools sind ähnlich unhandlich zu bedienen und haben unterschiedliche Arten, ihre Daten auszugeben. Dies reduziert für Nutzende ohne entsprechende Vorkenntnisse die Motivation, diese zu verwenden[26]. Um diese Störfaktoren zu reduzieren, wird im Verlauf dieser Arbeit SATI (Sentiment Analysis Tool Interface) entwickelt, ein visuelles Interface für Stimmungsanalyse-Tools. Es soll die Installation, die Bedienung und die Auswertung von Stimmungsanalyse-Tools vereinheitlichen, verständlich darstellen und Möglichkeiten für Direktvergleiche bieten. Zudem sollen durch optionale Eigenschaften für die zu evaluierenden Inhalte tiefere Einblicke in das Analyseergebnis zur Verfügung gestellt werden, wie beispielsweise der zeitliche Verlauf der Stimmung innerhalb eines Inhaltes.

1.1 Problemstellung

Stimmungsanalyse-Tools sind oftmals nicht anwenderfreundlich designt. Der Fokus liegt auf der Ausführung ihrer Aufgabe. Sie bieten oftmals kein visuelles Interface und erfordern exakte Eingaben in Form von Kommandozeilenbefehlen oder spezifisch formatierten Datensätzen. Ein mögliches Beispiel hierfür ist CoreNLP[33], ein anderes ist Senti4SD[10]. Senti4SD[10] ist ebenso wie CoreNLP[33] eines der fünf am häufigsten verwendeten Tools in wissenschaftlichen Arbeiten[34]. Die Kommandozeileneingaben werden über ein Script gesteuert, welches dies vereinfachen soll. Allerdings verwendet dieses Tool Bibliotheken, welche nicht automatisch installiert werden. Diese Eigenart hat zur Folge, dass die Ausführung ohne die Installation der Bibliotheken mit einer komplizierten Fehlermeldung fehlschlägt. Die möglichen Nutzenden solcher Software sind jedoch nicht zwangsläufig informationstechnisch bewandert und werden durch solche Unhandlichkeiten an der Bedienung dieser Tools gehindert[26]. Solche voraussehbaren Fehler und die Interaktion mit der Kommandozeile stellen unter anderem die Barrieren dar, welche in dieser Arbeit überwunden werden sollen. Zu diesem Zweck soll ein visuelles Interface programmiert und gestaltet werden. Dieses soll einen Lösungsweg anzubieten, die angesprochenen Barrieren zu überwinden.

1.2 Lösungsansatz

Es soll ein Interface erstellt werden, welches möglichst einfach die Installation von Stimmungsanalyse-Tools, ebenso wie deren Ausführung und die Evaluation der Ergebnisse vereinheitlicht. Das Programm sollte in einer weit verbreiteten Programmiersprache simpel strukturiert sein. Dadurch sollen mögliche weiterführende Anpassungen einfacher gestaltet werden. Innerhalb des Interfaces soll es möglich sein, Datensätze zu importieren und exportieren. Die Datensätze werden mithilfe von verschiedenen Stimmungsanalyse-Tools oder manuell evaluiert und die Ergebnisse davon miteinander verglichen. Die Installation der Stimmungsanalyse-Tools soll, wenn möglich, ebenfalls von diesem Programm durchgeführt werden und dabei auch fehlende Abhängigkeiten auflisten. Ebenfalls soll es möglich sein, anhand existierender Analyseergebnisse, neue Versionen von Tools zu trainieren. Diese neuen Versionen werden als eigene Tools gewertet und sollen die Art der Stimmungsanalyse von der zugrundeliegenden Analyse implementieren, indem das Tool sich an den Ergebnissen der zugrundeliegenden Analyse orientiert.

1.3 Ergebnisse der Arbeit

Die entwickelte Software vereinheitlicht auf Windows, Linux und macOS Plattformen die Bedienung der Stimmungsanalyse-Tools und reduziert diese auf eine geringe Anzahl an Schritten zum Erreichen einer Bewertung. Sie ist verständlich in einer verbreiteten Programmiersprache umgesetzt, gut dokumentiert und einfach zu bedienen. Zu der Evaluation der Software wurde eine Interviewstudie durchgeführt. In dieser Interviewstudie wurde die Software bedient und ihre Nutzbarkeit bewertet. Die Teilnehmenden berichten von einer großteils problemlosen Bedienung. Sie ziehen in Erwägung, die Software auch zukünftig zu verwenden, um Einblicke in die Stimmung innerhalb ihrer Teams zu erlangen, oder sie zu diesem Zwecke weiterzuempfehlen.

1.4 Struktur der Arbeit

Diese Arbeit behandelt in Kapitel 2 die Grundlagen der Stimmungsanalyse ebenso wie die Funktionsweise der eingebundenen Stimmungsanalyse-Tools, angewandte Verfahren zur Quantifizierung der Urteilsübereinstimmung und die verwendete Programmierumgebung. In Kapitel 3 wird die Planung und die Umsetzung der Software behandelt. In Kapitel 4 werden die Resultate einer Interviewstudie bezüglich des Programms dargestellt. Danach werden in Kapitel 5 ähnliche Projekte aufgeführt und deren unterschiedliche Herangehensweisen. In dem darauf folgendem Kapitel 6 werden die Ergebnisse kurz zusammengefasst und Schlussfolgerungen gezogen. Im Anschluss werden mögliche Weiterentwicklungen der Software in Aussicht gestellt.

Kapitel 2

Grundlagen

In diesem Kapitel wird die Stimmungsanalyse grundlegend dargestellt. Es wird erläutert, worin die Unterschiede zwischen Stimmungsanalyse-Tools bestehen und was die angewandten Tools auszeichnet. Weiterführend werden Methoden zur Quantifizierung der Übereinstimmungen von Bewertungen erklärt. Es wird deren Anwendung beschrieben, mit welcher die Übereinstimmung zwischen verschiedenen Bewertungen bestimmt werden kann. Abschließend wird die verwendete Programmiersprache und die Entwicklungsumgebung NodeJS[5] dargestellt.

2.1 Stimmungsanalyse

Emotionale Prozesse und Zustände sind sehr komplex und können von so vielen Standpunkten analysiert werden, dass es unmöglich scheint, eine vollständige und zutre□ende Einschätzung zu erlangen[29]. Die Stimmungsanalyse versucht eine möglichst objektive Einschätzung eines Textes, Kommentars oder Dialoges anhand von konstanten Bewertungskriterien zu erzeugen[15]. Die Bewertungskriterien versuchen, den emotionalen Informationsgehalt eines Textes neutral auszulesen. Grundlage hierfür bieten oftmals Lexika und andere linguistische Ressourcen[15]. Dabei werden für einzelne Sätze Bewertungen erzeugt, welche positiv, negativ oder neutral ausfallen können[44].

Alternativ lässt sich mit diesem Verfahren ein Text auf Emotionen untersuchen, um diese festzustellen[44]. Diese Emotionen können daraufhin zur Stimmungsklassifizierung genutzt werden[28]. Negative Emotionen sind zum Beispiel Nervosität oder Enttäuschung, während positive Beispiele Frohsinn und Optimismus sind[28].

Die im Text enthaltenen Stimmungen oder Emotionen können auch aufgrund von vorhergehenden Ausdrücken vorhergesehen werden. Beispielsweise folgen auf Beleidigungen eher negative Emotionen. Somit können die Antworten auf diese ebenfalls negativ gewertet werden[38].

Die so erhaltenen Informationen können dann zurate gezogen werden, um negative oder positive Inhalte zu identifizieren, oder um objektive von subjektiven Informationen zu unterscheiden[43].

2.2 Stimmungsanalyse-Tools

Stimmungsanalyse-Tools haben unterschiedliche Funktionsweisen. Sie können sich auf einzelne Wörter fokussieren und anhand ihrer Polarität eine Summe für den Satz erzeugen[15]. In diesem Falle spräche man von einem lexikonbasierten Tool[15]. Hierbei werden einer vorgegebenen Liste von Wörtern Wertungen zugeteilt. Diese Wertungen reflektieren die Stärke und die Richtung (positiv oder negativ) der im Wort enthaltenen Stimmung. Solch eine Liste wird dann als Referenz zurate gezogen, um einzelne Wörter zu evaluieren, anhand derer dann für einen Satz oder Text mittels der Bewertungen der totale Stimmungsgehalt bestimmt wird[15]. Dabei kann dieser Prozess sich je nach Herangehensweise auf einzelne Schlüsselworte beschränken, da der Informationsgehalt von verschiedenen Worten stark voneinander abweichen kann. Eine besondere Bewertung könnten beispielsweise verstärkende Wörter wie 'sehr' erhalten[10]. Dieses Wort würde keine eigene Polarität besitzen, jedoch den Wert der Polarität des folgenden Wortes verstärken. So wäre zum Beispiel 'sehr gut' positiver zu bewerten als 'gut' und 'sehr schlecht' negativer zu bewerten als 'schlecht'[10]. Solche Bewertungen würden dann mittels spezifischen Regelungen gesteuert werden, welche sich von Tool zu Tool unterscheiden können.

Um die den Tools zugrunde liegenden Lexika zu erstellen, sollte eine objektive Bewertung von Inhalten vorliegen. Maschinelles Lernen kann angewandt werden, um solch eine Bewertung zu erlangen oder zu verfeinern[44]. Hierbei werden anhand eines manuell bewerteten Datensatzes Bewertungskriterien erzeugt, welche auf weitere Daten angewendet werden können[44]. Manuell bewertete Datensätze, die als Grundlage für das maschinelle Lernen verwendet werden, oder die direkt als Lexikon verwendet werden, sollten von mehreren Personen erstellt worden sein und einem Goldstandard entsprechen[25].

Es handelt sich um einen Goldstandard, wenn die Unterschiede bei den Bewertungen zwischen den Bewertern so gering wie möglich sind. Die Übereinstimmung lässt sich beispielsweise mit den Vergleichsalgorithmen aus Abschnitt 2.3 quantifizieren. Je geringer die Übereinstimmung bei der Bewertung ist, desto weiter entfernt sich ihre Ausgabe von dem Status Goldstandard und sollte nicht als Grundlage für eine Analyse dienen[25].

Wenn der Datensatz für ein bestimmtes Umfeld oder Themengebiet erstellt wird, lässt sich durch Übereinstimmung mit manuellen Bewertungen feststellen, dass das resultierende Tool besser für Texte dieses Themengebietes geeignet ist[22]. Dies führt dazu, dass man für eine Vielzahl von The-

mengebieten eine Vielzahl von Toolvariationen benötigt, um bestmögliche Ergebnisse zu erzielen[13].

2.2.1 CoreNLP

Stanfords CoreNLP[33] ist eine Bibliothek, welche mittels eines Systems von Pipelines mehrere Prozessoren natürlicher Sprache hintereinander ausführen kann. Diese Prozessoren fügen dem Text Kommentare hinzu, welche aufeinander aufbauen können. Im Folgenden werden diese Prozessoren als 'annotator' bezeichnet[33].

Um die Stimmung eines Textes mithilfe von CoreNLP zu bestimmen, muss der Text mehrere dieser annotatoren durchlaufen. Der Text muss zunächst in einzelne Sätze unterteilt werden. Dies geschieht mithilfe des 'ssplit' annotators. Diese Sätze werden wiederum in die einzelnen Wörter und Satzzeichen getrennt. Dies geschieht mithilfe des 'tokenization' annotators. Den einzelnen Wörtern wird dann anhand der Position im Satz ein Typ zugeteilt. In dem Satz 'Marie was born in Paris' würden die Wörter folgende Typen zugewiesen bekommen: 'Marie' → NNP, 'was' → VBD, 'born' → VBN, 'in' → IN, 'Paris' → NNP. Anhand dieser Aufteilungen und Zuweisungen wird der 'parse' annotator den Text in einen binären Baum umwandeln, welcher im letzten Schritt von dem 'sentiment' annotator auf die enthaltene Stimmung bewertet wird. Diese Bewertung erfolgt basierend auf Socher et al's Stimmungsmodell. Dieses Modell weist jedem Blatt des binären Baums einen Einfluss zu und kombiniert diese Zuweisungen entsprechend ihres Zusammenhanges, um die Wertung zu erhalten[33][39].

Für die Eingabe wird eine Datei benötigt, welche den Text beinhaltet. Anhand von Satzzeichen würden die Sätze automatisch voneinander getrennt werden. Die Ausgabe ist die Gesamtheit der Ausgaben aller annotatoren für jeden einzelnen Satz.

