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

Analyse von Intrarater-Übereinstimmungen bei der Sentimentzuweisung in Softwareprojekten

Bachelorarbeit

im Studiengang Informatik

von

Timo Kubera

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

Hannover, 24.02.2023

Kurzfassung

Die Stimmung in Entwicklerteams ist ein wichtiger Faktor, der über Projekterfolg und -misserfolg und die Produktivität im Team entscheiden kann. Die Sentimentanalyse bietet zahlreiche Verfahren, um aus Textdaten automatisiert die Stimmung der Person beim Verfassen der Nachricht zu analysieren. Somit wird beispielsweise dem Management im Unternehmen ein Instrument gegeben, um bei schlechten Stimmungen im Team diese zu erkennen.
Grundsätzlich wird in der Sentimentanalyse zwischen lexikonbasierten Ansätzen und Ansätzen, die auf Modellen des Maschinellen Lernens beruhen, unterschieden. Letztzere verwenden typischerweise einen Supervised-Learning-Ansatz, bei dem die Modelle mit gelabelten Datensätzen trainiert werden.
Das Labeln der Datensätze erfolgt häufig manuell, mit menschlichen Bewertern, und kann im Ad-Hoc-Verfahren geschehen, oder indem bestimmte Guidelines befolgt werden. Vorherige Arbeiten haben gezeigt, dass die subjektive Einschätzung der Bewerter, trotz befolgter Guidelines, einen Faktor beim Labeln der Datensätze ausmacht.
In dem Zusammenhang kann es sinnvoll sein die Intrarater-Reliabilität zu berechnen. Vereinfacht ausgedrückt stellt diese ein Maß dar, welches aussagt wie wahrscheinlich es ist, dass ein Bewerter zu unterschiedlichen Bewertungszeitpunkten, ähnliche Bewertungen vornimmt.
Ziel der Bachelorarbeit ist es eine Software zu entwickeln, mit der Intrarater-Reliabilitätsuntersuchungen vorgenommen werden können und Datensätze gelabelt werden können. Darüber hinaus wird die Software im Rahmen der Bachelorarbeit eingesetzt, um Intrarater-Reliabilitätsuntersuchungen von zwei Datensätzen vorzunehmen, die aus zwei vorausgegangen Arbeiten hervorgegangen sind.

Abstract

The mood in development teams is an important factor that can determine about success or failure of projects and productivity in the team. Sentiment analysis offers numerous methods to automatically analyze a person's mood while writing a message. Thus, for example, the management in the company is given a tool to recognize bad moods in the team.
In sentiment analysis, a distinction is made between lexicon-based approaches and approaches that are based on machine learning models. The latter typically use a supervised learning approach, where the models are trained with labeled datasets.
The labeling is often done manually, with human raters, and can be done on an ad hoc basis or according to specific guidelines. Previous work has shown that despite compliance with the guidelines, the judges' subjective judgment plays a role in labelling. In this context, it can be useful to calculate the intrarater reliability. Put simply, it is a measure of how likely it is that a rater will make similar judgements at different times of observation.
The aim of the bachelor thesis is to develop a software that can be used to perform intrarater reliability studies and to label data sets. In addition, the software is used to perform intrarater reliability studies of two data sets that have emerged from previous work.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Stimmungen und Emotionen sind untrennbare Bestandteile des Menschen und können diesen in seinen Handlungen beeinflußen [10].
Auch in der Domäne der Softwareentwicklung ist die Stimmungsanalyse ein etabliertes Forschungsfeld [36], [25], [35].
Es ließ sich feststellen, dass die Stimmung von Entwicklern einen Einfluß auf deren Produktivität hat [18], [24] und bestimmte Verhaltensweisen eines Teammitgliedes die Stimmung des gesamten Teams beeinflußen kann [41].
So konnte nachgewiesen werden, dass dysfunktionale Kommunikation, wie z.B. Jammern, innerhalb von Meetings negativ mit der Meeting-Zufriedenheit aller Teilnehmenden, sowie dem organisatorischen Erfolg des Projekts korreliert [26].

Zudem steigt in Softwareprojekten die Komplexität der Software [36], sowie die logistische Komplexität, weil Entwicklerteams häufig global verteilt sind und somit zusätzlicher Planungsaufwand ensteht, um die Kommunikation der Teammitglieder zu organisieren [13].
Vor diesem Hintergrund ist es wenig überraschend, dass herkömmliche Methoden um die Stimmung von Entwicklern zu messen, wie Interviews, zu unbefriedigenden Ergebnissen geführt haben [9], [19].
Zum Einen werden die Entwickler durch die Interviewdurchführung in ihrem herkömmlichen Arbeitsablauf gestört und zum Anderen erschwert die globale Verteilung der Entwickler, beispielsweise in Open-Source-Projekten, das Messen der Stimmungen mit herkömmlichen Methoden.

Abhilfe schaffen Stimmungsanalyse-Tools (SA-Tools) wie SentiStrength-SE [25], oder SentiCR [1] mit denen es möglich ist mit weniger Aufwand die Stimmung von Entwicklern automatisiert zu messen.

1.1 Motivation

SA-Tools sind also ein Instrument, um die Stimmung von Entwicklern im Team zu messen und können für das Management von Bedeutung sein, um gegebenenfalls Maßnahmen zu ergreifen.

Im Wesentlichen wird zwischen lexikonbasierten Ansätzen und solchen, die auf Ansätzen des Supervised-Machine-Learnings beruhen, unterschieden [34].
Ein Beispiel für ein SA-Tool das einen lexikonbasierten Ansatz verwendet ist SentiStrength-SE [25] und SentiCR [1] basiert auf einem Supervised-Machine-Learning-Ansatz.
Obaidi et al. haben eine systematische Literaturrecherche unternommen unter Anderem, um herauszufinden welche Ansätze bei der Entwicklung von SA-Tools verwendet werden [37].
Dabei wurden insgesamt 92 themenrelevante Paper untersucht und die Autoren kamen zu dem Ergebnis, dass über 90% der Ansätze auf Verfahren des Maschinellen Lernens beruhen, während weniger als 10% auf lexikonbasierten Verfahren beruhen.
Der vermehrte Einsatz von Verfahren des Maschinellen Lernens lässt sich durch deren bessere Performance in vielen Metriken, darunter F-Measures, Precision, Recall und Accuracy begründen [35], [34], [4].

Damit die SA-Tools, die auf einem Ansatz des Maschinellen Lernens basieren, sinnvoll eingesetzt werden können, müssen sie mit gelabelten Datensätzen trainiert werden.
Das Setzen der Label erfolgt häufig manuell von menschlichen Bewertern und es lässt sich zwischen ad hoc annotierten Datensätzen unterscheiden und solchen bei denen die Labelsetzung gewissten Richtlinien folgte [35].
Eine beliebte Grundlage für solche Richtlinien bietet das Emotionsmodell von Shaver et al. [42].
Es lässt sich feststellen, dass die SA-Tools in der Regel bessere Ergebnisse liefern, wenn die Trainingsdatensätze nicht im ad hoc Verfahren gelabelt worden sind, sondern wenn die genannten Richtlinien umgesetzt worden sind [35].
Doch selbst beim Umsetzen von Richtlinien bleibt die Subjektvität der Bewerter ein entscheidender Faktor bei der Vergabe der Label [23].
So können nicht nur unterschiedliche Bewerter zu unterschiedlichen Einschätzungen bei der Vergabe der Label kommen, sondern auch der selbe Bewerter kann zu unterschiedlichen Bewertungszeitpunkten zu unterschiedlichen Ergebnissen kommen.
In diesen Zusammenhängen wird von der Inter-, bzw. der Intrarater-Reliabilität gesprochen.

Wärhend es in der Literatur bereits relativ viele Untersuchungen hinsichtlich der Interrater-Reliabilität gibt, gibt es relativ wenige Untersuchungen hinsichtlich der Intrarater-Reliabilität [22].
Da auch die Intrarater-Reliabilität eine wichtige Rolle für die Reproduzierbarkeit bei der Erstellung der Datensätze spielt [22], soll diese Bachelorarbeit einen Beitrag dazu leisten den Sachverhalt näher zu beleuchten.

1.2 Zielsetzung

Ziel der Bachelorarbeit ist die Software IIRA (**I**ntra and **I**nter **R**eliability **A**nalyses) zu entwickeln mit der die Intra- und Interrater-Reliabilität von gelabelten Datensätzen gemessen werden kann. Neben der Auswahl der Bewerter, bietet IIRA eine Auswahl an unterschiedlichen Metriken an, mit denen die Messung vorgenommen wird. Die Label der Datensätze können in einem beliebigen Skalenformat (nominal, ordinal, intervall, oder rational) vorliegen.
Darüber hinaus soll es möglich sein in der Software unterschiedliche Nutzerprofile anzulegen, um einen importierten Datensatz selber labeln zu können.
Anschließend sollen mithilfe der Software auf der Basis von zwei Datensätzen, die im Kapitel 5.1 vorgestellt werden, Intrarater-Reliabilitätsuntersuchungen vorgenommen werden.

1.3 Struktur der Arbeit

Die Bachelorarbeit besteht aus insgesamt sechs Kapiteln.
Zu Beginn wird auf Grundlagen der Sentimentanalyse und von Reliabilitätsmaßen eingegangen, sowie eine Definition von Skalenformaten präsentiert. Anschließend werden verwandte Arbeiten vorgestellt. Darunter sind zwei Arbeiten, aus denen die Datensätze hervorgehen, auf denen die Intrarater-Reliabilitätsuntersuchungen, am Ende der Bachelorarbeit, beruhen.
Das Kapitel Entwicklung der Software umfasst die Planung, die Implementierung, sowie das systematische Testen der Software.
Im Kapitel 5 werden die Datensätze, für die Intrarater-Reliabilitätsuntersuchungen, vorgestellt und die Ergebnisse der Untersuchungen formuliert.
Abschließend werden die wichtigsten Ergebnisse der Bachelorarbeit zusammengefasst und Ausblicke für mögliche Fortsetzungen der Arbeit aufgezeigt.

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen beschrieben, die für das weitere Verständnis der Bachelorarbeit von Bedeutung sind. Im ersten Unterkapitel wird die Herangehensweise bei der Sentimentanalyse im Software Engineering erklärt. Die weiteren Unterkapitel befassen sich mit Skalenformaten und mit Reliabilitätsmaßen, mit denen Intra-, bzw. Interrater-Reliabilitäten gemessen werden können.

2.1 Sentimentanalyse

Die Sentimentanalyse, als Teilbereich der Informatik, besteht grundsätzlich aus der Aufgabe Stimmungen von Autoren herauszufinden, die sie beim Verfassen von Text ausgedrückt haben. Pang und Lee unterscheiden dabei zwischen der Subjektivität und der Polarität von Text [38].
Die Subjektivität drückt aus, dass durch den Text Meinungen oder Emotionen übermittelt werden, die nicht sinnvoll objektiv verifizierbar sind. So kann eine Person in einer Produktrezension davon schreiben, dass ihr eine gelbe Jacke wegen der auffälligen Farbe gefällt und eine andere Person kann die Jacke aus dem gleichen Grund negativ bewerten.
Die Polarität beschreibt, ob im Text positive oder negative Stimmungen ausgedrückt worden sind. Häufig gibt es in der Stimmungsanalyse noch eine neutrale Polarität die dann zugeordnet wird, wenn sich aus dem Text weder eine positive-, noch eine negative Polarität ableiten lässt [3].
Im weiteren Verlauf der Bachelorarbeit wird von diesen drei Polaritätsklassen im Zusammenhang mit der Sentimentanalyse ausgegangen.

