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

Automatisierter Vorschlag für ein Stimmungsanalysetool basierend auf Nutzerangaben zu einem Entwicklerteam

Automated Suggestion for a Sentiment Analysis Tool based on User Input to a Development Team

Bachelorarbeit

im Studiengang Informatik

von

Mohammad Goudarzi Moghadam

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jill Ann-Christin Klünder
Betreuer: M.Sc. Martin Obaidi

Hannover, 09. Januar 2023

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 09.01.2023

_____

Mohammad Goudarzi Moghadam

Kurzfassung

Automatisierter Vorschlag für ein Stimmungsanalysetool basierend auf Nutzerangaben zu einem Entwicklerteam

Die Stimmung in einem Entwicklerteam hat einen großen Einfluss auf die Produktivität des Teams und somit auf den Gesamterfolg eines Softwareprojekts und die Qualität des resultierenden Softwareprodukts. Auch die Stimmung der Nutzenden einer Software ist für viele Entwicklerteams ein wichtiges Thema zu analysieren. Es gibt bereits mehrere Stimmungsanalysetools, die die Stimmungen der Mitglieder eines Software-Entwicklerteams feststellen können. Diese Stimmungsanalysetools werden mittels Datensätzen aus verschiedenen Domänen implementiert, trainiert und getestet. Aus diesem Grund zeigen diese Tools bei Eingabedaten aus verschiedenen Domänen unterschiedliche Performanzen. Um das bestmögliche Tool für Nutzende mit einer bestimmten Domäne zu finden, werden im Rahmen dieser Arbeit mehrere Tools mit Daten aus unterschiedlichen Domänen getestet und deren Performanzen verglichen.

Basierend auf den Gemeinsamkeiten und Unterschieden der vorhandenen Datensätze, Domänen und den beobachteten Performanzen der jeweiligen Tools wird ein Fragenkatalog erstellt, der dazu dient, die Domäne der Kommunikation eines Entwicklerteams zu identifizieren. Dieser Fragenkatalog wird in eine einfach zu bedienende Applikation eingebunden. Dadurch ermittelt diese Applikation einen Vorschlag für das beste Stimmungsanalysetool gemäß der identifizierten Domäne.

Um die Bedienbarkeit der geschaffenen Applikation und die Qualität des Fragenkatalogs zu bewerten, wird die Applikation mit mehreren Anwendungsszenarien von Personen mit verschiedenen Rollen in unterschiedlichen Entwicklerteams bedient und das Ergebnis dazu notiert.

Abstract

Automated Suggestion for a Sentiment Analysis Tool based on User Input to a Development Team

The sentiment in a development team has a great impact on the productivity of the team and thus on the overall success of a software project and the quality of a resulting software product. The sentiment of the users of a software is also an important topic to analyze for many development teams. Several sentiment analysis tools are available to detect the sentiment of the members of a software development team. These sentiment analysis tools are implemented, trained and tested by using datasets from different domains. As a result, these tools show different performances on input data from different domains. To find the best possible tool for users with a specific domain, this thesis aims to test several tools on different domains and comparing them regarding their performance.

Based on the similarities and the differences of the existing datasets, domains and the observed performance of the respective tools, a set of questions was designed to identify the domain of the communication a development team. This set of questions was integrated into an easy-to-use application. As a result, this application determines a suggestion for the best sentiment analysis tool according to the identified domain.

To evaluate the usability of the created application and the quality of the set of questions, the application is operated with several application scenarios by people with different roles in different development teams and the result of this is noted.

Inhaltsverzeichnis

Kapitel 1

1 Einleitung

1.1 Motivation

Um eine qualitativ akzeptierbare Software in einem Softwareentwicklerteam zu entwickeln, ist eine gute Teamarbeit notwendige Voraussetzung, da die Softwareentwicklung eine kollaborative Tätigkeit ist [1]. Studien haben gezeigt, dass die Stimmung in Entwicklerteams mit der Produktivität des Teams zusammenhängt [2]. Ein anderer wichtiger Faktor für den Erfolg eines SE-Teams ist, die Reviews der Nutzenden ihres Produkts zu analysieren und nachzuprüfen, ob die Nutzenden mit ihrem Produkt zufrieden sind [3]. Aus diesem Grund ist es für viele Führungskräfte der Softwareindustrie hilfreich zu wissen, wie die Stimmung der Entwickelnden im Software-Entwicklerteam und die Stimmung der Nutzenden der entwickelte Software ist [1] [3]. Um Stimmungen feststellen zu können, existieren bereits mehrere computergestützte Tools, die die Emotionen oder die Polarität der Emotionen in Eingabedaten bestimmen. Dabei wird zwischen solchen, die lexikonbasiert funktionieren, und den Tools, die einen Machine-Learning-Prozess nutzen, unterschieden [4]. Durch die spezifische Struktur der Konversationen in Software-Engineering-Teams wurden einige Stimmungsanalysetools basierend auf Datensätzen aus Domänen, wie zum Beispiel Jira [5], Stack Overflow [6] und Code-Reviews [7] entwickelt, um ein genaueres Ergebnis zu erhalten [8][9]. 2018 haben Novielli et al. [10] gezeigt, dass die Stimmungsanalysetools SentiStrength-SE [11], Senti4SD [8] und SentiCR [12] bei Datensätzen aus verschiedenen Domänen unterschiedliche Genauigkeiten haben. Das gilt auch für ein Stimmungsanalysetool, das mit Daten aus unterschiedlichen Domänen trainiert wird [10]. Grund dafür ist, dass die Implementierungsdetails der Klassifikatoren in den Tools unterschiedlich sind und auch emotionalen Eigenschaften in den Datensätzen versteckt sind [4]. Die Entwickelnden nutzen bei diversen Aktivitäten textbasierte Kanäle, wie z. B. Foren, Quellcode-Repositories, Issue-Tracking-Systeme (IST) und Code-Reviews für die Kommunikation in einem Software-Entwicklungszyklus [12]. In dieser Arbeit werden Stack Overflow [6], Jira [5], GitHub [13], Code-Reviews [7] und App-Reviews als relevante Domänen betrachtet [14]. Mit Domänen ist im Zuge dieser Arbeit die Quelle der Daten gemeint. Beim Code-Review-Datensatz ist es nicht bekannt, aus welcher Quelle die Daten gesammelt wurden. Aus diesem Grund wird Code-Reviews als eine eigene Domäne betrachtet.

Es ist in jedem Entwicklerteam wünschenswert, dasjenige Stimmungsanalysetool zu verwenden, das die Eingabedaten mit der höchstmöglichen Genauigkeit evaluiert und in entsprechende Emotionen oder Polaritäten der Emotionen übersetzt. Dazu sollte die Domäne gefunden werden, die am meisten der Art und Weise der

Kommunikation im Entwicklerteam oder der Kommunikation der anderen Personen (z. B. bei App-Reviews) entspricht. Somit kann auch das Tool mit der besten Performanz bezüglich der identifizierten Domäne der Kommunikation des Teams zur Stimmungsanalyse benutzt werden.

1.2 Lösungsansatz

Im Folgenden soll ein Fragenkatalog aufgestellt werden, um die Domäne der Kommunikation eines Software-Entwicklerteams zu identifizieren. Zu diesem Zweck müssen zuerst die zu berücksichtigenden Domänen und Datensätze auf Gemeinsamkeiten und Unterschiede geprüft werden. In dieser Arbeit werden SentiStrength [11], SentiStrength-SE [9], Senti4SD [8], SentiSW [15], DEVA [16], SentiCR [12], SEnti-Analyzer [17][18] und RoBERTa [19] als relevante Tools betrachtet. Die Genauigkeit dieser Tools wird mit Eingabedaten aus den im letzten Kapitel genannten Domänen bewertet und das jeweils beste Tool für die unterschiedlichen Domänen ermittelt. Dazu müssen einige Tools zuvor mit den Daten trainiert werden. Entsprechend der beobachteten Merkmale der Datensätze und Domänen müsste ein Fragenkatalog in Form einer Software erstellt werden, welcher die Domäne der Nutzenden identifiziert und das entsprechende Tool mit der höchsten Genauigkeit zur Stimmungsanalyse vorschlägt. Die Software müsste in einer weit verbreiteten Programmiersprache entwickelt werden.

1.3 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen der Stimmungsanalyse sowie die Funktionsweise der im Bereich des Software-Engineerings verwendeten Tools für Software-Engineering (SE) diskutiert. Ebenfalls werden in diesem Kapitel die vorhandenen Datensätze und Domänen beschrieben und anschließend die relevanten Metriken zur Evaluation der Tools und der statistischen Analyse der Datensätze vorgestellt. Schließlich werden in diesem Kapitel auch die zur Umsetzung der Anwendung genutzten Technologien besprochen. In Kapitel 3 werden verwandte Forschungsarbeiten behandelt. In Kapitel 4 werden die statistische und inhaltliche Analyse der Stichprobedaten aus den Datensätzen und deren Ergebnisse beschrieben. In Kapitel 5 beschäftigt sich diese Arbeit mit der Aufstellung des Fragenkatalogs. Das Kapitel 6 wird der Evaluation der Leistung aller betrachteten Stimmungsanalysetools und deren Ergebnissen gewidmet. Das Kapitel 7 geht auf die Umsetzung der zu entwickelnden Applikation ein. Das Kapitel 8 befasst sich mit der Evaluation der Applikation und deren Ergebnissen. In dem darauffolgenden Kapitel 9 werden die Ergebnisse diskutiert und interpretiert. Anschließend werden in Kapitel 10 die Ergebnisse dieser Arbeit zusammengefasst und ein Ausblick gegeben.

Kapitel 2

2 Grundlagen

In diesem Kapitel werden die für diese Arbeit relevanten Grundlagen erklärt. Zuerst werden Grundlagen und verschiedene Verfahren der Stimmungsanalyse dargelegt. Dabei werden auch wichtige Begrifflichkeiten, wie z. B. Emotion und Stimmung näher beschrieben. Da es bei dieser Arbeit hauptsächlich um die Stimmungsanalyse im SE geht, werden die Stimmungsanalysetools für SE genannt und eingehend untersucht und die zur Evaluation verwandten Datensätze vorgestellt. Anschließend werden die relevanten Metriken zur Quantifizierung der Genauigkeit der Tools und zur statistischen Analyse der vorhandenen Datensätze definiert. Abschließend werden die zur Umsetzung der Anwendung benutzte Programmiersprache und die anderen verwendeten Technologien erläutert.

2.1 Stimmungsanalyse

Die Stimmungsanalyse ist eine rechnerische Untersuchung von in Text ausgedrückten Meinungen, Emotionen und Stimmungen [20]. Eine Emotion ist ein psychischer Zustand, der nicht durch bewusste Bestrebung entsteht und normalerweise von psychischen Veränderungen begleitet wird [20]. Beispielsweise sind Angst, Freude, Liebe, Trauer und Überraschung wichtige Emotionen bei der Stimmungsanalyse [21][22]. Eine Meinung kann als ein Gefühl über einen bestimmten Aspekt einer bestimmten Entität zu einer bestimmten Zeit definiert werden. Die Stimmung ist das zugrundeliegende Gefühl oder die zugrundeliegende Emotion, das beziehungsweise die mit einer Meinung verbunden ist [20]. Eine Stimmung ist im Vergleich zu einer Emotion weniger intensiv und kann länger andauern. Die Polarität einer Stimmung kann positiv, negativ oder neutral sein. Neutrale Polarität bedeutet normalerweise die Abwesenheit von Gefühlen [20]. Den Emotionen können Polaritäten zugewiesen werden. Folglich sind Wut, Liebe und Freude Emotionen mit positiver Polarität und Angst und Trauer Emotionen mit negativer Polarität [20]. Einige Wissenschaftler betrachten die Überraschung bei der Stimmungsanalyse gar nicht, weil sie sowohl eine positive als auch eine negative Emotion bedeuten kann [23].
Ziel der Stimmungsanalyse ist, die Emotionen in einem Text festzustellen und somit die Polarität der Sätze zu bestimmen [24]. Dazu wurden Stimmungsanalysetools entwickelt, die das Verfahren der Stimmungsanalyse automatisieren und einem Text Polaritäten oder Emotionen zuordnen [25]. Diese Tools nutzen entweder lexikonbasierte oder Machine-Learning-gestützte Verfahren. Die lexikonbasierten Tools arbeiten mit einem Lexikon oder einer Liste von Wörtern mit der jeweiligen Emotion oder Polarität, die diese übertragen [26]. Außerdem wird den Emotionen

auch ein Gewicht zugewiesen, das die Intensität der Emotion angibt. Zum Beispiel weist die Kombination „sehr gut“ eine intensivere Polarität als das Wort „gut“ auf [8]. Die angesprochene Liste wird von einem Tool als eine Referenz benutzt, um die Stimmung der Wörter und damit die Stimmung der Sätze zu bestimmen [8].
Zum anderen machen einige Tools sich Machine-Learning-Verfahren zu Nutze, um die Stimmungsanalyse zu vereinfachen [26]. Bei diesen Tools sollte ein Datensatz zuerst manuell bewertet werden. Danach wird das Tool in einem Ansatz des überwachten Lernens durch den manuell gelabelten Datensatz trainiert [24]. Daher ist es wichtig, dass der zum Trainieren verwendete Datensatz von mehreren Personen erstellt und bewertet wird und einem Goldstandard entspricht [27]. Die Machine-Learning-Modelle wurden bei solchen Tools unterschiedlich gestaltet, was dazu führt, dass diese Tools bei Datensätzen von verschiedenen Themengebieten oder Umgebungen nicht die gleiche Leistung aufweisen [9]. Der Vorteil von lexikonbasierten Tools ist es, dass sie nicht trainiert werden müssen und deren Leistung bei verschiedenen Domänen seltener abfällt. Jedoch gibt es weniger lexikonbasierte Tools, die für bestimmte Gebiete wie Software-Engineering geeignet sind [1].

2.2 Stimmungsanalyse in Software-Engineering

Die Software-Entwickelnden und auch andere Beteiligte in einem Entwicklungsprojekt arbeiten normalerweise in Teams [1]. Die Mitglieder dieser Teams dürfen an verschiedenen Standorten arbeiten. Demzufolge kommunizieren die Entwickelnden über digitale Kanäle, wie Jira, Stack Overflow und GitHub [1]. Recherchen haben ergeben, dass Entwickelnde mit einer positiven Stimmung entwicklungsbezogene Probleme mit einer höheren Leistung lösen können [2]. Daher ist es für viele Teams oder Führungskräfte wünschenswert, dass die Entwickelnden ihre positive Stimmung in verschiedenen Phasen des Entwicklungszyklus bewahren [18]. Es ist für die SE-Teams auch sehr wichtig zu wissen, wie die entwickelte Anwendung von Nutzenden bewertet wird [3]. Die Nutzenden der Apps schreiben Reviews, um ihre Zufriedenheit oder Unzufriedenheit mit der App zu äußern oder Fehler der App zu melden. Daher könnte einem SE-Team oder SE-Unternehmen vorteilhaft sein, die Reviews der Nutzenden zu analysieren und die Stimmung der Nutzenden zu evaluieren [3]. So können SE-Teams in nächsten Releases die gemeldeten Fehler beheben und die meistgefragte Funktionen entwickeln [3]. Untersuchungen haben ergeben, dass die unmodifizierten Stimmungsanalysetools keine genauen Abschätzungen für die Datensätze aus dem Bereich SE ermitteln. Das liegt daran, dass diese Tools nicht an das SE-spezifische Vokabular angepasst worden waren [8]. Zum Beispiel ist Stanford CoreNLP [28] mit den Datensätzen von Film-Reviews trainiert. Das hat die Notwendigkeit geschaffen, Stimmungsanalysetools zu entwickeln, die am SE ausgerichtet worden sind [29].

2.3 Stimmungsanalyse-Tools

Im Folgenden werden Stimmungsanalysetools für den Bereich SE betrachtet und deren Funktionsweise beschrieben.

2.3.1 SentiStrength und SentiStrength-SE

SentiStrength-SE [9] ist eine für SE modifizierte Version von SentiStrength [11]. SentiStrength ist ein lexikonbasiertes Tool, das mit einer Liste von 298 positiven und 456 negativen englischen Wörtern und Slang-Ausdrücken arbeitet. Diese Wörter haben eine Gewichtung von 2 bis 5 bzw. von -2 bis -5. Wörter mit den Gewichtungen 1 oder -1 werden entfernt, weil sie einen zu geringen Informationsgehalt haben. Die Gewichtungen bezeichnen auch die Polarität jeweiliger Wörter in der Liste. Die Klassifikationen dieser Liste wurden manuell durchgeführt [11].
Der Algorithmus von SentiStrength korrigiert die falsch geschriebenen Wörter und entfernt alle Buchstaben, die mehr als zweimal hintereinander in einem Wort auftreten. Zum Beispiel wird das Wort „Hellllloooo“ zu „Hello“ umgewandelt [11]. Wenn ein Buchstabe normalerweise seltener doppelt auftritt, wird dieser entfernt, wenn er mehr als einmal in einem Wort vorkommt. Zum Beispiel wird das Wort „hii“ als „hi“ identifiziert [11]. Darüber hinaus entfernt der Algorithmus doppelte Buchstaben in einem Wort, falls das resultierende Wort am Ende zu einem normalen Wort wird. Zum Beispiel wird das Wort „nnice“ zu „nice“ transformiert [11]. Auch eine Liste von sogenannten Boosting-Wörtern steht dem Algorithmus zur Verfügung, deren Elemente die Emotion des darauffolgenden Wortes verstärken bzw. abschwächen [11]. SentiStrength wertet Wörter mit sich unnötigerweise wiederholenden Buchstaben stärker, weil dies normalerweise eine Betonung anzeigt. Emojis werden hierbei mit Gewichtungen von 1 bis 2 bzw. -1 bis -2 berücksichtigt. Ausrufezeichen nach einem emotionalen Wort verstärken die von SentiStrength wahrgenommene Emotion des Wortes [11].
Die Polarität eines Satzes ergibt sich aus der Summe der Polaritäten aller Wörter in diesem Satz. Bei Fragen mit Fragezeichen werden alle negativen Wertungen von SentiStrength ignoriert, weil solche Sätze wegen enthaltener negativer Wörter fälschlicherweise negativ eingestuft werden würden [11].
Durch Modifizierung der zugrundeliegenden Liste von Wörtern wird SentiStrength auf einen bestimmten Bereich spezialisiert. So benutzt SentiStrength-SE eine Liste, die aus 5600 manuell evaluierten Kommentaren aus Jira besteht [9].

2.3.2 Senti4SD

Senti4SD [8] verwendet Machine-Learning-Algorithmen und wurde am Anfang mit einem Gold-Standard-Datensatz von 4423 Posts aus Stack Overflow trainiert. Jeder Post besteht aus vier Klassen textueller Elemente, nämlich Fragen, Antworten, Fragekommentaren und Antwortkommentaren [8]. Ein Merkmal eines akzeptablen Training-Datensatzes ist die gleichmäßige Verteilung der Daten über die zu

betrachtenden Klassen. Um das zu erreichen, haben Calefato et al. SentiStrength verwendet, um zu erfahren, ob ein Post emotionale Informationen enthält [8]. Danach wurden zu jeder Polaritätsklasse gleich viele zufällige textuelle Elemente genommen. Die manuelle Evaluation des Datensatzes wurde von zwölf Personen unter Anwendung der Richtlinien von Shaver et al. [30] durchgeführt. Eine positive Polarität sollte angegeben werden, falls die Auswertenden die Emotionen Freude oder Liebe und eine negative Polarität, falls die Auswertenden die Emotionen Angst oder Trauer oder Ärger im Text festgestellt hatten. Im Falle von Überraschung sollten die Auswertenden auf den Kontext achten. Eine neutrale Polarität sollte bei der Abwesenheit aller anderen Emotionen angegeben werden. Ein zusätzliches Label „mixed“ wurde verwendet, um Beiträge mit entgegengesetzten Emotionen (i.e. Freude und Trauer) zu kennzeichnen [8].
Senti4SD nutzt drei verschiedene Arten von Features: Ein generisches Lexikon, Schlüsselwörter und ein verteilungsbezogenes semantisches Modell (DSM) [8]. Zu jedem Post werden die Werte von Features im generischen Lexikon berechnet. Dazu gehört unter anderem die Anzahl der Tokens mit positiver oder negativer Polarität, Bewertung des letzten Emoticons im Post und Boolean-Werte, die zeigen, ob das Post mit einem positiven bzw. negativen Token und einem Ausrufezeichen endet [8].
Schlüsselwortbasierte Features enthalten verschiede Bigramme und Unigramme und die entsprechende Anzahl ihres Vorkommens [8]. Dazu gehören unter anderem die Anzahl der Einträge mit Großbuchstaben (i. e. „BAD“), die Gesamtanzahl der Einträge mit unnötigerweise wiederholenden Buchstaben und Nutzererwähnungen [8].
Beim semantischen Modell werden die Texteinheiten und die Wörter als Vektoren definiert. Also wird ein Dokument auf Stack Overflow in diesem Modell als Vektorsumme aller in diesem Dokument vorkommenden Wörter repräsentiert, indem der Überlagerungsoperator benutzt wird [8]. Die semantischen Features erfassen die Ähnlichkeit zwischen den Vektordarstellungen der Dokumente und den Prototyp-Vektoren, die die Polaritätsklassen in einem DSM darstellen [8]. Um den positiven Prototyp-Vektor zu berechnen, werden die Vektoren für Wörter mit positiver Polaritätsbewertung im gewählten Lexikon summiert. In der gleichen Art und Weise werden der negative Prototyp-Vektor und der neutrale Prototyp-Vektor berechnet [8]. Anhand der Open-Source-Bibliothek Liblinear [31] in der Programmiersprache R [32] wird ein Klassifikationsmodell generiert. Dieses Modell kann dann benutzt werden, um die Stimmung eines Textes zu bestimmen [8]. Senti4SD bietet die Möglichkeit, das Tool auf neue Datensätze zu trainieren [8].