2.2.2 EmoTxT

EmoTxT[11] ist ein Tool, welches, anstelle zwischen negativer und positiver Stimmung zu unterscheiden, auf die zugrundeliegenden Emotionen prüft. So kann das Tool verwendet werden, um beispielsweise Freude, Liebe oder Ärger in Texten zu erkennen. Es bezieht sich wie andere Stimmungsanalyse-Tools auf ein Lexikon, welches anstelle von Polaritäten Emotionen klassifiziert. Das Lexikon besteht aus mehreren Teillexika, welche jeweils eine der Emotionen bestimmen können[11].

Das Lexikon basiert auf Bewertungen von Stack Overflow Einträgen von 12 Individuen, welche für Inhalte aus den Jahren 2008 bis 2015 die Anwesenheit oder die Abwesenheit von sechs grundlegenden Emotionen notieren sollten[11]. Jeder Eintrag wurde von genau drei Individuen evaluiert. Über den resultierenden Datensatz war für die einzelnen Emotionen eine

Übereinstimmung von 0.86 bis 0.98 zu beobachten, wobei 1 eine absolute Übereinstimmung wäre, was die Zuverlässigkeit der Daten unterstreicht[11].

Für jede Emotion läuft EmoTxT[11] bei der Bewertung einmal, um dann die Ergebnisse miteinander zu kombinieren. Innerhalb eines Durchlaufes wird das Liblinear[18] Framework angewandt, um die Klassifikationen zu bestimmen[11][18].

Die Eingabe erfolgt über die Kommandozeile und als Ausgabe wird, für jede Emotion, eine CSV Datei ausgegeben. Diese CSV Dateien enthalten die zu bewertenden Zeilen und ein binäres Ergebnis ('YES, 'NO'), welches darstellt, ob die betrachtete Emotion in der Zeile enthalten ist.

2.2.3 Fairseq

Fairseq[3] ist ein Sequenz Modellierungs Toolkit für Sprachmodellierung. Es umfasst mehrere Modelle und Herangehensweisen, um Stimmung anhand eines vortrainierten Modells zu analysieren. Von sich aus kann Fairseq[3] keine Stimmungen klassifizieren und dient lediglich als Trainer, welcher beispielsweise das RoBERTa[31] Modell darauf trainieren kann, Sätze zu klassifizieren, um dann damit Stimmungen zu bestimmen[3].

Das RoBERTa[31] Modell ist eine Weiterentwicklung des BERT[14] Modells, welches für ein Training von bidirektionalen Transformatoren für Sprachverständnis ausgelegt ist. Das BERT[14] Modell zeichnet sich durch zwei Trainingsabläufe aus. Zu Beginn wird das Training ohne Labels durchgeführt. Der Text wird interpretiert, ohne irgendwelche Schlussfolgerung vorzugeben. Dadurch wird das generelle Textverständnis des Modells verbessert. Als Nächstes wird das so erhaltene Modell im 'Finetuning' mit Labels auf die zu bewältigende Aufgabe spezialisiert. Hierbei wird das vorher erlangte generelle Textverständnis genutzt, um anhand von Zuweisungen einer Eingabe zu einem Ergebnis Zusammenhänge herzustellen, also in diesem Falle das Bestimmen der vorliegenden Stimmung in einer Zeile. Das BERT Modell ist somit für eine Mehrzahl an Aufgaben einsetzbar, wobei nur der 2. Schritt angepasst werden muss. Diese Aufgaben können neben Stimmungsanalyse auch das Beantworten von Fragen, das Übersetzen von Sätzen oder das Vervollständigen von Text sein[14].

Das RoBERTa[31] Modell unterscheidet sich von dem BERT[14] Modell in vier wesentlichen Punkten: (1.) das Modell wird länger und mit größeren Datenabschnitten über mehr Daten trainiert, (2.) die Möglichkeit, den nächsten Satz vorherzusagen wurde entfernt, (3.) trainiert wird auf längeren Sequenzen und (4.) dynamische Änderungen an dem Maskierungsverhalten während des Trainings[31].

Die Fairseq Bibliothek[3] bietet ein umfangreiches Kommandozeileninterface. Mit diesem lassen sich der Trainingsprozess und viele Aufgaben mit vortrainierten Datensätzen durchführen, zum Beispiel Übersetzungen oder den folgenden Satz vorhersagen. Alternativ kann programmatisch mit der

Bibliothek interagiert werden und dadurch weitere Aufgaben durchgeführt werden, wie zum Beispiel die Klassifikation eines Satzes auf die enthaltene Stimmung. Die Ausgabe variiert je nach angewandter Methode.

2.2.4 Senti4SD

Das Analysetool Senti4SD[10] fokussiert sich auf drei Methoden der Bewertung: (1.) ein generisches Stimmungslexikon, (2.) Schlüsselwörter und (3.) ein semantisches Modell, trainiert mit Daten aus der Softwareentwicklung[10].

In dem verwendeten Lexikon muss für jeden Eintrag eine Bewertung existieren. Anhand dieser Bewertungen werden verschiedene Werte für den zu analysierenden Satz aufgestellt. Diese Werte umfassen die Anzahl an positiv und negativ bewerteten Wörtern, Adjektive mit negativen oder positiven Auswirkungen, die Bewertung des zuletzt verwendeten Emojis, die positive und negative Summe der Bewertungen der verwendeten Wörter, die positivste und negativste Bewertung und die Wertung des letzten Wortes. Letztere wird zusätzlich verstärkt, wenn der Satz mit einem Ausrufezeichen endet[10].

Die Schlüsselwörter umfassen Unigrame, Bigrame, großgeschriebene Wörter, Ausdrücke für Gelächter, künstlich verlängerte Wörter, Satzzeichenwiederholungen, Nutzererwähnungen und die Tatsache, ob der Satz oder die Nachricht mit einem Ausrufezeichen beendet wurde[10].

Das semantische Modell besteht aus Vektoren, welche die Auswirkungen der Wortwahl in positiven, negativen und neutralen Sätzen umschreiben. Zusätzlich gibt es einen subjektiven Vektor, welcher eine Kombination aus dem negativen und positiven Vektor darstellt, um neutrale Sätze von anderen zu unterscheiden. Diese Vektoren wurden mittels word2vec[37] von einem Stack Overflow Datensatz[10] mit 3,8 Millionen Fragen und dazu 5,9 Millionen Antworten automatisch generiert[10].

Anhand der Bewertungen werden mithilfe der Open Source Bibliothek Liblinear[18] lineare Klassifikationen generiert. Diese Klassifikationen können dann verwendet werden, um die Stimmungen in Texten zu erkennen.[10].

Senti4SD[10] ist in der Lage, neue Datensätze zu trainieren. Hierfür muss ein Datensatz vorliegen, welcher für einzelne Zeilen die Polaritäten neutral, negativ und positiv enthält. Anhand dieser Vorlage kann ein neues Lexikon generiert werden, welches entsprechend dem zugrunde liegenden Datensatz neue Gewichte beinhaltet[10].

Senti4SD[10] wird in mehreren Schritten entweder mit Python oder mit R-Script und Java über die Kommandozeile ausgeführt. Als Eingabe dient eine Datei, die für jeden zu bewertenden Inhalt eine Zeile Text enthält. Die Ausgabe erfolgt über eine CSV Datei mit einer Zeilenidentifikation und der ausgewerteten Stimmung für die betre□ende Zeile. Es gibt drei mögliche Resultate: 'Positive', 'Negative' und 'Neutral'.

2.2.5 SEntiAnalyzer

SEntiAnalizer[19][20] ist ein Tool, welches mehreren Tools die Evaluation der Eingaben überlässt und deren Ergebnisse miteinander kombiniert. Für deutsche Eingaben werden GerVADER[42], BertDE[17], TextBlob-DE[7] und SentiStrength-DE[36] verwendet, während für englische Eingaben SentiStrength-SE[22], Senti4SD[10], SentiStrength[41] und TextBlob[32] angewendet werden. Die Ausgaben dieser Tools werden miteinander kombiniert, um einen Durchschnittswert zu erhalten, welcher dann seinerseits in positiv, neutral oder negativ übersetzt wird. Um geringen negativen Einwirkungen zuvorzukommen, werden vier Sonderfälle definiert, welche die Fälle abdecken, in denen die Hälfte der Ausgaben einer negativen oder positiven Polarität entsprechen und der anderen Hälfte dieser mit neutraler oder neutraler und positiver Polarität widerspricht. Dies wurde so eingebunden, da der SEntiAnalizer[19][20] eine Funktion beinhaltet, welche live Audio aufnehmen und in Echtzeit evaluieren kann. Bei dieser soll verhindert werden, dass sie bei zu geringer Negativität ausschlägt und den Nutzer durch solch einen Ausschlag fälschlicherweise alarmiert[19][20].

Die Eingabe erfolgt über die Kommandozeile, wobei die zu bewertenden Sätze als Datei vorliegen müssen. Die Ausgabe ist eine CSV Datei mit den individuellen Wertungen, einer ID, dem analysierten Satz und dem Median der Wertungen.

2.2.6 SentiStrength

Die Grundlage des SentiStrength[41] Tools bildet eine Liste von 298 positiven und 465 negativen Wörtern, klassifiziert mit Gewichtungen von 2 bis 5 oder -2 bis -5. Diese Gewichtungen spiegeln die Polarität der in dem Wort enthaltenen Stimmung wider. Wörter mit einer Wertung von 1 bis -1 haben einen zu geringen Informationsgehalt, um berücksichtigt zu werden, darum wurden diese entfernt. Die Klassifikationen wurden anhand von menschlichem Urteil vorgenommen. Sie wurden in einer weiterführenden maschinellen Trainingsphase automatisch angepasst. Diese Wörter umfassen englische Wörter und Slangausdrücke, welche häufig auf der online Plattform MySpace verwendet wurden[41].

Bevor diese Liste jedoch angewandt wird, korrigiert ein Algorithmus falsch geschriebene Wörter. Es werden immer alle Buchstaben entfernt, die mehr als zweimal in Folge auftreten. Für Buchstaben, die seltener doppelt auftreten, wie zum Beispiel 'c', wird dies bereits durchgeführt, wenn sie mehr als einmal in Folge auftreten. Ebenso werden doppelte Buchstaben entfernt, wenn das Wort, welches nach dieser Operation entsteht, eines wäre, das dem Algorithmus bekannt ist. Die Wörter mit unnötigerweise wiederholenden Buchstaben werden zusätzlich noch einen Punkt stärker gewertet, da dies üblicherweise ein Anzeichen für einen energetischen Ausdruck ist[41].

Zusätzlich wird eine Wortliste mit verstärkenden Wörtern verwendet, um Wörter zu markieren, welche die Emotion des darauf folgenden Wortes verstärken, ebenso wie eine Liste von negierenden Wörtern, welche die Emotion ihres Nachfolgers umkehren. Neben Wörtern wurden auch noch Emojis mit Gewichtungen von 1 bis 2 versehen[41].

Exzessive verwendete Satzzeichen oder ein Ausrufezeichen nach einem emotionalen Wort verstärken die Bewertung dieser Emotion zusätzlich um einen Punkt. Wenn es sich jedoch um eine Frage handelt, wird jegliche negative Wertung ignoriert, da Fragen wie 'are you angry?' aufgrund der Verwendung des Wortes 'angry' negativ eingestuft werden würden, es sich jedoch um eine neutrale Frage handelt[41].

SenitStrength[41] ist in der Lage anhand von anders bewerteten Datensätzen, seine Gewichtungen neu zu verteilen. Diese ersetzen das zugrundeliegende Lexikon, verändern somit die Betrachtungsweise und können andere Resultate erzeugen. SentiStrength-SE[22] hat solch einen modifizierten Datensatz. Dieser wurde im Gegensatz zu dem herkömmlichen SentiStrength[41] Datensatz auf Software-Engineering Daten spezialisiert und mit einem Datensatz von 5.600 händisch evaluierten Kommentaren auf der JIRA Plattform getestet[22]. Über diese Funktionalität kann auch die Sprache von SentiStrength geändert werden. So verwendet der Datensatz SentiStrength-DE[36] nur deutsche Wörter und ist somit anwendbar auf deutsche Texte[36].

Die Eingabe erfolgt über die Kommandozeile, wobei die zu bewertenden Sätze als Datei vorliegen müssen. Die Ausgabe ist eine CSV Datei in welcher ein positiver und ein negativer Wert zusammen mit dem Text ausgegeben werden. Die bestimmte Polarität ergibt sich, wenn beide Werte kombiniert werden.