In der Sentimentanalyse werden SA-Tools eingesetzt, um automatisiert die Stimmungen von Autoren zu ermitteln, die sie beim Verfassen des Textes ausgedrückt haben. Ein Beispieltext der bewertet werden kann ist der folgende Satz:

„Thank you, that was really helpful!“

Das SA-Tool Sentistrength [44] würde dem Text die positive Polaritätsklasse zuordnen, da sowohl das Wort „Thank“, als auch das Wort „helpful“, als positive Indikatoren bewertet werden und in dem Satz keine Indikatoren für die negative Polaritätsklasse gefunden worden sind [3].

Neben der Sentimentanalyse auf Satzebene können die Dokumentebene und die Aspektebene betrachtet werden [12]. Bei der Sentimentanalyse auf Dokumentebene wird beispielsweise einem Social-Media-Eintrag, der aus mehreren Sätzen besteht, oder einem Blogeintrag im Internet eine Polaritätsklasse zugeordnet. Dabei wird vereinfachend davon ausgegangen, dass sich das gesamte Dokument einer Polaritätsklasse zuordnen lässt, je nachdem ob positive oder negative Indikatoren überwiegen [12].
Die Sentimentanalyse auf Aspektebene kommt beispielsweise bei Produktrezension zum Einsatz. Hier ist es sinvoll einzelne Aspekte des zugrundeliegenden Textes zu extrahieren und diesen jeweils eine Polarität zuzuordnen. In einer Rezension von einem Smartphone könnte beispielsweise das Display als positiv, aber die Akkulaufzeit als zu gering und damit als negativ bewertet werden. Würden nun nicht die einzelnen Aspekte analysiert werden, sondern lediglich, ob im gesamten Review überwiegend positive, neutrale oder negative Stimmungen ausgedrückt werden, würden dabei Informationen verloren gehen, die unter Umständen von Bedeutung sind [12].

Es gibt unterschiedliche Möglichkeiten, wie dem Text eine Polarität zugeordnet werden kann. Ein Ansatz ist das lexikonbasierte Verfahren mittels eines SA-Tools. Tools, die das lexikonbasierte Verfahren verwenden, besitzen ein Sentiment-Lexikon, das aus Wörtern besteht, denen eine Polarität zugeordnet wurde [38]. Ein Beispiel für solch ein Tool ist SentiStrength [25]. In dem Lexikon von SentiStrength wird jedem negativ konnotierten Wort ein Wert zwischen -2 und -5 zugeordnet. Analog dazu wird jedem positiv konnotierten Wort ein Wert zwischen $+2$ und $+5$ zugeordnet und neutralen Wörtern werden die Werte ± 1 zugeordnet. Diesem Ansatz folgend, wird jedem Satz eine positive und eine negative Wertung zugeordnet. Die Gesamtwertungen für positive-, bzw. negative Bewertungen, entsprechen dem Maximum, bzw. dem Minimum, der Bewertungen aller Sätze des zu analysierenden Textes. Dabei werden Ausdrucksverstärkungen wie „sehr“ berücksichtigt, indem die entsprechende Wertung erhöht wird. Darüber hinaus werden Negierungen berücksichtigt,

indem dem negierten Wort die inverse Polaritätswertung zugeordnet wird. Zum Schluss wird vom Tool eine Gesamtpunktzahl ausgegeben, die ausdrückt, ob der analysierte Text insgesamt als positiv (+1), neutral (0), oder als negativ (-1) bewertet wird. Dabei wird die positive Gesamtwertung mit der negativen Gesamtwertung verglichen [3].
Andere Ansätze, die in SA-Tools zum Einsatz kommen können, sind Verfahren des maschinellen Lernens. Dabei werden die Tools typischerweise mit Daten trainiert, die neben dem Text, noch ein Label der Polaritätsklasse enthalten, die dem Text zugeorndet wurde [12].
Die einfachste Möglichkeit den Text zu labeln, stellt das ad hoc Verfahren dar. Dabei entscheidet ein Bewerter, auf der Basis seiner subjektiven Einschätzung, welche Polaritätsklasse er dem Text zuordnet [34]. Eine Alternative stellen Verfahren dar, bei denen der Text nach bestimmten Richtlinien einer der drei Klassen zugeordnet wird.
Lazarus unterscheidet beispielsweise zwischen neun negativen Emotionen (anger, fright, anxiety, guilt, shame, sadness, envy, jealousy, und disgust) und sieben positiven Emotionen (happiness, pride, relief, love, hope, compassion, und gratitude) [28]. Auf dieser Grundlage können Bewerter untersuchen, ob mindestens eine der insgesamt 16 Emotionen im Text vorkommen und dementsprechend den Text als positiv, oder negativ, bzw. als neutral labeln, falls keine der Emotionen vorkommt. Es ließ sich feststellen, dass die Abwesenheit von Richtlinien beim Labeln der Trainingsdatensätze einen negativen Einfluß auf die Leistung der SA-Tools hat [43], [34], [35].

2.1.1 Sentimentanalyse im Software Engineering

Wenn ein SA-Tool wie SentiStrength, das allgemeinen Text analysieren soll, in der Domäne des Software Engineering eingesetzt wird, kann es zu Problemen bei den Bewertungen kommen [25].
So gibt es domänenspezifische Wörter wie beispielsweise „Exception“, „Error“, oder „Fault“, denen bei der herkömmlichen Sentimentanalyse möglicherweise eine negative Polarität zugeordnet wird, die im Software Engineering allerdings als neutral zu werten sind. Die Erstellung von domänenspezifischen Wörterbüchern ist eine Praxis, um dem Problem entgegenzuwirken [39], [16].
Darüber hinaus hat sich gezeigt, dass SA-Tools, die auf einem Ansatz des maschinellen Lernens beruhen und plattformübergreifend eingesetzt werden, schlechtere Ergebnisse in vielen Metriken erzielen [34]. Ein Tool das mit Daten von Github trainiert wurde, würde beispielsweise schlechtere Ergebnisse erzielen, wenn es StackOverflow-Kommentare auswerten sollte. Daher sind neben den domänenspezifischen Anpassungen, auch Anpassungen an die jeweilige Plattform sinnvoll, auf der das Tool eingesetzt wird, um bessere Ergebnisse zu erzielen. Das kann erreicht werden, indem der Klassifizierer der Machine-Learning-Modelle mit Daten der Plattform, auf der das Tool eingesetzt wird, erneut trainiert wird [34].

2.2 Skalenformate

Daten können unterschiedlich strukturiert sein und in Abhängigkeit von der Struktur können einige mathematische Operationen auf den Daten sinnvoll sein und andere nicht. Beispielsweise lässt sich angeben, dass der Abstand vom 01.01.2023 zum 02.01.2023 genau so groß ist wie der Abstand vom 01.02.2023 zum 02.02.2023. Es macht aber keinen Sinn die einzelnen Datumsangaben miteinander zu multiplizieren oder zu dividieren.
Man kann zwischen den folgenden vier Skalen unterscheiden [40]. Auf jeder nachfolgenden Skala sind die Operationen der vorherigen Skalen zusätzlich erlaubt.

- **Nominalskala**: Die schwächste Skala erlaubt es nur Werte miteinander zu vergleichen. Es sind neben der Äquivalenzrelation allerdings keine weiteren mathematischen Operationen erlaubt und es ist keine Rangfolge der Elemente definiert.
  Beispielelemente liefert die Menge {„Rot“, „Apfel“, „Haus“}.

- **Ordinalskala**: Auf Ordinalskalen ist zusätzlich eine Reihenfolge der Elemente definiert. Beispielsweise lassen sich die Polaritätsklassen aus der Sentimentanalyse in dieser Reihenfolge anordnen:
  Negativ $<$ Neutral $<$ Positiv.
  Weitere mathematische Operationen sind allerdings nicht erlaubt. Terme wie 2 * Positiv = Negativ, oder Negativ + Neutral = Positiv sind schließlich nicht definiert.

- **Intervallskala**: Die Abstände bei Intervallskalen entsprechen den Abständen der Ausprägung des zu messenden Merkmals.
  Neben den eingangs erwähnten Datumsangaben sind Temperaturangaben in Celsius ein weiteres Beispiel für intervallskalierte Daten. So ist der Temperaturabstand zwischen 0°C und 1°C genau so groß, wie der Abstand zwischen 15°C und 16°C.
  Allerdings gibt es bei diesem Skalenformat keinen natürlichen Nullpunkt, weshalb keine Multiplikation oder Division mit den Daten erlaubt ist. Schließlich wäre die Aussage 10°C ist halb so warm wie 20°C eine falsche Aussage.

- **Rationalskala**: Bei Rationalskalen dürfen die Daten miteinander multipliziert und dividiert werden, weshalb auch Verhältnisse gebildet werden können.
  Ein Beispiel ist die Temperaturangabe in Kelvin. In dem Fall darf angegeben werden, dass 200 K doppelt so warm ist wie 100 K, weil ein natürlicher Nullpunkt gegeben ist, der nicht unterschritten werden kann.

2.3 Reliabilitätsmaße

In dem folgenden Unterkapitel werden ausgewählte Reliabilitätsmaße vorgestellt, mit denen es möglich ist Urteilerübereinstimmungen zu messen. Die vorgestellten Maße lassen sich in zwei Szenarien einsetzen. Das erste Szenario ist der Einsatz als Interrater-Reliabilitätsmaß. Dabei soll der Grad der Übereinstimmung zwischen zwei, oder mehreren, Bewertern gemessen werden. Im zweiten Szenario ist der Ansatz ähnlich, allerdings werden nicht Übereinstimmungen zwischen mehreren Bewertern gemessen, sondern das Maß an Übereinstimmung ein und desselben Bewerters, wenn er zu mehreren Zeitpunkten die gleichen Sachverhalte bewertet. Man spricht in diesem Kontext von der Intrarater-Reliabilität. Weitere Limitierungen der einzelnen Maße, sowie Besonderheiten, werden in den entsprechenden Unterkapiteln erläutert.