2.3.3 SentiSW

SentiSW [15] ist ein Stimmungsanalysetool, welches sowohl die Stimmung eines Satzes als auch die Entitäten, zu denen die Meinungen geäußert werden, erkennt und das Ergebnis in Form der Tupel <Stimmung, Entität> ausgibt. Der Datensatz, der zum Trainieren des Modells benutzt wurde, besteht aus 3000 Issue-Kommentaren von 10 verschiedenen Projekten auf GitHub. Jeder Kommentar wurde von

mindestens zwei Auswertenden separat hinsichtlich der Polarität evaluiert [15]. GitHub wurde deshalb als Datenquelle hierbei verwendet, weil dieses mehr Informationen als zum Beispiel Jira vermittelt. Dazu gehört die „thumbs-up"-Reaktion, welche auf eine implizite Stimmung hinweist [15]. Vor der eigentlichen Evaluation wird der Text von SentiSW vorverarbeitet. Dies beinhaltet die Löschung der nicht englischen Buchstaben, Entfernung der Code-Ausschnitte und Stop-Words und Expansion der verkürzten Wörter [15].

2.3.4 DEVA

DEVA [16] ist ein Tool, welches sowohl die Emotionspolaritäten als auch die Emotionszustände, wie zum Beispiel Entspannung, Depression, Stress und Aufregung, erkennen kann. Es ist ein lexikonbasiertes Tool und nutzt die Kombination von Software Engineering Arousal (SEA) [33] und Affective Norms of English Words (ANEW) [34] als Lexikon. Zur Evaluation von DEVA wurde ein Datensatz aus 1795 Issue Kommentaren auf Jira erstellt und benutzt [16].

2.3.5 SentiCR

SentiCR [12] wurde speziell für die Analyse der Code-Reviews und anhand von Supervised-Learning-Algorithmen entwickelt. Für das Training des Tools wurde ein Datensatz aus zufällig ausgewählten 2000 Reviews von 20 Projekten und von drei Auswertenden manuell auf Polarität gelabelt [12]. SentiCR benutzt auch einen Vorverarbeitungsprozess vor der eigentlichen Evaluation, wobei unter anderem URLs entfernt und Emoticons besonders behandelt werden [12]. Um den besten Algorithmus für das Supervised-Learning zu finden, wurden anhand des erstellten Feature-Vektors verschiedene Algorithmen evaluiert. Die beste Performanz wurde bei der Verwendung des Gradient-Boosting-Tree-Algorithmus erreicht [12].

2.3.6 SEnti-Analyzer

Um textuelle Daten auf die Polaritäten zu mappen, evaluiert SEnti-Analyzer [17][18] die Ergebnisse von vier anderen Tools und kombiniert danach diese Ergebnisse in einer Polarität. Für englischsprachige Texte sind diese vier Tools Senti4SD [8], SentiStrength-SE [9], SentiStrength [11] und TextBlob [35]. SEnti-Analyzer wurde zu dem Zweck entwickelt, Stimmungen von Teilnehmenden in einem Meeting in Echtzeit zu analysieren, wo die Teilnehmenden nicht immer das SE-spezifische Vokabular benutzen. Das erklärt die Tatsache, dass zwei nicht SE-spezifische Tools zur Evaluation angewandt werden [18]. BertDE [36], GerVADER [37], SentiStrength-DE [38] und TextBlob-DE [39] sind die Tools, die für die deutschsprachigen Texte verwendet werden. Die resultierende Polarität ergibt sich aus der Funktion, die in Abbildung 1 definiert ist. In dieser Funktion sind a, b, c und d jeweils Bewertungen der anderen vier Tools [18].

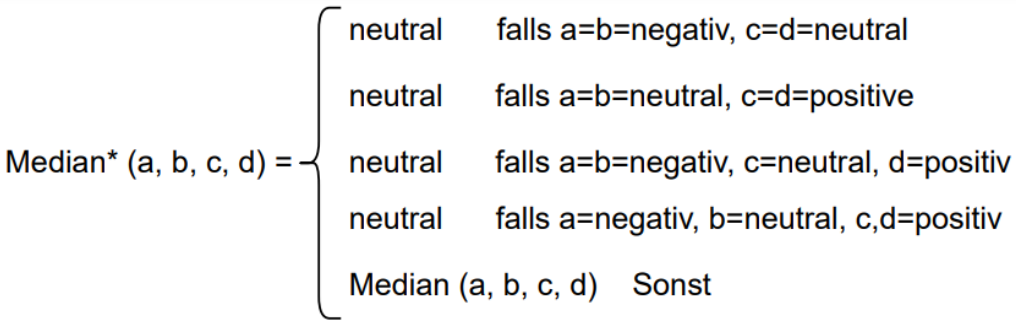

Abbildung 1: Die Funktion für die Berechnung der Polarität in SEnti-Analyzer

2.3.7 RoBERTa

RoBERTa [19] ist ein nicht SE-spezifisches Tool für die Stimmungsanalyse, welches laut Recherchen eine hohe Genauigkeit bei der Analyse der SE-spezifischen Datensätze zeigt [14]. Forscher von Facebook AI und der Universität Washington haben die BERT-Architektur modifiziert, um eine höhere Genauigkeit bei der Stimmungsanalyse zu erhalten [19]. Bidirectional Encoder Representation (BERT) [40] ist ein Machine-Learning-Framework für Natural Language Processing, welches mit Daten aus Wikipedia vortrainiert wurde und feingetunt werden kann. BERT stützt sich auf Transformers, ein Deep-Learning-Modell, in dem jedes Ausgabeelement mit jedem Eingabeelement verbunden ist und die Gewichtung zwischen ihnen dynamisch auf der Grundlage ihrer Verbindung berechnet wird [40]. BERT ist also in der Lage, eine Sequenz von textuellen Eingaben in beiden Richtungen zu lesen [40].
RoBERTa trainiert das Modell länger als BERT und mit größeren Datenabschnitten. Außerdem entfernt RoBERTa die Next-Sentence-Prediction-Klassifikation (NSP). NSP prognostiziert, ob zwei Segmente im Text aufeinander folgen [19]. RoBERTa unterscheidet sich von BERT auch dadurch, dass das Modell auf längere Sequenzen von Daten trainiert wird und das Maskierungsmuster, das auf die Trainingsdaten angewandt wird, dynamisch geändert wird [19].

2.4 Relevante Domänen und verwandte Datensätze

Zu verschiedenen Domänen haben Forschende bereits Gold-Standard-Datensätze erstellt und öffentlich zur Verfügung gestellt. Diese Datensätze wurden genutzt, um Tools zu trainieren oder auch ein Lexikon zu erstellen und zu testen. In dieser Arbeit werden die folgenden Datensätze als Eingabedaten benutzt, um die Leistung der Tools zu bewerten. Die Tabelle 1 zeigt die Übersicht der verwandten Datensätze.

2.4.1 App-Reviews

Die App-Reviews werden von Nutzenden geschrieben, die mit dem von einem Entwicklerteam aufgebauten Softwareprodukt gearbeitet haben. Es ist einem Entwicklerteam nicht nur wichtig, die Stimmung der Entwickelnden positiv zu halten, sondern auch die Stimmung der Nutzenden ihres Software-Produktes.

Anhand von App-Reviews wissen die Entwickelnden, wo und wie sie ihre Software verbessern können [29]. Villarroel et al. [3] kategorisieren die App-Reviews in drei Kategorien:

1) Bug Reporting,
2) Vorschlag für neue Features und
3) Sonstiges.

Die letzte Kategorie beinhaltet Reviews, die keine für das Entwicklerteam nützlichen Informationen liefern [3].

Lin et al. [29] haben zufällig 341 Android-App Reviews aus einem größeren Datensatz gewählt und nach Polarität gelabelt. Der ursprüngliche Datensatz wurde von Villarroel et al. [3] zu Verfügung gestellt und beinhaltet 1763 App-Reviews von über 200 Android-Apps [3]. Drei Auswertende haben die Polarität der Reviews ohne Richtlinien bestimmt. Daraus haben sich 130 negative, 25 neutrale und 186 positive Reviews ergeben [29].

2.4.2 Code-Reviews

Code-Review ist eine Tätigkeit, bei welcher der Code von einem anderen Teammitglied analysiert wird, um zu beurteilen, ob der Code von ausreichender Qualität ist, um in die Hauptcodebase des Projektes integriert zu werden [7]. Viele Open Source Softwares (OSS) benutzen heutzutage auch Bots, um diese Aufgabe zu automatisieren [12]. Ein Code-Review wird in verschiedenen Phasen der Entwicklung durchgeführt [41]. Normalerweise wird bei einem Code-Review auch darauf geachtet, dass der Code lauffähig und richtig getestet worden ist [41].

Ein Datensatz wurde von Ahmed et al. [12] erstellt, um SentiCR zu trainieren und auch die Leistung von sieben anderen Tools zu bewerten. Der Datensatz besteht aus 2000 von Menschen geschriebenen Code-Review-Kommentaren von 20 beliebten Open Source Softwares (OSS)**.** Jeder Kommentar besteht aus mindestens 50 Buchstaben. Aus jedem Projekt wurden 100 Kommentare zufällig ausgewählt, weil eine proportionale Auswahl den Datensatz zugunsten des Vokabulars der größeren Projekte verzerrt hätte [12]. Die Reviews wurden von drei Auswertenden manuell klassifiziert. Nach dem Labeling-Prozess bestand der Datensatz aus 7,7 % positiven, 19,9 % negativen und 72,4 % neutralen Kommentaren [12]. Dabei wurden die Einträge in die zwei Klassen „Negativ“ und „Nicht negativ“ eingestuft. Um den Datensatz ausgewogen zu gestalten, wurden 400 neutrale Kommentare gestrichen [12].

2.4.3 GitHub

GitHub ist eine OSS-Community, welche einen Code-Hosting-Service und ein Issue-Tracking-System anbietet [13]. Entwickelnde können mit Hilfe von GitHub an einem Projekt zusammenarbeiten und ihren Code hochladen bzw. committen. Die

Entwickelnden können zu den jeweiligen Issues und Commits auch Kommentare schreiben, um mit den anderen Entwickelnden zu kommunizieren bzw. ihren Commit zu beschreiben [15]. GitHub ermöglicht es Entwickelnden, mehrere Zweige (Branch) eines Projektes zu haben. Um die Zweige zu mergen, müssen die Entwickelnden einen so genannten „Pull Request“ einreichen, der auch einen Kommentar beinhalten kann [42].

1) Um die Leistung verschiedener Tools zu messen, haben Novielli et al. [23] einen Datensatz von 7122 Pull Requests und Commit-Kommentaren auf GitHub erstellt. Dieser Datensatz wurde nach den Richtlinien von Shaver et al. [30] von drei Auswertenden auf Polarität annotiert. Dabei haben die Auswertenden jeweils einen ganzen Kommentar als eine Einheit annotiert [23]. Der Datensatz besteht aus 2013 positiven, 3022 neutralen und 2087 negativen Einträgen [23]. In dieser Arbeit wird dieser Datensatz abgekürzt mit „GitHub-1“ referenziert.

2) Ein Datensatz wurde von Ding et al. [15] erstellt und besteht aus 2962 Issue-Kommentaren von zehn OSS auf GitHub und wurde benutzt, um SentiSW zu evaluieren. Die Einträge sind von mindestens zwei Auswertenden separat auf Polarität bewertet worden und umfassen 19,9 % positive, 66,6 % neutrale und 13,5 % negative Kommentare [15]. In dieser Arbeit wird dieser Datensatz abgekürzt mit „GitHub-2“ referenziert.

3) Ein weiterer Datensatz wurde von Imtiaz et al. [43] erstellt und besteht aus 589 Pull Request- und Code-Review-Kommentaren auf GitHub. Die Einträge wurden aus verschiedenen öffentlichen Projekten ausgewählt und von zwei Auswertenden auf Polaritäten gemappt. Am Ende wurden 93 Einträge positiv, 419 neutral und 73 negativ eingestuft [43]. In dieser Arbeit wird dieser Datensatz abgekürzt mit „GitHub-3“ referenziert.

2.4.4 Jira

Jira ist das beliebteste ITS in SE und wird weltweit von großen Unternehmen und OSS-Entwickelnden benutzt, um verschiedene Informationen zu einem Projekt zu verwalten [5]. Zu diesen Informationen gehören die Beschreibung, der Status (Neu, In Arbeit, Erledigt usw.), der Issue-Typ und die Sequenz von Kommentaren von jedem Issue [26].

1) Dieser Datensatz wurde von Islam und Zibran [16] verwendet, um die Genauigkeit von DEVA zu messen und beinhaltet 1795 Issue-Kommentare auf Jira. Die Kommentare wurden von drei Auswertenden ohne bestimmte Richtlinien auf die Emotionen Aufregung, Stress, Depression, Entspannung und auf die neutralen Emotionen gemappt. Diese Emotionen können wiederum in Polaritäten transformiert werden [16]. Der Datensatz ist eine

Teilmenge eines größeren Datensatzes, der zwei Millionen Einträge enthält und von Ortu et al. [22] erstellt wurde [16]. In dieser Arbeit wird dieser Datensatz abgekürzt mit „Jira-1“ referenziert.

2) Ortu et al. [22] haben einen Datensatz aus Jira-Kommentaren und -Sätzen erstellt und den in drei Gruppen aufgeteilt. Die erste Gruppe besteht aus 382 Kommentaren, die von 16 Auswertenden mit Emotionen „Joy“, „Love“, „Sadness“, „Anger“, „Fear“ und „Surprise“ gelabelt wurden [22]. Die zweite Gruppe umfasst 2112 Kommentare, die von drei Auswertenden gelabelt wurden. Die Labels waren hierbei nur „Joy“, „Love“ und „Sadness“ [22]. Die dritte Gruppe beinhaltet 3806 Sätze, die ebenfalls von drei Auswertenden gelabelt wurden, die zusätzlich zu den Emotionen in der zweiten Gruppe auch „Anger“ als Label benutzten [22]. In dieser Arbeit werden diese drei Gruppen abgekürzt mit „Jira-2“ referenziert.

2.4.5 Stack Overflow

Stack Overflow ist eine Q & A-Plattform, die von vielen Entwickelnden benutzt wird, um Fragen zum Thema Software-Entwicklung zu stellen oder Fragen anderer Nutzenden zu beantworten [6]. Generell sind auf Stack Overflow Fragen, Fragekommentare, Antworten und Antwortkommentare einsehbar [6].

1) Der erste Datensatz besteht aus 1500 Sätzen, die zur Evaluation von Stanford CoreNLP SO benutzt wurden [29]. Der Datensatz wurde von fünf Auswertenden ad-hoc gelabelt und umfasst 178 positive, 1191 neutrale und 496 negative Sätze [29] In dieser Arbeit wird dieser Datensatz abgekürzt mit „Stack-Overflow-1“ referenziert.

2) Der zweite Datensatz besteht aus ca. 4688 Fragen, Antworten und Kommentaren auf Stack Overflow und wurde von Novielli et al. [21] erstellt. Die Daten wurden von zwölf Auswertenden nach den Richtlinien von Shaver et al. [30] gelabelt. Die Labels sind „Love“, „Joy“, „Surprise“, „Anger“, „Sadness“ und „Fear“. Ein Eintrag ohne eine bestimmte Emotion wird als neutral bewertet [21]. Nach der Transformation der Emotionen in Polaritäten ergeben sich 1527 positive, 1694 neutrale und 1202 negative Einträge. Liebe und Freude wurden positiv und Ärger, Trauer und Angst negativ bewertet. Alle Einträge mit dem Label „Überraschung“ wurden entfernt, weil diese sowohl positiv als auch negativ gedeutet werden können [44]. 35 % der Einträge präsentieren positive, 27 % negative und 38 % neutrale Emotionen [44]. In dieser Arbeit wird dieser Datensatz abgekürzt mit „Stack-Overflow-2“ referenziert.

Datensatz	Anzahl der Einträge	Anzahl positiver Einträge	Anzahl negativer Einträge	Anzahl neutraler Einträge	Richtlinie beim Labeling
App-Reviews	341	186	130	25	ad-hoc
Code-Reviews	1600	-	398	1202	ad-hoc
GitHub-1	7122	2013	2087	3022	Shaver et al. [30]
GitHub-2	2962	598	401	1968	ad-hoc
GitHub-3	585	93	73	419	Mohammad [45]
Jira-1	1795	638	541	616	ad-hoc
Jira-2	6300	1357	778	4165	Parrott [46]
Stack-Overflow-1	1500	178	131	1191	ad-hoc
Stack-Overflow-2	4688	1595	1145	1947	Shaver et al. [30]

Tabelle 1: Übersicht der verwandten Datensätze

2.5 Metriken

Im Folgenden werden die relevanten Metriken zu der Evaluation der Tools und der statistischen Analyse der Datensätze vorgestellt.

2.5.1 Metriken zur Evaluation der Stimmungsanalysetools

Um die Performanz verschiedener Stimmungsanalysetools zu bewerten, müssen zuerst die relevanten Metriken festgelegt werden. Für diese Arbeit sind Micro- und Macro-averaged-F1-Score die relevanten Metriken. Als die entscheidende Metrik für den Vergleich der Leistungen der Tools wird der Durchschnittswert von Micro- und Macro-averaged F1-Score benutzt. Der Grund dafür ist, dass bei manchen Anwendungsfällen eines Stimmungsanalysetools jede Klasse und bei manchen anderen jede Probe gleich gewichtet werden muss. In dieser Arbeit wird diese Metrik Overall-Score genannt.
Um diese Metriken zu definieren, sollten zuerst andere grundlegende Begriffe zur Evaluation eines Klassifikationsmodells bezüglich einer hypothetischen Klasse A definiert werden:

True Positive (TP): Die Eingaben, die zur Klasse A gehören und auch in diese Klasse eingestuft werden [47].

False Positive (FP): Die Eingaben, die nicht zur Klasse A gehören aber fälschlicherweise in diese Klasse eingestuft werden [47].

True Negative (TN): Die Eingaben, die zu anderen Klassen als A gehören und auch in diese Klassen eingestuft werden [47].

False Negative (FN): Die Eingaben, die zur Klasse A gehören aber in andere Klassen eingestuft werden [47].