2.2.7 VADER

VADER[21] ist ein Stimmungsanalysetool, das sich auf Texte aus sozialen Medien spezialisiert. Es basiert auf einem menschlich erstellten Lexikon. Es ist darauf optimiert, schnell mit Daten umzugehen, ohne dabei signifikant an Präzision zu verlieren. Das Lexikon wurde auf Basis von etablierten Stimmungsdaten erstellt und umfasst Abkürzungen, Emoticons und üblicherweise verwendeten Slang mit Stimmungsinformationen. Insgesamt handelt es sich um über 7.500 Einträge. Diese wurden bewertet auf einer Skala von -4 bis 4 mit mehr als 90.000 Bewertungen. Einträge mit einer Wertung von 0 wurden entfernt. Anhand dieses Lexikons wird ein Wert für den zu bewertenden Satz erstellt und angepasst, basierend auf internen Regelungen[21].

Diese Regeln wurden, ebenso wie das Lexikon, anhand von menschlichen Bewertungen von Sätzen aufgestellt und umfassen: (1.) Zeichensetzung, besonders das Ausrufezeichen hat eine verstärkende Wirkung, (2.) Großschreibung, besonders komplett großgeschriebene Worte werden stärker

gewichtet, (3.) Adverbien wie 'sehr' oder 'extrem', welche den Einfluss des nächsten Wortes verstärken, (4.) invertierende Wörter, beispielsweise 'but' signalisiert, dass der folgende Text stärker gewichtet werden sollte als der vorhergehende und (5.) Trigramme vor einem gewichteten Wort. Diese helfen dabei, fast 90 % der invertierten Stimmungen aufzudecken[21].

VADER[21] wird als Python Bibliothek bereitgestellt und somit erfolgt die Eingabe und Ausgabe über ein Python Programm.

Für VADER[21] wurde eine deutschsprachige Version erstellt, mit dem Namen GerVADER[42]. Dessen Spezialität ist, ebenso wie beim englischsprachigem Original, die Analyse von Texten aus sozialen Medien[42].

GerVADER[42] umfasst im Gegensatz zu VADER[21] ein Kommandozeileninterface, über welches eine Datei eingelesen werden kann, deren Zeilen individuell bewertet werden.

2.3 Urteilsübereinstimmung

Zur Bestimmung der Übereinstimmung unterschiedlicher 'Bewertungen der Polarität von Datensätzen' können statistische Verfahren zur Quantifizierung der Übereinstimmung bei Mengen von zwei oder größeren Anzahlen von Bewertungen verwendet werden. In dem in dieser Arbeit entwickelten Interface wurden vier dieser Verfahren implementiert. Cohens Kappa[12] und Kendalls Tau[23] sind auf die Bewertung der Übereinstimmung zwischen zwei Gruppen von Bewertungen beschränkt. Fleiss' Kappa[16] und Kendalls W[24] sind äquivalente Bewertungen, jedoch können diese für eine beliebig große Anzahl von Gruppen von Bewertungen angewendet werden. Der resultierende Wert dieser Verfahren reicht von +1 bis -1, wobei ein größerer Wert eine größere Übereinstimmung signalisiert. Landis und Koch haben 1977 eine geläufige Interpretation dieser Werte aufgestellt, diese ist in Tabelle 2.1 abgebildet[27]. Diese Interpretation ist jedoch nicht zweifelsfrei zutre□end, da es sich um individuelle Einschätzungen handelt.

Wert	Stärke der Übereinstimmung
<0	schlecht übereinstimmend
0.01 – 0.20	gering übereinstimmend
0.21 – 0.40	einigermaßen übereinstimmend
0.41 – 0.60	mittelmäßig übereinstimmend
0.61 – 0.80	stark übereinstimmend
0.81 – 1.00	fast perfekt übereinstimmend

Tabelle 2.1: Übersetzte Interpretation nach Landis und Koch[27]

2.3.1 Cohens Kappa

Cohens Kappa[12] ist ein statistisches Maß, um die Übereinstimmung von zwei Gruppen von Bewertungen widerzuspiegeln. Hierbei werden ein gemessener Übereinstimmungswert und ein erwarteter Übereinstimmungswert erzeugt. Der erwartete Übereinstimmungswert spiegelt die Übereinstimmung wider, falls die Bewertungen zufällig vergeben wurden. Ein negativer Wert des Kappas weist darauf hin, dass die Übereinstimmung noch geringer ist als aufgrund einer zufälligen Verteilung zu erwarten war[12].

Cohens Kappa[12] ist definiert als:

$$= \frac{p_o \quad p_e}{1 \quad p_e}$$

Die erwartete Übereinstimmung wird dargestellt von p_e, und p_o stellt die tatsächliche Übereinstimmung dar. Die erwartete Übereinstimmung wird berechnet, indem man das prozentuale Vorkommen einer möglichen Wertung multipliziert und dies aufsummiert[12].

In einem fiktiven Beispiel gibt es die möglichen Bewertungen 0 und 1, und Bewerter 1 hat in 60 % der Fälle 1 ausgegeben, während Bewerter 2 in 40 % der Fälle 1 ausgegeben hat. Mit diesen Informationen lassen sich die erwarteten Übereinstimmungen für 0 und 1, p_0 und p_1, bestimmen:

$$p_1 = 0{:}4 \quad 0{:}6 = 0{:}24$$

$$p_0 = 0{:}6 \quad 0{:}4 = 0{:}24$$

Diese können aufsummiert werden, um p_e zu bestimmen:

$$p_e = 0{:}24 + 0{:}24 = 0{:}48$$

Es ergibt sich, dass 48 % der Bewertungen aufgrund von Zufall übereinstimmen sollten. Nun muss p_o bestimmt werden. Hierfür muss der prozentuale Anteil der Fälle bestimmt werden, in denen die Bewertungen übereinstimmen. In dem Beispiel nehmen wir an, dass dies 70 % beträgt. Somit wäre Cohens Kappa Wert für dieses Beispiel wie folgt:

$$= \frac{0{:}7 \quad 0{:}48}{1 \quad 0{:}48} = 0{:}52$$

Dieser Wert wäre nach Tabelle 2.1 als mittelmäßige Übereinstimmung zu interpretieren.

2.3.2 Fleiss' Kappa

Fleiss' Kappa[16] ist ein weiteres Maß, um die Übereinstimmung von mehreren Bewertungen widerzuspiegeln. Im Gegensatz zu Cohens Kappa[12] ist

dieses jedoch anwendbar für einen Vergleich von zwei und mehr Gruppen von Bewertungen. Ähnlich wie Cohens Kappa[12] betrachtet Fleiss' Kappa[16] die Abweichung von der erwarteten Übereinstimmung und ist identisch definiert, allerdings unterscheidet sich die Bestimmung von der erwarteten Übereinstimmung p_e und der tatsächlichen Übereinstimmung p_o. Hierbei werden zunächst folgende Werte festgelegt[16]:

N = Anzahl der zu bewertenden Objekte

n = Anzahl der Gruppen von Bewertungen

k = Anzahl der möglichen Arten von Bewertungen

x_{ij} = die Anzahl der Bewertungen für Objekt i von einer Art j

Um p_e zu bestimmen, werden die prozentualen Verteilungen der verschiedenen Arten von Bewertungen quadriert und aufsummiert[16].

$$p_j = \sum_{i=1}^{N} \frac{x_{ij}}{nN}$$

$$p_e = \sum_{j=1}^{k} p_j^2$$

Um p_o zu bestimmen, muss bestimmt werden, wie weit die Mengen von Bewertungen bezüglich der zu bewertenden Objekte übereinstimmen. Dies wird realisiert, indem das Maß der Übereinstimmung der Bewertungen für das Objekt i mit allen möglichen Arten (n (n - 1)) von Bewertungen verglichen wird[16].

$$P_i = \frac{1}{n(n-1)} \left(\sum_{j=1}^{k} x_{ij}^2 - n \right)$$

$$p_o = \frac{1}{N} \sum_{i=1}^{N} P_i$$

In Tabelle 2.2 ist eine beispielhafte Verteilung von Stimmungsauswertungen von drei fiktiven Bewertenden dargestellt, aufgeteilt auf die Bewertungsmöglichkeiten positiv (1), neutral (0) und negativ (-1). Zusätzlich wurde der prozentuale Anteil von positiven, neutralen und negativen Bewertungen p_j und das Maß P_i, zu dem die Bewertenden bei einem einzelnen zu bewertenden Objekt übereinstimmen, bestimmt.

	-1	0	1	P_i
1	0	2	1	0.33
2	1	1	1	0
4	0	3	0	1
5	0	0	3	1
6	3	0	0	1
7	0	3	0	1
8	0	0	3	1
9	0	1	2	0.33
10	0	2	1	0.33
Total	4	15	11	
p_j	0.133	0.5	0.366	

Tabelle 2.2: Fleiss' Kappa Zuweisung für Beispielbewertungen

Nun lässt sich die durchschnittliche und die erwartete Übereinstimmung bestimmen, ebenso wie der anhand dieser Übereinstimmungen resultierende Kappa Wert.

$$p_e = 0{:}133^2 + 0{:}5^2 + 0{:}366^2 = 0{:}4021$$

$$p_o = \frac{1}{10}(0{:}33 + 0 + 1 + 1 + 1 + 1 + 1 + 0{:}33 + 0{:}33) = 0{:}7$$

$$= \frac{0{:}7 \quad 0{:}4021}{1 \quad 0{:}4021} = 0{:}458$$

Dieser Wert wäre nach Tabelle 2.1 als eine mittelmäßige Übereinstimmung zu interpretieren.

2.3.3 Kendalls Tau

Kendalls Tau[23], oder auch der Rangkorrelationskoe□zient, bewertet die Übereinstimmung bei einer Vergabe von Rängen, ohne die Berücksichtigung der Wahrscheinlichkeitsverteilung der Variable. Die möglichen Bewertungen müssen ordinal sein, damit die zu bewertenden Objekte auf Grundlage dieser geordnet werden können. Die Position der Objekte in der so entstehenden Reihenfolge ist der Rang des Objektes. Stärkere Abweichung in der Platzierung fallen hierbei stärker ins Gewicht als eine Vielzahl von geringeren Abweichungen, da diese weniger abweichende Ränge erzeugen als eine stärkere Abweichung, welche alle unter sich platzierten Evaluationen mit Zweifeln behaftet[23].

Kendalls Tau[23] ist definiert als:

$$= \frac{n_c \quad n_d}{\frac{n}{2}}$$

n_c = Anzahl der Konkordanzpaare

n_d = Anzahl der Diskordanzpaare

Die Anzahl der Konkordanzpaare wird bestimmt, indem die Ränge der 2. Menge entsprechend denen der 1. Menge sortiert werden und dann für jeden Rang nachgezählt wird, wie viele Ränge nach einem Rang kommen und welche nach der Wertung der 2. Menge nach ihm kommen sollten. Alle, die nicht diesem Schema entsprechen, werden als Diskordanzpaare gezählt. Dies wird für jeden Eintrag durchgeführt und aufsummiert[23].

Um Gleichstände bei Rängen zu kompensieren, wird die b Variante[9] verwendet, wobei die Anzahl der unentschiedenen Ränge in der Berechnung berücksichtigt wird. Ebenfalls werden bei der Aufstellung der Paare nur solche gewertet, welche nicht in einem Gleichstand mit dem zu betrachtenden Wert stehen[9].

$$= \frac{n_c \quad n_d}{\sqrt{(n_0 \quad n_1)(n_0 \quad n_2)}}$$

$$n_0 = \frac{n}{2}$$

$$n_n = \sum_i t_{n,i}(t_{n,i} \quad 1)=2$$

$t_{n,i}$ = Anzahl der Gleichstände in der i-ten Gruppe bei Menge n

Um den Zufall bei Gleichständen zu reduzieren, werden die Bewertungen innerhalb eines Gleichstandes bei der 1. Menge von Bewertungen nach den Rängen innerhalb der 2. Menge sortiert. Auf diese Weise wird das Ergebnis konstant sein und keinem Zufall ausgesetzt. Dies führt jedoch auch zu einem höheren Endergebnis.

i	R1	R2	Konkordant	Diskordant
1	1	1	8	-
2	1	1	8	-
3	1	1	8	-
4	1	1	8	-
5	1	1	8	-
6	1	1	8	-
7	1	0	1	1
8	0	1	7	-
9	0	0	1	-
10	0	0	1	-
11	0	0	1	-
12	0	0	1	-
13	0	-1	-	2
14	-1	0	-	-
15	-1	0	-	-
			60	3

Tabelle 2.3: Bestimmung von Konkordanz und Diskordanz

Ein Beispiel mit sortierten Bewertungen für die Aufstellung der Konkordanzpaare und Diskordanzpaare ist in Tabelle 2.3 dargestellt. Anhand dieser Aufstellung wurden in Tabelle 2.4 die Gleichstände für die möglichen Werte bestimmt.