2.3.1 Cohen's Kappa

Cohen's Kappa ist ein Reliabilitätsmaß, das auf nominalen und ordinalen Skalen definiert ist und die Übereinstimmung von zwei unterschiedlichen Bewertern misst [6], [22]. Dabei können die unterschiedlichen Bewertungsobjekte unterschiedlichen Kategorien zugeordnet worden sein. Da es nur eine endliche Zahl an Kategorien gibt, für die sich die Bewerter entscheiden können, nimmt das Maß zusätzlich eine Korrektur hinsichtlich der Zufallsübereinstimmungswahrscheinlichkeit vor. Im Kontext der Sentimentanalyse, wo es nur drei Polaritätsklassen, oder Kategorien, gibt, ist die Wahrscheinlichkeit, dass zwei Bewerter zufälligerweise die gleiche Kategorie auswählen, schließlich nicht zu vernachlässigen.
Das Maß lässt sich als Intrarater-Reliabilitätsmaß einsetzen, indem statt der unterschiedlichen Bewerter zwei unterschiedliche Bewertungszeitpunkte betrachtet werden [22]. Da in der Bachelorarbeit der Fokus auf der Intrarater-Analyse liegt, werden die folgenden Beispiele unterschiedliche Bewertungszeitpunkte beinhalten.
In der Tabelle 2.1 wurde allgemein dargestellt wie ein Bewerter, zu zwei unterschiedlichen Bewertungszeitpunkten, m Bewertungsobjekte in q Kategorien einordnet, mit $m, q \in \mathbb{N}$.
Cohen's Kappa ist definiert als [6]:

$$\kappa = \frac{p_a - p_e}{1 - p_e} \tag{2.1}$$

Dabei ist

$$p_a = \frac{1}{m} \cdot \sum_{k=1}^{q} a_{kk} \tag{2.2}$$

die Gesamtübereinstimmungswahrscheinlichkeit. Die Wahrscheinlichkeit drückt aus wie wahrscheinlich es ist, dass der Bewerter zu beiden

Erster Bewertungszeitpunkt Kategorien	Zweiter Bewertungszeitpunkt Kategorien					
	1	...	k	...	q	$\sum$
1	a_{11}	$\cdots$	a_{1k}	$\cdots$	a_{1q}	$\sum_{i=1}^{q} a_{1i}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
k	a_{k1}	$\cdots$	a_{kk}	$\cdots$	a_{kq}	$\sum_{i=1}^{q} a_{ki}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
q	a_{q1}	$\cdots$	a_{qk}	$\cdots$	a_{qq}	$\sum_{i=1}^{q} a_{qi}$
$\sum$	$\sum_{i=1}^{q} a_{i1}$	$\cdots$	$\sum_{i=1}^{q} a_{ik}$	$\cdots$	$\sum_{i=1}^{q} a_{iq}$	m

Tabelle 2.1: Bewertung von m Objekten in Abhängigkeit vom Bewertungszeitpunkt und den ausgewählten Kategorien

Bewertungszeitpunkten die gleiche Kategorie ausgewählt hat.
Die Zufallsübereinstimmungswahrscheinlichkeit ist definiert als:

$$p_e = \frac{1}{m^2} \cdot \sum_{k=1}^{q} (\sum_{i=1}^{q} a_{ki} \cdot \sum_{i=1}^{q} a_{ik}) \tag{2.3}$$

Der Wertebereich von Cohen's Kappa ist $W_\kappa = [-1, 1]$ und die Ergebnisse lassen sich wie in Tabelle 2.2 dargestellt interpretieren [32].

Metrikwert	Grad der Übereinstimmung
<0	schlecht
0 - 0.2	geringfügig
0.2 - 0.4	mäßig
0.4 - 0.6	moderat
0.6 - 0.8	substanziell
0.8 - 1.0	fast perfekt

Tabelle 2.2: Interpretation der Metriken nach Landis und Koch

2.3.2 Fleiss' Kappa

Fleiss beschäftigte sich mit dem Problem, dass die Anwendung von Cohen's Kappa auf die Fälle limitiert ist, in denen zu genau zwei Bewertungszeitpunkten, bzw. von genau zwei Bewertern, gemessen wurde [14]. Das Reliabilitätsmaß Fleiss' Kappa hebt diese Limitierung auf und es ist ebenfalls auf nominalen und ordinalen Skalen definiert.
In der Tabelle 2.3 wurde dargestellt wie ein Bewerter zu n unterschiedlichen

Bewertungszeitpunkten m Bewertungsobjekte in q Kategorien einordnet.

Bewertungsobjekt	Kategorien					
	1	...	k	...	q	$\sum$
1	a_{11}	$\cdots$	a_{1k}	$\cdots$	a_{1q}	n
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	a_{i1}	$\cdots$	a_{ik}	$\cdots$	a_{iq}	n
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
m	a_{m1}	$\cdots$	a_{mk}	$\cdots$	a_{mq}	n
$\sum$	$\sum_{i=1}^{m} a_{i1}$	$\cdots$	$\sum_{i=1}^{m} a_{ik}$	$\cdots$	$\sum_{i=1}^{m} a_{iq}$	$m \cdot n$

Tabelle 2.3: Bewertung von mn Objekten in Abhängigkeit vom Bewertungsobjekt und der ausgewählten Kategorien

Die Definitionen von Fleiss' Kappa und der Gesamtübereinstimmungswahrscheinlichkeit sind identisch mit den Definitionen aus Gleichung 2.1, bzw. aus Gleichung 2.2 [22].
Lediglich die Definition der Zufallsübereinstimmungswahrscheinlichkeit ändert sich:

$$\begin{aligned} p_e &= \textstyle\sum_{k=1}^{q} p_k^2 \text{ , mit} \\ p_k &= \tfrac{1}{m \cdot n} \textstyle\sum_{i=1}^{m} a_{ik} \end{aligned} \tag{2.4}$$

Da sich Fleiss' Kappa für den Fall $n = 2$ nicht auf Cohen's Kappa reduzieren lässt, kann es sinnvol sein Cohen's Kappa für den Fall $n = 2$ Bewertungszeitpunkte, bzw. Bewerter, und Fleiss' Kappa für den Fall $n > 2$ Bewertungszeitpunkte, bzw. Bewerter, zu verwenden [33].
Der Wertebereich von Fleiss' Kappa ist gegeben durch $W_\kappa = [-1, 1]$ und die Ergebnisse lassen sich ebenfalls wie in Tabelle 2.2 dargestellt interpretieren [32].

2.3.3 Gwet's AC_1

Die beiden Kappa-Koeffizienten haben die Eigenschaft an sich, dass sie unter Umständen unangemessen niedrige Werte ergeben, obwohl die Daten erwarten ließen, dass ein hohes Maß an Übereinstimmung vorliegt. Die Phänomene sind in der Literatur als Kappa-Paradoxien bekannt [21], [11], [2].
Aus diesem Grund soll zusätzlich Gwet's AC_1-Reliabilitätsmaß vorgestellt werden, welches resistenter gegenüber den Paradoxien ist [21], [22].
Gwet's AC_1 ist auf nominalen und ordinalen Skalen definiert und lässt

sich berechnen wie in Gleichung 2.1 dargestellt. Auch die Gesamtübereinstimmungswahrscheinlichkeit ist äquivalent zu der in Gleichung 2.2. Die Zufallsübereinstimmungswahrscheinlichkeit basiert auf der Gleichung 2.4 mit den folgenden Unterschieden:

$$p_e = \frac{1}{q-1} \sum_{k=1}^{q} p_k \cdot (1 - p_k) \tag{2.5}$$

Die Interpretation der Ergebnisse erfolgt wieder wie in Tabelle 2.2 dargestellt [32], [21]. Dementsprechend ist auch der Wertebereich von Gwet's AC_1 gegeben durch $W_{AC_1} = [-1, 1]$.

2.3.4 Gewichtete Reliabilitätsmaße

Von den drei vorgestellten Reliablitätsmaßen existieren gewichtete Versionen, die eine Verallgemeinerung der vorgestellten Maße darstellen und deren Einsatz auf Ordinalskalen sinnvoller machen können [17], [21]. Man spricht in dem Zusammenhang vom gewichteten Cohen's-, bzw. vom gewichteten Fleiss'-Kappa und von Gwet's AC_2-Reliabilitätsmaß. Die Gewichte ermöglichen es Assoziationen zwischen den Kategorien herzustellen [17].
Beispielsweise kann man durch die Wahl von geeigneten Gewichten erreichen, dass die Auswahl der Kategorien „Positiv“ und „Neutral“, aus der Sentimentanalyse, in einem höheren Grad an Übereinstimmung resultieren, als die Auswahl der Kategorien „Positiv“ und „Negativ“.
Um die gewichteten Reliabilitätsmaße einsetzen zu können ist es erforderlich jedem Eintrag a_{ik}, mit $1 \leq i \leq q$, $1 \leq k \leq m$ aus Tabelle 2.3 ein Gewicht $w_{ik} \in [0, 1]$ zuzuordnen.
Da die Auswahl der Gewichte einen Einfluß auf das Ergebnis der Reliabilitätsanalyse haben kann und die Ergebnisse bei der Wahl unterschiedlicher Gewichte schwerer miteinander vergleichbar sind, sollte die Wahl der Gewichte gut überlegt sein [17]. Maclure und Willet [30] schlagen eine quadratische Gewichtung als die vernünftigste Wahl der Gewichte in den meisten Fällen vor. Die quadratische Gewichtung wurde auch von Cohen und Fleiss genutzt, um zu zeigen, dass das gewichtete Cohen's Kappa unter bestimmten Bedingungen asymptotisch äquivalent zum ICC [27] ist [15].
Bezugnehmend auf Tabelle 2.3 werden die quadratischen Gewichte wie folgt berechnet [17]:

$$w_{ik} = 1 - \frac{(i-k)^2}{(q-1)^2} \tag{2.6}$$

In der Tabelle 2.4 ist ein Beispiel für die Kategorisierung von 50 Bewertungsobjekten zu zwei unterschiedlichen Zeitpunkten zu sehen und in der Tabelle 2.5 wurde die dazugehörige Gewichtsmatrix dargestellt.

Erster Bewertungszeitpunkt	Zweiter B.z.				
	1	2	3	4	5
1	0	10	0	0	0
2	0	0	10	0	0
3	0	0	0	10	0
4	0	0	0	0	10
5	0	0	0	0	10

Tabelle 2.4: Kategorisierung von 50 Bewertungsobjekten in fünf Kategorien zu zwei Bewertungszeitpunkten

	1	2	3	4	5
1	1	$\frac{15}{16}$	$\frac{3}{4}$	$\frac{7}{16}$	0
2	$\frac{15}{16}$	1	$\frac{15}{16}$	$\frac{3}{4}$	$\frac{7}{16}$
3	$\frac{3}{4}$	$\frac{15}{16}$	1	$\frac{15}{16}$	$\frac{3}{4}$
4	$\frac{7}{16}$	$\frac{3}{4}$	$\frac{15}{16}$	1	$\frac{15}{16}$
5	0	$\frac{7}{16}$	$\frac{3}{4}$	$\frac{15}{16}$	1

Tabelle 2.5: Quadratische 5x5 Gewichtsmatrix

Die gewichteten Reliabilitätsmaße lassen sich auf eine ähnliche Weise berechnen wie in Gleichung 2.1 dargestellt, allerdings müssen zusätzlich die Gewichte berücksichtigt werden. Es ergibt sich die folgende Gleichung, wenn die Gewichte hinzugefügt werden [17], [21]:

$$\mu = \frac{p_{a_w} - p_{e_w}}{1 - p_{e_w}} \tag{2.7}$$

für $\mu \in \{\kappa_{cohen_w}, \kappa_{fleiss_w, AC_2}\}$.
Am Beispiel vom gewichteten Cohen's Kappa soll gezeigt werden, an welchen Stellen in der Gleichung die Gewichte relevant sind. Das Vorgehen bei den anderen Maßen ist analog. Für die Gesamtübereinstimmungswahrscheinlichkeit ergibt sich die Formel [17]:

$$p_{a_w} = \frac{1}{m} \cdot \sum_{i=1}^{q} \sum_{j=1}^{q} w_{ij} \cdot a_{ij} \tag{2.8}$$

und die Zufallsübereinstimmungswahrscheinlichkeit ist definiert als:

$$p_{e_w} = \frac{1}{m^2} \cdot \sum_{i=1}^{q} \sum_{j=1}^{q} w_{ij} \cdot \sum_{k=1}^{q} a_{ik} \cdot \sum_{k=1}^{q} a_{kj} \tag{2.9}$$

Es lässt sich feststellen, dass das ungewichtete Cohen's Kappa ein Sonderfall des gewichteten Cohen's Kappa ist, wenn die identitäre Gewichtsmatrix verwendet wird [17]. Dieser Zusammenhang gilt auch für Fleiss' Kappa und Gwet's AC [21]. Mithilfe der in Tabelle 2.6 dargestellten identitären Gewichtsmatrix soll dieser Zusammenhang verdeutlicht werden.