Precision: Precision ist definiert als der Anteil der Eingaben, die korrekt vom Modell in die Klasse A eingestuft wurden, an allen mit der Klasse A gelabelten Daten [47].

$$Precision = \frac{TP}{TP+FP}$$

Recall: Recall ist definiert als der Anteil der Eingaben, die korrekt vom Modell in die Klasse A eingestuft wurden, von allen Daten, die tatsächlich zur Klasse A gehören. [47].

$$Recall = \frac{TP}{TP + FN}$$

Damit können weitere Kennzahlen für die Multiclass-Klassifizierung wie folgt definiert werden:

Macro-averaged Precision und Macro-averaged Recall: Diese beiden Kennzahlen werden als arithmetisches Mittel von Precision bzw. Recall der einzelnen Klassen berechnet. [47].

$$Macro - avg.Precision = \frac{\sum_{k=1}^{K} Precision_k}{K} \; ;$$
$$K = Anzahl\ der\ Klassen.$$

$$Macro - avg.Recall = \frac{\sum_{k=1}^{K} Recall_k}{K} \; ;$$
$$K = Anzahl\ der\ Klassen.$$

Macro-averaged F1-Score: Diese Kennzahl ist das harmonische Mittel von Macro-averaged-Precision und -Recall. [47].

$$Macro - avg.F1 - Score = 2 * \left(\frac{Macro - averaged\ Precision * Macro - averaged\ Recall}{Macro - averaged\ Precision + Macro - averaged\ Recall}\right)$$

Micro-averaged Precision und Micro-averaged Recall: Bezüglich der betrachteten Klassen werden diese beiden Kennzahlen wie folgt berechnet:

$$Micro - avg.Precision = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + FP_k}$$

$$Micro - avg.Recall = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + FN_k}$$ [47].

Micro-averaged F1-Score: Diese Kennzahl ist das harmonische Mittel von Micro-averaged-Precision und -Recall [47].

$$Micro - avg.F1 - Score = 2 * \left(\frac{Micro - averaged\ Precision * Micro - averaged\ Recall}{Micro - averaged\ Precision + Micro - averaged\ Recall}\right)$$

2.5.2 Metriken zur statistischen Analyse der Stichprobedaten

Auch statistische Informationen können bei der Identifizierung der Domäne hilfreich sein, indem sie die Form der Sätze beschreiben können [48]. Folgende Metriken wurden von Julian Horstmann [48] und Alexander Specht [49] in ihren Masterarbeiten verwendet:

1) Die durchschnittliche Länge der Einträge in Buchstaben,
2) die durchschnittliche Länge von Wörtern in Buchstaben,
3) die durchschnittliche Anzahl von Wörtern,
4) die durchschnittliche Anzahl von großgeschriebenen Wörtern,
5) die durchschnittliche Anzahl von Rechtschreibfehlern und
6) die durchschnittliche Anzahl von Emoticons.

Die durchschnittliche Länge der Einträge und Wörtern und die durchschnittliche Anzahl von Wörtern liefern Informationen über die Struktur von Sätzen. Durchgehend groß geschriebene Wörter werden normalerweise benutzt, um eine negative Stimmung deutlich zu machen [49]. Formale Texte beinhalten weniger Rechtschreibfehler [9]. Daher zeigt diese Metrik, wie formal der Text ist. Außerdem sind Emoticons Zeichen für verschiedene Stimmungen [50].

Zusätzlich zu diesen Statistiken werden auch folgende Statistiken berechnet:

7) Die durchschnittliche Anzahl von Fragezeichen und
8) die durchschnittliche Anzahl von Ausrufezeichen.

Die Rechtschreibfehler werden mithilfe der Python-Bibliothek pyspellchecker [51] geprüft. Die Statistiken 1 bis 8 werden mithilfe der Python-Bibliotheken pandas [52] und NumPy [53] berechnet. Zur Berechnung der Anzahl von Emoticons wurde die von Wang und Castanon [54] erstellte Liste von Emoticons als Referenz verwendet. Diese Liste besteht aus 34 Emoticons [54].

2.6 Java, JavaFX und FXML

Für die Einbindung des Fragenkatalogs in eine Anwendung werden die Programmiersprache Java und die Library JavaFX verwendet. Java ist eine verbreitete Programmiersprache, die von Betriebssystem unabhängig ist und für die Entwicklung von Desktop-, Mobile- und Web-Applikationen benutzt werden kann. Der in Java geschriebene Quellcode lässt sich in eine .jar-Datei konvertieren, die dann auf verschieden Betriebssystemen ausgeführt werden kann. Voraussetzung dafür ist, dass Java auf dem System installiert worden ist [55].

JavaFX ist eine Open Source Library von Java zur Entwicklung von Desktop-, Mobil- und Embedded-Applikationen. JavaFX bietet vielfältige Komponenten zur Gestaltung eines modernen Graphical User Interface (GUI). Programme, die mit JavaFX geschrieben worden sind, sind auf Windows, Linux und MacOS lauffähig [56]. Durch diese Auswahl von verwendeten Technologien für die Entwicklung der Anwendung ist die entwickelte Applikation ohne große Schwierigkeiten erweiterbar.

FXML ist eine skriptfähige, XML-basierte Markup-Sprache, die für die Gestaltung einer Benutzerschnittstelle in JavaFX benutzt werden kann. [57][58].

Kapitel 3

3 Verwandte Arbeiten

In diesem Kapitel werden wissenschaftliche Arbeiten besprochen, die sich mit der Analyse der Genauigkeit für Stimmungsanalysetools auf Eingabedaten aus verschiedenen Domänen beschäftigt oder eine Literaturübersicht zum Thema Stimmungsanalyse in SE zur Verfügung gestellt haben.

Novielli et al. [10] haben 2018 anhand von vier Datensätzen eine Benchmark-Studie durchgeführt und dabei die Stimmungsanalysetools SentiStrength [11], SentiStrength-SE [9], Senti4SD [8] und SentiCR [12] miteinander verglichen. Die Datensätze bestanden aus 4423 Posts aus Stack Overflow, 5869 Sätzen aus Jira, 1600 Code-Review-Kommentaren und weiteren 1500 Sätzen aus Stack Overflow zum Thema Java-Bibliotheken [10]. Die Ergebnisse dieser Studie zeigten, dass Senti4SD für Eingabedaten aus Stack Overflow und SentiCR für Eingabedaten aus Jira und Code-Reviews die höchste Genauigkeit erreicht haben [10].

2020 befassten sich Novielli et al. [23] damit, wie genau verschiedene Stimmungsanalysetools bei einer Cross-Platform-Einstellung funktionieren. In diesem Fall ist also der Test-Datensatz aus einer anderen Domäne als der Trainings-Datensatz [23]. Dazu wurden drei Gold-Standard Datensätze aus Jira mit ca. 6000 Kommentaren und GitHub mit ca. 4000 und 7000 Kommentaren verwendet [23]. In dieser Arbeit wurden Senti4SD [8], SentiCR [12], SentiStrength-SE [9] und DEVA [16] mit Eingabedaten evaluiert. Ergebnisse zeigten, dass in der Within-Platform-Einstellung Senti4SD für GitHub und Stack Overflow die beste Leistung aufgewiesen haben, und SentiCR für Jira [23]. In der Cross-Platform-Einstellung performten die lexikonbasierten Tools besser als die Machine-Learning-basierten Tools. Ausnahme hierbei war der Fall, in dem das Modell mit dem Stack-Overflow-Datensatz trainiert und mit dem GitHub-Datensatz getestet wurde. In diesem Fall hatte Senti4SD die höchste Performanz [23].

Wu et al. [59] haben das vortrainierte Modell BERT [40] für Textklassifikation in SE nachtrainiert und das resultierende Framework BERT-FT genannt. Um die Leistung von BERT-FT zu bewerten, wurde es mit sechs anderen Stimmungsanalysetools, nämlich NLTK [60], SentiStrength [11], SentiStrength-SE [9], Stanford CoreNLP [28], SentiCR [12] und Senti4SD [8] verglichen. Zu diesem Zweck wurden der Jira-Datensatz von Ortu et al. [61] mit 4000 Sätzen, der API-Reviews-Datensatz von Uddin et al. [62] mit 4522 Sätzen über APIs aus Stack Overflow, der Stack-Overflow-Datensatz (SO-Lib) von Lin et al. [29] mit 1500 Sätzen, der Code-Review-Datensatz von Ahmed et al. [12] mit 1600 Kommentaren, der GitHub-Datensatz von Guzman et al. [63] mit 7122 Sätzen aus Pull-Requests und Commit-Kommentaren

und der Stack-Overflow-Datensatz von Calefato et al. [8] mit 4423 Posts benutzt. Die höchste Genauigkeit hatte das BERT-FT-Modell bei allen Datensätzen. Dennoch hat beim API-Reviews-Datensatz Stanford CoreNLP [28] das beste Ergebnis auf als neutral gelabelten Daten erreicht [59].
Außerdem wurde das Experiment mit BERT-FT, SentiCR [12] und Senti4SD [8] in einer Cross-Platform-Einstellung wiederholt. Dazu wurden einmal der GitHub-Datensatz für das Training und SO-Lib für das Testing benutzt und einmal andersherum. Auch in diesem Fall hat BERT-FT besser als die anderen Tools abgeschnitten [59].

Um die Performanz von Senti4SD [8] zu bewerten, haben Calefato et al. einen Gold-Standard-Datensatz aus 4423 Posts aus Stack Overflow erstellt und die Polarität der Einträge manuell festgelegt [8]. Dieser Datensatz wurde dann für Training und Testing von Senti4SD [8] verwendet. In einem Experiment wurde die Polarität derselben Daten mittels SentiStrength [11] und SentiStrength-SE [9] abgeschätzt. Insgesamt zeigte Senti4SD [8] die höchste Genauigkeit im Vergleich mit den anderen beiden Tools [8].

Zhang et al. [14] versuchten herauszufinden, wie präzise und wie effizient die Transformermodelle im Vergleich zu anderen Stimmungsanalysetools sind. Als vortrainierte Transformermodelle wurden BERT [40], RoBERTa [19], XLNet [64] und ALBERT [65] und als Tools mit anderen Verfahren zur Stimmungsanalyse Stanford CoreNLP [28], SentiStrength [11], SentiStrength-SE [9], SentiCR [12] und Senti4SD [8] betrachtet. Vor der Durchführung des Experiments wurden die Transformermodelle mit Daten aus SE-Domänen feingetunt. Die genutzten Datensätze zur Evaluation sind bis auf einen dieselben wie die bei der Studie von Wu et al. [59]. Zhang et al. haben anstatt der API-Reviews [62] die App-Reviews von Lin et al. [29] mit 341 Reviews genutzt [14].
Im Allgemeinen haben die Transformermodelle bei allen Datensätzen besser abgeschnitten als die anderen Tools. Unter den nicht transformerbasierten Tools hatte SentiCR bei allen Datensätzen die beste Performanz [14].

Im Rahmen einer anderen Studie haben Islam und Zebran [66] drei Stimmungsanalysetools, nämlich SentiStrength-SE [9], EmoTxT [67] und Senti4SD [8] bezüglich ihrer Leistung miteinander verglichen. Hierzu nutzten sie die Gruppe 2 und 3 des Jira-Datensatzes von Ortu et al. [22], den Stack-Overflow-Datensatz von Calefato et al. [8] und den Code-Review-Datensatz von Ahmed et al. [12] [66]. Die Ergebnisse dieser Studie zeigten, dass SentiStrength-SE [9] beim Jira-Datensatz und beim Code-Review-Datensatz besser als die anderen beiden Tools abschneidet. Beim Stack-Overflow-Datensatz deutet Senti4SD [8] die höchste Genauigkeit an [66].

Anhand des eigenen Datensatzes aus GitHub mit 589 Kommentaren haben Imtiaz et al. [43] die Genauigkeit von SentiStrength [11], NLTK [60], Alchemy, Stanford

NLP [68], Senti4SD [8] und SentiCR [12] miteinander verglichen. Als Ergebnis dieser Studie wurde die beste Performanz bei Senti4SD [8] gesehen [43].

Obaidi und Klünder [1] haben in ihrer Studie 80 verschiedene Arbeiten analysiert und die wichtigsten Anwendungsszenarien für die Stimmungsanalyse in SE, den Zweck der Stimmungsanalyse in den ausgewählten Arbeiten, die benutzten Daten für die Stimmungsanalyse, die Ansätze zur Entwicklung der Stimmungsanalysetools und die Schwierigkeiten bei der Entwicklung dieser Tools ermittelt [1]. Laut der Studie basieren die meisten der Papers zur Stimmungsanalyse in SE auf OSS und berücksichtigen weniger als ein Drittel entweder industrielle Projekte oder den akademischen Bereich [1]. Es hat sich herausgestellt, dass es bezüglich des Hauptzwecks der Papers drei Arten von Papers gibt: Entwicklung der Stimmungsanalysetools, Vergleich der Tools und Anwendung der Tools. Die meisten der Papers befassten sich mit dem Anwendungstyp und 35 % der Papers betrachteten die Entwicklung oder den Vergleich der Tools [1]. Insgesamt wurden bei diesen 80 Papers 48 unterschiedliche Datensätze verwendet. Jira, GitHub und Stack Overflow wurden am häufigsten zum Trainieren und Testen der Tools verwendet [1]. In den Papers wurden 28 Stimmungsanalysetools verwendet, wobei SentiStrength [11] hervorsticht [1]. Vier Papers haben die nicht ideale Performanz der Tools in einer Cross-Platform-Einstellung als eine Schwierigkeit erwähnt [1].

In einer ähnlichen Arbeit haben Lin et al. [69] 185 Papers zum Thema Opinion-Mining in SE systematisch reviewt. Laut dieser Studie wird Opinion-Mining am häufigsten in den Qualitätssicherungsprozessen verwendet [69]. In diesen Papers wurden 19 Stimmungsanalysetools adoptiert bzw. entwickelt, die die Polarität der Daten feststellen. Von diesen 19 Tools wurden sechs Tools anhand von Daten aus SE gestaltet [69].

Kapitel 4

4 Inhaltliche und statistische Analyse der Stichprobedaten

In diesem Kapitel wird der Prozess der inhaltlichen und statistischen Analyse der Stichprobedaten aus den Domänen Stack Overflow, Jira, GitHub, Code-Reviews und App-Reviews betrachtet. In der vorliegenden Arbeit ist mit der Domäne die Quelle der Kommunikationen gemeint. Zu jeder Domäne gibt es einen Datensatz oder mehrere Datensätze, die analysiert wurden. Die Merkmale der Domänen und der Datensätze sind eine Grundlage für die Ableitung der Fragen im Fragenkatalog, die zur Identifizierung der Domäne der Kommunikation von Nutzenden benutzt wird.

4.1 Datenbereinigung

Im ersten Schritt wurden die Datensätze, die aus derselben Domäne stammen, zusammengefügt. Dazu wurden in Datensätzen, die mit Emotionen gelabelt waren, die Emotionen in Polaritäten umgestellt. Beim Datensatz von Islam und Zibran [16] wurden basierend auf deren vorgestelltem Mapping die Emotionen „Excited“ und „Relax“ als die positive und Emotionen „Depression“ und „Stress“ als negative Stimmungen gekennzeichnet. Alle Einträge mit dem ursprünglichen Label „Neutral“ wurden somit als neutrale Stimmung markiert [16].
Im Datensatz von Imtiaz et al. [43] wurden alle Einträge mit dem Label „Sarcasm“ entfernt, weil diese Einträge auch von der Autorenschaft beim Labeling-Prozess entfernt wurden [43].
Im Datensatz von Ortu et al. [22] wurden zuerst in der dritten Gruppe die nach Emotionen gelabelten Daten und partitionierten Dateien zusammengefügt. Diese Gruppe der Daten beinhaltet 4000 Sätze, die von drei Auswertenden gelabelt wurden und „Anger“, „Joy“, „Love“ und „Sadness“ als Labels benutzten [22]. Danach wurden nach dem Prinzip von Novielli et al. [23] die Einträge mit den Labels „Anger“ und „Sadness“ in negative und die Einträge mit den Labels „Joy“ und „Love“ in positive Polarität umgewandelt. Nach dem gleichen Prinzip wurden auch die Einträge in der ersten und zweiten Gruppe angepasst. In der zweiten Gruppe wurden auch alle anderen Kommentare in allen Spalten außer in der Spalte „Comment N“ entfernt. Die entfernten Kommentare dienten beim Labeling-Prozess dazu, den Kontext der Kommunikation zu bestimmen. Die Ergebnisse der Arbeit von Murgia et al. [70] zeigten, dass der Kontext keine große Rolle bei der Evaluation der Einträge spielt. In allen drei Gruppen wurden die Einträge, denen keine Emotion zugewiesen worden war, als neutral markiert.
Beim Stack-Overflow-Datensatz von Novielli et al. [21] wurden die Labels „Love“ und „Joy“ als positive und die Labels „Sadness“, „Anger“ und „Fear“ als negative

Stimmungen markiert. Auch hier wurden Einträge ohne zugewiesene Emotionen als neutral betrachtet.

4.2 Extrahierung der Eigenschaften der Stichprobedaten

Nach dem Zusammenfügen der Daten in den jeweiligen Domänen wurden aus jeder Domäne zufällig 90 Einträge zur Analyse genommen. Diese Einträge sind gleichmäßig auf die Polaritäten verteilt, d. h. die Probe besteht aus jeweils 30 Einträgen zu den jeweiligen Polaritäten und somit aus insgesamt 90 Einträgen.
Um inhaltliche Eigenschaften der Kommunikationen in den Stichprobedaten zu extrahieren, wurden zuerst deutliche Eigenschaften der Einträge identifiziert. Als Nächstes wurde bei jeder Eigenschaft in jeder Domäne überprüft, ob die Eigenschaft für den Fragenkatalog geeignet ist. Danach wurden basierend auf den geeigneten Eigenschaften Fragen entworfen und die relevanten Eigenschaften bei allen Domänen analysiert. Das Ziel bei der Auswahl der Eigenschaften war es, den Fragenkatalog möglichst klein zu halten. Nach der Analyse der Probedaten in jeder Domäne, waren folgende Eigenschaften in den Daten erkennbar:

4.2.1 App-Reviews

1) In 64 % der Einträge werden die Emotionen direkt und ohne Umweg angegeben. Ein Beispiel dazu ist: „*invaluable! I use this app daily and can't imagine not having it*" (ID 177 in Stichprobedaten).
2) In 43 % der positiven Einträge werden Emotionen hervorgehoben ausgedrückt. Ein Beispiel dazu ist der folgende Satz: „*…cool wow…i love [th]is…*" (ID 225 in Stichprobedaten).
3) In 80 % der negativen Reviews begründen die Nutzenden ihr Review und teilen somit für Entwickelnde nützliche Informationen mit. Ein Beispiel dazu ist der Satz: *„…Suddenly after downloading an update pack I cannot login and said that loading failed. [C]heck your connection network. But my connection is working finely*" (ID 3 in Stichprobedaten). Da in anderen Domänen nicht nur Reviews geschrieben werden, wurde in anderen Domänen auch überprüft, ob in den Einträgen explizit einen Grund für die negative Polarität geäußert wurde, die auch nützliche Informationen für andere Personen beinhaltet.
4) In 24 % der Einträge wird explizit eine Anfrage zum Bug-Fix ausgedrückt.
5) Die Reviews haben keinen technischen Kontext.
6) 31 % der im Allgemeinen negativ gemeinten Reviews beinhalten auch Komplimente an die Apps.
7) In 97 % der Reviews gibt es Komplimente an Produkte oder Personen.

4.2.2 Code-Reviews

1) 97 % der Reviews haben ein technisches Thema.
2) Nur in 6 % der Reviews werden Emotionen direkt ausgedrückt.

3) In 20 % der Reviews wird nach dem Grund einer Entscheidung gefragt, z. B. „*why do we json_encode() $scopeData here?*“ (ID 1500 in Stichprobedaten).

4.2.3 GitHub

1) Es gibt Benutzernamen in 14 % der Einträge. Benutzernamen beginnen normalerweise mit einem „@“-Zeichen.
2) 21 % der Einträge beinhalten URLs. Diese Eigenschaft wird für den Fragenkatalog nicht berücksichtigt, weil bei einigen anderen Datensätzen URLs im Rahmen der Vorverarbeitung von Datensatzerstellern entfernt wurden.
3) Bei 23 % der Einträge wird über den Code gesprochen (ID 1102 in Stichprobedaten).
4) Bei 16 % der Einträge geht es darum, den anderen Entwickelnden bei ihren Problemen zu helfen oder um Hilfe zu bitten.

4.2.4 Stack Overflow

1) Es geht bei 93 % dieser Daten um Fragen oder Antworten auf Fragen.
2) Es geht bei 78 % dieser Daten um technische Themen.

4.2.5 Jira

1) Bei 12 % der positiven Kommentare geht es darum, andere Personen über einen Fortschritt bzw. Erfolg zu informieren.
2) 63 % der positiven Reviews zeigen Dankbarkeit gegenüber anderen Personen.
3) In 32 % der Einträge gibt es Namenserwähnungen. Sowohl Vornamen als auch Vollständige Namen wurden im Rahmen dieser Arbeit als Namenserwähnungen definiert.