Gleichstände 1		Gleichstände 2	
Wert	Zähler	Wert	Zähler
1	7	1	7
0	6	0	7
-1	2	-1	1

Tabelle 2.4: Gleichstände unter den Rängen

Anhand auf diese Weise aufgestellter Daten lassen sich nun die Werte n_1 und n_2 bestimmen, was folgendes Tau ergeben würde:

$$n_1 = \frac{7\ (7\ \ 1)}{2} + \frac{6\ (6\ \ 1)}{2} + \frac{2\ (2\ \ 1)}{2} = 37$$

$$n_2 = \frac{7\ (7\ \ 1)}{2} + \frac{7\ (7\ \ 1)}{2} + \frac{1\ (1\ \ 1)}{2} = 42$$

$$= \frac{60\ \ 3}{\sqrt{(105\ \ 37)(105\ \ 42)}} = 0{:}871$$

Dieser Wert wäre nach Tabelle 2.1 als fast perfekte Übereinstimmung zu interpretieren.

2.3.4 Kendalls W

Kendalls W[24], oder auch Kendallscher Konkordanzkoe□zient, vergleicht ebenso wie Kendalls Tau die Platzierung der Bewertungen in einer Rangliste, ohne die Wahrscheinlichkeitsverteilung zu berücksichtigen. Bei Kendalls W[24] besteht jedoch die Möglichkeit, zwei oder mehr Bewertungen zu vergleichen. Gleichstände werden in dieser Berechnung kompensiert, indem allen identischen Bewertungen ihr durchschnittlicher Rang verliehen wird. Also würden Bewertungen wie 10, 9, 9, 9, 8 nicht als Platz 1, 2, 3, 4, 5, sondern als 1, 3, 3, 3, 5 interpretiert, da Rang 3 der durchschnittliche Rang der Wertung 9 ist[24].

Kendalls W[24] ist definiert als:

$$W = \frac{12S}{m^2(n^3 \quad n)}$$

S ist die quadratische Abweichung von dem Durchschnitt der Bewertungen und wird wie folgt berechnet:

$$S = \sum_{t=1}^{n} (R_i \quad \overline{R})^2$$

R_i ist der aufsummierte Rang, den alle Gruppen von Bewertungen einem Objekt i zugewiesen haben, und $\overline{R}$ ist der Median all dieser Ränge.

$$R_i = \sum_{j=1}^{m} r_{i,j}$$

$$\overline{R} = \frac{1}{n} \sum_{i=1}^{n} R_i$$

Objekt	WertA	RangA	WertB	RangB	WertC	RangC	R_i
1	1	3	1	2	1	3	8
2	1	3	1	2	1	3	8
3	1	3	1	2	1	3	8
4	1	3	0	6.5	1	3	12.5
5	1	3	0	6.5	1	3	12.5
6	0	6.5	0	6.5	0	8	21
7	0	6.5	0	6.5	0	8	21
8	-1	9	0	6.5	0	8	23.5
9	-1	9	0	6.5	0	8	23.5
10	-1	9	-1	10	0	8	27

Tabelle 2.5: Rangaufstellung für Kendalls W mit Anpassungen für Gleichstände

Die Werte in Tabelle 2.5 erfassen die Daten von drei fiktiven Parteien. Ihre Ränge wurden auf Grundlage der enthaltenen Gleichstände angepasst und die aus diesen resultierenden totalen Ränge R_i aufgestellt. Um den Median zu bestimmen, müssen nun die totalen Ränge aufsummiert und durch die Anzahl der Objekte geteilt werden:

$$\overline{R} = \frac{8 + 8 + 8 + 12{:}5 + 12{:}5 + 21 + 21 + 23{:}5 + 23{:}5 + 27}{n} = 16{:}5$$

Dieser Wert wird dann für die Berechnung des S Wertes verwendet:

$$S = (8 \quad 16{:}5)^2 + (8 \quad 16{:}5)^2 + (8 \quad 16{:}5)^2 + (12{:}5 \quad 16{:}5)^2 + ::: + (27 \quad 16{:}5)^2$$

$$S = 497{:}5$$

Dies resultiert in folgendem Wert für den Konkordanzkoe□zient:

$$W = \frac{12 \quad 497{:}5}{3^2(10^3 \quad 10)} = 0{:}67$$

Dieser Wert wäre nach Tabelle 2.1 als eine starke Übereinstimmung zu interpretieren.

2.4 NodeJS

Für die Umsetzung des visuellen Interfaces wird NodeJS[5] verwendet. Hierbei handelt es sich um eine JavaScript-Laufzeitumgebung[5]. Diese ist weitestgehend von dem Betriebssystem unabhängig. Da JavaScript die am weitesten verbreitetste Programmiersprache in Open-Source-Projekten ist[4], bietet sich diese an, um ein potenziell erweiterbares Programm zu entwickeln.

Um in dieser Umgebung eine visuelle Oberfläche zu ö□nen, kann Electron[2] verwendet werden. Electron[2] ist ein Framework, welches Chromium[1] und NodeJS[5] vereint, um eine Desktopanwendung zu erzeugen, welche mittels Webtechnologien ein visuelles Interface erzeugt. Diese Konstellation bietet eine Abstraktionsschicht, welche das Interface vor unterschiedlichen Einflüssen von Betriebssystemarchitekturen bewahrt und eine universelle Schnittstelle zu diesen bereitstellt. Eine Electron[2] Anwendung lässt sich für Windows, Linux und macOS gleichermaßen compilieren[2].

Unter Webtechnologien ist React[6] eine weit verbreitete Frontend-Javascript Bibliothek, um Webseiten zu gestalten, welche funktionale Interaktionen bieten, sogenannte Web-Apps. Es bietet schnelle Interaktionen und vollständige Kontrolle über die Beschaffenheit des Interfaces. Dies macht es zu einer beliebten Lösung, um in einer Elektron-Umgebung das Interface zu gestalten und die Interaktionen zu steuern[6].

Kapitel 3

Umsetzung

In diesem Kapitel wird zunächst die Planung des Tools beschrieben. Darauf folgt die Beschreibung der Struktur der zugrunde liegenden Plattform, in welche die Funktionen der Tools integriert werden. Abschließend wird die Integration der Tools beschrieben.

3.1 Planung

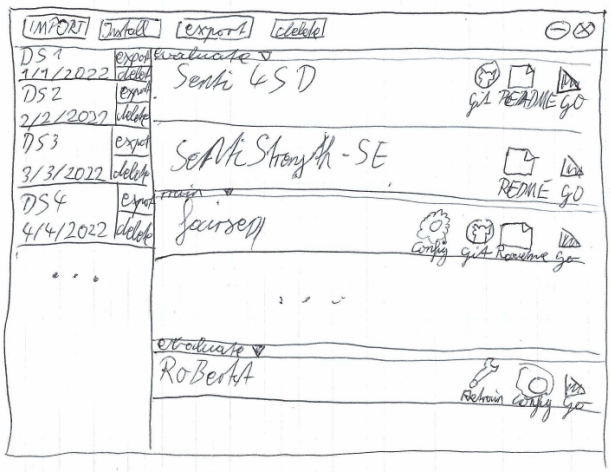

Abbildung 3.1: Der ’Paper Prototype’

Bevor die Umsetzungsphase des Tools beginnt, wird eine Planungsphase durchlaufen. Diese dient dazu, um eine Vorstellung davon zu erlangen, wie das Resultat aussehen soll. Zu Beginn der Planungsphase werden elementare Use Cases erstellt, welche die wichtigsten Funktionen des Interfaces beschreiben. Die wichtigsten Funktionen, welche auf diese Weise dokumentiert wurden, sind das Installieren der Software, das Importieren

von zu bewertenden oder bereits bewerteten Daten aus einer CSV Datei, das Installieren der Stimmungsanalyse-Tools und deren Ausführung. Die Use Cases sind dargestellt in Anhang A.1, A.2 und A.3.

Um diese Use Cases in einem Interface zu bündeln, wurde ein sogenannter Paper-Prototyp erstellt, welcher einen Leitfaden darstellt, wie die Oberfläche gestaltet werden sollte. Dieser ist in Abbildung 3.1 abgebildet. Er umfasste zwei miteinander interagierende Listen und eine Steuerungsfläche. In einer Liste sollten die Datensätze aufgelistet sein und in der anderen die installierten Tools. Alle weiterführenden Interaktionen sollten über die Steuerungsfläche erreichbar sein. Die weiterführenden Interaktionen sollten Import und Export von Daten umfassen, ebenso wie die Installation von Tools und deren Abhängigkeiten.

3.2 Plattform

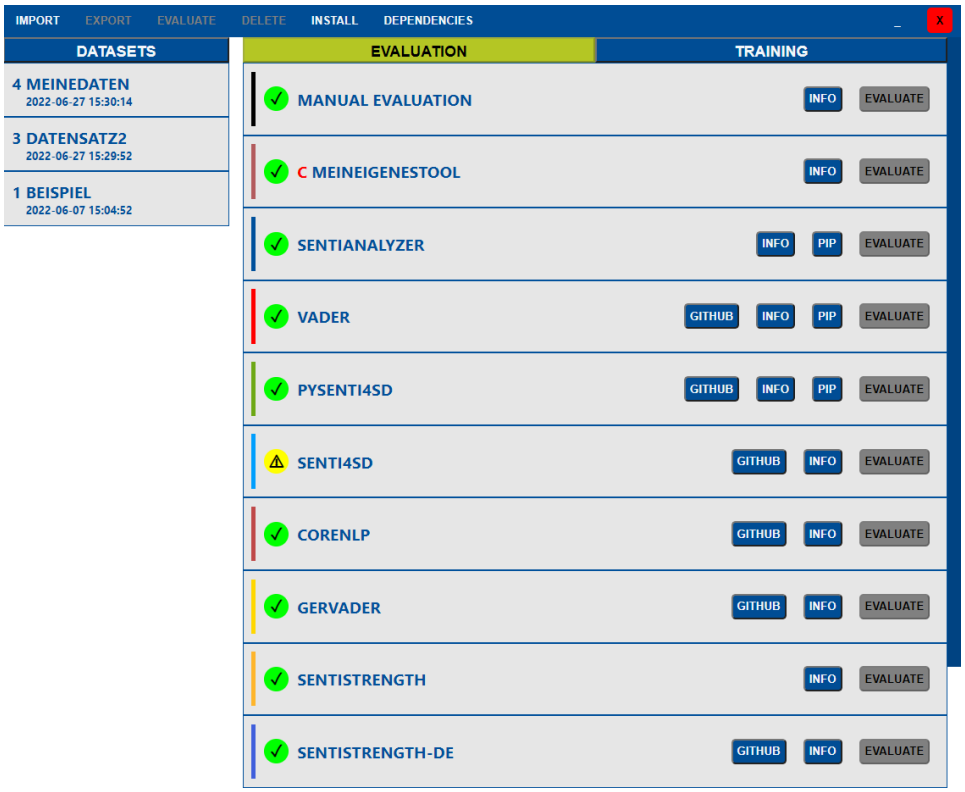

Abbildung 3.2: SATI Startbildschirm

Der Startbildschirm der entstandenen Plattform ist in Abbildung 3.2 abgebildet. Er umfasst die geplanten zwei Listen mit Datensätzen und Tools.