	1	2	3	4	5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Tabelle 2.6: Identitäre 5x5 Gewichtsmatrix

Es ist zu erkennen, dass durch die identitäre Gewichtsmatrix bei der Berechnung von p_{a_w} und p_{e_w} nur die Diagonaleinträge berücksichtigt werden, was den Definitionen von p_a, bzw. von p_e, entspricht.
In der Literatur wird darauf hingewiesen, dass der Einsatz von gewichteten Reliabilitätsmaßen im Allgemeinen eher den Grad an Assoziationen, statt den Grad an Übereinstimmungen misst [17].
In der Tabelle 2.4 ist ein Beispiel zu sehen, bei dem ein Bewerter zu zwei unterschiedlichen Bewertungszeitpunkten in nur 10 Fällen mit sich vollständig übereinstimmt, aber es gibt einen relativ hohen Grad an Assoziationen bei seinen Bewertungen. Diese Zusammenhänge sind auch in den entsprechenden Kappa Werten zu erkennen. So ergibt sich für das ungewichtete Cohen's Kappa $\kappa_{cohen} = 0$, während sich für das quadratisch gewichtete Kappa $\kappa_{cohen_w} = 0.8$ ergibt.

Kapitel 3

Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten aus den Bereichen der Sentimentanalyse und der Urteilerübereinstimmung vorgestellt, die die Grundlage für diese Bachelorarbeit bilden.

3.1 Sentimentanalyse

Pang und Lee haben im Jahr 2008 ein umfangreiches Paper zum Thema der Sentimentanalyse herausgebracht [38]. Dort wird von einem „land rush“ geschrieben, den das Forschungsfeld der Sentimentanalyse zu dem Zeitpunkt durchlebte. Dafür wurde der Einsatz vom maschinellen Lernen in den Themenbereichen des Information Retrievals und des Natural Language Processings verantwortlich gemacht, sowie die Verfügbarkeit von großen Datenmengen für das Training der Machine-Learning-Modelle, die aus dem Aufblühen des World Wide Web hervorgegangen sind.
Darüber hinaus wurden Schwierigkeiten in der Sentimentanalyse angesprochen, wie das Erkennen von Sarkasmus und Ironie, die zum Teil noch heutige Systeme und Forschende herausfordern [25]. Ferner wurden Möglichkeiten aufgezeigt, wie automatisiert Emotionen und Stimmungen ausgelesen werden können, die Autoren beim Verfassen von Text ausgedrückt haben. Dabei wurde sowohl auf lexikonbasierte Verfahren, als auch auf Verfahren des maschinellen Lernens eingegangen.
Ein bekanntes SA-Tool, das auf einem lexikonbasierten Verfahren beruht, ist das im Jahr 2010 von Thelwall et al. [45] publizierte Tool SentiStrength, welches bereits im Kapitel 2.1 kurz vorgestellt wurde.
Von Murgia et al. [33] wurde im Jahr 2014 ein Paper herausgebracht, das sich mit der automatisierten Sentimentanalyse im Software-Engineering beschäftigte. Es wurden fast 800 Issue-Kommentare analysiert und es ließ sich deren Annahme bestätigen, dass auch im Software-Engineering die Entwickler ihre Stimmung beim Verfassen von Issue-Kommentaren ausdrücken. Es wurde herausgefunden, dass die Emotionen „Liebe“, „Freude“

und „Traurigkeit“ relativ gut von SA-Tools in den Issue-Kommentaren herausgefunden werden konnten, während andere Emotionen mehr Probleme bereiteten [33]. Die Zusammenhänge wurden überprüft, indem die Übereinstimmungswerte zwischen den Ergebnissen der SA-Tools und den Bewertungen von menschlichen Bewertern miteinander verglichen worden sind [33].
Es folgten viele weitere Paper die sich mit der Sentimentanalyse im Software-Engineering beschäftigten. So wurde beispielsweise das Tool SentiCR [1] im Jahr 2017 vorgestellt, welches entwickelt wurde, um Stimmungen aus Code Reviews auslesen zu können. Oder auch das Tool SentiStrength-SE [25], das eine Weiterentwicklung des Tools SentiStrength [45] ist und unter Anderem um ein Lexika ergänzt wurde, welches Fachausdrücke aus dem Software Engineering beinhaltet.
Hermann et al. [23] haben sich in ihrem Paper mit der Fragestellung auseinandergesetzt, wie stark die Interrater-Reliabilität bei Datensätzen aus der Sentimentanalyse ausgeprägt ist. Dafür wurden zwei annotierte Datensätze verwendet, die jeweils aus StackOverflow- und aus GitHub-Kommentaren bestehen und im Forschungsfeld der Sentimentanalyse als Trainings- und Testdatensätze verwendet worden sind [34], [29]. Aus diesen Datensätzen wurde eine Umfrage erstellt, in der Personen, mit Bezug zum Software Engineering, den Einträgen eine Polarität zuordnen sollten [23]. So sollte überprüft werden, inwieweit die Personen mit den annotierten Labeln übereinstimmen. Die Autoren kamen zu den Ergebnissen, dass im Median 62.5% der Personen mit den annotierten Labeln übereinstimmen, es aber gleichzeitig keinen einzigen Teilnehmer gibt, der vollständig mit den Annotationen übereinstimmt [23]. Zudem konnte gezeigt werden, dass die Probanden größere Übereinstimmungswerte mit dem GitHub-Datensatz aufweisen, der auf der Basis des Emotionsmodells von Shaver et al. [42] annotiert wurde. Der Stackoverflow-Datensatz wurde hingegangen nach dem ad hoc Verfahren gelabelt.
Martensen [31] hat sich in seiner Bachelorarbeit mit der Fragestellung auseinandergesetzt welche Einflüße es in der Sentimentzuweisung bei Entwicklern gibt. Dafür hat er eine Umfrage erstellt an der Teilnehmer des Software-Projekts der Leibniz Universität Hannover mitwirken konnten. Dabei wurden die Probanden an bis zu vier unterschiedlichen Zeitpunkten des Software-Projekts zu ihrer Lebenssituation, ihrer Stimmung im Allgemeinen, ihrer Stimmung in den letzten 7 Tagen und der Gruppendynamik im Team des Software-Projekts befragt. Zusätzlich sollten die Probanden 30 ausgewählte Sätze labeln, die ebenfalls aus den GitHub-, bzw. aus StackOverflow-Datensätzen, stammen, die auch von Hermann et al. [23] verwendet worden sind. Martensen [31] konnte unter Anderem zeigen, dass Probanden Aussagen weniger häufig als positiv bewerten und häufiger als negativ bewerten, wenn deren Stimmung in den letzten sieben Tagen schlecht war. Die Korrelationskoeffizienten seien mit 0.16, bzw. -0.19, aber relativ gering. Außerdem wurden

Intrarater-Reliabilitätsanalysen unternommen, um zu untersuchen inwiefern die Probanden an den bis zu vier Umfrageteilnahmen mit sich selbst bei der Labelvergabe übereinstimmen. Dafür wurden die Fleiss' Kappa Werte berechnet und miteinander verglichen. Dabei konnte festgestellt werden, dass Probanden bei mehrfacher Teilnahme einen Lerneffekt erzielten, weil sie im Durchschnitt neutrale Sätze besser erkennen konnten [31].

3.2 Urteilerübereinstimmung

Cohen [6] brachte im Jahr 1960 ein Paper heraus, aus dem das bereits in dieser Bachelorarbeit vorgestellte Reliabilitätsmaß Cohen's Kappa hervorging. Es wurde als Maß vorgestellt mit dem Inter-Rater-Übereinstimmungen von zwei Bewertern auf nominalskalierten Daten berechnet werden können. Eine Besonderheit an Cohen's Kappa ist, dass es das erste bekannte Maß ist, das eine zufällige Übereinstimmungswahrscheinlichkeit der Bewerter berücksichtigt [6].
Allerdings gab es beim klassischen Cohen's Kappa nur zwei Möglichkeiten hinsichtlich der Beurteilung der Bewertung, wenn ein einzelnes Bewertungsobjekt betrachtet wird. Entweder die beiden Bewerter haben dem Bewertungsobjekt die gleiche Kategorie zugeordnet und stimmten miteinander überein, oder sie haben dem Objekt unterschiedliche Kategorien zugeordnet und stimmten nicht miteinander überein.
Im Jahr 1968 wurde eine Verallgemeinerung, das gewichtete Cohen's Kappa, in einem Paper vorgestellt [7]. Damit ist es möglich, wie in Kapitel 2.3.4 näher beschrieben, abgestufte Formen der Übereinstimmung, oder Assoziationen, abzubilden. Folglich ist das Maß auf ordinalskalierten Daten sinnvoll einsetzbar [7].
Fleiss [14] hat sich mit dem Problem beschäftigt, dass die Anwendung von Cohen's Kappa auf die Fälle limitiert ist, in denen Urteilerübereinstimmungen von genau zwei Bewertern gemessen werden. Er stellte im Jahr 1971 sein Fleiss' Kappa vor, welches auf Cohen's Überlegungen basiert, allerdings für zwei oder mehr Bewerter eingesetzt werden kann [14].
In dem Zusammenhang sei die Arbeit von Conger [8] erwähnt. Er kritisierte an Fleiss' Kappa, dass es nicht die gleichen Ergebnisse wie Cohen's Kappa liefert, wenn zwei Bewerter betrachtet werden. Das motivierte Conger [8] dazu seine Verallgemeinerung von Cohen's Kappa vorzustellen, welches ebenfalls für zwei oder mehr Bewerter eingesetzt werden kann und die gleichen Ergebnisse wie Cohen's Kappa für $n = 2$ Bewerter liefert.

Gwet [21] hat in seinem Paper auf das folgende Beispiel aufmerksam gemacht:

	Bewerter 2		
Bewerter 1	+	-	Total
+	118	5	123
-	2	0	2
Total	120	5	125

Tabelle 3.1: Verteilung von 125 Teilnehmern nach Bewerter und Antwortkategorie, übersetzt aus Gwet [21]

Intuitiv würde man von einer hohen Übereinstimmung zwischen den beiden Bewertern ausgehen, da 118, von insgesamt 125 Bewertungsobjekten, identisch bewertet worden sind. Doch Cohen's Kappa liefert in dem Beispiel sogar einen negativen Wert in Höhe von -0.0023. Das Beispiel steht für ein Problem welches zu einer Problemklasse gehört, die in die Literatur als Kappa-Paradoxien einging [21]. Das nahm Gwet zum Anlass, in dem genannten Paper, sein Reliabilitätsmaß vorzustellen, welches resistenter gegenüber den Paradoxien ist. Es exisitert eine ungewichtete Version seines Reliabilitätsmaßes, welches als Gwet's AC_1 bekanntgeworden ist, und eine gewichtete Version, bekannt als Gwet's AC_2 [21].
Zum Abschluss sei eine Arbeit erwähnt, die sich exklusiv mit dem Problemfeld der Intrarater-Reliabilitätsanalyse beschäftigt [22]. In dem Paper wurde aufgezeigt warum sich einige Interrater-Maße auch als Intrarater-Maße einsetzen lassen, darunter die in dieser Bachelorarbeit vorgestellten Reliabilitätsmaße.
Darüber hinaus wurde in dem Paper eine Übersicht erstellt für welches Skalenformat welche Reliabilitätsmaße einsetzbar sind und es wurden Empfehlungen zum Studiendesign einer Intrarater-Reliabilitätsuntersuchung gegeben [22].