4.2.6 Ergebnisse der statistischen Analyse

Die Stichprobedaten wurden statistisch analysiert, um weitere strukturelle Eigenschaften der Texte in jeweiliger Domäne herauszufinden. Das Ergebnis dieser Analyse ist in der Tabelle 2 zusammengefasst.

- Laut der Ergebnisse sind die Sätze in App-Reviews am längsten.
- Die Sätze in Stack Overflow sind fast so lang wie die in App-Reviews.
- Die Wörter in Code-Reviews sind mit kleinem Abstand durchschnittlich die längsten.
- Jeder Eintrag in App-Reviews beinhaltet durchschnittlich die höchste Anzahl von Wörtern.
- Pro Eintrag hatte die Stichprobe von Stack Overflow die höchste Anzahl von großgeschriebenen Wörtern.
- Pro Eintrag wurden in GitHub ungefähr drei Wörter falsch geschrieben.

- Die Nutzenden von GitHub benutzen die Emoticons mehr als die in anderen Domänen.
- In Stack Overflow werden mehr Fragen gestellt als in anderen Domänen.
- Auch die Ausrufezeichen waren in der Stichprobe von Stack-Overflow-Daten zahlreicher als in anderen Domänen.

Statistiken (durchschnittlich pro Eintrag)	App-Reviews	Code-Reviews	Jira	Stack Overflow	GitHub
Länge von jedem Eintrag in Buchstaben	176, 36	165,74	104,21	176,32	165,55
Länge von jedem Wort in Buchstaben	4,14	4,71	4,61	4,54	4,53
# Wörter	33,15	28,86	17,43	32,08	27,78
# Großgeschriebene Wörter	0	0,37	0,22	0,66	0,44
# Rechtschreibfehler	0,76	1,98	1,06	1,7	2,97
# Emoticons	0,02	0,07	0,14	0,07	0,4
# Fragezeichen	0,29	0,32	0,14	0,41	0,24
# Ausrufezeichen	0,32	0,03	0,21	0,51	0,48

Tabelle 2: Ergebnisse der statistischen Analyse der Stichproben

Es ist bei der Analyse aufgefallen, dass pyspellchecker auch die Namen der Technologien oder Tools, Abkürzungen und Benutzernamen als falschgeschriebene Wörter kennzeichnet. pyspellchecker ist eine Python-Bibliothek, die zur Prüfung von Rechtschreibfehlern benutzt werden kann [51]. Das ist für diese Arbeit nicht problematisch, weil auch die Eingabedaten in der zu entwickelnden Anwendung dieselbe Bibliothek bei der Analyse verwenden.

Kapitel 5

5 Fragen des Fragenkatalogs

Basierend auf den inhaltlichen Eigenschaften der Stichprobedaten wurden die Fragen des Fragenkatalogs gestaltet. Die Tabelle 3 beschreibt, welche Eigenschaften bei den Fragen betrachtet wurden.

Frage	Eigenschaften
1	App-Reviews (1), Code-Reviews (2)
2	App-Reviews (2)
3	App-Reviews (5), Code-Reviews (1), GitHub (3), Stack Overflow (2)
4	App-Reviews (6)
5	Jira (1)
6	Jira (2)
7	Code-Reviews (3)
8	GitHub (4), Stack Overflow (1)
9	App-Reviews (6), App-Reviews (7)
10	App-Reviews (4)
11	App-Reviews (3)
12	GitHub (1)
13	Jira (3)

Tabelle 3: Betrachtete Eigenschaften bei den Fragen des Fragenkatalogs

Die Fragen sind auf Englisch und werden in der zu entwickelnden Anwendung auf Englisch geschrieben, weil die Datensätze nur aus Sätzen in der englischen Sprache bestehen. Die Fragen beziehen sich auf die zu evaluierenden Kommunikationsdaten. Die Fragen des Fragenkatalogs sind:

1. In your communications, the participants express their emotions in a direct way. Examples of direct expressions of Emotions: "*cool wow...I love this...good job*" or "*I absolutely hate how tabs work in Xcode*".

2. Participants in a communication express their positive opinions in an emphasized way.
   Examples of emphasized expression of opinions: " *woooow it is a greaaaaaaate code, wonderful*".
   Example of not-emphasized expression of opinions: "G*ood app for Indian users*".

3. The communications have predominantly technical topics. Example for a technical topic: Communications about the code, programming languages and libraries, technologies, and IDEs.
   Example for a non-technical topic: Communications that relate to the software product from the user's point of view do not have a technical context.

4. If someone has expressed his generally negative feedbacks or comments in communications about a topic, he has mentioned positive points about that topic as well.

5. In Communications progress and achievements, e.g. bug fixes, new commits, or patches are shared.

6. In the positively intended comments in communications, thankfulness is expressed to others (Not compliments).

7. In the communications, questions are asked about specific code-level decisions.

8. The communications are mainly about asking for help with problems or helping other people with their problems.

9. The positively intended comments include compliments to other people's work (Not thankfulness).

10. In communications, there are explicit requests to fix the bugs.

11. In your communications, negatively intended feedbacks are justified in the feedbacks and deliver useful information for others.

12. In your communications, participants write the username of the person, who they want to mention or talk to.

13. In your communications, participants write the name of the person, who they want to mention or talk to. It could be the full name of that person or just his first name.

Als nächstes wurde für die jeweiligen Stichprobedaten berechnet, bei welchem Anteil der Daten die als Frage definierte Sätze gelten. Die Ergebnisse dieser Analyse sind in der Tabelle 4 dargestellt.

Als Antwort stehen fünf Antwortmöglichkeiten zur Verfügung:

„Fully true“, „More likely to be true“, „More unlikely to be true“, „Not true at all” und „Not Specify”.
Wenn die anderen Antwortmöglichkeiten für den Nutzenden nicht gelten, kann „Not Specify“ ausgewählt werden. Diese Antwortmöglichkeiten bestimmen letztendlich, welchen Anteil an den Einträgen, die den Fragen entsprechenden Eigenschaften haben.

Frage	App-Reviews	GitHub	Jira	Stack Overflow	Code-Reviews
1	**64%**	26 %	23 %	33 %	**6 %**
2	**43 %**	2 %	1 %	17 %	**0 %**
3	**0 %**	63 %	64 %	78 %	**97 %**
4	**31 %**	0 %	0 %	1 %	**0 %**
5	0 %	0 %	**12 %**	0 %	**0 %**
6	13 %	43 %	**63 %**	10 %	**3 %**
7	**0 %**	6 %	0 %	1 %	**20 %**
8	0 %	16 %	9 %	**93 %**	**0 %**
9	**97 %**	20 %	23 %	37 %	**1 %**
10	**24 %**	0 %	1 %	0 %	**0 %**
11	**80 %**	17 %	43 %	**16 %**	53 %
12	0 %	**14 %**	0 %	8 %	**0 %**
13	**0 %**	7 %	**32 %**	6 %	3 %

Tabelle 4: Anteil der Einträge in jeweiligen Stichprobedaten, für die die Sätze in jeweiligen Fragen gelten. Das Maximum und Minimum der Werte bei jeder Frage sind fett markiert. Zahlen sind auf die nächste natürliche Zahl gerundet.

Um jeder Antwortmöglichkeit der Fragen eine Domäne zuzuweisen, wurden zuerst der maximale und der minimale Wert der Anteile aus der Tabelle 4 identifiziert. Danach wurde der Wert D als die Differenz des maximalen und minimalen Wertes berechnet. Als nächster Schritt wurde D durch vier dividiert. Somit wurde bei jeder Frage festgestellt, welchem Intervall jede Antwortmöglichkeit entspricht. Die untenstehenden Formeln beschreiben, wie die Größe der Intervalle berechnet wurde. Die Intervalle wurden, wie in Tabelle 5 dargestellt, den Antwortmöglichkeiten zugewiesen. Somit konnten anhand der Tabelle 4 und der Tabelle 5 bei jeder Frage die Domänen den Antwortmöglichkeiten zugeteilt werden. Falls bei einer Frage keine Domäne der Antwortmöglichkeit „More likely to be true“ bzw. „More unlikely to be true“ zugeteilt werden kann, wird dieser Antwortmöglichkeit die Domänen bzw. die Domäne der Antwortmöglichkeit „Fully true“ bzw. „Not true at all“ zugeteilt.

Fully true	More likely to be true	More unlikely to be true	Not true at all
3*d bis max	2*d bis 3*d	d bis 2*d	min bis d

Tabelle 5: Zuteilung der berechneten Intervalle auf die Antwortmöglichkeiten der Fragen

$$max = Max.Wert\ der\ Tabelle\ 3\ bei\ Frage\ i$$
$$min = Min.Wert\ der\ Tabelle\ 3\ bei\ Frage\ i$$
$$D = max - min$$
$$d = D / 4$$

Eine Übersicht über die entsprechenden Domänen bei der Auswahl der jeweiligen Antwortmöglichkeiten ist aus der Tabelle 6 zu entnehmen.

Der Fragenkatalog beinhaltet auch einen zweiten Teil, wobei die Befragten die im Kapitel 4.2.6 berechneten Statistiken zu ihren Kommunikationen angeben sollen.
Die Antworten auf die 13 Fragen des Fragenkatalogs und die anzugebenden Statistiken haben alle eine Gewichtung von eins.

Frage	Fully true	More likely to be true	More unlikely to be true	Not true at all
1	App-Reviews	App-Reviews	Jira, GitHub, Stack Overflow	Code-Reviews
2	App-Reviews	App-Reviews	Stack Overflow	Code-Reviews, Jira, GitHub
3	Code-Reviews, Stack Overflow	GitHub, Jira	App-Reviews	App-Reviews
4	App-Reviews	App-Reviews	Stack Overflow, GitHub, Jira, Code-Reviews	Stack Overflow, GitHub, Jira, Code-Reviews
5	Jira	Jira	GitHub, Stack Overflow, App-Reviews, Code-Reviews	GitHub, Stack Overflow, App-Reviews, Code-Reviews
6	Jira	GitHub	Stack Overflow, App-Reviews, Code-Reviews	Stack Overflow, App-Reviews, Code-Reviews

7	Code-Reviews	Code-Reviews	GitHub	Stack Overflow, App-Reviews, Jira
8	Stack Overflow	Stack Overflow	App-Reviews, Code-Reviews, Jira, GitHub	App-Reviews, Code-Reviews, Jira, GitHub
9	App-Reviews	App-Reviews	Stack Overflow	Code-Reviews GitHub, Jira,
10	App-Reviews	App-Reviews	GitHub, Jira, Stack Overflow, Code-Reviews	GitHub, Jira, Stack Overflow, Code-Reviews
11	App-Reviews	Code-Reviews	Jira	GitHub, Stack Overflow
12	GitHub	Stack Overflow	Code-Reviews, App-Reviews	Code-Reviews, App-Reviews
13	Jira	Jira	GitHub, Stack Overflow, App-Reviews, Code-Reviews	GitHub, Stack Overflow, App-Reviews, Code-Reviews

Tabelle 6: Die entsprechende Domäne bei der Auswahl der jeweiligen Antwortmöglichkeiten der Fragen

Kapitel 6

6 Performanzanalyse der Stimmungsanalysetools

In diesem Kapitel wird beschrieben, wie die Evaluation der Genauigkeit der Stimmungsanalysetools durchgeführt wurde und wie dessen Ergebnisse aussehen. Diese Evaluation wird später bei der Umsetzung der Software verwendet, um das Tool mit der höchsten Genauigkeit vorzuschlagen.

6.1 Datenbereinigung

Um die Genauigkeit der Tools zu berechnen, wurden die im Kapitel 2.4 beschriebenen Datensätze verwendet. Vor der Evaluation wurden folgende Schritte durchgeführt:

1. Bei allen Datensätzen wurden die Duplikate entfernt.
2. New-Line-Zeichen ("\n") wurden aus allen Einträgen entfernt.
3. Einträge ohne zugewiesene Labels wurden aus allen Einträgen entfernt.
4. Einträge, die nur aus leerem String bestanden (""), wurden aus allen Datensätzen entfernt.
5. Alle Spalten außer den Spalten der Polaritäten und Texte wurden entfernt.
6. Die Spalten der Polaritäten und Texte wurden bei allen Datensätzen entsprechend auf „Text“ und „Polarity“ umbenannt.
7. Im GitHub-3-Datensatz wurden alle Einträge mit dem Label „sarcasm“ entfernt.
8. In den Datensätzen, die nicht mit den Zahlen 1, 0, und -1, sondern mit „positive“, „neutral“ und „negative“ bzw. „Positive“, „Neutral“ und „Negative“ gelabelt worden waren, wurden die Labels entsprechend auf das Zahlenformat angepasst.
9. Bei den Test-Daten für die Evaluation von SentiSW wurden die Spalten entsprechend in „text“ und „Annotation“ umbenannt, weil dies vom Tool so erwartet war.
10. Beim Jira-1-Datensatz wurden die Emotionslabels „Excited“, „Stress“, „Relax“, „Depression“ und „Neutral“ entsprechend durch Polaritäten 1, -1, 1, -1 und 0 ersetzt.
11. Bei der dritten Gruppe des Jira-2-Datensatzes, der aus vier nach Emotionen aufgeteilten und gelabelten Dateien besteht, wurden die Labels „love“, „joy“, „sadness“ und „anger“ entsprechend durch 1, 1, -1 und -1 ersetzt. Alle Einträge, die gar keine Labels hatten, wurden als neutral gesehen und mit 0 gelabelt.
12. Bei der ersten und zweiten Gruppe des Jira-2-Datensatzes wurden alle Einträge, die mit „surprise“ gelabelt waren, entfernt.

13. Bei der zweiten Gruppe des Jira-2-Datensatzes wurden die Labels „love“, „joy“, „anger“, „sadness“, „fear“ und „neutral“ entsprechend durch 1, 1, -1, -1, -1 und 0 ersetzt.
14. Beim Stack-Overflow-2-Datensatz wurden die Labels „LOVE“, „JOY“, „ANGER“, „SADNESS“ und „FEAR“ entsprechend durch 1, 1, -1, -1 und -1 ersetzt. Alle Einträge, die gar keine Labels hatten, wurden als neutral gesehen und mit 0 gelabelt.
15. Zur Evaluation der Datensätze mit RoBERTa wurde bei allen negativen Einträgen das Label -1 durch 2 ersetzt.

Nach der Evaluation der Datensätze mit den Tools wurden folgende Schritte durchgeführt:

16. Die von SentiStrength-SE [9] und SentiStrength [11] berechneten Werte wurden in drei Polaritäten umgewandelt, indem alle Werte kleiner Null mit -1, alle Werte größer Null mit 1 und alle Werte gleich Null mit 0 gelabelt wurden.
17. Auch bei Deva [16] wurden die resultierenden Emotionen in Polaritäten umgewandelt, indem die Emotionen „STRESSED“ und „DEPRESSED“ als -1, „RELAXED“ und „EXCITED“ als 1 und „NEUTRAL“, „NEUTRAL (HA)“ und „NEUTRAL (LA)“ als 0 eingestuft wurden. Das Label „POS VALENCE“ wurde zu 1 und das Label „NEG VALENCE“ zu -1 transformiert. Außerdem wurden alle Ergebnisse von DEVA vom File-Format in das einheitliche CSV-Format transformiert.
18. Die von Senti4SD [8] zugewiesenen Labels wurden von „positive“, „negative“ und „neutral“ in 1, -1 und 0 umgeändert.
19. Die zugewiesenen Labels von allen Stimmungsanalysetools beim Code-Review-Datensatz wurden entsprechend von „0“ und „1“ in „0“ umgeändert. Alle mit „-1“ gelabelten Einträge haben ihr Label behalten.

6.2 Performanzanalyse der lexikonbasierten Tools

Bei der Performanzanalyse der lexikonbasierten Tools wurden alle Daten der zusammengefügten Datensätze verwendet. Eine Übersicht über die Ergebnisse ist in der Tabelle 7 dargestellt.

6.3 Trainieren und Testen der Machine-Learning-geschützten Tools

Um die Genauigkeit der Machine-Learning-gestützten Tools zu berechnen, wurde jeder Datensatz in Test-Daten und Train-Daten gesplittet. Train-Daten besteht aus jeweils 80% und Test-Daten aus jeweils 20% der gesamten Daten eines Datensatzes ohne Duplikate. Die Tools wurden mit den jeweiligen Train-Daten trainiert und schließlich mit dem Test-Daten aus demselben Datensatz getestet. Zur Evaluation der Datensätze mit RoBERTa [19] wurde das vortrainierte Model „roberta-base“ genutzt. Dieses Model verwendet die BERT-base-Architektur [71]. Um den Datensatz zu in

Train-Daten und Test-Daten zu splitten und die Genauigkeit der Tools zu bewerten wurde die Python-Bibliothek scikit-learn [72] verwendet.

6.4 Ergebnisse der Evaluation

Die Ergebnisse der Evaluation der Genauigkeiten sind in der Tabelle 7 dargestellt. RoBERTa schneidet bei App-Reviews, Code-Reviews, GitHub-1, GitHub-2, Jira-1, Jira-2, Stack-Overflow-1 und Stack-Overflow-2 besser als andere Tools ab. Bei GitHub-3 hat das SEnti-Analyzer eine höhere Genauigkeit.
Es war nicht möglich, die drei GitHub-Datensätze zu konkatenieren und die konkatenierten Daten als einen Datensatz zu betrachten, weil diese Datensätze mit unterschiedlichen Richtlinien gelabelt worden waren. Als Beispiel wurde GitHub-1 basierend auf ein Emotionsmodell gelabelt, während GitHub2 ad-hoc gelabelt wurde [23] [15]. Um das genaueste Tool für GitHub zu identifizieren, wurde bei jedem Tool der Durchschnittswert von Overall-Scores der drei GitHub-Datensätze berechnet. Die Tabelle 8 stellt die Ergebnisse dar. Wie es sich aus der Tabelle 8 zu entnehmen lässt, ist RoBERTa das genaueste Tool bei GitHub im Allgemeinen. Basierend auf diesen Informationen kann festgelegt werden, dass die zu entwickelnde Software bei der Identifizierung von App-Reviews, Code-Reviews, GitHub, Jira und Stack Overflow als Domäne RoBERTa zum Zweck der Stimmungsanalyse vorschlagen soll.

Datensatz	Metriken	SentiStrength-SE	SentiStrength	Deva	SentiCR	Senti4SD	SentiSW	SEnti-Analyzer	RoBERTa
App-Reviews	Micro-averaged F1	59 %	62 %	61 %	74 %	87 %	80 %	59 %	**94 %**
	Macro-averaged F1	50 %	49 %	47 %	52 %	57 %	56 %	53 %	**63 %**
	Overall-Score	54 %	55 %	54 %	63 %	72 %	68 %	56 %	**78 %**
Code-Reviews	Micro-averaged F1	76 %	70 %	73 %	79 %	81 %	78 %	77 %	**88 %**
	Macro-averaged F1	58 %	58 %	58 %	72 %	74 %	69 %	55 %	**84 %**
	Overall-Score	67 %	64 %	65 %	75 %	77 %	74 %	66 %	**86 %**
GitHub-1	Micro-averaged F1	70 %	61 %	74 %	82 %	89 %	78 %	72 %	**92 %**
	Macro-averaged F1	68 %	62 %	74 %	81 %	90 %	79 %	69 %	**92 %**
	Overall-Score	69 %	62 %	74 %	81 %	89 %	79 %	71 %	**92 %**
GitHub -2	Micro-averaged F1	67 %	52 %	43 %	74 %	75 %	74 %	72 %	**83 %**
	Macro-averaged F1	60 %	52 %	38 %	65 %	61 %	62 %	57 %	**75 %**
	Overall-Score	65 %	52 %	41 %	69 %	68 %	68 %	64 %	**79 %**
GitHub -3	Micro-averaged F1	69 %	52 %	62 %	68 %	71 %	67 %	72 %	**73 %**
	Macro-averaged F1	54 %	50 %	54 %	**55 %**	41 %	47 %	54 %	38 %
	Overall-Score	61 %	51 %	58 %	61 %	56 %	57 %	**63 %**	55 %
Jira-1	Micro-averaged F1	83 %	74 %	89 %	87 %	88 %	86 %	79 %	**95 %**
	Macro-averaged F1	83 %	74 %	90 %	85 %	88 %	86 %	78 %	**94 %**
	Overall-Score	83 %	74 %	89 %	86 %	88 %	86 %	78 %	**95 %**

Datensatz	Metriken	SentiStrength-SE	SentiStrength	Deva	SentiCR	Senti4SD	SentiSW	SEnti-Analyzer	RoBERTa
Jira-2	Micro-averaged F1	80 %	65 %	74 %	85 %	81 %	84 %	80 %	**86 %**
	Macro-averaged F1	79 %	69 %	73 %	78 %	77 %	80 %	74 %	**82 %**
	Overall-Score	79 %	67 %	73 %	81 %	79 %	82 %	77 %	**84 %**
Stack-Overflow-1	Micro-averaged F1	78 %	69 %	74 %	83 %	86 %	83 %	80 %	**89 %**
	Macro-averaged F1	42 %	54 %	41 %	66 %	70 %	64 %	46 %	**79 %**
	Overall-Score	60 %	61 %	57 %	75 %	78 %	73 %	62 %	**84 %**
Stack-Overflow-2	Micro-averaged F1	77 %	76 %	74 %	80 %	82 %	79 %	83 %	**87 %**
	Macro-averaged F1	78 %	79 %	74 %	78 %	81 %	78 %	83 %	**86 %**
	Overall-Score	78 %	78 %	74 %	79 %	81 %	68 %	83 %	**87 %**

Tabelle 7: Ergebnisse der Evaluation der Genauigkeit der Stimmungsanalysetools. Die höchsten Werte bei jeder Metrik sind fett markiert. Zahlen sind auf die nächste natürliche Zahl gerundet.