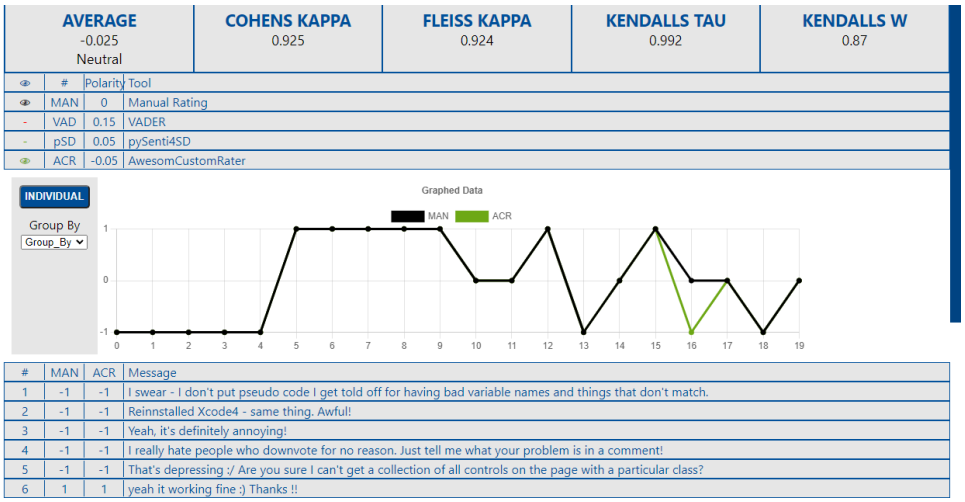

Abbildung 3.3: SATI Evaluationsübersicht

Zusätzlich kann bei den Tools zwischen Evaluatoren und Trainern unterscheiden werden. Die Trainer können anhand eines evaluierten Datensatzes neue Evaluatoren hinzufügen. Über die Steuerungsfläche kann der Import von neuen Datensätzen gestartet werden. Zudem können Operationen mit dem aktiven Datensatz ausgeführt werden. Diese Operationen sind: (1.) das Exportieren des aktiven Datensatzes, (2.) das Anzeigen der Evaluation von diesem anhand von bereits durchlaufenden Evaluationsprozessen und (3.) das Löschen des aktiven Datensatzes und aller zugehörigen Daten. Zusätzlich können von hier aus die bekannten Abhängigkeiten überprüft werden. In dem Fall, dass eine bekannte Abhängigkeit nicht erfüllt wird, kann eingesehen werden, wie diese erfüllt werden kann, oder von wo sie installiert werden muss. Wenn alle Voraussetzungen für ein Tool erfüllt sind, kann dieses von der 'INSTALL' Fläche aus installiert werden. Für alle online verfügbaren Tools ist dies möglich. Nach der Ausführung einer Evaluation, oder, wenn für einen Datensatz alle Evaluationen angezeigt werden sollen, wird eine Übersicht aufgerufen. In dieser Ansicht wird grafisch dargestellt, welche Polaritäten vergeben wurden. Ebenfalls werden die Urteilsübereinstimmungen berechnet und angezeigt. Zusätzlich wird eine Liste aller Nachrichten des Datensatzes mit den zugehörigen Bewertungen ausgegeben. Die grafische Anzeige kann abgeändert werden, um anstelle der individuellen Bewertungen eine Aufsummierung dieser anzuzeigen, oder nach einer vergebenen Eigenschaft, wie beispielsweise Autoren, zu gruppieren, um die von dieser Gruppe ausgehenden Stimmungen einzusehen. Diese Ansicht ist in Abbildung 3.3 dargestellt.

3.3 Integration

Die Integration der Tools erfolgt über zwei bis vier Dateien. Diese umfassen eine Indexdatei, welche dazu dient, das Tool zu beschreiben und die Verknüpfung zu den weiterführenden Funktionen herzustellen. In dieser Datei werden Abhängigkeiten aufgelistet, Quellen angegeben und toolspezifische Designelemente festgelegt. Die andere notwendige Datei ist die Datei, welche für die Ausführung des Tools zuständig ist. In dieser wird definiert, wie ein Datensatz formatiert werden muss, um von dem Tool verarbeitet zu werden. Außerdem werden die Schritte festgelegt, die das Tool ausführen muss, und es wird festgelegt, wie das Ergebnis zu interpretieren ist. Eine weitere optionale Datei behandelt die Automatisierung der Installation des Tools, falls dieses öffentlich verfügbar ist. Falls das Tool in der Lage ist, trainiert zu werden, wird für die Steuerung des Trainierens eine weitere Datei benötigt. In dieser muss der Ablauf des Trainierens einer neuen Variation des Tools anhand von einem evaluierten Datensatz programmiert werden.

Kapitel 4

Evaluation

Um die Software zu evaluieren, wurde eine Interviewstudie durchgeführt. Innerhalb dieser sollten die Befragten das Interface auf einem Windows Computer anwenden und den nötigen Aufwand bewerten. Ein Windows System wurde verwendet, da dieses Betriebssystem das verbreitetste System ist und SATI für dieses am meisten getestet wurde [8]. Zu Beginn wurden Fragen bezüglich des demografischen Hintergrunds der Teilnehmenden gestellt. Danach wurden die Aufgaben durchgeführt und abschließend Fragen zu dem Gesamteindruck gestellt. Des Weiteren wurden Prioritäten für weitere Entwicklungen und Verbesserungsvorschläge erfragt. Alle Ergebnisse wurden von dem Interviewenden notiert und zusätzliche Informationen durch Beobachtung erfasst, um präzisere Einblicke in potenzielle Fehler und Probleme zu erhalten. Die Ergebnisse dieser Interviewstudie werden in diesem Kapitel wiedergegeben.

4.1 Teilnehmende

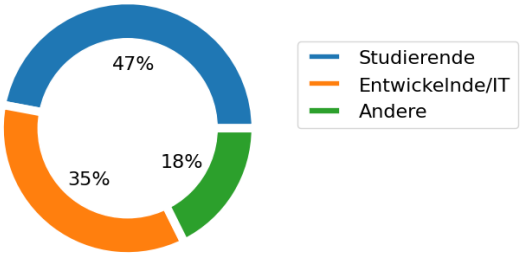

Abbildung 4.1: Verteilung der beruflichen Tätigkeit der Teilnehmenden

Es haben 17 Personen an der Interviewstudie teilgenommen. Diese waren zum Zeitpunkt der Studie 19 bis 39 Jahre alt. Die Teilnehmenden waren großteils Studierende der Universität Hannover und informationstechnisch Beschäftigte, siehe Abbildung 4.1.

Die Befragten sollten ihre eigene Programmiererfahrung einschätzen. Die Ergebnisse dieser Selbsteinschätzung sind in Abbildung 4.2 dargestellt. Die Einschätzungen der Teilnehmenden sind recht gleichmäßig verteilt, wobei die Programmiererfahrung eine durchschnittliche Bewertung von 3.06 erhält. Dies entspricht mittlerer Erfahrung. Da die einzelnen Antworten zusätzlich von 1 bis 5 verteilt sind, ist dies ist ein ausgeglichener Wert, um eine große Abdeckung von verschiedenen Erfahrungsständen zu haben.

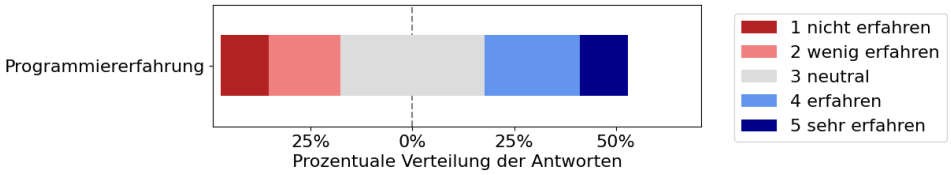

Abbildung 4.2: Programmiererfahrung

Ebenso wie die Programmiererfahrung sollten die Teilnehmenden ihre Nutzung der Kommandozeile einschätzen. Die Ergebnisse hiervon sind in Abbildung 4.2 enthalten. Ebenso wie bei der Programmiererfahrung sind die Teilnehmenden bezüglich der Kommandozeilennutzung gleichmäßig verteilt, mit einem Median von 3.29. Dies entspricht ebenfalls der mittleren Wertung, und die Abdeckung von verschiedenen Erfahrungsständen ist ebenfalls gewährleistet.

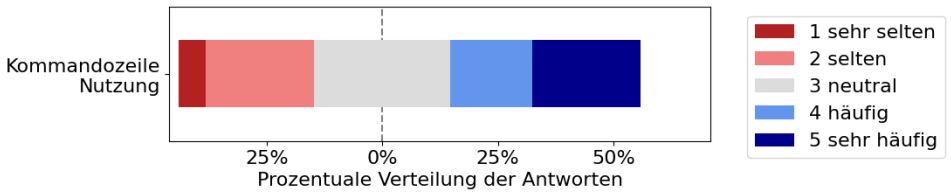

Abbildung 4.3: Nutzung der Kommandozeile

Circa 65 Prozent der Teilnehmenden haben vor der Umfrage bereits von 'Natural Language Processing' gehört, und ein wenig über 50 Prozent haben von dem Gebiet der Stimmungsanalyse bereits etwas gehört. Dieses Verhältnis ist in Abbildung 4.4 dargestellt.

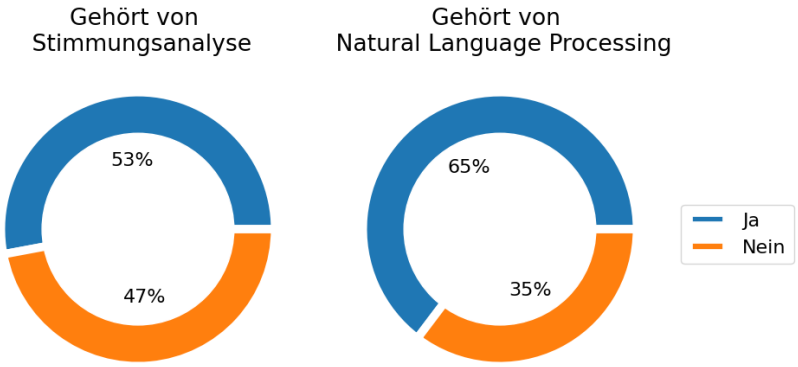

Abbildung 4.4: Vorerfahrung mit Stimmungsanalyse

Von allen Teilnehmenden hat nur einer in der Vergangenheit Stimmungsanalysesoftware erfolgreich angewandt. Der häufigste Grund dafür, dass ein Tool, von dem ein Teilnehmender erfahren hatte, nicht angewandt wurde, ist, dass dessen Ausführung nicht notwendig war. Ebenfalls angegeben wurde, dass die Ausführung fehlschlug, ebenso wie, dass zwar Interesse an der Ausführung bestand, jedoch nicht ersichtlich war, wie diese durchgeführt werden sollte. Dies wurde jedoch jeweils nur von einer Person angegeben. Ein Teilnehmender war in der Situation, dass jemand anderes das Tool für ihn ausgeführt hatte. Dieses Verhältnis ist in Abbildung 4.5 dargestellt

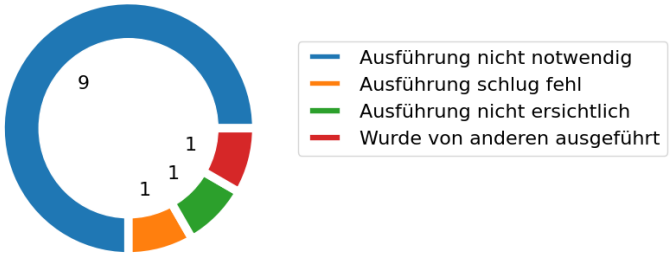

Abbildung 4.5: Grund für Nichtausführung bekannter Tools

4.2 Aufgaben

Die Aufgaben, welche in der Studie durchgeführt werden sollten, wurden in zwei Blöcke aufgeteilt. Jede Aufgabe sollte von 1 bis 5 bewertet werden. Der Wert 5 entspricht: 'die Aufgabe war sehr gut zu bewältigen', und 1 entspricht: 'die Aufgabe war sehr schwer zu bewältigen'. Die Bewertungen innerhalb des ersten Blocks sind in Abbildung 4.6 dargestellt. In diesem