3.3 Abgrenzung der Arbeit

Der Fokus in der vorliegenden Bachelorarbeit liegt darauf die Software IIRA zu entwickeln, mit der Reliabilitätsuntersuchungen vorgenommen werden können und Daten gelabelt werden können. Dabei werden Limitierungen und Besonderheiten im Entwicklungsprozess berücksichtigt, die aus den zuvor erwähnten Arbeiten hervorgegangen sind. Beispielsweise sind die unterschiedlichen Reliabilitätsmaße nur für bestimmte Skalenformate definiert. Darüber hinaus wird IIRA verwendet, um Intrarater-Reliabilitätsuntersuchungen auf den Datensätzen vorzunehmen, die aus den Arbeiten von Hermann et al. [23] und Martensen [31] entstanden sind. Bei den Untersuchungen von Hermann

et al. [23] lag der Fokus auf der Untersuchung der Interrater-Reliabilität, während in der vorliegenden Bachelorarbeit die Intrarater-Reliabilität der unterschiedlichen Bewerter analysiert wird.
Martensen [31] hat in seiner Arbeit auch die Intrarater-Reliabilität berechnet. Allerdings sollen in der vorliegenden Arbeit, neben dem Berechnen der Fleiss' Kappa Werte, zusätzlich die Gwet's AC_1 Werte des Datensatzes berechnet werden.

Kapitel 4

Entwicklung der Software

Dieses Kapitel widmet sich der im Rahmen dieser Bachelorarbeit erstellen Software IIRA.
Zunächst wird auf den Planungsprozess der Software eingegangen. Dabei liegt der Fokus beim Aufstellen der Anforderungen, sowie der Darstellung eines Design-Konzeptes.
Bei der Implementierung, im Unterkapitel 4.2, wird die abgabebereite Software vorgestellt. Neben der reinen Vorstellung der Software, wird in diesem Kapitel auf Besonderheiten und Herausforderungen, die bei der Implementierung aufgekommen sind, eingegangen.
Abschließend wird sich in dem Unterkapitel 4.3 damit auseinandergesetzt wie die Software getestet wurde.

4.1 Planung

In diesem Unterkapitel werden zunächst die Stakeholder von IIRA vorgestellt. Anschließend werden die definierten Anforderungen aufgezeigt, sowie eine Priorisierung der Anforderungen vorgenommen. Zum Schluss folgt die Darstellung eines Design-Konzepts in Form von Papierprototypen.

4.1.1 Stakeholder

Wie eingangs erwähnt, gibt es zwei übergeordnete Anwendungsfälle der Software. Einerseits das Erstellen von Reliabilitätsuntersuchungen und andererseits das Labeln von Datensätzen.
Demnach richet sich IIRA in erster Linie an Personen, die Intra-, bzw. Interrater-Reliabilitätsuntersuchungen durchführen wollen. Solche Untersuchungen werden beispielsweise in klinischen Studien unternommen, um Aussagen über die Qualität der Studien treffen zu können [22]. Ein anderes Szenario sind Datensätze, die in Supervised-Machine-Learning-Systemen zum Training der Modelle verwendet werden [37]. Falls das Labeln der Datensätze

nach subjektiven Maßstäben erfolgte, wie es in der Sentimentanalyse der Fall ist, kann es auch in diesem Fall sinnvoll sein Reliabilitätsuntersuchungen mit den Datensätzen durchzuführen.
Der zweite Anwendungsfall richtet sich an Personen, die beim Labeln der Daten, äußere Einflüße möglichst gering halten wollen. Die Personen haben die Möglichkeit sich die Bewertungsobjekte isoliert anzeigen zu lassen und die Reihenfolge bei der Bewertung zu randomisieren. So soll sichergestellt werden, dass sich die Bewerter, bei der Bewertung, an keine Patterns gewöhnen können.

4.1.2 Anforderungen

Die folgenden Anforderungen haben sich aus mehreren Iterationen ergeben. Zunächst wurden erste Anforderungen in Zusammenarbeit mit dem Betreuer der Bachelorarbeit aufgestellt. Diese dienten als Grundlage, um Papierprototypen zu erstellen, die im Kapitel 4.1.4 dargestellt worden sind. Daraufhin wurden Experteninterviews mit zwei Mitarbeitern der Leinbiz Universität Hannover geführt, die sich unter anderem mit der Sentimentanalyse beschäftigen. Zusätzlich wurden zwei weitere Interviews mit Informatikstudierenden der Leibniz Universität Hannover geführt. Schließlich ergaben sich die folgenden Anforderungen an die Software:

[R01] Die Anwendung soll Excel- , LibreOffice-Calc- und CSV-Dateien importieren können.
Die importierten Dateien können gemäß den Anforderungen [R02], [R03] und [R12] analysiert, bzw. bewertet, werden.

[R02] Die Anwendung soll eine Intrarater-Reliabilitätsanalyse auf strukturierten Daten vornehmen können.
Wie die Daten strukturiert sein müssen ergibt sich aus den Anforderungen [R04] - [R08].

[R03] Die Anwendung soll eine Interrater-Reliabilitätsanalyse auf strukturierten Daten vornehmen können.
Wie die Daten strukturiert sein müssen ergibt sich aus den Anforderungen [R04] - [R08].

[R04] Der SW-User kann zwischen den Skalentypen nominal, ordinal, intervall und rational wählen.
Erforderlich, um [R02] und [R03] ausführen zu können.

[R05] Der SW-User kann bei nominal- oder ordinalskalierten Daten Kategorienamen festlegen.
Erforderlich, um [R02], [R03] und [R12] ausführen zu können.

[R06] Der SW-User kann festlegen von welchen Bewertern die Intrarater-Reliabilitätsanalysen (bzw. Interrater-Reliabilitätsanalyse) vorgenommen werden sollen.
Erforderlich, um [R02] und [R03] ausführen zu können.

[R07] Der SW-User kann festlegen welche Bewertungsobjekte analysiert werden sollen.
Erforderlich, um [R02], [R03] und [R12] ausführen zu können. Die Bewertungsobjekte sind in der in [R01] importierten Datei enthalten.

[R08] Der SW-User kann auswählen welche Metriken er verwenden möchte, um die Intrarater-Reliabilitätsanalyse (bzw. Interrater-Reliabilitätsanalyse) vorzunehmen.
Erforderlich, um [R02] und [R03] ausführen zu können.

[R09] Die Anwendung soll eine Vorauswahl treffen, so dass in [R08] nur die Metriken angezeigt werden, die zusammen mit den Eingaben aus [R04] - [R07] einsetzbar sind.
Soll die Bedienbarkeit des Programmes erhöhen, da [R04] – [R07] vorgeben welche Metriken verwendet werden dürfen und welche nicht.

[R10] Die in [R08] angezeigten Metriken sollen von der Anwendung erklärt werden können.
Stellt eine Hilfestellung für User dar, die unter Umständen die Metriken noch nicht kennen.

[R11] Der SW-User soll Daten selber bewerten können.
Durch die Bewertung der Daten können beispielsweise im Anschluß eigene Reliabilitätsuntersuchungen vorgenommen werden.

[R12] Es können in der Anwendung unterschiedliche Nutzerprofile angelegt werden.
Damit unterschiedliche Nutzer, innerhalb der gleichen Installation der Software, Bewertungen unter unterschiedlichen Pseudonymen vornehmen können.

[R13] Die Anwendung kann alle geforderten Daten im Excel-Format exportieren.
Bewertungssessions können exportiert werden und die Auswertung aus [R02] und [R03] kann exportiert werden.

[R14] Die Anwendung soll in einem modernen Design erscheinen.
Für das Jahr 2023 angemessene Labels und Buttons verwenden.

[R15] Die Anwendung kann auf unix-ähnlichen Betriebssystemen, sowie auf Windows-Betriebssystemen verwendet werden.
Anwendung kann auf allen gängigen Betriebssystemen verwendet werden.

4.1.3 Priorisierung der Anforderungen

In der Tabelle 4.1 wurde dargestellt wie die 15 Anforderungen priorisiert worden sind.

Anforderung	Priorität
[R01]	Hoch
[R02]	Hoch
[R03]	Hoch
[R04]	Hoch
[R05]	Hoch
[R06]	Hoch
[R07]	Hoch
[R08]	Hoch
[R09]	Hoch
[R10]	Mittel
[R11]	Hoch
[R12]	Hoch
[R13]	Hoch
[R14]	Niedrig
[R15]	Mittel

Tabelle 4.1: Priorisierung der Anforderungen

4.1.4 Papierprototypen

Die im folgenden Abschnitt vorgestellten Papierprototypen stellen einen ersten Designentwurf der Software dar. Einige Aspekte wurden von den Prototypen in der Implementierungsphase übernommen und andere wurden verworfen, da sie sich als unvorteilhaft erwiesen haben. Um dem Inhalt des Kapitels 4.2 nicht vorwegzugreifen, werden die Papierprototypen nur beschrieben und nicht explizit darauf hingeweisen welche Aspekte übernommen und welche verworfen worden sind.

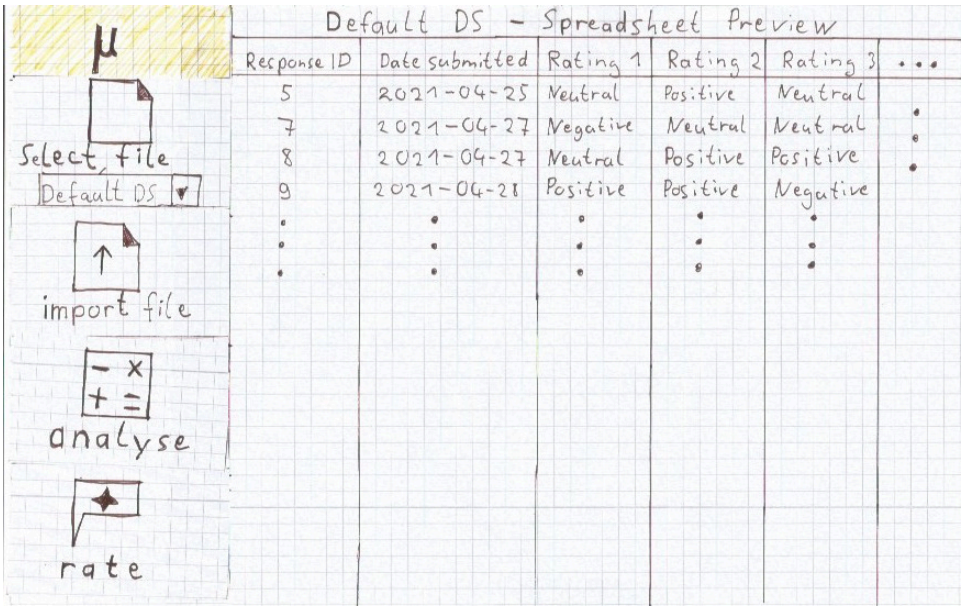

Abbildung 4.1: Hauptfenster

Das Hauptfenster ist zugleich das Startfenster der Applikation. Auf der linken Seite des Fensters sind mehrere Buttons übereinander angeordnet. Der μ-Button, ganz oben, ist in jedem Fenster sichtbar und stellt das Home-Icon der Anwendung dar. Ein Linksklick auf diesen Button wird den User zurück zum Hauptfenster springen lassen.
Direkt darunter sind zwei Buttons zu sehen, mit denen Dateien importiert, bzw. ausgewählt, werden können. Über den Datei-Importieren-Button sollen Dateien in die Software importiert werden können. Importierte Dateien sollen mit dem Datei-Auswählen-Dropdown-Menü schließlich ausgewählt werden können.
Es folgen die beiden Buttons, mit denen die übergeordneten Anwendungsfälle von IIRA angesteuert werden können. Nämlich das Erstellen von Reliabilitätsuntersuchungen (Analyse-Button), bzw. das Labeln von Bewertungsobjekten (Rate-Button).
Das restliche Fenster besteht aus einer Vorschau der importierten Datei. Diese soll eine Hilfestellung für den SW-User darstellen. Möglicherweise kann sich ein SW-User nicht mehr daran erinnern, welcher Inhalt in welcher Datei vorhanden ist und anstatt sie manuell zu öffnen, könnte der Inhalt direkt in IIRA geprüft werden. Es wurde allerdings kritisiert, dass das Design gegen das Prinzip des Information Hidings verstößt. Schließlich werden hier unter Umständen Informationen angezeigt, die für die Nutzung des Programmes nicht erforderlich sind.