Datensatz	Metrik	SentiStrength-SE	SentiStrength	Deva	SentiCR	Senti4SD	SentiSW	SEnti-Analyzer	RoBERTa
GitHub	Overall-Score	65 %	55 %	58 %	70 %	71 %	68 %	66 %	**75 %**

Tabelle 8: Durchschnittswerte von Overall-Scores von GitHub-1, GitHub-2 und GitHub-3 für jedes Tool. Der höchste Wert der Metrik ist fett markiert. Zahlen sind auf die nächste natürliche Zahl gerundet.

Kapitel 7

7 Umsetzung der Anwendung

In diesem Kapitel wird der Prozess der Umsetzung der Anwendung beschrieben. Zuerst wird die Planungsphase der Entwicklung dargestellt und anschließend wird die Implementierung der jeweiligen Funktionen der Anwendung beschrieben.

7.1 Planung

Bei der Entwicklung einer Software ist die Planung des Entwicklungsprozesses eine wesentliche Phase. In diesem Zusammenhang sollten zuerst die passende Plattform und das passende Framework sowie die zu verwendenden Bibliotheken ausgewählt werden. Windows ist mit über 75 % Marktanteil weltweit das meist genutzte Betriebssystem [73]. Aus diesem Grund und der Erfahrung mit der Entwicklung der Desktop-Anwendungen wurden Java und JavaFX als die Programmiersprache und Software-Plattform für die zu entwickelnde Anwendung ausgewählt. Die initialen Anforderungen der Anwendung wurden vom Betreuer der vorliegenden Arbeit spezifiziert und wurden regelmäßig um neue Anforderungen ergänzt. Als nächster Schritt wurden die Hauptfunktionen der Anwendung als Use Cases beschrieben, um die Funktionsweise der Anwendung besser entwickeln und die potenziellen Fehlverhalten der Anwendung früher erkennen zu können. Um das Graphical User Interface (GUI) der Anwendung und die Hauptfunktionen zu veranschaulichen, wurden Paper-Prototypen gezeichnet und als eine Grundlage für die Gestaltung des GUI verwendet, siehe Anhang. Nachdem eine erste Version der Anwendung entwickelt wurde, wurde die Anwendung von zwei Probanden testweise benutzt, um herauszufinden, inwiefern die Anforderungen erfüllt sind. Die Anwendung wurde durch die Feedbacks dieser Probanden nach den durchgeführten Probeläufen verbessert: so wurde beispielsweise die Position eines Buttons auf dem GUI geändert und Fehlfunktionen der Anwendung korrigiert. An diesen Probeläufen haben zwei Personen teilgenommen und haben die Use Cases der Anwendung einmal durchgeführt und ihre Feedbacks mündlich geäußert. Diese Personen sind beide Studierenden der Informatik bzw. Unternehmensentwicklung und haben bisher keine Erfahrungen im Bereich der Stimmungsanalyse.

7.2 Use Cases

Hauptakteure aller Use Cases sind Mitglieder eines Software-Entwicklerteams. Der einzige Use Case der Anwendung auf der Hauptebene (Tabelle 9), die die Use Cases auf der Ebene der Teilfunktionen (Tabelle 10 und Tabelle 11) beinhaltet, ist das Beantworten der Fragen des Fragenkatalogs und der Erhalt eines Vorschlags für ein

Stimmungsanalysetool. Zuerst startet der Nutzende die Recommender-Funktion, indem der Nutzende auf den Recommender-Button klickt. Danach beantwortet der Nutzende die 13 Fragen des Fragenkatalogs. Durch den Klick auf Next-Button navigiert der Nutzende zur nächsten Szene, wo der Nutzende Statistiken der Kommunikation entweder explizit eingeben oder von der Anwendung berechnen lassen kann. Dem Nutzenden ist es durch Klick auf Back-Button möglich, wieder zur vorherigen Szene zu navigieren. Um die Statistiken von der Anwendung berechnen zu lassen, klickt der Nutzende auf Select-File-Button und wählt die CSV-Datei aus. Anschließend klickt der Nutzende ein Button, um die Berechnung der Statistiken zu starten. Nach der Auswahl der Statistiken klickt der Nutzende den Submit-Button. Die Anwendung zeigt anschließend das vorgeschlagene Tool, die Domäne der Kommunikation des Nutzenden und Erklärungen zum Tool sowie die Kommunikationsart des Nutzenden. Der Nutzende klickt das PDF-Icon an, um eine Übersicht seiner Antworten, Statistiken und weitere Kommunikationsinformationen zu speichern. Das geschieht dadurch, dass der Nutzende einen Speicherpfad und einen Dateinamen angibt.

7.3 Implementierung der Funktionen

Zuerst wurde der Fragenkatalog implementiert. Um die Antworten auf die Fragen bei der Navigation zwischen den Szenen zwischenzuspeichern, wurde das Modell-View-Controller-Pattern (MVC) genutzt. In dieser Anwendung sind die Antworten auf die 13 Fragen des Fragenkatalogs als Enum-Objekte, Statistiken der Kommunikation als Zahlen und der Name der hochgeladenen CSV-Datei als String die Attribute des Hauptmodells. Um die Eigenschaften der GUI-Elemente zwischenzuspeichern, wurde ein anderes Modell definiert, das alle GUI-Elemente der Szenen der Recommender-Funktion beinhaltet. Jede als FXML-Datei definierte Szene der Anwendung hat entsprechend eine Controller-Java-Klasse.
Zu jeder der 13 Fragen des Fragenkatalogs stehen im GUI die Antwortmöglichkeiten als eine Toggle-Group und die einzelnen Antworten als Radio-Buttons zur Verfügung. Entsprechend den angeklickten Radio-Button und der im Kapitel 5 dargestellten Tabelle 5 bekommt eine Domäne in einem Scoring-Prozess bei jeder Frage einen positiven Punkt. Bei allen Fragen ist „Not Specify“ voreingestellt. Für die Navigation zwischen den Szenen des Recommenders gibt es einen Next- und einen Back-Button. Die Attribute der genannten Modelle der Anwendung werden bei der Navigation zwischen den Szenen zwischengespeichert und beim Laden der Szenen wiederhergestellt. Die Statistiken der Kommunikation können auf der zweiten Szene des Recommenders von Nutzenden in Text-Fields eingegeben werden. Die Anwendung prüft die Eingabe der Nutzenden ständig und färbt den Hintergrund der Text-Fields rot, falls die Eingaben keine Zahlen sind. Falls der Nutzende eine CSV-Datei zur statistischen Analyse auswählt, wird der Pfad zu dieser Datei in das Modell gespeichert. Nach dem Klick auf den Calculate-Statistics-Button auf der zweiten Szene des Recommenders läuft ein Python-Skript durch, das die Statistiken für die ausgewählte Datei berechnet. Während der Berechnung der

Statistiken wird dem Nutzenden mittels Multi-Threading-Fähigkeiten von JavaFX einen Indicator angezeigt. Die berechneten Statistiken werden in Text-Fields eingesetzt und somit werden die Text-Fields erstmal deaktiviert. Durch einen Klick auf den Delete-Button ist es möglich, den Pfad zur hochgeladenen CSV-Datei zu löschen und die Statistiken wieder explizit einzugeben. Die Statistiken werden durch den Klick auf den Submit-Button von der Anwendung in das Modell gespeichert und mit den vorhandenen Statistiken jeder Domäne im Scoring-Prozess verglichen. Die Statistiken der Domänen sind in ein Array gespeichert. Wenn eine berechnete Statistik derselben Statistik einer Domäne am nächsten ist, bekommt diese Domäne einen positiven Punkt. Falls die hochgeladene CSV-Datei keine Spalten mit dem Namen „Text“ hat, sieht der Nutzende eine Fehlermeldung und wird darauf hingewiesen. Die erste Szene des Recommenders ist in Abbildung 2 veranschaulicht.

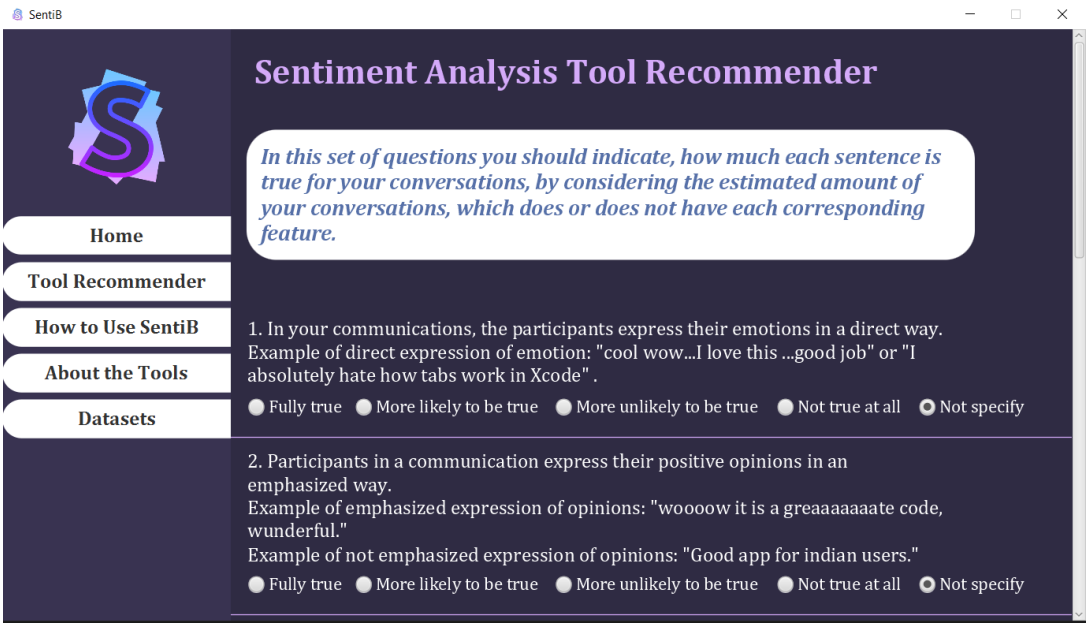

Abbildung 2: Screenshot von der ersten Recommender-Szene

Sobald der Nutzende auf den Submit-Button klickt, wird der Scoring-Prozess gestartet, der letztendlich feststellt, welche Domäne die meisten Punkte hat und somit der Kommunikationsart der Nutzenden am stärksten ähnelt.
Falls Nutzende keine CSV-Datei hochladen und keine Statistiken explizit eingeben und mindestens sieben Fragen mit „Not Specify“ beantworten, wird ihnen sowohl RoBERTa [19] als auch SentiStrength-SE [9] als Tool vorgeschlagen, da es zu wenige Informationen gibt, um die Domäne der Kommunikation der Nutzenden zu identifizieren. In dem Fall, dass der Nutzende keine Fragen beantwortet, aber mindestens eine der Statistiken angibt, wird trotzdem die Domäne der Kommunikationen des Nutzenden identifiziert. Nach dem Scoring-Prozess sehen Nutzende eine neue Szene, die das vorgeschlagene Tool, die identifizierte Domäne und einen Link zum vorgeschlagenen Tool beinhaltet. Über das Menü können Nutzende in eine neue Szene, zu den „Datasets“, navigieren, wo sie auf die in dieser Arbeit verwendeten Datensätze zugreifen können. Diese Datensätze können zum

Training des Tools benutzt werden, falls die Nutzenden keinen schon gelabelten Datensatz ihrer Kommunikation besitzen. Dazu ist es möglich durch Klick auf den PDF-Button auf der finalen Szene einen Bericht von den Antworten auf die Fragen, den Statistiken, der gefundenen Domäne und das vorgeschlagene Tool in Form einer PDF-Datei zu speichern. In dieser Datei werden auch zusätzliche Informationen über die Kommunikation von Nutzenden ermittelt. Zu diesen Informationen gehört eine Erklärung des Prozesses der Identifizierung der Domäne und das genaueste Tool. Zur Erstellung der PDF-Datei wurde die Java-Bibliothek PDFBox [74] benutzt. Angesichts der gefundenen Domäne wird mittels PDFBox Textzeilen in die PDF-Datei geschrieben. Über die Tools-Szene besteht die Möglichkeit, durch den Klick auf den entsprechenden Button zu jedem Tool auf die Stimmungsanalysetools zuzugreifen. Nach dem Klick auf den Button wird der Link zu jedem Tool im Browser geöffnet. Die finale Szene der Anwendung ist in der Abbildung 3 veranschaulicht.

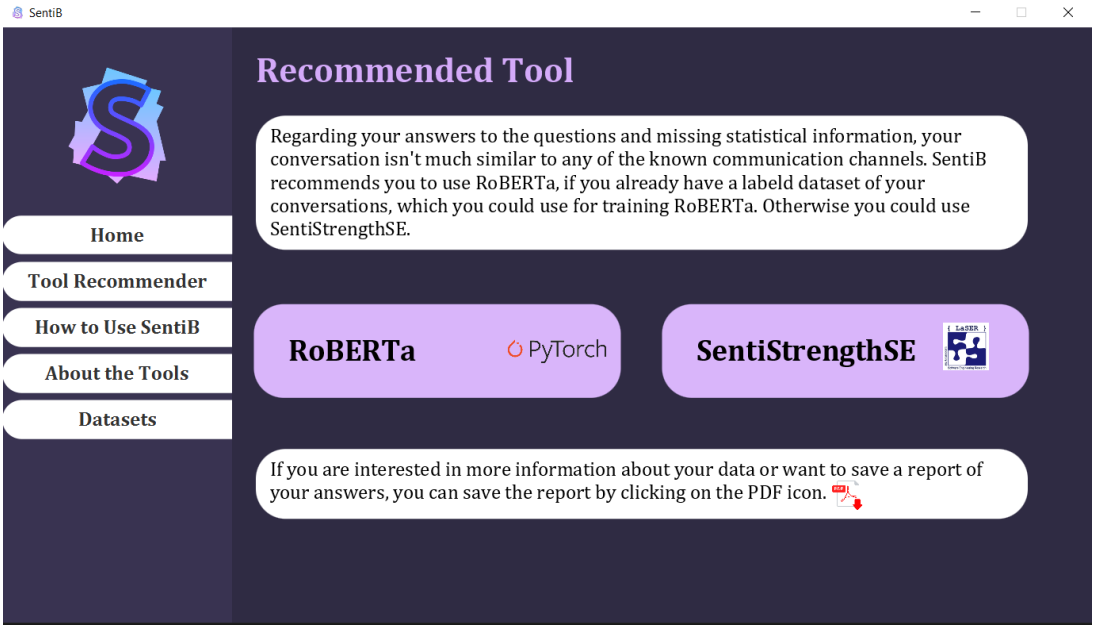

Abbildung 3: Screenshot von der Final-Szene

Kapitel 8

8 Evaluation der Anwendung

In diesem Kapitel wird die Evaluation der Anwendung und deren Ergebnisse dargestellt. Bei der Evaluation lag der Fokus hauptsächlich auf der Bedienbarkeit der Anwendung und der Qualität des Fragenkatalogs im Sinne der Verständlichkeit und Vollständigkeit.
Die Teilnehmenden der Evaluation haben im Rahmen von vier Aufgaben, basierend auf den Use Cases der Anwendung, mit der Anwendung gearbeitet. Um die Bedienbarkeit der Anwendung und die Qualität des Fragenkatalogs zu bewerten, wurde auch ein Fragebogen entwickelt, den die Teilnehmenden nach Abschluss der Bearbeitung der Aufgaben beantwortet haben. Zu Beginn dieses Fragebogens wurden Fragen zum demografischen Hintergrund der Teilnehmenden und ihrer Erfahrung im Gebiet der Stimmungsanalyse gestellt. Danach haben die Teilnehmenden Fragen zu verschiedenen Teilaspekten der Bedienbarkeit bei der Anwendung beantwortet. Bei diesen Fragen sollten die Teilnehmenden bestimmen, wie gut jeder Teilaspekt in der Anwendung umgesetzt worden ist. Diese Teilaspekte, die in ISO 9241-110 [75] definiert sind, sind Verfügbarkeit, Selbstbeschreibungsfähigkeit, Fehlertoleranz, Steuerbarkeit, Lernerförderlichkeit und Erwartungskonformität eines User-Interfaces. Jeder Teilaspekt wurde den Teilnehmenden im Fragebogen anhand von Beispielbildern definiert, damit die Teilnehmenden die Teilaspekte besser nachvollziehen konnten. Abschließend wurden Verbesserungsvorschläge und potenzielle neue Features für die Anwendung erfragt. Die Antworten der Teilnehmenden wurden anonymisiert gespeichert. Im Folgenden werden die Ergebnisse der Evaluation erörtert. Die Teilnehmenden werden in folgenden Teilkapiteln mit T1 bis T11 anonymisiert.

8.1 Demografischer Hintergrund

Es haben acht männliche und drei weibliche und somit insgesamt elf Personen an der Evaluation der entwickelten Anwendung teilgenommen, die entweder mindestens einmal in einem Software-Entwicklerteam gearbeitet haben oder zurzeit in so einem Team tätig sind. Die Teilnehmenden waren zum Zeitpunkt der Evaluation 25 bis 36 Jahre alt. Das Histogramm des Alters der Teilnehmenden ist in der Abbildung 4 dargestellt. Sechs Teilnehmende haben ein Bachelorstudium und zwei Teilnehmende ein Masterstudium abgeschlossen. Zwei Teilnehmenden haben die weiterführende Schule als höchsten Bildungsgrad angegeben und ein Teilnehmender die Ausbildung. Die Teilnehmenden wurden auch nach ihrer Rolle im Software-Entwicklerteam befragt. Sieben Personen sind als Entwickelnde und zwei Personen als Testende tätig. Als Führungskraft und Mitarbeitende mit einer fachlichen Rolle arbeitet jeweils einer der Teilnehmenden. Auf die Frage, ob die Teilnehmenden bereits vom Gebiet

der Stimmungsanalyse gehört haben, haben sechs Personen mit „Ja“ und fünf Personen mit „Nein“ geantwortet. Nur eine Teilnehmende Person hatte Stimmungsanalysetools, nämlich SentiStrength-SE und GerVADER [37], zu einem wissenschaftlichen Zweck verwendet und war teilweise mit deren Ergebnissen zufrieden, da es zur Zeit seiner wissenschaftlichen Arbeit keine besseren Tools gab. Alle Teilnehmenden haben sich dazu geäußert, dass sie innerhalb ihres Teams textuelle Kommunikationskanäle benutzen und haben diese angegeben, siehe Abbildung 5.