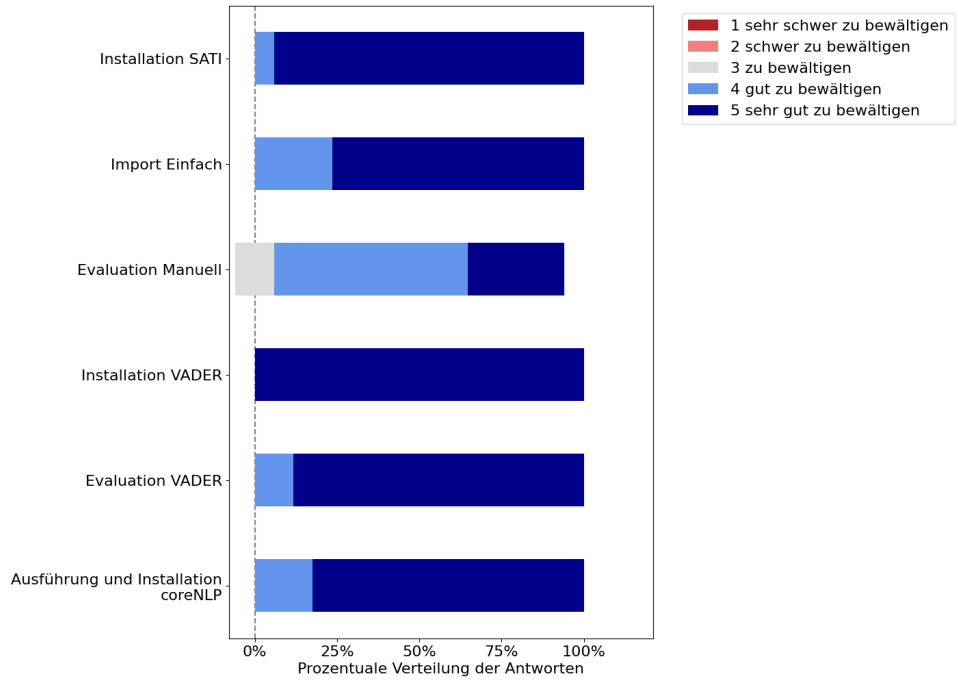

Abbildung 4.6: Bewertung des ersten Aufgabenblocks

wurden die einfacheren Interaktionen durchgeführt wie das Installieren, das manuellen Evaluieren und das automatische Evaluieren mithilfe des Tools. Als Erstes sollte das Programm SATI installiert werden, danach sollte ein Datensatz, der nur Nachrichten enthält, importiert werden. Dieser sollte manuell evaluiert werden. Nachdem dies erledigt wurde, sollte das Tool VADER[21] installiert und angewandt werden, um den Datensatz automatisch zu evaluieren. Abschließend sollte das Tool CoreNLP[33] installiert und verwendet werden, um einen Vergleich zwischen den Resultaten der Tools und der manuellen Bewertung zu ermöglichen. Besonders negativ fällt bei diesen Vorgängen nur die manuelle Evaluation auf. Die Teilnehmenden hatten Schwierigkeiten sofort zu verstehen, was genau von ihnen verlangt wurde und schlugen eine bessere Beschriftung der Knöpfe vor, da nicht auf den ersten Blick ersichtlich war, wofür 1, 0 und -1 stehen soll. Hinzu kam, dass es sich um einen lästigen händischen Prozess handelte, und dass die Sätze teilweise schwer zu deuten waren. Die verwendeten Sätze waren Englischsprachig und aus dem Bereich des Software-Engineerings. Ebenfalls negativ aufgefallen ist die Importanzeige während des zweiten Schrittes, in welchem ein Datensatz importiert werden sollte. Doch da an dieser Stelle dort noch keine Aktionen innerhalb dieser Ansicht des Interfaces notwendig waren, ist die Bewertung nicht negativ ausgefallen. Es gab noch weitere

Schwierigkeiten beim Verstehen davon, wie genau die Evaluation gestartet werden sollte, doch diese waren nicht so schwerwiegend, als dass sie die Bewertung beeinflussten. Ein paar Teilnehmende haben sich an dem Mangel einer Fortschrittsanzeige oder einer Information darüber, dass der Download von CoreNLP[33] länger dauert, gestört, was jedoch nach Bewertung der Teilnehmenden die Nutzung des Programms nicht erschwert hat.

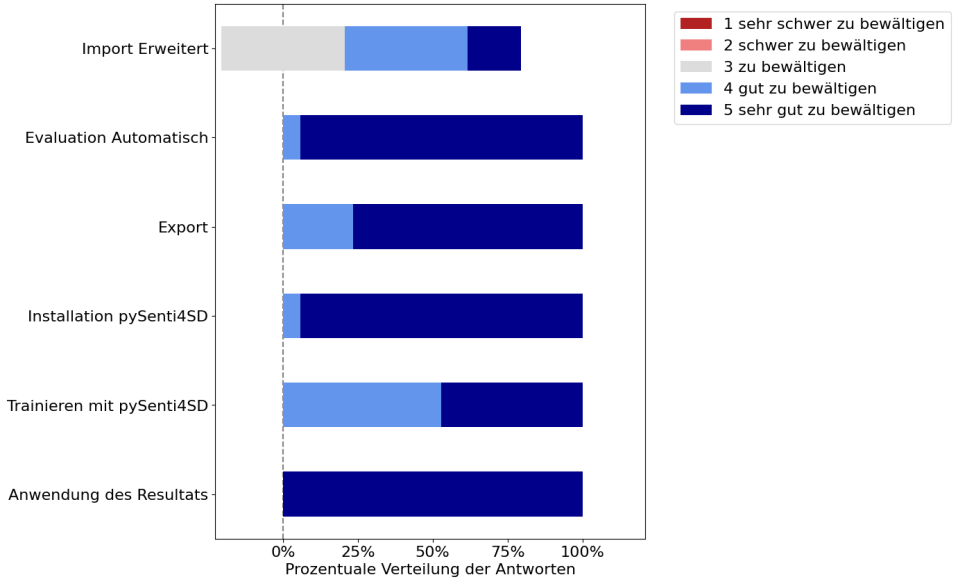

Abbildung 4.7: Bewertung des zweiten Aufgabenblocks

Die Bewertungen der individuellen Schritte des zweiten Blocks sind in Abbildung 4.7 dargestellt. In diesem Block sollte ein erweiterter Datensatz importiert werden, welcher zusätzlich zu jeder Nachricht einen Autor und eine manuelle Bewertung enthielt. Dieser Datensatz sollte dann von einem Tool evaluiert und anschließend exportiert werden. Danach sollte die Python Version des Tools Senti4SD[10] mit dem Namen pySenti4SD installiert werden. Diese sollte dann verwendet werden, um anhand einer manuellen Bewertung eine eigene Version von Senti4SD[10] zu generieren, welche dann als neues Tool anwendbar ist. Das entstandene Tool sollte anschließend verwendet werden, um einen Datensatz zu evaluieren. Hierbei fällt der erweiterte Importschritt im Verhältnis zu den anderen Schritten negativ auf. In diesem mussten die Anwendenden einen Datensatz mit zusätzlichen Informationen importieren. Es gab hierbei vermehrt Schwierigkeiten, die richtigen Zuweisungen festzulegen, doch sobald das Prinzip verstanden wurde, verlief alles ohne Probleme. Schwierigkeiten traten ebenfalls beim Trainieren auf. Hier waren in den Voreinstellungen bei der Eingabe des Tool-Namens und der Abkürzung die Felder nicht zweifelsfrei beschriftet.

Zusätzlich wurde die Ansicht, in welcher die Tools mit der Funktionalität zu trainieren aufgelistet sind, oftmals nicht sofort gefunden.

Über alle Aufgaben in dem ersten und zweiten Block hinweg entstand eine durchschnittliche Bewertung von 4.7, was 'sehr gut zu bewältigen' entspricht.

4.3 Gesamtwertung

Der Gesamteindruck der Bedienung von SATI wurde separat bewertet und wird in Abbildung 4.8 veranschaulicht. Die Teilnehmenden hatten keine schwerwiegenden Schwierigkeiten, die Aufgaben durchzuführen. Die negativeren Kritikpunkte lagen eher bei der Intuitivität und dem allgemeinen Design.

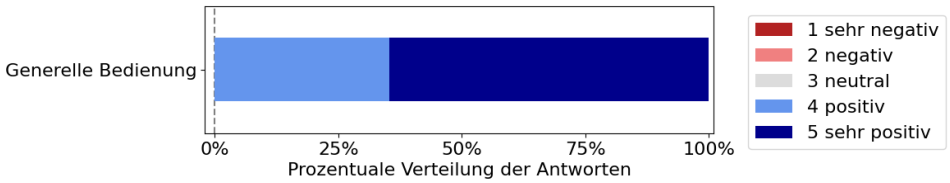

Abbildung 4.8: Bewertung des Gesamteindrucks

Ebenfalls separat bewertet wurde die Übersichtlichkeit der Evaluationsübersicht in Abbildung 4.9. Bei dieser wurde bemängelt, dass bei einer größeren Datenmenge eventuell zu viel auf einmal angezeigt wird, und dass bei geringer Größe des Fensters die Inhalte von einzelnen besonders langen Nachrichten nicht einzusehen sind.

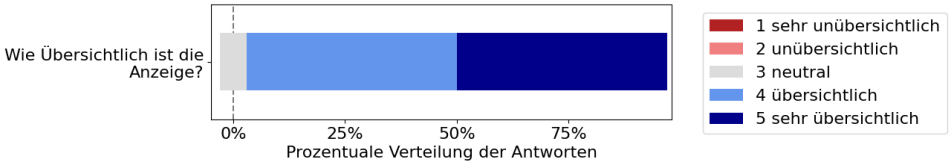

Abbildung 4.9: Bewertung der Evaluationsübersicht

Zusätzlich sollten die Befragten sich dazu äußern, ob sie lieber die Kommandozeile verwenden würden oder das Programm SATI, um eine Stimmungsanalyse durchzuführen. Die Ergebnisse hiervon werden in Abbildung 4.10 dargestellt. Ein Großteil der Teilnehmenden waren der Meinung, dass sie SATI lieber verwenden würden als die Kommandozeile. Eine kleinere Gruppe merkten jedoch an, dass es wahrscheinlich Szenarien gebe, in denen trotzdem die Ausführung mittels Kommandozeile notwendig wäre, da

Parameter übergeben werden müssten, die über das Programm noch nicht zu übergeben sind. Doch unabhängig davon haben die Teilnehmenden dennoch überwiegend das Interface gewählt.

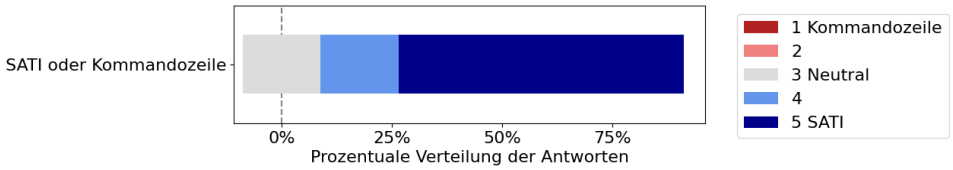

Abbildung 4.10: Bewertung der Präferenz

Als letzte Bewertung über die Gesamtheit des Programms wurde die Intuitivität der Bedienung des Programms bewertet, was in Abbildung 4.11 veranschaulicht wird. Rund 70 Prozent der Anwendenden fanden, dass das Interface intuitiv zu bedienen ist, auch wenn es ein paar Stellen gab, an denen nicht sofort ersichtlich war, was getan werden sollte. Dies betri□t vor allem die Zuweisungen bei einem erweiterten Import und die Anzeige mit den Voreinstellungen für den Trainingsprozess.

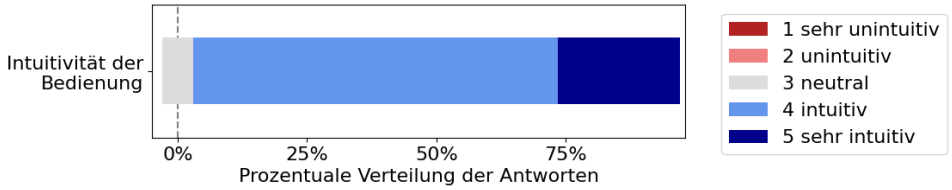

Abbildung 4.11: Bewertung der Intuitivität

4.4 Nutzbarkeit und Verbesserungen

In Abbildung 4.12 ist dargestellt, wie die Teilnehmenden bewerteten, ob sie das Programm außerhalb dieser Umfrage verwenden würden und ob sie es weiterempfehlen würden. Die Mehrzahl der Nutzenden hielt es für unwahrscheinlich, dass sie das Programm außerhalb der Umfrage verwenden würden. Dies begründet sich dadurch, dass sie es für unwahrscheinlich halten, in eine Situation zu geraten, in der sie es verwenden würden. Im Gegensatz dazu ist sich die Mehrzahl der Nutzenden einig darüber, dass sie das Programm weiterempfehlen würden.

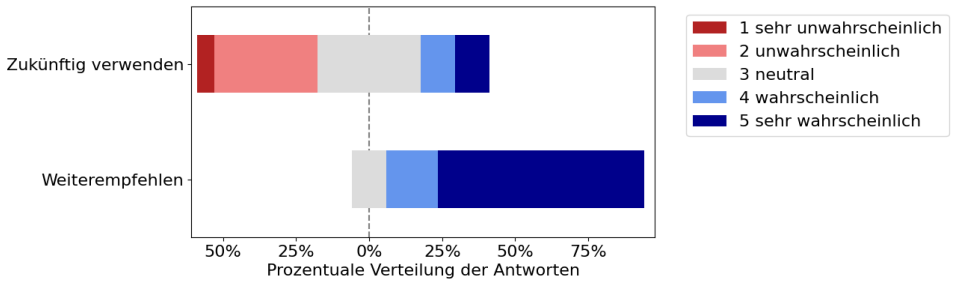

Abbildung 4.12: In Zukunft nutzen und weiterempfehlen

Die Teilnehmenden wurden über mögliche zusätzliche oder verbesserte Funktionen befragt, welche während der Entwicklung diskutiert wurden. Diese Features waren: ’Mehr Einstellungsmöglichkeiten’, ’Auslagern des Trainingsprozesses auf ein stärkeres System’, ’Export von spezifischen Feldern’, ’live Audio Aufnahme mit Evaluation in Echtzeit’ und ’In App crawlen von Webseiten zur Datensatzerstellung’. Am meisten wurde die Funktion zum Crawlen von Webseiten priorisiert, um auf diesen Wegen Datensätze zu erstellen und nicht den Umweg über eine CSV Datei zu gehen. Das Ergebnis ist in Abbildung 4.13 dargestellt.