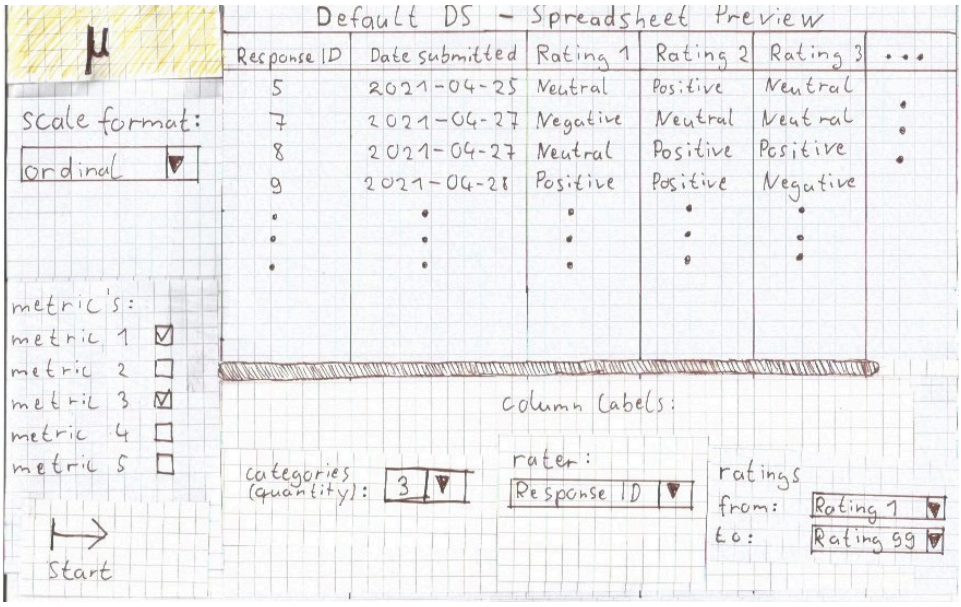

Abbildung 4.2: Analysieren-Fenster

Das Analysieren-Fenster wird angezeigt, wenn im Hauptfenster auf den Analyse-Button gedrückt wird. In dieser Ansicht hat der SW-User die Möglichkeit alle Informationen anzugeben, die notwendig sind, um Reliabilitätsuntersuchungen vorzunehmen.
Auf der linken Seite, unter dem Home-Icon, kann über ein Dropdown-Menü das Skalenformat ausgewählt werden.
Direkt darunter ist es möglich Metriken auszuwählen, mit denen die Reliabilitätsuntersuchungen vorgenommen werden sollen. Beispielsweise könnte hier, in der Anwendung, Cohen's κ ausgewählt werden, falls das Skalenformat dies zulässt.
Ganz unten links ist ein Start-Button zu sehen, mit dem die Untersuchungen gestartet werden können. Im unteren Bereich der Ansicht kann der User festlegen, in welchen Spalten der importierten Datei die Bewerter, bzw. die Bewertungsobjekte, zu finden sind.
Außerdem kann über ein Dropdown-Menü die Anzahl der Kategorien festgelegt werden.

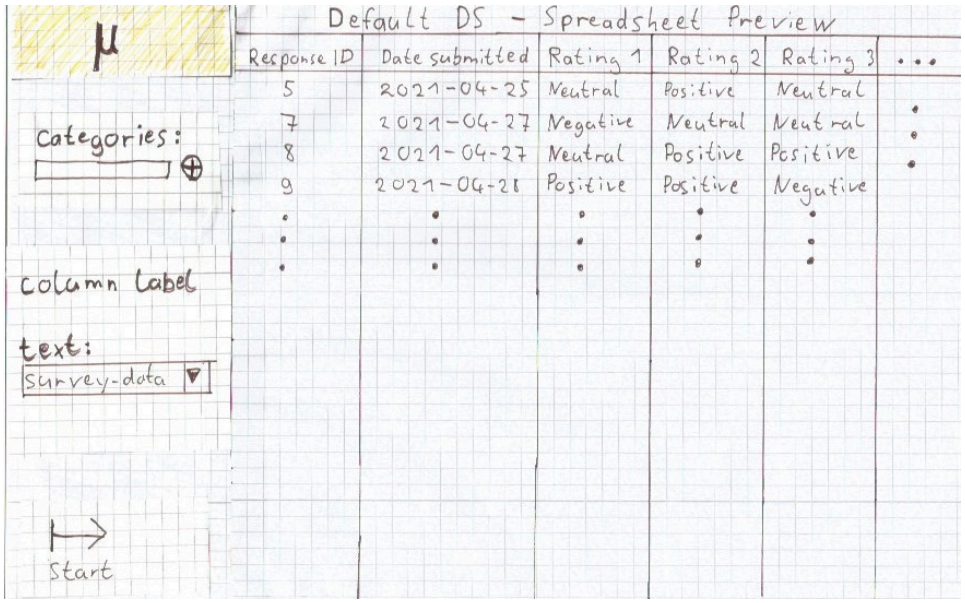

Abbildung 4.3: Bewertungsvorbereitung-Fenster

Die vorliegende Ansicht wird angezeigt, wenn der SW-User im Hauptfenster auf den Rate-Button drückt. In diesem Fenster können Vorbereitungen getroffen werden, um die Bewertungssession zu starten. Bevor mit der Bewertung begonnen werden kann, ist es erforderlich festzulegen welche Bewertungsobjekte bewertet werden sollen und welche Kategorien bei der Bewertung auswählbar sind.
Die Kategorien sollen im Papierprototypen, durch ein Eingabefeld, direkt vom User eingegeben und durch das Drücken auf den Plus-Button hinzugefügt werden können. Falls Text im Sinne der Sentimentanalyse bewertet werden soll, würde der SW-User hier also typischerweise die Kategorien „Positiv“, „Neutral“ und „Negativ“ eingeben.
Direkt unter dem Eingabefeld der Kategorien, kann durch ein Drop-Down-Menü ausgewählt werden, in welcher Spalte die Bewertungsobjekte zu finden sind. Wie damit umgegangen wird, wenn die Bewertungsobjekte in mehreren Spalten, oder zeilenweise, statt spaltenweise, organisiert sind, kann vom Prototypen noch nicht beantwortet werden. Die Details werden in der Implementierungsphase in Kapitel 4.2 beantwortet.
Sobald alle Angaben ausgefüllt worden sind, hat der User die Möglichkeit mit dem Start-Button, unten links, die Bewertungssession zu beginnen.

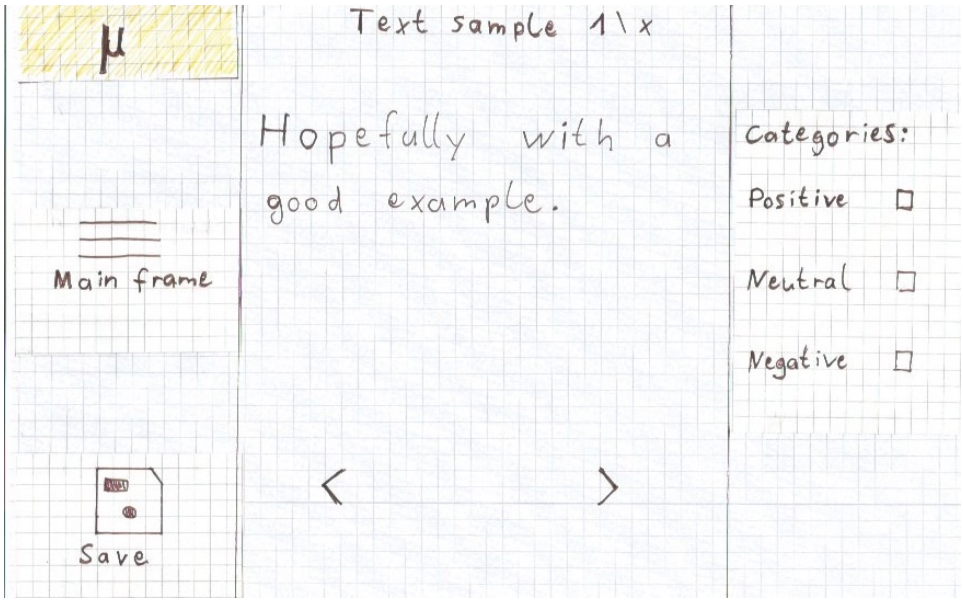

Abbildung 4.4: Bewerten-Fenster

Im Bewerten-Fenster wird die Bewertungssession vom SW-User vorgenommen. Auf der linken Seite sind erneut mehrere Buttons zu sehen. Wichtig ist nur der Save-Button, mit dem die Bewertungssession gespeichert wird. Der Mainframe-Button, direkt darüber, dient lediglich als Platzhalter im Prototypen für einen weiteren Button, der gegebenenfalls an der Stelle hinzugefügt werden kann.
Im unteren Bereich der Ansicht sind jeweils ein Pfeil nach rechts, bzw. ein Pfeil nach links, zu sehen, mit denen zum vorherigen, bzw. zum nächsten, Objekt, das bewertet werden soll, gesprungen werden kann. In der Mitte ist das Bewertungsobjekt zu sehen und oben wird eine Erfolgsanzeige dargestellt, die dem User Feedback darüber gibt, wie viele Bewertungsobjekte bereits bewertet worden sind.
Die rechte Seite ist mit den Kategorien, in Form von Radiobuttons gefüllt, die im Bewertungsvorbereitung-Fenster vom SW-User eingegeben worden sind. Über die Radiobuttons erfolgt schließlich die Bewertung.

4.2 Implementierung

Das Unterkapitel beginnt mit dem Vorstellen der Softwarearchitektur. Hier wird auf die grundlegende Package-Struktur des Projektes, sowie auf die verwendete Programmiersprache und die wichtigsten Bibliotheken eingegangen. Es folgen Erläuterungen dazu, wie die wichtigsten Softwaremodule implementiert worden sind und zum Schluss wird auf Herausforderungen bei der Implementation hingewiesen.