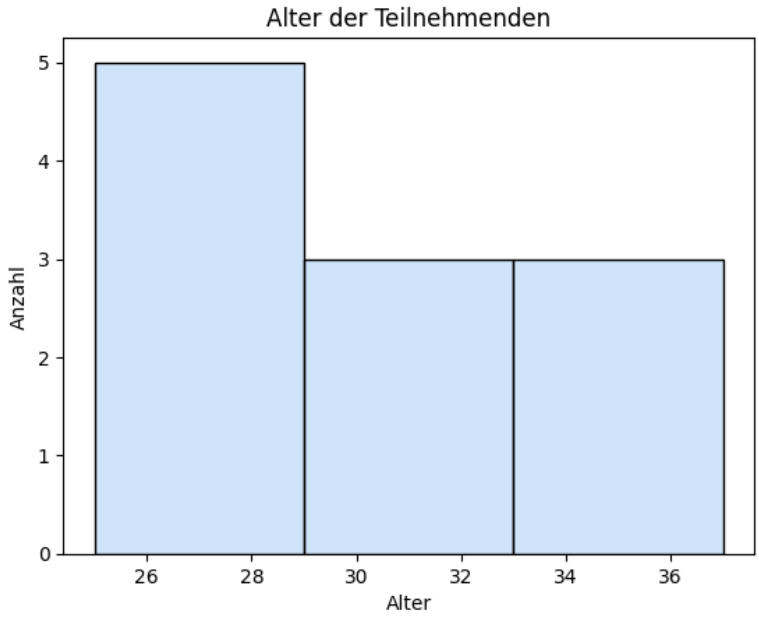

Abbildung 4: Histogramm des Alters der Teilnehmenden an der Evaluation

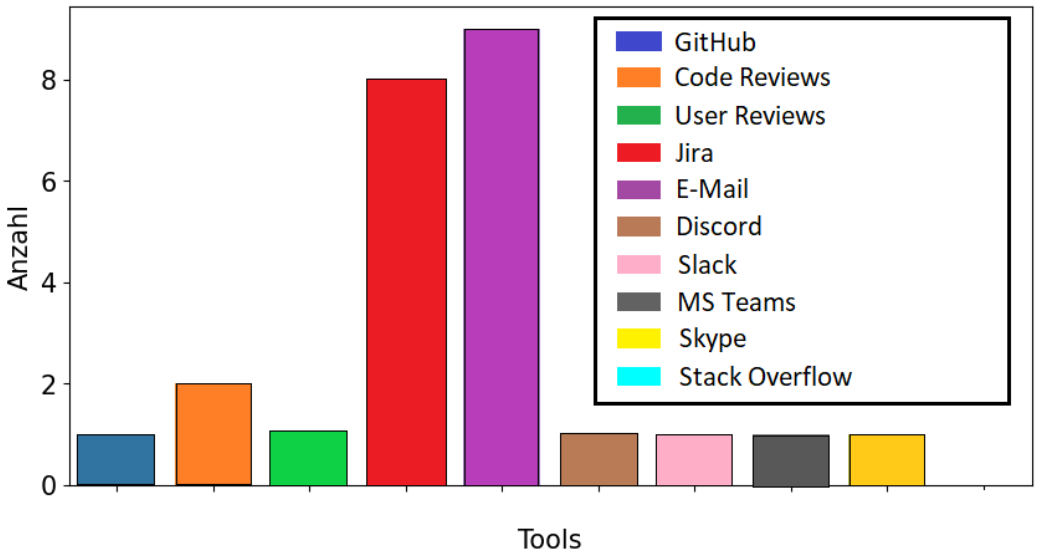

Abbildung 5: Verteilung der verwendeten Kommunikationskanäle der Teilnehmenden

8.2 Bedienbarkeit der Anwendung

Die Teilnehmenden sollten jeden Teilaspekt des User-Interface-Designs mit einer 5er-Skala als Antwortoptionen bewerten. Die Antwortoptionen waren „Sehr gut“, „Gut“, „Mäßig“, „Schlecht“ und „Sehr schlecht“. Zusätzlich zu den Antwortoptionen konnten die Teilnehmenden bei jeder Frage auch Kommentare hinterlassen. Die Bewertung der Teilaspekte wurde in Abbildung 6 dargestellt. Fast alle Teilnehmenden haben alle Teilaspekte mit „Sehr gut“ oder „Gut“ bewertet. Diese Bewertung zeigt, dass die entwickelte Anwendung leicht bedienbar ist.

- T5 hat außerdem geäußert, dass kleine Icons auf dem Menü die Selbstbeschreibungsfähigkeit der Anwendung verbessern würden.
- T9 wünschte sich bei der Selbstbeschreibungsfähigkeit, dass der Button für Recommender hervorgehoben wird.
- T11 und T8 merkten an, dass sie durch die Icons in der Datasets-Szene verwirrt wurden.
- Bei der Lernerförderlichkeit hat T11 angegeben, dass sie den Hilfetext der Statistiken nicht ganz verstanden hat. Dieser Hilfetext wurde in der Anwendung eingebaut, um die Bedeutung des Begriffs „Entry“ zu erklären.

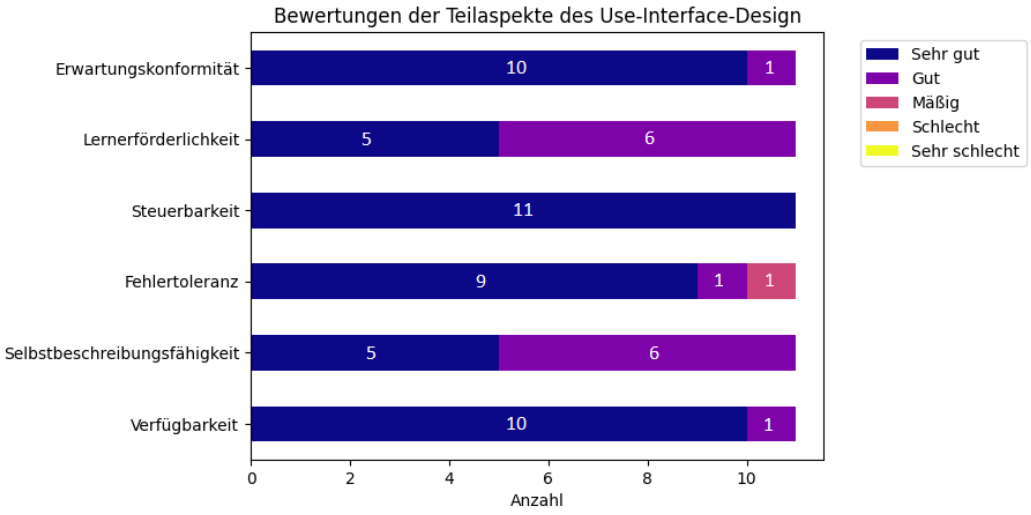

Abbildung 6: Die Bewertung der Teilaspekte des User-Interface-Designs

8.3 Qualität des Fragenkatalogs

Für alle Teilnehmenden waren die Fragen des Fragenkatalogs insgesamt gut verständlich, siehe Abbildung 7.

- Von T5 wurde bemängelt, dass die Sätze noch mehr entschärft werden und weniger komplexe Sätze verwendet werden sollten.
- T10 wünschte sich, dass die Fragen auch eine deutsche Version hätten.
- T11 hat geschrieben, dass nicht alle Fragen auf den ersten Blick verständlich waren. Sie hat ergänzt, dass es vielleicht an ihren Englischkenntnissen liegt.
- T6 fand es besser, mehr Antwortmöglichkeiten angeboten zu bekommen, um die Fragen besser beantworten zu können.

- T9 wünschte sich eine ordinale Skala mit Zahlen als Antwortmöglichleiten der Fragen.
- T11 hat vorgeschlagen, die Antwortmöglichkeiten in Form eines Balkens mit Bereichsschieber darzulegen, damit die Nutzenden ihre Antworten genauer eingeben können.
- T15 hat vorgeschlagen, eine Frage bezüglich der Kommunikation in den stressigen Phasen der Entwicklung, wie z. B. bei Deadlines, zum Fragenkatalog hinzuzufügen.
- T15 hat auch empfohlen, die Wiederholung der Wörter oder Sätze als eine Statistik zu berechnen. Als Beispiel präsentierte sie den folgenden Satz zur Stimmungsanalyse: „Du musstest das bis zum 01.12 machen, BIS ZUM 01.12!!!!!“.
- Als eine weitere Statistik empfahl T5, die Analyse der Häufigkeit und der Anzahl an Antworten oder Reaktionen auf einen Beitrag oder Kommentar zu beachten.

Von allen Teilnehmenden fanden sechs Personen die Informationen in der exportierten PDF-Datei sehr nützlich und fünf Personen nützlich.

- T3 wollte gerne aus dieser Datei Informationen erhalten, die den Vorschlag genauer begründen und die Ähnlichkeiten zu jeweiliger Domäne zeigen.
- T9 war der Meinung, dass der Name der CSV-Datei in der Datei stehen sollte, falls so eine Datei hochgeladen wurde.
- T11 fand es nützlich, den Link auf das vorgeschlagene Tool in die PDF-Datei zu schreiben.

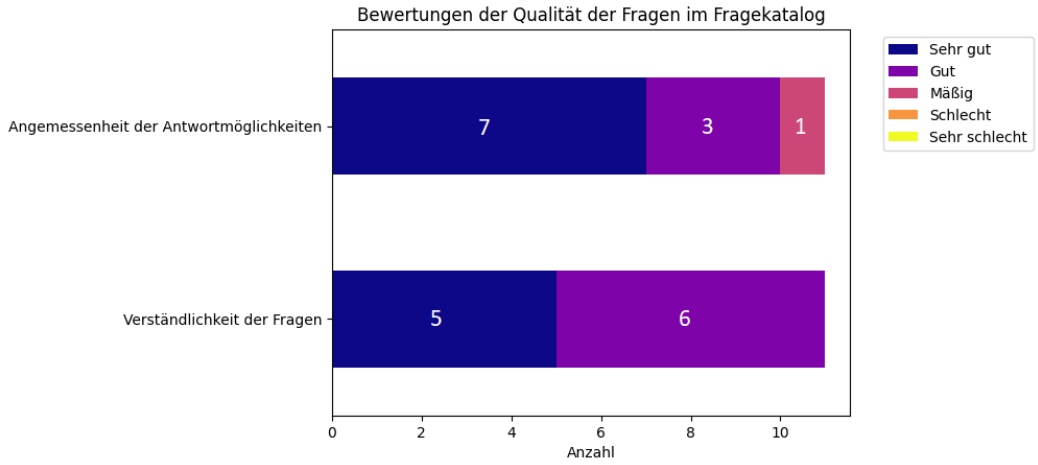

Abbildung 7: Bewertung der Qualität des Fragenkatalogs

8.4 Vorschläge für die Erweiterung der Anwendung

- Als weitere Features empfehlen T2 und T7 eine detailliertere Erklärung in der Anwendung, wie genau die Anwendung das beste Tool für ihre Kommunikation wählt.
- T7 und T9 ergänzen die Möglichkeit, einen begonnenen Vorgang zu speichern als ein interessantes neues Feature.

- T10 und T9 wünschen sich, die Anwendung auch in deutscher Sprache benutzen zu können.
- T8 hätte gern eine Möglichkeit bekommen, neue Stimmungsanalysetools zur Anwendung hinzuzufügen.
- T8 würde auch eine Product-Tour in die Anwendung einbauen lassen, die den Nutzenden zeigt, wie die Anwendung zu benutzen ist.
- Um die Anwendung zu verbessern, haben T5 und T9 mehr Bilder und weniger Text in der Anwendung gewünscht.
- Als einen weiteren Verbesserungsvorschlag hat T9 geschrieben, dass eine Verlaufsfunktion umgesetzt werden sollte.
- T11 hätte gern die Möglichkeit bekommen, alle Eingaben in der Anwendung wieder zurückzusetzen.

Den meisten der Teilnehmenden hat es gefallen, dass die Anwendung übersichtlich und verständlich aufgebaut ist und durch die Erklärungen und Beschriftung der Interaktionselemente gut und schnell bedienbar ist und Fehler abgefangen werden. Eine direkte Verlinkung der Tools und die geeignete Farbauswahl wurden von Teilnehmenden als weitere positive Punkte genannt.
Die Farbauswahl, die Größe der GUI-Elemente und die Formulierung der Fragen des Fragenkatalogs haben einigen Teilnehmenden hingegen nicht gefallen.

8.5 Weiterempfehlung und Verwendung der Anwendung in der Zukunft

Alle Teilnehmenden haben angegeben, dass sie die Anwendung in einem den durchgeführten Aufgaben ähnlichen Szenario benutzen und auch anderen Personen weiterempfehlen würden. Die Begründung für diese Entscheidung war zumeist, dass im Hinblick auf die Stimmungsanalysetools wenige Kenntnisse vorlagen und durch die entwickelte Anwendung der Aufwand, das beste Tool Online zu suchen, reduziert werden kann.
Einer der Teilnehmenden hat erklärt, dass er die Anwendung grundsätzlich weiterempfiehlt, kennt er aber keine andere Anwendung mit der gleichen Funktion, um diese mit der im Rahmen dieser Arbeit entwickelte Anwendung zu vergleichen.

Kapitel 9

9 Diskussion

Im Rahmen dieser Bachelorarbeit wurden bestimmte statistische Merkmale der Datensätze aus den Domänen App-Reviews, Code-Reviews, Jira, GitHub und Stack Overflow berechnet und die Domänen bezüglich ihrer Unterschiede bzw. Gemeinsamkeiten analysiert. Außerdem wurden die Genauigkeiten von sieben Stimmungsanalysetools im Bereich SE anhand von vorhandenen Datensätzen festgestellt. Basierend auf diesen Befunden wurde ein Fragenkatalog erstellt, um die Domäne der Kommunikation der Befragten zu identifizieren. In diesem Kapitel werden die Ergebnisse aus den Kapiteln 4, 5, 6 und 8 diskutiert. Anschließend werden Validity Threats dieser Ergebnisse nach Wohlin et al. [76] erörtert.

9.1 Interpretation der Ergebnisse

Die Ergebnisse der Kapitel 4 und 5 zeigen, dass sich die Domäne der Kommunikation eines Software-Entwicklerteams durch die Analyse unterschiedlicher statistischen und inhaltlichen Eigenschaften identifizieren lässt. Die Eigenschaften der Kommunikation sind in jeder Domäne unterschiedlich, weil in jeder Domäne mit einem bestimmten Kontext und zu einem bestimmten Zweck kommuniziert wird. Weitere Informationen, wie die Zeitstempel der Kommunikation, die Verbindung der Einträge in der Kommunikation oder die Analyse der Höflichkeit der Kommunikation können dazu beitragen, weitere Kenntnisse der Domäne der Kommunikation eines Software-Entwicklerteams zu erwerben.

Die Ergebnisse aus dem Kapitel 6 bestätigen die Ergebnisse der Studien von Zhang et al. [14] und Wu et al. [59]: transformerbasierte Tools schneiden bei der Stimmungsanalyse besser ab als andere Tools. Novielli et al. [10] haben gezeigt, dass Senti4SD [8] in Kombination mit Stack-Overflow-Daten als SentiCR, SentiStrength-SE und SentiStrength präziser arbeitet. Im Rahmen derselben Arbeit wurde bewiesen, dass SentiCR in Kombination mit Jira-2-Daten am besten performt. Die Ergebnisse aus Kapitel 6 widerlegen aber die Analyse der Genauigkeit von SentiCR in Kombination mit Code-Reviews. Der Grund dafür können die unterschiedlichen Einstellungen in der Training-Phase der Tools sein. Ebenfalls wurden die Ergebnisse der anderen wissenschaftlichen Arbeiten von Novielli et al. [23] und der Arbeit von Calefato et al. [8] bestätigt. In der letzteren Arbeit wurde gezeigt, dass Senti4SD in Kombination mit Stack-Overflow-Daten besser performt als SentiCR, DEVA, SentiStrength-SE und SentiStrength. Bei den Ergebnissen aus Kapitel 6 fällt auf, dass sich die Genauigkeiten von SentiStrength und SentiStrength-SE von denen in den

oben genannten Arbeiten teilweise deutlich unterscheiden. Das könnte daran liegen, dass bei diesen Arbeiten andere bzw. keine Datenbereinigungen vor der Evaluation durchgeführt wurden oder nicht die ganzen Daten evaluiert wurden. Es ist außerdem zu beachten, dass bei keinem Datensatz ein lexikonbasiertes Tool besser performt als ein Machine-Learning-gestütztes Tool.

Fast die Hälfte der Teilnehmenden der Evaluation der Anwendung hatten schonmal vom Gebiet der Stimmungsanalyse gehört, aber nur eine Person hatte mindestens einmal ein Stimmungsanalysetool im Rahmen ihrer Abschlussarbeit benutzt. Das kann bedeuten, dass die Stimmungsanalysetools bis jetzt hauptsächlich zu wissenschaftlichen Zwecken benutzt wurden und in betrieblichen Umgebungen eher unbekannt sind. Es hat sich herausgestellt, dass die meisten der Teilnehmenden über E-Mails mit anderen Teammitgliedern kommunizieren; und über Skype wird ebenso viel kommuniziert, wie über Code-Reviews. Deswegen könnten E-Mails und Skype-Chats für die künftigen Forschungen in der Stimmungsanalyse interessante Domänen darstellen. Es hat sich außerdem gezeigt, dass potenzielle Nutzende der Anwendung bereits einen Datensatz aus den Kommunikationen in seinem Team haben und sich einen Überblich über diesen Datensatz verschafft haben sollten, um die Fragen des Fragenkatalogs nach bestem Wissen beantworten zu können. Viele Teilnehmende der Evaluation der Anwendung haben den Wunsch geäußert, die Fragen des Fragenkatalogs auf Deutsch zu lesen und zu beantworten; dafür sind Gold-Standard-Datensätze im SE-Bereich zu erstellen und diese zu analysieren.

Ziel dieser Arbeit war, das genaueste Stimmungsanalysetool für jede Domäne zu identifizieren und anhand der Eigenschaften der Domänen einen Fragenkatalog zu erstellen, der dazu dient, die Domäne der Kommunikation der Befragten zu identifizieren. Dieser Fragenkatalog sollte am Ende in eine Anwendung eingebunden werden. Die Ergebnisse dieser Arbeit und deren Vergleich mit den Ergebnissen ähnlicher Arbeiten zeigen, dass dieses Ziel erreicht wurde: die entwickelte Anwendung kam bei den Probanden allgemein gut an und kann in Zukunft mit relativ geringem Aufwand weiterentwickelt werden. Die statistischen und inhaltlichen Eigenschaften der Datensätze und Domänen waren informativ und können die Grundlage für weitere Forschungsaktivitäten bilden.

9.2 Threats to Validity

Unter den vorhandenen Datensätzen wurden in dieser Arbeit nur die im Kapitel 2.4 vorgestellten Datensätze analysiert (Threat to Construct Validity). Zusätzlich zu diesen Datensätzen war ein Java-API-Reviews-Datensatz [62] vorhanden, der in dieser Arbeit nicht mitbetrachtet wurde. Der Grund für diese Entscheidung war, dass dieser Datensatz anscheinend von Erstellenden des Datensatzes vorverarbeitet wurde. Dennoch wurde der Vorverarbeitungsprozess in der wissenschaftlichen Arbeit von Uddin und Khomh [62] nicht beschrieben und es ist nicht bekannt, wie die Daten

vorverarbeitet wurden. Dadurch kann die Gefahr entstehen, dass die Daten nicht repräsentativ genug sind.

Die meisten der analysierten Datensätze sind unausgewogen. Das kann dazu führen, dass die Ergebnisse der Klassifizierung zugunsten einer Klasse verzerrt werden (Threat to Conclusion Validity). Aus diesem Grund wurde der F1-Score als die relevante Metrik ausgewählt, der das Gleichgewicht zwischen Precision und Recall hält.

Die Größe der Datensätze der jeweiligen Domänen sind unterschiedlich. Das kann dazu führen, dass nach dem Zusammenfügen und bei der zufälligen Auswahl der 90 Stichprobedaten aus der zusammengefügten Datei die Einträge der größeren Datensätze mit einer höheren Wahrscheinlichkeit ausgewählt werden als die Einträge aus der kleineren Datensätze (Threat to Conclusion Validity). Somit können die Stichprobedaten mehr dem Inhalt und der Art und Weise der Kommunikation des größeren Datensatzes entsprechen.

Eine weitere Schwierigkeit kann die Größe der Stichprobedaten aus den Datensätzen sein (Threat to Construct Validity). Aus jedem Datensatz wurden, abgesehen von der Größe des Datensatzes, ca. 90 Einträge berücksichtigt. Um aussagekräftigere Schlussfolgerungen aus den Stichproben zu ziehen, wäre eine Analyse größerer Stichproben notwendig.

Bei der Feststellung der inhaltlichen Eigenschaften der Domänen wurde keine bestimmte Richtlinie benutzt, um die Struktur der Einträge der Stichprobedaten zu analysieren und die Merkmale zu finden (Threat to Construct Validity). Alle Merkmale in den Domänen wurden durch statistische Herangehensweise und ohne Richtlinien herausgefunden und auch in anderen Domänen erforscht.