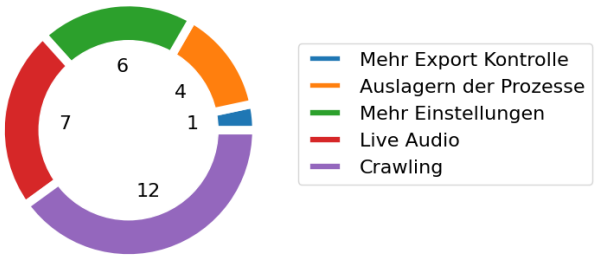

Abbildung 4.13: Neue Funktionen

Abschließend wurde eine o□ene Frage nach Verbesserungen gestellt. Der Fokus der Antworten lag hierbei auf der Verbesserung der Anzeige. Es wurde ein moderneres Design gewünscht, zusätzliche Informationen an Statusmeldungen wie zum Beispiel Zeitstempel, ebenso wie Tooltips und Überschriften an den Feldern und Listen, bei denen nicht sofort klar war, was deren Funktion ist. Dies soll vor allem die Intuitivität fördern, wobei in der Regel auch angemerkt wurde, dass es zwar unklar war, was die Felder machen, man jedoch nach kurzem Experimentieren die Funktion schnell herausfinden konnte.

Kapitel 5

Verwandte Arbeiten

In diesem Kapitel werden andere visuelle Benutzeroberflächen für Stimmungsanalyse-Tools beschrieben, welche auf die Anwendung eines einzelnen Tools spezialisiert sind. In der Regel haben diese Tools einen spezielleren Funktionsumfang und bieten andere Möglichkeiten, Stimmungen zu analysieren.

5.1 Stimmungsanalyse-Tool Interfaces

SentiStrength[41] umfasst in der öffentlichen Version ein visuelles Interface für die Anwendung von kompatibelen Datensätzen. Dieses Interface ist jedoch limitiert auf die Anwendung innerhalb von Windows Systemen und ersetzt Kommandozeilen Interaktionen mit der Java Jar, welche die eigentliche Funktionalität beinhaltet. Da es sich um ein zusammenhängendes Bündel handelt, muss keine Software nachinstalliert werden.

Hiervon gibt es eine Abwandlung, die als Java Jar Datei vorliegt und konzipiert wurde, um mit dem SentiStrength-SE[22] Datensatz zu arbeiten, welche denselben Funktionsumfang bietet wie das originale SentiStrength[41] Programm. Diese Anwendung läuft jedoch aufgrund der Tatsache, dass es sich um eine Java Jar handelt, auf allen Betriebssystemen mit einer Java Installation. Dies hat jedoch auch zur Folge, dass sie oftmals über die Kommandozeile geöffnet werden muss, da Betriebssysteme mit Java Installation nicht immer Java assoziieren, um damit Jar Dateien auszuführen.

Bei beiden Programmen ist das einzige anwendbare Tool SentiStrength[41], wobei der angewandte Datensatz ausgetauscht werden kann, um, für Texte in anderen Sprachen, eine speziellere Wertung vorzunehmen, oder, im Falle des SentiStrength-SE[22] Datensatzes, für Texte aus dem Bereich des Software-Engineerings.

Ein grafisches Nutzer Interface für SEntiAnalyzer[19] wurde von Juri Linnemann im Rahmen der Arbeit 'Entwicklung einer grafischen Benutzeroberfläche zur Analyse von Stimmungen in Softwareprojektmeetings'[30]

entwickelt, um speziell für die Funktionalität des SEntiAnalyzers[19] Audio in Echtzeit aufzunehmen und zu evaluieren ein Interface zu erstellen, und auf diesem Wege eine einfache Bedienung zu ermöglichen.

Eine Browsererweiterung für Firefox wurde von Elham Salajegheh Tezerji im Rahmen der Arbeit 'Analyse und Visualisierung der Stimmung innerhalb Open-Source-Projekten durch Entwicklung einer Firefoxerweiterung'[40] entwickelt. Diese verwendet Senti4SD[10], um Daten von der aktuell geö□neten Github Seite direkt zu evaluieren. Dadurch, dass es sich um eine Erweiterung für Firefox handelt, ist diese Lösung sehr zugänglich, jedoch beschränkt auf die Interpretation der Daten mit Senti4SD[10] und somit auf englischsprachige Texte. Ebenfalls bietet die Erweiterung nur eine Integration für Github Inhalte. Somit ist die Nutzung auf englische Github Seiten beschränkt.

5.2 Abgrenzung der Arbeit

Die spezialisierten Interfaces haben Flexibilität in den Einstellungen und Funktionen, welche in dem momentanen Umfang von SATI nicht enthalten ist.

In den Interfaces für SentiStrength[41] können mittels Einstellungen unterschiedliche Datensätze als Grundlage für die Evaluation verwendet werden. Im Gegensatz dazu wird innerhalb von SATI, um einen reibungslosen Ablauf zu garantieren, jeder angepasste Datensatz als separates Tool gewertet. Somit müssen diese installiert werden, da es in der aktuellen Ausführung keine Möglichkeit gibt, Einstellungen zu hinterlegen, um dies zu bewerkstelligen.

Das Interface für den SEntiAnalyzer[19] kann live Audio aufnehmen und dieses in Echtzeit bewerten[30]. Diese Funktionalität wird von SATI in der aktuellen Ausführung nicht angeboten.

Der Vorteil der Firefox Erweiterung liegt darin, dass sie einen kompletten Ablauf bietet, welcher in der aktuellen SATI Version nicht angeboten wird, da das am meisten priorisierte Feature, das Crawlen von Webseiten, noch nicht implementiert ist.

Für diese spezielleren Anwendungsbereiche sind die anderen Tools geeigneter. Jedoch bietet keines von ihnen die Möglichkeit, neue Tools auf verschiedenen Wegen zu generieren, um die eigene Art der Bewertung widerzuspiegeln. Ebenfalls bieten die Tools keinen automatisierten Vergleich zwischen unterschiedlichen Tools. Jedes Tool hat seine Vorzüge. Um diese in SATI einzubinden, ist eine Weiterentwicklung nötig. Es ist auch denkbar, dass die Programme als Unterprogramme ausgeführt werden, um auf diese Weise ihre Funktionalitäten zu implementieren.

Kapitel 6

Diskussion

In diesem Kapitel werden zunächst die zentralen Aussagen aus der Evaluation bezüglich der entwickelten Software wiederholt, um diese anschließend zu interpretieren und mögliche weiterführende Entwicklungsziele festzulegen.

6.1 Zusammenfassung der Ergebnisse

Im Rahmen dieser Arbeit wurde ein visuelles Interface mit dem Namen SATI entwickelt, mit welchem verschiedene Stimmungsanalyse-Tools installiert, evaluiert und deren Ergebnisse miteinander verglichen werden können. Diese Software wurde in einer Interviewstudie evaluiert. In dieser Evaluation wurde die Bedienbarkeit und die Intuitivität von SATI bewertet. Hierzu wurde in der Interviewstudie eine Reihe von Teilnehmenden mit unterschiedlicher Vorerfahrung aufgefordert, die Software zu bedienen und daraufhin dazu befragt. Ein Großteil dieser Nutzenden hat in der Vergangenheit noch nie ein Stimmungsanalyse-Tool angewandt, hauptsächlich aufgrund der Tatsache, dass die Ausführung nicht notwendig war. Während des Interviews war die Rückmeldung über mehrere Interaktionen mit der Software hinweg positiv. Ebenso verhielt es sich mit der generellen Rückmeldung zu der Bedienbarkeit der Software. Nur an wenigen Stellen ist eine unterdurchschnittliche Bewertung feststellbar. Die allgemeine Bewertung ist ebenfalls positiv ausgefallen, wobei die Intuitivität noch ein wenig verbessert werden könnte.

Andere visuelle Interfaces für Stimmungsanalyse-Tools fokussieren sich auf andere Aspekte, welche spezifisch für deren angewandtes Tool sind. Diese Funktionalitäten könnten in SATI aufgenommen werden, oder es könnten die Programme als Unterprogramm ausgeführt werden, um die Funktionen auf diesem Wege zu integrieren.

6.2 Interpretation

Das Problem, das in dieser Arbeit behandelt wurde, war, dass Stimmungsanalyse-Tools nicht anwenderfreundlich designt sind. Die entwickelte Software bietet eine mögliche Alternative zu der ursprünglichen Art der Anwendung. Den Ergebnissen der Interviewstudie zufolge ist die Software gut zu bedienen und angemessen intuitiv. Es wurden keine klar negativen Bewertungen für die Bedienbarkeit festgestellt. Die beiden au□älligen Stellen, Import und die Vorauswahl des Trainings, sind dennoch Schwachstellen, welche künftig bearbeitet werden sollten, da diese häufig nicht optimal bewertet wurden. Ebenfalls betro□en ist die manuelle Evaluation. Dies leitet sich jedoch eher von dem in dieser Aufgabe enthaltenen manuellen Aufwand ab, als dass die Software die Ursache dafür wäre. Nichtsdestotrotz sollte die Tatsache, dass dieser Prozess negativ au□ällt, berücksichtigt werden und alle mit diesem Vorgang in Verbindung stehenden Komponenten optimiert werden, damit für die Nutzenden eine möglichst optimale Anwendung gewährleistet wird.

Zu beachten ist, dass die Vorerfahrung der Nutzenden sehr ausgewogen verteilt war. Die tatsächliche Vorerfahrung der schlussendlich Nutzenden wird wahrscheinlich nicht so ausgewogen verteilt sein wie in dieser Studie und mehr Nutzende mit weniger Programmiererfahrung und geringerer genereller Nutzung der Kommandozeile beinhalten, denn in dieser Umfrage wurden zu großen Teilen nur Studierende und Beschäftigte in informationstechnischen Bereichen befragt. Dies sollte sich jedoch positiv auf das Ergebnis der Frage auswirken, ob man lieber die Software oder die Kommandozeile verwendet, denn Anwender, die nicht häufiger die Kommandozeile verwenden, schrecken vor ihrer Nutzung zurück[26].

Bei der Interviewstudie ist jedoch auch aufgefallen, dass Stimmungsanalyse nicht häufig durchgeführt wird. Welchen Hintergrund das hat, ist nicht klar. Es könnte an der Verteilung der Teilnehmenden liegen, dass diese in der Vergangenheit nicht die Notwendigkeit für eine Durchführung hatten, oder daran, dass die Stimmungsanalyse als Gebiet nicht ö□entlich bekannt genug ist, um häufiger angewandt zu werden.

Als meist priorisierte Verbesserung sticht das Crawlen von Webseiten hervor, welches ein komplett neues Feature darstellt. Dieses Feature komplettiert den Ablauf der Analyse von Inhalten von Webseiten und würde somit die Anwendung vereinfachen, da der Import momentan zu den kritisierten Stellen des Programms gehört. Dies könnte in Verbindung stehen mit der mangelnden Vorerfahrung im Bereich der Stimmungsanalyse. Dadurch könnte die Priorisierung beeinflusst worden sein, da durch dieses Feature mögliche Anwendungsbereiche kreiert werden. Neben den vorgeschlagenen Weiterentwicklungen wurde am häufigsten kleinere Designänderungen gewünscht, um die Intuitivität und Nutzbarkeit noch weiter zu verbessern.

Im Großen und Ganzen ist das Problem, welches dieser Arbeit zugrunde

lag, der Evaluation zufolge gelöst worden. SATI bietet eine anwenderfreundliche Variante, eine Stimmungsanalyse mithilfe von Software automatisch durchzuführen und zu personalisieren. Ohne besonderes Vorwissen können mehrere Stimmungsanalyse-Tools verwendet und deren Ergebnisse miteinander verglichen werden, um auf diese Weise das Tool zu finden, welches die eigene Interpretation am besten reflektiert, oder welches am angemessensten für den Kontext des Inhaltes ist.