4.2.1 Softwarearchitektur

Wegen der hohen Portabilität der Programmiersprache Python, wurde die Sprache für das Softwareprojekt verwendet. Für die graphische Nutzeroberfläche (GUI) wurde das, zur Standardbibliothek gehörende, Package Tkinter verwendet.
Die Package-Struktur des Projektes ist an dem Model-View-Controller-Architekturmuster angelehnt und kann, wie im Anhang A.1 dargestellt, visualisiert werden. Der Controller befindet sich auf der höchsten Architekturebene, also direkt im IIRA-Package. Er steuert den gesamten Programmablauf und stellt beispielsweise sicher, dass die richtigen GUI-Elemente angezeigt werden. Die GUI-Elemente liegen in dem GUI-Package und stellen die View des Softwareprojektes dar. Ein Model wird im Core-Package generiert, sobald der User eine Datei importiert hat. Die importierte Datei wird zunächst validiert, um anschließend alle Informationen zu extrahieren, die für die beiden übergeordneten Anwendungsfälle, Bewerten oder Analysieren, von Bedeutung sind. Die extrahierten Informationen werden wiederum in der GUI dargestellt. So kann der SW-User beispielsweise beim Analysieren-Anwendungsfall die Bewerter-ID's in der GUI auswählen, von denen eine Reliabilitätsuntersuchung vorgenommen werden soll. Für die Analyse wird im Core-Package erneut ein geeignetes Model generiert und auch die Berechnungen für die Reliabilitätsuntersuchung, werden in dem Package durchgeführt. Die Reliabilitätsmaße werden größtenteils durch das externe Package irrCAC[1] importiert. Lediglich der Intra-Klassen-Korrelationskoeffizient (ICC) wird durch das Package pingouin[2] importiert. Zusätzlich gibt es ein Data-Package, in dem Bilder und Icons liegen, die in der GUI verwendet werden. Außerdem ist in dem Package das Theme zu finden, das in IIRA verwendet wird. Darüber hinaus liegt in dem Data-Package eine kleine Datenbank, in Form von einer CSV-Datei, in dem die unterschiedlichen Nutzerprofile gespeichert werden, die in der Applikation angelegt werden können.

[1] https://pypi.org/project/irrcac/
[2] https://pingouin-stats.org/build/html/generated/pingouin.intraclass_corr.html

4.2.2 Umsetzung

In dem Unterkapitel werden die wichtigsten Module der Software vorgestellt und dabei wird aufgezeigt wie die Anforderungen aus Kapitel 4.1.2 umgesetzt worden sind.

Hauptfenster

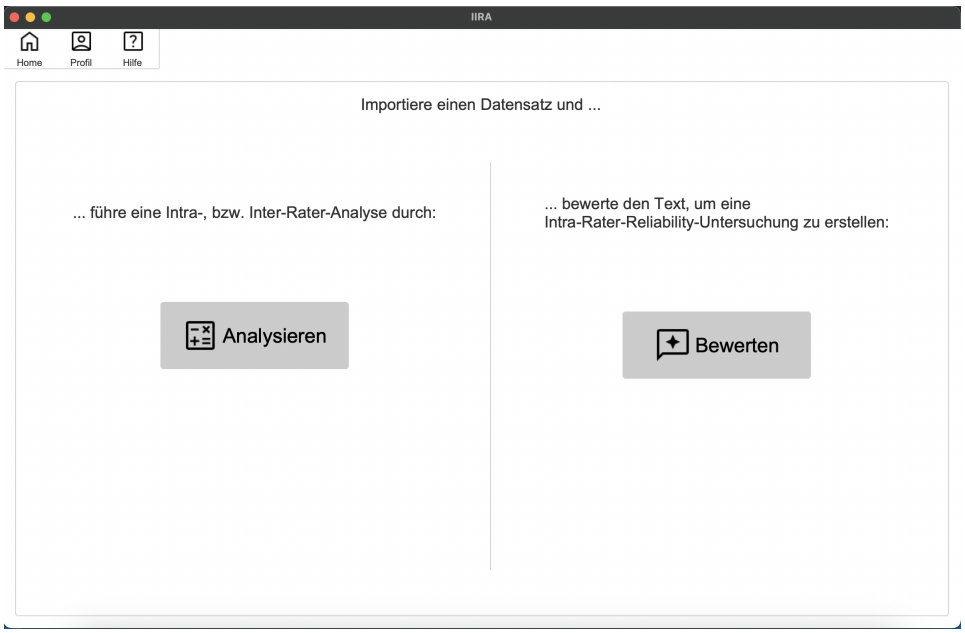

Abbildung 4.5: Hauptfenster

Das Hauptfenster besteht aus insgesamt fünf Elementen, mit denen der SW-User interagieren kann. Durch das Drücken auf den Analysieren-Button, kann der User die in der Anforderung [R02], bzw. [R03], spezifizierte Intra-, bzw. Interrater-Reliabilitätsuntersuchungen, beginnen. Mit dem Bewerten-Button kann die, in der Anforderung [R11] spezifizierte, Bewertungssession begonnen werden.
Darüber hinaus befindet sich am oberen Rand der Applikation eine Menübar, die in jeder Ansicht angezeigt wird. Die Menübar besteht links aus einem Home-Button, mit dem der SW-User jederzeit zum Hauptfenster springen kann. Mittig ist ein Profil-Button zu sehen, mit dem das im Anhang A.13 dargestellte Fenster geöffnet werden kann. In dem Fenster kann neben einem Profilwechsel, ein neues Profil gemäß [R12] angelegt werden, sowie das aktuell ausgewählte Profil gelöscht werden. Rechts in der Menübar ist ein Help-Button zu sehen. Durch Drücken des Help-Buttons öffnet sich ein Hilfefenster, welches Erklärungen zu den in der GUI dargestellten Elementen

liefert. Im Anhang A.11 ist exemplarisch ein Hilfefenster dargestellt, welches Erklärungen zu Skalenformaten und Gewichten liefert.

Skalenformat und Gewichte auswählen

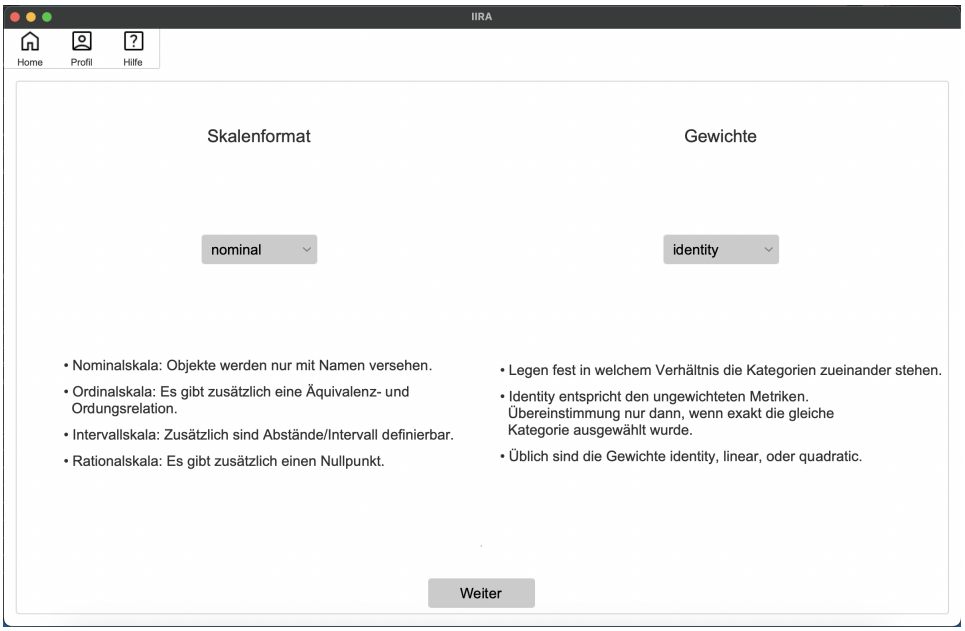

Abbildung 4.6: Skalenformat und Gewichte auswählen

Wenn der SW-User eine Reliabilitätsuntersuchung vornehmen möchte, ist es zunächst erforderlich das Skalenformat gemäß [R04] festzulegen. Die Auswahl wird im dargestellten Fenster durch ein Dropdown-Menü ermöglicht. Es kann zwischen den in Kapitel 2.2 vorgestellten Skalenformaten ordinal, nominal, intervall und rational ausgewählt werden.
Zusätzlich hat der User die Möglichkeit zwischen acht verschiedenen Gewichten auszuwählen, die bei den Berechnungen, wie in Kapitel 2.3.4 erläutert, verwendet werden.
Falls der SW-User im vorherigen Fenster den Bewerten-Button, statt den Analysieren-Button, gedrückt hat, gelangt er in eine leicht modifizierte Ansicht. Da für die Bewertungssession keine Auswahl an Gewichten notwendig ist, wird in der modifizierten Ansicht lediglich die Auswahl des Skalenformates dargestellt. Im Anhang A.4 wurde zugehörige Ansicht dargestellt.

Datei importieren

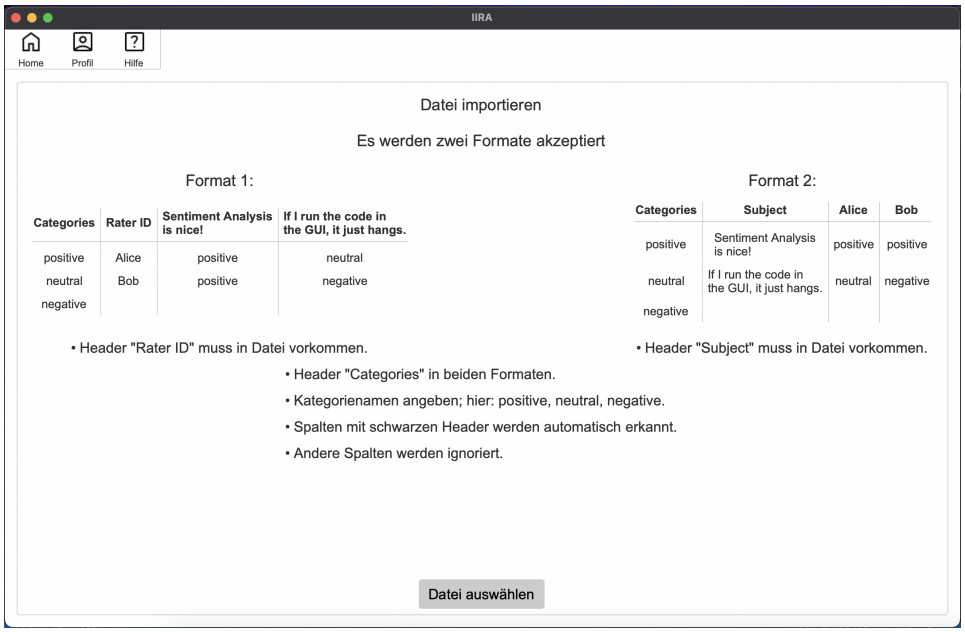

Abbildung 4.7: Datei importieren

Die nächste Ansicht ermöglicht es, die in [R01] erwähnte Datei hochzuladen. Auf dem Bild wurde dargestellt wie das Fenster aussieht, wenn der SW-User im vorherigen Schritt ein nominales oder ordinales Skalenformat ausgewählt hat. Bei intervall- oder rationalskalierten Daten wird die Ansicht, wie im Anhang A.6 dargestellt, angezeigt.
Wenn der Datei-Auswählen-Button gedrückt wird, öffnet sich ein Filedialog. Um Dateiformatfehler auszuschließen, können im sich öffnenden Dialog lediglich CSV-, Excel und LibreOffice-Calc-Dateien ausgewählt werden. Des Weiteren ist es erforderlich, dass die zu importierende Datei dem dargestellten Format 1 oder dem Format 2 entspricht. In beiden Formaten ist es zwingend erforderlich, dass in der Datei eine Spalte mit dem Header „Categories“ existiert. In dieser Spalte werden gemäß [R05] die Kategorienamen festgelegt, die für Reliabilitätsuntersuchungen, bzw. für das Bewerten, von Bedeutung sind.
Im Format 1 muss darüber hinaus eine Spalte mit dem Header „Rater ID“ vorkommen und im Format 2 eine Spalte mit dem Header „Subject“. Die beiden akzeptierten Formate unterscheiden sich also in der Hinsicht, dass die Rater ID’s, bzw. Bewertungsobjekte (Subjects), zeilenweise, bzw. spaltenweise, organisiert worden sind. Ausgehend vom Dateiformat 1 findet das Programm automatisch alle Bewertungsobjekte, indem in der Datei nach Spalten gesucht wird, in denen alle Zellen entweder leer sind, oder

aus Einträgen bestehen, die gültige Kategorien darstellen. Auf diese Weise wird [R07] implementiert ohne, dass der SW-User explizit die Spalten der Bewertungsobjekte angeben muss.
Das Vorgehen beim Format 2 ist analog, allerdings werden bei dem Format auf diese Weise die Rater ID's gesucht. Alle weiteren Spalten sind für IIRA nicht von Bedeutung und werden ignoriert.