Bei der Evaluation der Genauigkeiten der Tools wurden alle Einträge der vorhandenen Datensätze von lexikonbasierten Stimmungsanalysetools klassifiziert. Um die Genauigkeit der Machine-Learning-gestützten Tools zu messen, wurden 80 % der Daten jedes Datensatzes in der Training-Phase und 20 % der Daten des Datensatzes in der Test-Phase verwendet. Es könnte sein, dass bei einer z. B. 70 % - 30 % Verteilung der Train-Test-Daten und der Anwendung der K-Fold-Cross-Validation genauere Ergebnisse bei der Evaluation erreicht werden (Threat to Construct Validity).

Bei einigen Datensätzen wurden vor der Veröffentlichung der Datensätze von den Erstellern Code-Ausschnitte und URLs entfernt und bei einigen anderen Datensätzen sind Code-Ausschnitte und URLs Teile der Daten. Das heißt, dass die Datensätze unterschiedlich vorverarbeitet wurden (Threat to Internal Validity). Das führt dazu, dass die Anzahl der von pyspellchecker gefundenen Rechtschreibfehler bei Datensätzen mit Code-Ausschnitten fälschlicherweise erhöht wird, da bestimmte

technische Wörter und Code-Ausschnitte von pyspellchecker als Rechtschreibfehler gemeldet werden (Threat to Conclusion Validity).

Es könnte bei der Evaluation der entwickelten Anwendung eine Schwierigkeit darstellen, dass die Interviews mit den Probanden vom Entwickler der Anwendung durchgeführt wurden. Das könnte dazu geführt haben, dass die Probanden die Anwendung in Anwesenheit des Entwicklers positiver bewerten, obwohl sie die Anwendung in der Abwesenheit des Entwicklers eigentlich nicht so positiv bewerten würden (Threat to Conclusion Validity). Um diesen Threat zu minimieren, wurden alle Teilnehmenden trotzdem vor dem Interview darum gebeten, die Anwendung ehrlich zu bewerten und nicht auf die Anwesenheit des Entwicklers zu achten.

Insgesamt haben elf Personen an der Evaluation der entwickelten Anwendung teilgenommen. Bei einer größeren Anzahl an Teilnehmenden könnten die Ergebnisse der Evaluation besser verallgemeinerbar sein (Threat to Conclusion Validity). Außerdem waren nur eine Teilnehmende mit einer fachlichen Rolle und ein Teilnehmende als Führungskraft in Software-Entwicklerteams tätig. Es könnten sich mehr Informationen bei der Evaluation der Anwendung ergeben, falls mehr Führungskräfte an der Evaluation teilgenommen haben, da Führungskräfte eine wichtige Zielgruppe dieser Anwendung sind (Threat to External Validity).

Die Teilnehmenden der Evaluation der Anwendung sprechen Englisch nicht als Muttersprache und kommunizieren in ihren Entwicklerteams fast ausschließlich auf Deutsch. Dass die Fragen des Fragenkatalogs auf Englisch sind, könnte das Verständnis der Fragen erschweren (Threat to Conclusion Validity). Um diese Gefahr zu minimieren, wurde den Teilnehmenden bei Unklarheiten im Fragenkatalog geholfen.

Bei der Evaluation der entwickelten Anwendung sollten die Teilnehmenden angeben, wie gut die jeweiligen Teilaspekte des User-Interface-Designs in der Anwendung umgesetzt wurden. Da jede Person Begriffe wie „gut“ oder „schlecht“ anders definieren und interpretieren kann, wurden die Teilaspekte des User-Interface-Designs von Teilnehmenden subjektiv bewertet (Threat to Construct Validity). Daher wurde im Fragebogen versucht, diese Teilaspekte zu definieren und anhand von Beispielbildern näher zu beschreiben.

Es könnte eine Schwierigkeit sein, dass bei der Performanzanalyse der lexikonbasierten Tools alle Daten der Datensätze evaluiert wurden (Threat to Construct Validity). Die Ergebnisse der Performanzanalyse der lexikonbasierten Tools könnten anders aussehen, falls bei diesen auch 20 % der Daten evaluiert worden wären.

Kapitel 10

10 Fazit und Ausblick

In diesem abschließenden Kapitel wird ein Überblick über den Inhalt und die Ergebnisse dieser Arbeit und ein Ausblick auf künftige Forschungen und die Weiterentwicklung der Anwendung gegeben.

10.1 Fazit

Um Stimmungen in Kommunikationsdaten aus dem SE-Bereich zu analysieren, sind mehrere Stimmungsanalysetools entwickelt worden. Aufgrund der unterschiedlichen Implementierungsdetails dieser Tools unterscheidet sich die Genauigkeit der jeweiligen Tools in Anhängigkeit davon, mit welchen Daten sie trainiert werden bzw. welche Daten von diesen Tools klassifiziert werden sollen. Natürlich ist es für jeden, der eine Stimmungsanalyse durchführen möchte, wünschenswert, das genaueste Instrument für diesen Zweck zu verwenden.
Im Rahmen dieser Arbeit sollte das genaueste Stimmungsanalysetool für Kommunikation in den Domänen Stack Overflow, Jira, App-Reviews, Code-Reviews und GitHub gefunden werden und basierend auf den vorhanden Datensätzen aus diesen Domänen ein Fragenkatalog erstellt werden. Mit der Beantwortung dieses Fragenkatalogs sollten Nutzende die für ihre Art der Kommunikation am besten geeignete Domäne identifizieren lassen können. Dazu wurden im Zuge dieser Arbeit Datensätze aus den Domänen Stack Overflow, Jira, App-Reviews, Code-Reviews und GitHub analysiert und ihre Merkmale in inhaltlicher und statistischer Hinsicht untersucht und mit anderen Domänen verglichen. Außerdem wurde für jede Domäne das genaueste Tool identifiziert. Es hat sich herausgestellt, dass bei allen Domänen RoBERTa das genaueste Stimmungsanalysetool ist. Anhand der gefundenen Merkmale in den Domänen wurde ein Fragenkatalog entworfen und als eine Software umgesetzt, die von Mitgliedern eines Software-Entwicklerteams benutzt werden kann, um das für sie bestmöglich passende Tool vorgeschlagen zu bekommen. Im Anschluss wurde die entwickelte Anwendung bezüglich ihrer Bedienbarkeit und der Qualität des enthaltenen Fragenkatalogs von elf Probanden evaluiert. Alle Probanden haben entweder mindestens einmal in einem Software-Entwicklerteam gearbeitet oder arbeiten heute in einem Entwicklerteam. Die Ergebnisse dieser Evaluation haben darauf hingewiesen, dass die Anwendung gut bedienbar ist und der Fragenkatalog zum Zweck der Identifizierung der Domäne der Kommunikation der Nutzenden geeignet ist.

10.2 Ausblick

In weiteren Forschungen können durch die Anwendung von Natural-Language-Processing, Machine-Learning-Algorithmen und bestimmte Richtlinien noch mehr inhaltliche und statistische Eigenschaften verschiedener Domänen festgestellt werden, um dem Fragenkatalog weitere Fragen hinzuzufügen. Auch ist es denkbar, weitere Datensätze und Domänen in den künftigen Forschungen zu betrachten, die in dieser Arbeit nicht berücksichtigt wurden. Des Weiteren wäre eine ähnliche Forschung und Analyse mithilfe deutscher Datensätze sinnvoll. So könnte ein neuer Fragenkatalog für die Kommunikationen in deutscher Sprache gestaltet und in die Software eingebunden werden. Es wäre weiterhin möglich, das UI/UX der bestehenden Anwendung zu optimieren und die Anwendung mit einem besseren Framework und in eine andere Programmiersprache umzuschreiben, damit die Anwendung dynamischer bedienbar wird. Auch wird empfohlen, die im Rahmen dieser Arbeit entwickelte Anwendung mit echten Kommunikationsdaten von Software-Entwicklerteams zu testen, um nachzuprüfen, ob die Anwendung auch für echte Kommunikationsdaten tatsächlich das genaueste Tool vorschlägt. In Zukunft können die lexikonbasierten Tools anhand von neuen Lexika angepasst werden, um die Performanzanalyse der Tools zu verbessern. Somit kann die im Rahmen dieser Arbeit entwickelte Anwendung Nutzenden sowohl ein lexikonbasiertes als auch ein Machine-Learning-gestütztes Tool vorschlagen.

Anhang

Use Cases

UC1	Fragen des Fragenkatalogs beantworten und den Vorschlag bekommen
Umfeld	Ein Raum in einem Unternehmen oder einer Universität
Ebene	Hauptebene
Hauptakteure	Mitglieder eines Software-Entwicklerteams
Stakeholder u. Interessen	Mitglied eines Software-Entwicklerteams möchte wissen, mit welcher Stimmung im Projekt gearbeitet wird.
Voraussetzung	Die Anwendung ist bereits installiert und wurde durch den Doppelklick auf das Icon geöffnet. Die Home-Szene wird gezeigt.
Garantie	Der Nutzende bekommt ein Stimmungsanalysetool vorgeschlagen oder sieht eine entsprechende Fehlermeldung.
Erfolgsfall	Der Nutzende bekommt ein Stimmungsanalysetool vorgeschlagen.
Auslöser	Der Nutzende klickt auf "Recommender" auf dem Menü.
Beschreibung	1. Der Nutzende klickt auf "Recommender" auf dem Menü.
	2. Die Szene mit 9 Fragen wird angezeigt.
	3. Der Nutzende beantwortet die 9 Fragen, indem er zu jeder Frage entsprechendes Radio-Button anklickt.
	4. Zu jeder Frage wird die ausgewählte Antwort markiert.
	5. Der Nutzende klickt auf "Next"-Button.
	6. Zweite Szene von Recommender wird angezeigt.
	7. Der Nutzende gibt Statistiken explizit ein.
	8. Der Nutzende klickt auf "Submit"-Button.
	9. Anwendung zeigt die finale Szene mit dem vorgeschlagenen Tool und die Möglichkeit, einen Bericht zu speichern.
Erweiterungen	2.a WENN der Nutzende den Recommender verlassen will, DANN klickt der Nutzende eine andere Option aus dem Menü.
	2.a.1 Die Anwendung fragt, ob Nutzende den Recommender verlassen will.
	2.a.2 Der Nutzende wählt "Yes" und verlässt somit den Recommender.
	6.a WENN der Nutzende die Statistiken nicht kennt, DANN CSV-Datei hochladen.
	6.b WENN der Nutzende die Fragen der ersten Szene des Recommenders anders beantworten will, klickt er "Back"-Button an.
	6.b.1 Dem Nutzende wird die erste Szene von

	Recommender angezeigt.
	9.a WENN der Nutzende den Bericht herunterladen möchte, DANN Bericht herunterladen.

Tabelle 9: Use Case 1

UC2	CSV-Datei hochladen
Umfeld	Ein Raum in einem Unternehmen oder einer Universität
Ebene	Teilfunktion
Hauptakteure	Mitglieder eines Software-Entwicklerteams
Stakeholder u. Interessen	Mitglied eines Software-Entwicklerteams möchte wissen, mit welcher Stimmung im Projekt gearbeitet wird.
Voraussetzung	Der Nutzende hat die Anwendung gestartet und zweite Szene vom Recommender wird angezeigt.
Garantie	Der Nutzende sieht den Namen der gewählten CSV-Datei sieht eine entsprechende Fehlermeldung.
Erfolgsfall	Der Nutzende sieht den Namen der gewählten CSV-Datei.
Auslöser	Der Nutzende klickt "Select File" an.
Beschreibung	1. Der Nutzende klickt auf "Select File"
	2. File-Chooser wird geöffnet.
	3. Der Nutzende navigiert zum gewünschten Pfad und wählt die Datei aus.
	4. Der Name der Datei wird angezeigt und der Nutzende kann die Statistiken nicht mehr direkt eingeben.
Erweiterungen	2.a WENN keine CSV-Datei ausgewählt wird, DANN wird der Nutzende darauf hingewiesen.
	4.a WENN die Datei nicht das richtige Format hat, DANN wird dem Nutzenden eine Fehlermeldung gezeigt.
	4.b WENN der Nutzende die Datei löschen möchte, DANN klickt er "Delete File" an.
	4.c WENN der Nutzende zuerst nur die Statistiken berechnen lassen will, DANN klickt er "Calculate Statistics" an.
	4.c.1 Die Anwendung berechnet die Statistiken anhand der Datei und füllt die Textfelder automatisch aus.

Tabelle 10: Use Case 2

UC3	PDF-Bericht exportieren
Umfeld	Ein Raum in einem Unternehmen oder einer Universität
Ebene	Teilfunktion
Hauptakteure	Mitglieder eines Software-Entwicklerteams
Stakeholder u. Interessen	Mitglied eines Software-Entwicklerteams möchte wissen, mit welcher Stimmung im Projekt gearbeitet wird.
Voraussetzung	Der Nutzende hat die Anwendung gestartet sieht die finale Szene des Recommenders.
Garantie	Die PDF-Datei wird im von Nutzenden angegebenen Pfad gespeichert oder dem Nutzenden wird eine Fehlermeldung angezeigt.
Erfolgsfall	Die PDF-Datei wird im von Nutzenden angegebenen Pfad gespeichert.
Auslöser	Der Nutzende klickt auf PDF-Icon.
Beschreibung	1. Der Nutzende klickt auf PDF-Icon.
	2. Path-Chooser wird geöffnet.
	3. Der Nutzende navigiert zum gewünschten Pfad und wählt den Pfad aus und gibt einen Namen ein.
	4. Die PDF-Datei wird im angegebenen Pfad gespeichert.
Erweiterungen	Keine

Tabelle 11: Use Case 3

Paper-Prototypen

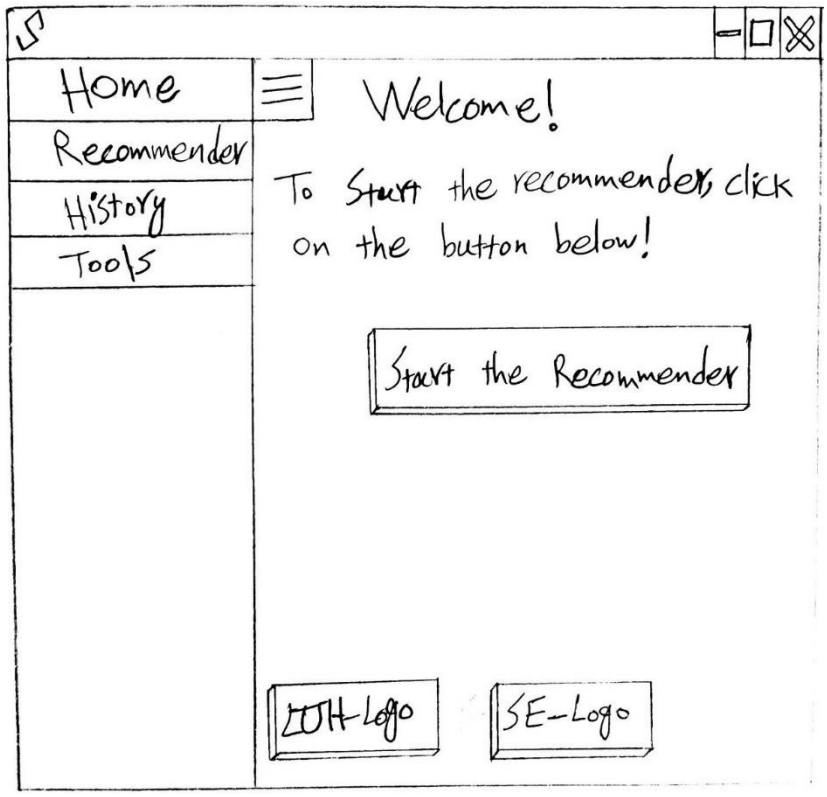

Abbildung 8: Paper-Prototyp der Welcome-Szene

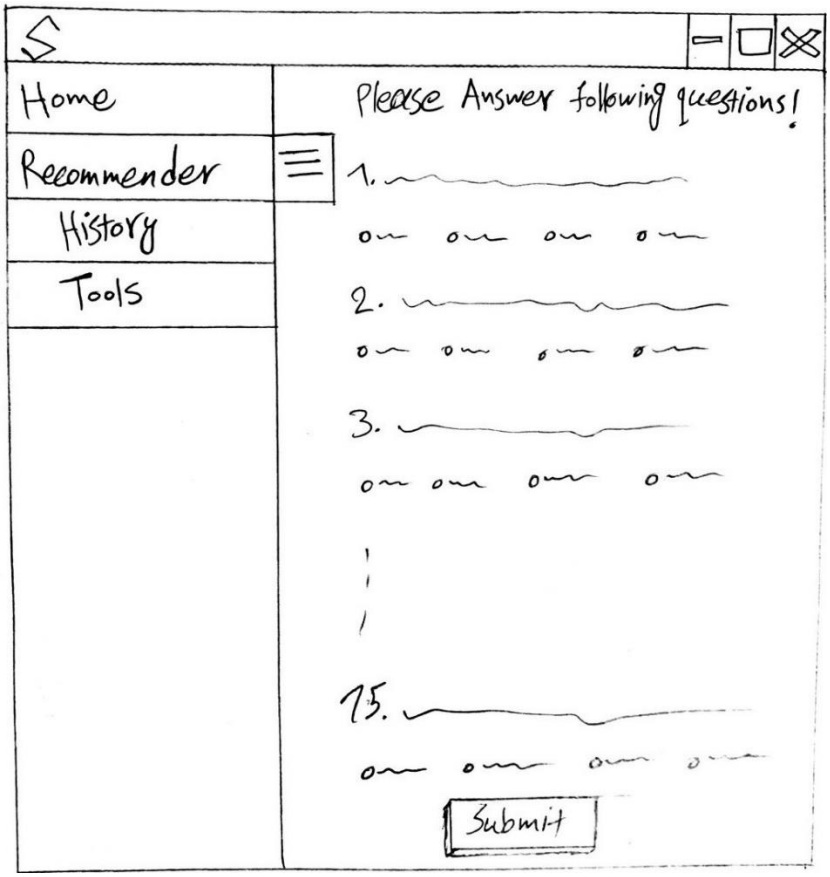

Abbildung 9: Paper-Prototyp der Recommender-Szene

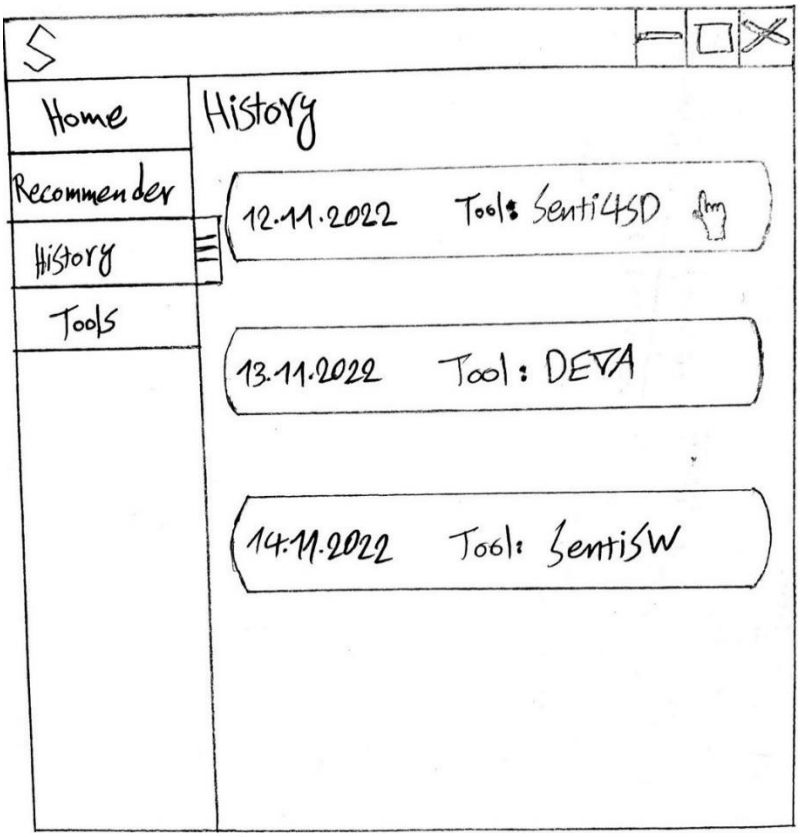

Abbildung 10: Paper-Prototyp der History-Szene

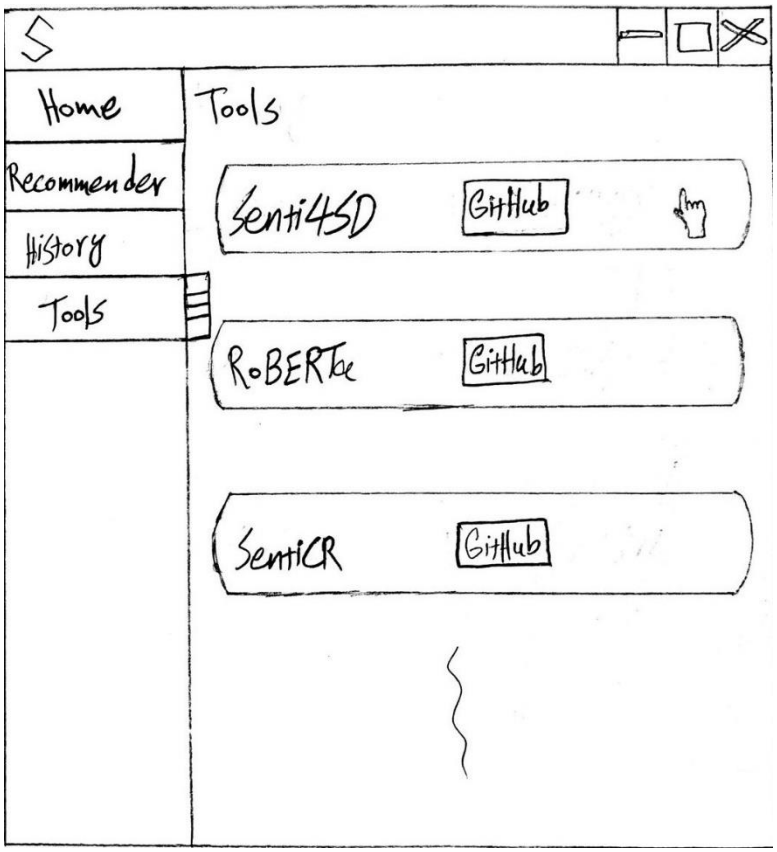

Abbildung 11: Paper-Prototyp der Tools-Szene

Aufgaben bei der Evaluation der Anwendung

Du bist ein Mitglied eines Software-Entwicklerteams und kommunizierst mit anderen Teammitgliedern. Das Team entscheidet sich, anonym Stimmungen der Teammitglieder zu messen, um im Falle der negativen Stimmung im Team gegensteuern zu können. Leider funktionieren die meisten der vorhandenen Tools mit Machine-Learning-Algorithmen, die nur mit einem bestimmten Datensatz für Training gut performen.
Dir wird eine Anwendung gegeben, welche dir verspricht, dir ein geeignetes Stimmungsanalysetool basierend auf deinen Antworten auf bestimmte Fragen vorzuschlagen.