6.3 Limitationen

Die Software deckt nur eine Auswahl an Stimmungsanalyse-Tools ab und dies nur auf eine bestimmte Art der Ausführung. Es könnten weitere Tools in der Software enthalten sein und die Interaktionen mit diesen könnte besser kontrollierbar sein, zum Beispiel durch Einstellungsmöglichkeiten. Doch da der zeitliche Rahmen dieser Arbeit ebenfalls limitiert ist, musste der Fokus auf einige Tools gerichtet werden, um beispielhafte Implementationen einzubinden, an denen sich weitere Tools orientieren können. Zudem fehlten zum Zeitpunkt der Evaluation noch ein paar Anzeigen für gängige Metriken wie beispielsweise das F Maß eines trainierten Tools, um eine Metrik zu haben, die widerspiegelt, was von dem Tool zu erwarten ist. Ebenfalls anzumerken ist, dass die Software zwar auf Windows, Linux und macOS gleichermaßen läuft, jedoch hauptsächlich unter Windows getestet wurde, weshalb auf den anderen Systemen weitere unbekannte Komplikationen entstehen können.

Durch die Anzahl von 17 Teilnehmenden ist die Aussagekraft der Interviewstudie limitiert und müsste für weitere statistische Signifikanz in größerem Rahmen erfolgen. Zudem hatten die Teilnehmenden ausgeglichene Vorerfahrungen und wurden geboren nach 1980. Dies lässt schlussfolgern, dass diese mit Computern aufgewachsen sind und somit eher in der Lage sind, ein neues Programm erfolgreich zu verwenden. Viele der möglichen Anwendenden werden eher geringe bis keine Vorerfahrung haben und sind möglicherweise älter. Von daher könnte es sein, dass die Software für sie schwieriger zu bedienen ist als die Ergebnisse der Evaluation es vermuten lassen. Hinzu kommt, dass die Studie in einer kontrollierten Umgebung durchgeführt wurde. In dieser Umgebung waren alle Eigenarten bekannt und keine unerwarteten Ereignisse traten auf. Dies entspricht nicht realen Anwendungsszenarios. In diesen werden unbekannte Einflüsse die Erfahrung verschlechtern. Die Teilnehmenden der Studie hatten zu großen Teilen keine Vorerfahrung mit der Ausführung von Stimmungsanalyse-Tools. Dies könnte sich positiv auf die Frage bezüglich der Präferenz im Vergleich zu der Kommandozeile ausgewirkt haben. Erfahrene Nutzende würden eventuell eher die bewährte und freier zu bedienende Interaktion über die Kommandozeile bevorzugen.

Kapitel 7

Zusammenfassung und Ausblick

Dieses Kapitel dient dazu, einen abschließenden Überblick über die Inhalte und die wichtigsten Ergebnisse der Arbeit zu geben, ebenso wie über mögliche weiterführende Entwicklungen.

7.1 Zusammenfassung

Stimmungsanalyse-Tools werden üblicherweise mit Kommandozeileneingaben und speziell formatierten Dateien bedient. Dies stellt eine Barriere dar, die überwunden werden sollte, indem eine grafische Benutzeroberfläche entwickelt wird.

Die entwickelte grafische Benutzeroberfläche zur Analyse von Stimmungen SATI umfasst eine Plattform, die den Umgang mit Daten und Stimmungsanalyse-Tools ermöglicht. In dieser können Auswertungen von Tools durchgeführt, deren Ergebnisse miteinander verglichen und neue Variationen von Tools trainiert werden.

Die Software wurde in einer Interviewstudie evaluiert. Diese Evaluation ist sehr positiv ausgefallen. Die Aufgaben innerhalb der Studie wurden im Schnitt als 'sehr gut zu bewältigen' bewertet und der Gesamteindruck wurde ebenfalls sehr positiv bewertet. Lediglich die Intuitivität wurde nur positiv bewertet.

Hieraus lässt sich schlussfolgern, dass die Software SATI ihren grundlegenden Zweck, eine Möglichkeit zu bieten, die Barrieren von der Nutzungen von Stimmungsanalyse zu reduzieren, erfüllt. Diesbezüglich gibt es jedoch Vorbehalte, da die Teilnehmendenzahl und deren Verteilung nicht groß genug ist, damit eine vollständige Abdeckung gewährleistet werden kann.

Es gibt mögliche Verbesserungen, die umgesetzt werden könnten, damit der Funktionsumfang größer und vollständiger wird. Der Evaluation zufolge ist die meist priorisierte von diesen das programminterne Crawlen von

Webseiten.

7.2 Ausblick

Um die Software noch vielseitiger einsetzbar und nutzungsfreundlicher zu gestalten, können Funktionen von anderen bereits existierenden Tools eingearbeitet werden. Der Interviewstudie zufolge scheint das am ehesten gewünschte zusätzliche Feature das Crawlen von Webseiten zu sein. Dieses würde den Arbeitsablauf weiter vereinfachen. Das Tool könnte dann als selbstständiges Werkzeug verwendet werden, um den gesamten Prozess zu bewältigen, ohne ein spezielles Vorwissen zu besitzen. Weitere Tools können eingebunden werden, um beispielsweise andere Sprachen außer Deutsch und Englisch abzudecken. Ebenso können Tools eingebunden werden, welche auf spezielle Bereiche, wie zum Beispiel Social Media, optimiert sind. Diese könnte den Nutzungswert des Programms steigern.

Solche Entwicklungen könnten in weiteren Arbeiten oder durch Open Source Beiträge realisiert werden. Entwickelnde von Stimmungsanalyse-Tools könnten für ihr Tool direkt einplanen, dass es innerhalb von SATI ausführbar ist, damit ihre Anwendung gut zugänglich ist. Falls eine Aufgabe durch die Ausführung von SATI nicht bewältigt werden kann, muss der Nutzende leider auf die Anwendung der Kommandozeile zurückgreifen. Solch ein Nutzender könnte SATI erweitern, sodass die Aufgabe durch das Interface zu bewältigen wäre und zukünftige Anwendende nicht auf die Nutzung der Kommandozeile zurückgreifen müssen. Es würde bereits reichen zu dokumentieren, worin das Problem bestand, damit jemand mit der notwendigen Expertise sich mit der Lösung für dieses befassen kann.

Die Software sollte in Zukunft durch das Feedback der Nutzenden um die gewünschten Features erweitert werden, um ein vollständiges Tool zu scha□en. Dieses kann für viele Anwendende die Arbeit angenehmer gestalten und die Stimmungsanalyse für vielerlei Anwendungsbereiche zugänglich machen.

Anhang A

Anhang

Actor	System User
Precondition	The user has downloaded SATI
Postcondition	SATI is installed and includes the Tool Senti4SD
Main path (M)	1. user executes the downloaded SATI file 2. user selects install 3. user selects senti4SD install 4. SATI promps installation of dependecies 5. installation of dependencies executed by user or SATI 6. SATI downloads Senti4SD

Abbildung A.1: Use Case Installation

Actor	System User
Precondition	The user has a comma separated value file (csv) with messages in each row and columns separated by ';'. SATI is running
Postcondition	The data from the csv is stored inside the SATI Database
Main path (M)	1. user selects import 2. user selects the csv file 3. SATI or user inputs a name 4. SATI or user assigns the column including the message to be the message 5. user clicks 'OK'

Abbildung A.2: Use Case Import

Actor	System User
Precondition	in the SATI database is a dataset, Senti4SD is installed and SATI is running
Postcondition	evaluation view for selected dataset is shown
Main path (M)	1. user selects dataset of interest 2. user clicks on 'EVALUATE' in Senti4SD 3. sati displays progression 4. sati displays completition 5. user clicks 'CONTINUE'

Abbildung A.3: Use Case Exekution

Literaturverzeichnis

[1] Chromium an open-source browser project that aims to build a safer, faster, and more stable way for all internet users to experience the web. https://www.chromium.org/Home/. Accessed: 2022-05-05.

[2] Electron plattformübergreifende desktop-anwendungen mit javascript, html und css entwickeln. https://www.electronjs.org/. Accessed: 2022-03-26.

[3] fairseq. https://github.com/pytorch/fairseq. Accessed: 2022-04-03.

[4] Github Language Stats a small place to discover languages in github. https://madnight.github.io/githut/#/pull_requests/2021/4. Accessed: 2022-03-26.

[5] Node.js node.js is a javascript runtime built on chrome's v8 javascript engine. https://nodejs.org/en/. Accessed: 2022-03-26.

[6] React a javascript library for building user interfaces. https://reactjs.org/. Accessed: 2022-03-26.

[7] textblob-de, the german language extension for textblob. https://textblob-de.readthedocs.io/en/latest/index.html. Accessed: 2022-05-05.

[8] W3Counter global web stats. https://www.w3counter.com/globalstats.php?year=2020&month=10. Accessed: 2022-03-26.

[9] A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

[10] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli. Sentiment polarity detection for software development. *Empirical Software Engineering*, 23(3):1352–1382, 2018.

[11] F. Calefato, F. Lanubile, and N. Novielli. Emotxt: A toolkit for emotion recognition from text. In *2017 Seventh International Conference on A ective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 79–80, 2017.

[12] J. Cohen. A coe□cient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[13] K. Dashtipour, S. Poria, A. Hussain, E. Cambria, A. Y. Hawalah, A. Gelbukh, and Q. Zhou. Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cognitive computation*, 8(4):757–771, 2016.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[15] R. Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, apr 2013.

[16] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[17] O. Guhr, A.-K. Schumann, F. Bahrmann, and H. J. Böhme. Training a broad-coverage German sentiment classification model for dialog systems. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1627–1632, Marseille, France, May 2020. European Language Resources Association.

[18] T. Helleputte and P. Gramme. Liblinear: Linear predictive models based on the liblinear c/c++ library. *R package version*, 2:10–18, 2017.

[19] 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, Sep. 2021.

[20] M. Herrmann, M. Obaidi, and J. Klünder. Senti-analyzer: Joint sentiment analysis for text-based and verbal communication in software projects, 2022.

[21] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014.

[22] M. R. Islam and M. F. Zibran. Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. *Journal of Systems and Software*, 145:125–146, 2018.

[23] M. G. Kendall. Rank correlation methods. 1948.

[24] M. G. Kendall and B. B. Smith. The problem of m rankings. *The Annals of Mathematical Statistics*, 10(3):275–287, 1939.

[25] K. Kenyon-Dean, E. Ahmed, S. Fujimoto, J. Georges-Filteau, C. Glasz, B. Kaur, A. Lalande, S. Bhanderi, R. Belfer, N. Kanagasabai, R. Sarrazingendron, R. Verma, and D. Ruths. Sentiment analysis: It's complicated! In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1886–1895, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[26] G. V. Kissel. The e□ect of computer experience on subjective and objective software usability measures. In *Conference companion on Human factors in computing systems*, pages 284–285, 1995.

[27] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[28] R. S. Lazarus. *Emotion and adaptation*. Oxford University Press, 1991.

[29] R. S. LAZARUS. Hope: An emotion and a vital coping resource against despair. *Social Research*, 66(2):653–678, 1999.

[30] J. Linnemann. Entwicklung einer grafischen benutzeroberfläche zur analyse von stimmungen in softwareprojektmeetings. Bachelor's thesis, (Gottfried Wilhelm Leibniz University of Hannover) - Germany, 2022.

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[32] S. Loria et al. textblob documentation. *Release 0.15*, 2:269, 2018.

[33] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[34] M. Obaidi and J. Klünder. Development and application of sentiment analysis tools in software engineering: A systematic literature review. In *Evaluation and Assessment in Software Engineering*, EASE 2021, page 80–89, New York, NY, USA, 2021. Association for Computing Machinery.

[35] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive? empirical study of a□ectiveness vs. issue fixing time. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 303–313, 2015.

[36] H. Pirker. Sentistrength_de: German version of lexica for sentistrength. *http://www. ofai. at/research/interact/resources/SentiStrength_DE/download_form. html, checked on*, 22(08):2012, 2012.

[37] X. Rong. word2vec parameter learning explained, 2014.

[38] P. Shaver, J. Schwartz, D. Kirson, and C. O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061, 1987.

[39] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[40] E. Tezerji. Analyse und visualisierung der stimmung innerhalb open source projekten durch entwicklung einer firefox erweiterung. Master's thesis, (Gottfried Wilhelm Leibniz University of Hannover) - Germany, 2022.

[41] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[42] K. Tymann, M. Lutz, P. Palsbröker, and C. Gips. Gervader-a german adaptation of the vader sentiment analysis tool for social media texts. In *LWDA*, pages 178–189, 2019.

[43] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin. A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60(2):617–663, 2019.

[44] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.