1. Zuerst willst du wissen, was diese Anwendung macht und wie sie es macht. Versuche das mit Hilfe der Anwendung herauszufinden.

2. Danach willst du sehen, welche Stimmungsanalysetools diese Anwendung im Allgemeinen den Nutzenden vorschlägt. Versuche das mit Hilfe der Anwendung herauszufinden.

3. Jetzt willst du einen Vorschlag für ein Stimmungsanalysetool von der Anwendung erhalten und eine Übersicht dieses Vorschlags auf Desktop speichern und zu schauen. Benutze bitte die Anwendung, um dieses Ziel zu erreichen. Statistiken zu

den Kommunikationen deines Teams findest du unter “Desktop/sample 1.csv” und “Desktop/sample 2.csv”. Versuche auch diese bei dieser Aufgabe zu benutzen.

4. Versuche nun, die für deine Kommunikationen geeigneten Datensätze in der Anwendung zu finden.

Gerne kannst du auch versuchen, die weiteren Fähigkeiten der Anwendung während der Bearbeitung dieser Aufgaben zu testen und zu sehen, wie sich die Anwendung in unerwünschten Situationen verhält. 🙂

Vielen Dank!

Literatur

[1] M. Obaidi und J. Klünder, “Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review,” in *Evaluation and Assessment in Software Engineering*, Trondheim Norway, R. Chitchyan, J. Li, B. Weber und T. Yue, Hg., 2021, S. 80–89, doi: 10.1145/3463274.3463328.

[2] D. Graziotin, X. Wang und P. Abrahamsson, “Happy software developers solve problems better: psychological measurements in empirical software engineering,” *PeerJ*, Early Access. doi: 10.7717/peerj.289.

[3] L. Villarroel, G. Bavota, B. Russo, R. Oliveto und M. Di Penta, “Release planning of mobile apps based on user reviews,” in *Proceedings of the 38th International Conference on Software Engineering*, Austin Texas, L. Dillon, W. Visser und L. Williams, Hg., 2016, S. 14–24, doi: 10.1145/2884781.2884818.

[4] Munir Ahmad, Shabib Aftab, Syed Shah Muhammad und Sarfraz Ahmad, *Machine Learning Techniques for Sentiment Analysis: A Review* (8), 2017. [Online]. Verfügbar unter: https://www.researchgate.net/profile/shabib-aftab-2/publication/317284281_machine_learning_techniques_for_sentiment_analysis_a_review

[5] Atlassian. “Jira Overview.” https://www.atlassian.com/software/jira/guides/getting-started/overview (Zugriff am: 2. Okt. 2022).

[6] Stack Overflow. “About Stack Overflow.” https://stackoverflow.co/ (Zugriff am: 2. Okt. 2022).

[7] A. Bosu und J. C. Carver, “Impact of Peer Code Review on Peer Impression Formation: A Survey,” in *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Baltimore, Maryland, 2013, S. 133–142, doi: 10.1109/ESEM.2013.23.

[8] F. Calefato, F. Lanubile, F. Maiorano und N. Novielli, “Sentiment Polarity Detection for Software Development,” *Empir Software Eng*, Jg. 23, Nr. 3, S. 1352–1382, 2018. doi: 10.1007/s10664-017-9546-9. [Online]. Verfügbar unter: https://link.springer.com/content/pdf/10.1007/s10664-017-9546-9.pdf

[9] M. R. Islam und M. F. Zibran, “SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text,” *Journal of Systems and Software*, Jg. 145, S. 125–146, 2018, doi: 10.1016/j.jss.2018.08.030.

[10] N. Novielli, D. Girardi und F. Lanubile, “A benchmark study on sentiment analysis for software engineering research,” in *Proceedings of the 15th International Conference on Mining Software Repositories*, Gothenburg Sweden, A. Zaidman, Y. Kamei und E. Hill, Hg., 2018, S. 364–375, doi: 10.1145/3196398.3196403.

[11] M. Thelwall, K. Buckley, G. Paltoglou, Di Cai und A. Kappas, “Sentiment strength detection in short informal text,” *J. Am. Soc. Inf. Sci.*, Jg. 61, Nr. 12, S. 2544–2558, 2010, doi: 10.1002/asi.21416.

[12] T. Ahmed, A. Bosu, A. Iqbal und S. Rahimi, “SentiCR: A customized sentiment analysis tool for code review interactions,” in *ASE'17: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering : October 30-November 3, 2017, Urbana-Champaign, IL, USA*, Urbana, IL, G. Rosu, M. Di Penta und T. N. Nguyen, Hg., 2017, S. 106–111, doi: 10.1109/ASE.2017.8115623.

[13] GitHub. “GitHub Introduction.” https://docs.github.com/en/get-started/quickstart/hello-world (Zugriff am: 4. Okt. 2022).

[14] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo und L. Jiang, “Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?,” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Adelaide, Australia, 2020, S. 70–80, doi: 10.1109/ICSME46990.2020.00017.

[15] J. Ding, H. Sun, X. Wang und X. Liu, “Entity-level sentiment analysis of issue comments,” in *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, Gothenburg Sweden, A. Begel, A. Serebrenik und D. Graziotin, Hg., 2018, S. 7–13, doi: 10.1145/3194932.3194935.

[16] M. R. Islam und M. F. Zibran, “DEVA: sensing emotions in the valence arousal space in software engineering text,” in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau France, H. M. Haddad, R. L. Wainwright und R. Chbeir, Hg., 2018, S. 1536–1543, doi: 10.1145/3167132.3167296.

[17] M. Herrmann und J. Klünder, “From Textual to Verbal Communication: Towards Applying Sentiment Analysis to a Software Project Meeting,” 2021, doi: 10.48550/arXiv.2108.01985.

[18] M. Herrmann, M. Obaidi und J. Klünder, “SEnti-Analyzer: Joint Sentiment Analysis For Text-Based and Verbal Communication in Software Projects,” 2022, doi: 10.48550/arXiv.2206.10993.

[19] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 2019, doi: 10.48550/arXiv.1907.11692.

[20] B. Liu, “Sentiment Analysis and Subjectivity,” in *Handbook of Natural Language Processing*, N. Indurkhya und F. J. Damerau, Hg., Chapman and Hall/CRC, 2010, S. 627–666.

[21] N. Novielli, F. Calefato und F. Lanubile, “A Gold Standard for Emotion Annotation in Stack Overflow,” 2018, doi: 10.48550/arXiv.1803.02300.

[22] M. Ortu *et al.*, “The emotional side of software developers in JIRA,” in *Proceedings of the 13th International Conference on Mining Software Repositories*, Austin Texas, M. Kim, R. Robbes und C. Bird, Hg., 2016, S. 480–483, doi: 10.1145/2901739.2903505.

[23] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi und F. Lanubile, “Can We Use SE-specific Sentiment Analysis Tools in a Cross-Platform Setting?,” in *Proceedings of the 17th International Conference on Mining Software*

Repositories, Seoul Republic of Korea, 2020, S. 158–168, doi: 10.1145/3379597.3387446.

[24] R. Feldman, “Techniques and applications for sentiment analysis,” *Commun. ACM*, Jg. 56, Nr. 4, S. 82–89, 2013, doi: 10.1145/2436256.2436274.

[25] W. Medhat, A. Hassan und H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Engineering Journal*, Jg. 5, Nr. 4, S. 1093–1113, 2014, doi: 10.1016/j.asej.2014.04.011.

[26] A. Valdez, H. Oktaba, H. Gomez und A. Vizcaino, “Sentiment Analysis in Jira Software Repositories,” in *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Chetumal, Mexico, 2020, S. 254–259, doi: 10.1109/CONISOFT50191.2020.00043.

[27] K. Kenyon-Dean *et al.*, “Sentiment Analysis: It’s Complicated!,” in *Proceedings of the 2018 Conference of the North American Chapter of*, New Orleans, Louisiana, M. Walker, H. Ji und A. Stent, Hg., 2018, S. 1886–1895, doi: 10.18653/v1/N18-1171.

[28] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard und D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Stroudsburg, PA, USA, 2014, doi: 10.3115/v1/p14-5010.

[29] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza und R. Oliveto, “Sentiment analysis for software engineering,” in *Proceedings of the 40th International Conference on Software Engineering*, Gothenburg Sweden, I. Crnkovic, Hg., 2018, S. 94–104, doi: 10.1145/3180155.3180195.

[30] P. Shaver, J. Schwartz, D. Kirson und C. O'Connor, “Emotion knowledge: Further exploration of a prototype approach,” *Journal of Personality and Social Psychology*, Jg. 52, Nr. 6, S. 1061–1086, 1987, doi: 10.1037/0022-3514.52.6.1061.

[31] C.-J. Lin. “LIBLINEAR Introduction.” https://www.csie.ntu.edu.tw/~cjlin/liblinear/ (Zugriff am: 4. Okt. 2022).

[32] The R Foundation. “Introduction to R.” https://www.r-project.org/about.html (Zugriff am: 4. Okt. 2022).

[33] M. V. Mantyla, N. Novielli, F. Lanubile, M. Claes und M. Kuutila, “Bootstrapping a Lexicon for Emotional Arousal in Software Engineering,” in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, Buenos Aires, 2017, S. 198–202, doi: 10.1109/MSR.2017.47.

[34] A. B. Warriner, V. Kuperman und M. Brysbaert, “Norms of valence, arousal, and dominance for 13,915 English lemmas,” *Behav Res*, Jg. 45, Nr. 4, S. 1191–1207, 2013. doi: 10.3758/s13428-012-0314-x. [Online]. Verfügbar unter: https://link.springer.com/article/10.3758/s13428-012-0314-x

[35] S. Loria. “TextBlob: Simplified Text Processing.” https://textblob.readthedocs.io/en/dev/ (Zugriff am: 24. Sep. 2022).

[36] O. Guhr, A.-K. Schumann, F. Bahrmann, and H. J. Böhme., “Training a Broad-Coverage German Sentiment Classification Model for Dialog Systems,” in *LREC 2020 Marseille: Twelfth International Conference on Language Resources and Evaluation*, N. Calzolari, Hg., Paris: The European Language Resources Association (ELRA), 2020, S. 1627–1632.

[37] K. Tymann, M. Lutz, P. Palsbröker, C. Gips, Hg. *GerVADER - A German Adaptation of the VADER Sentiment Analysis Tool for Social Media Texts*, 2019.

[38] H. Priker. “SentiStrength-DE: A collection of German lexicon files to be used for sentiment classification with SentiStrength.” https://www.ofai.at/resources/sentistrength_de (Zugriff am: 24. Sep. 2022).

[39] M. Killer. “textblob-de: the german language extension for textblob.” https://textblob-de.readthedocs.io/en/latest/ (Zugriff am: 24. Sep. 2022).

[40] J. Devlin, M.-W. Chang, K. Lee und K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” 2018, doi: 10.48550/arXiv.1810.04805.

[41] T. Winters, T. Manshreck und H. Wright, *Software Engineering at Google*, 1. Aufl. Erscheinungsort nicht ermittelbar, Boston, MA: Upfront Books; Safari, 2021. [Online]. Verfügbar unter: https://learning.oreilly.com/library/view/-/1492082791/?ar

[42] GitHub. “About Pull Requests.” https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests (Zugriff am: 2. Okt. 2022).

[43] N. Imtiaz, J. Middleton, P. Girouard und E. Murphy-Hill, “Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people,” in *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, Gothenburg Sweden, A. Begel, A. Serebrenik und D. Graziotin, Hg., 2018, S. 55–61, doi: 10.1145/3194932.3194938.

[44] F. Calefato, F. Lanubile, N. Novielli und L. Quaranta, “EMTk - The Emotion Mining Toolkit,” in *2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, 2019, doi: 10.1109/semotion.2019.00014.

[45] S. Mohammad, “A Practical Guide to Sentiment Annotation: Challenges and Solutions,” in *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, San Diego, California, A. Balahur, E. van der Goot, P. Vossen und A. Montoyo, Hg., 2016, S. 174–179, doi: 10.18653/v1/W16-0429.

[46] W. G. Parrott, Hg. *Emotions in social psychology: Essential readings* (Key readings in social psychology). Philadelphia, Pa.: Psychology Press, 2001. [Online]. Verfügbar unter: http://www.loc.gov/catdir/enhancements/fy0652/00042544-d.html

[47] M. Grandini, E. Bagli und G. Visani, *Metrics for Multi-Class Classification: an Overview*. arXiv, 2020.

[48] J. Horstmann, *Computer-gestützte Analyse des Kommunikationsverhaltens in Entwicklerteams unter Berücksichtigung digitaler Medien, Masterarbeit*, 2019.
[49] A. Specht, *Identifikation von relevanten Metriken zur Analyse von Kommunikation in Entwicklerteams, Masterarbeit*, 2021.
[50] M. Claes, M. Mäntylä und U. Farooq, "On the use of emoticons in open source software development," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, Oulu Finland, M. Oivo, D. Méndez und A. Mockus, Hg., 2018, S. 1–4, doi: 10.1145/3239235.3267434.
[51] pyspellchecker. "pyspellchecker Documentation." https://pyspellchecker.readthedocs.io/en/latest/quickstart.html (Zugriff am: 11. Okt. 2022).
[52] pandas. "pandas API Reference." https://pandas.pydata.org/docs/reference/index.html (Zugriff am: 11. Okt. 2022).
[53] NumPy. "NumPy API Reference." https://numpy.org/doc/stable/reference/index.html#reference (Zugriff am: 11. Okt. 2022).
[54] H. Wang und J. A. Castanon, "Sentiment expression via emoticons on social media," in *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA, 2015, S. 2404–2408, doi: 10.1109/BigData.2015.7364034.
[55] Oracle. "About Java." https://www.oracle.com/java/ (Zugriff am: 24. Sep. 2022).
[56] JavaFX. "JavaFX Documentation." https://openjfx.io/openjfx-docs/ (Zugriff am: 24. Sep. 2022).
[57] w3schools. "Introduction to XML." https://www.w3schools.com/xml/xml_whatis.asp (Zugriff am: 30. Sep. 2022).
[58] Oracle. "Introduction to FXML." https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html#overview (Zugriff am: 30. Sep. 2022).
[59] J. Wu, C. Ye und H. Zhou, "BERT for Sentiment Classification in Software Engineering," in *2021 International Conference on Service Science (ICSS)*, Xi'an, China, 2021, S. 115–121, doi: 10.1109/ICSS53362.2021.00026.
[60] E. Loper und S. Bird, "NLTK: The Natural Language Toolkit," 2002, doi: 10.48550/arXiv.cs/0205028.
[61] M. Ortu, G. Destefanis, B. Adams, A. Murgia, M. Marchesi und R. Tonelli, "The JIRA Repository Dataset," in *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering*, Beijing China, A. Bener, L. Minku und B. Turhan, Hg., 2015, S. 1–4, doi: 10.1145/2810146.2810147.
[62] G. Uddin und F. Khomh, "Automatic Mining of Opinions Expressed About APIs in Stack Overflow," *IIEEE Trans. Software Eng.*, Jg. 47, Nr. 3, S. 522–559, 2021, doi: 10.1109/TSE.2019.2900245.
[63] E. Guzman, D. Azócar und Y. Li, "Sentiment analysis of commit comments in GitHub: an empirical study," in *Proceedings of the 11th Working Conference on*

Mining Software Repositories - MSR 2014, Hyderabad, India, P. Devanbu, S. Kim und M. Pinzger, Hg., 2014, S. 352–355, doi: 10.1145/2597073.2597118.

[64] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov und Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," 2019, doi: 10.48550/arXiv.1906.08237.

[65] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma und R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," 2019, doi: 10.48550/arXiv.1909.11942.

[66] M. R. Islam und M. F. Zibran, "A comparison of software engineering domain specific sentiment analysis tools," in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Campobasso, 2018, S. 487–491, doi: 10.1109/SANER.2018.8330245.

[67] F. Calefato, F. Lanubile und N. Novielli, "EmoTxt: A toolkit for emotion recognition from text," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, San Antonio, TX, 2017, S. 79–80, doi: 10.1109/ACIIW.2017.8272591.

[68] *Recursive deep models for semantic compositionality over a sentiment treebank*, 2013. [Online]. Verfügbar unter: https://aclanthology.org/d13-1170.pdf

[69] B. Lin, N. Cassee, A. Serebrenik, G. Bavota, N. Novielli und M. Lanza, "Opinion Mining for Software Development: A Systematic Literature Review," *ACM Trans. Softw. Eng. Methodol.*, Jg. 31, Nr. 3, S. 1–41, 2022, doi: 10.1145/3490388.

[70] A. Murgia, P. Tourani, B. Adams und M. Ortu, "Do developers feel emotions? an exploratory analysis of emotions in software artifacts," in *Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014*, Hyderabad, India, P. Devanbu, S. Kim und M. Pinzger, Hg., 2014, S. 262–271, doi: 10.1145/2597073.2597086.

[71] "RoBERTa in fairseq." https://github.com/facebookresearch/fairseq/tree/main/examples/roberta (Zugriff am: 15. Nov. 2022).

[72] scikit-learn. "Scikit-Learn Machine Learning in Python." https://scikit-learn.org/stable/ (Zugriff am: 16. Dez. 2022).

[73] Statcounter GlobalStats. "Desktop Operating System Market Share Worldwide." https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202204-202204-bar (Zugriff am: 28. Nov. 2022).

[74] PDFBox. "PDFBox Getting Started." https://pdfbox.apache.org/2.0/getting-started.html (Zugriff am: 28. Nov. 2022).

[75] iso.org. "ISO 9241-110 Grundsätze der Dialoggestaltung." https://www.iso.org/obp/ui/#iso:std:iso:9241:-110:ed-2:v1:en (Zugriff am: 10. Dez. 2022).

[76] C. Wohlin P. Runeson M. Höst M. C. Ohlsson B. Regnell A. Wesslén, *Experimentation in software engineering*. Berlin, Heidelberg: Springer, 2012.