Reliabilitätsuntersuchung vorbereiten

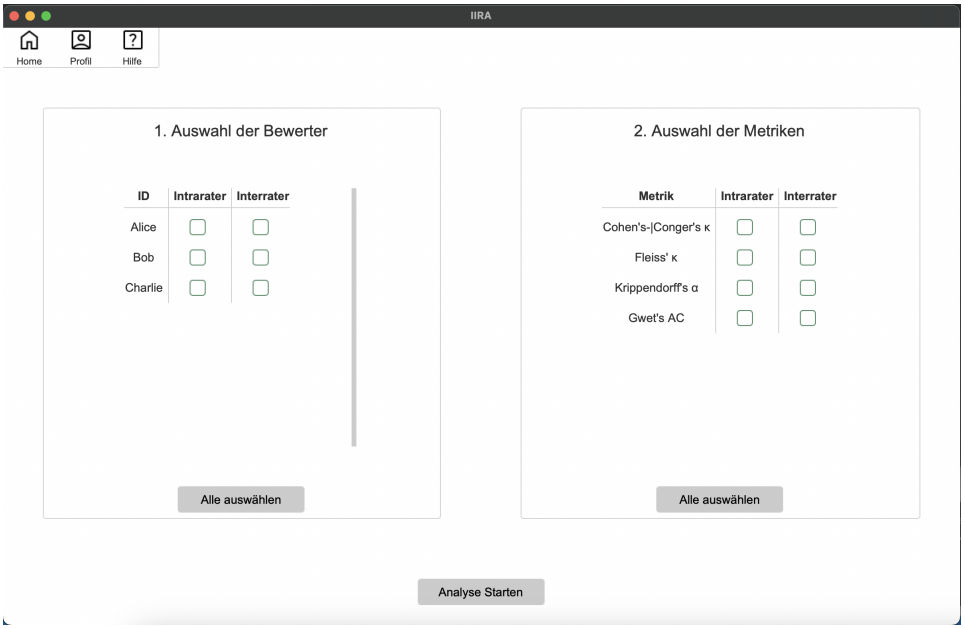

Abbildung 4.8: Reliabilitätsuntersuchungen vorbereiten

Wie in [R06] definiert, ist es erforderlich anzugeben, von welchen Bewertern die Reliabilitätsuntersuchungen vorgenommen werden sollen. Die dargestellte Ansicht ermöglicht es, durch Checkbuttons, die Bewerter auszuwählen, von denen Intrarater-, bzw. Interrater-Reliabilitätsuntersuchungen, vorgenommen werden sollen. Für jeden ausgewählten Intrarater wird eine eigene Intrarater-Reliabilitätsuntersuchung erstellt. Bei der Auswahl mehrerer Interrater hingegen, wird eine einzelne Untersuchung mit allen ausgewählten Bewertern erstellt.
Darüber hinaus ist es möglich in dem Fenster gemäß [R08] die Metriken auszuwählen, mit denen die Intra-, bzw. die Interrater-Reliabilität, berechnet werden soll. Wie in [R09] gefordert, werden in der Ansicht nur die Metriken dargestellt, die auf dem zuvor festgelegten Skalenformat definiert sind. Bei nominal- oder ordinalskalierten Daten werden die im Bild dargestellten Metriken angezeigt und bei intervall- oder rationalskalierten

Daten steht der ICC zur Auswahl. Zusätzlich wird intern geprüft, ob eine Reliabilitätsuntersuchung von genau zwei Bewertern vorgenommen wird. In dem Fall wird Cohen's Kappa berechnet, falls die entsprechende Metrik ausgewählt wurde und andernfalls Conger's Kappa. Die beiden Metriken wurden zusammengefasst, da Conger's Kappa eine echte Verallgemeinerung von Cohen's Kappa ist. Da Cohen's Kappa namentlich die wohl bekannteste Reliabilitätsmetrik ist, sollte allerdings auf dessen Erwähnung nicht verzichtet werden. Die Anforderung [R10] wird durch das Hilfefenster umgesetzt, in dem Erklärungen zu den Metriken zu finden sind.
Darüber hinaus gibt es in der Ansicht zwei Togglebuttons, mit denen alle Bewerter, bzw. alle Metriken, ausgewählt werden können. Falls bereits alle Bewerter, bzw. alle Metriken, ausgewählt sind, bieten die Buttons die Option alle Bewerter, bzw. alle Metriken, abzuwählen.

Ergebnisse

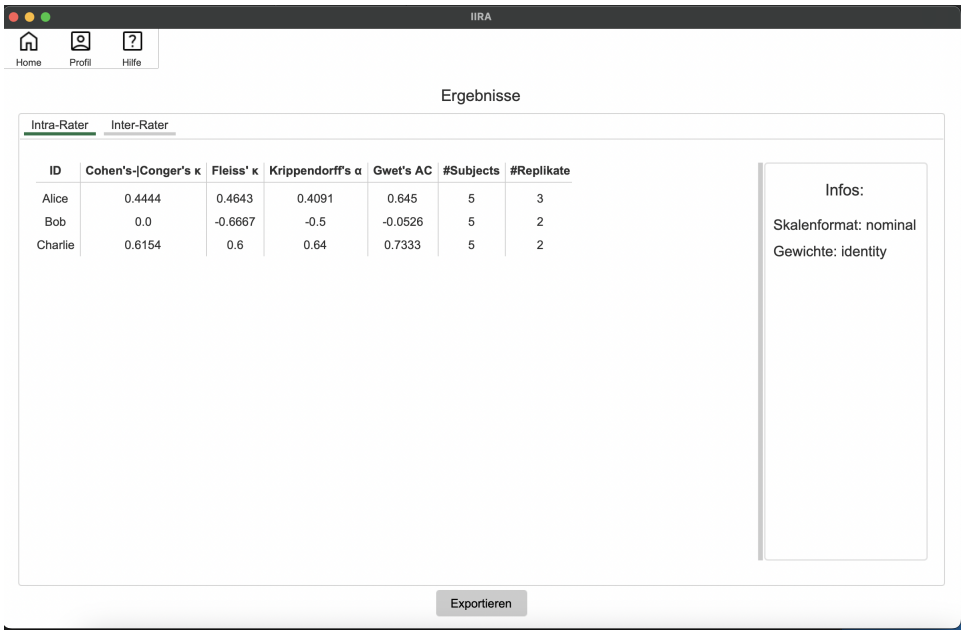

Abbildung 4.9: Ergebnisse

Nachdem alle Angaben für die Reliabilitätsuntersuchungen eingegeben worden sind, können die Ergebnisse berechnet und dargestellt werden. Die Ergebnis-Ansicht besteht aus einem Intrarater-Tab, in dem die Ergebnisse der Intrarater-Reliabilitätsuntersuchungen tabellarisch dargestellt werden. Für jede ausgewählte Intrarater-ID werden die Werte der Metriken berechnet, die im vorherigen Fenster ausgewählt worden sind. Zusätzlich wird in der Tabelle für jede Intrarater-ID angegeben wie viele Bewertungsobjekte

(#Subjects), wie oft (#Replikate) bewertet worden sind. Alice hat beispielsweise fünf Bewertungsobjekte jeweils drei mal bewertet, während die beiden anderen Bewerter die Bewertungsobjekte nur zwei mal bewertet haben.
Rechts in der Info-Box werden wichtige Informationen zu den Reliabilitätsuntersuchungen dargestellt. Das zuvor ausgewählte Skalenformat, sowie die ausgewählten Gewichte werden immer in der Info-Box dargestellt. Zusätzlich kann es vorkommen, dass von einer ausgewählten Rater ID keine Untersuchung vorgenommen werden kann, weil sie keine Bewertungsobjekte mehrfach bewertet hat. Darauf würde in der Info-Box hingewiesen werden, statt die Rater ID in der Tabelle darzustellen.
Der Interrater-Tab ist analog aufgebaut und im Anhang A.9 abgebildet. Ein Unterschied besteht lediglich darin, dass nicht unterschiedliche Bewertungszeitpunkte, sondern unterschiedliche Bewerter betrachtet werden, wie in Kapitel 2.3 erläutert.
Die Reliabilitätsuntersuchungen können gemäß [R13] als CSV-, Excel-, oder LibreOffice-Calc-Datei exportiert werden, wenn auf den Exportieren-Button gedrückt wird. Neben den auf dem Bild dargestellten Informationen, wird in der exportierten Datei das 95%-Konfidenzintervall, sowie der p-Wert der jeweiligen Metrik angezeigt. Ausschnitte aus der exportierten Datei sind im Anhang A.14 und A.4 zu finden.

Bewerten

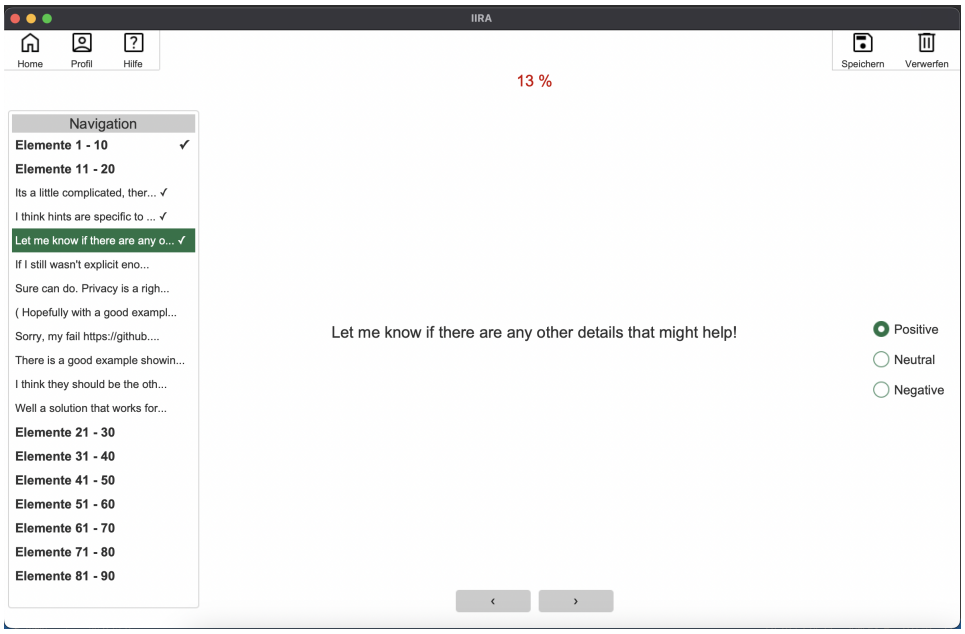

Abbildung 4.10: Bewerten

Durch die dargestellte Ansicht wird die Anforderung [R11], Daten bewerten zu können, umgesetzt. Bevor die Ansicht gefüllt wird, wird der SW-User gefragt, ob die Reihenfolge der zu bewertenden Elemente randomisiert werden soll. Mittig werden Bewertungsobjekte angezeigt. Damit der Text vollständig angezeigt werden kann, mussten an geeigneten Stellen Zeilenumbrüche eingefügt werden. Oberhalb des Textes gibt es eine Statusanzeige, die prozentual angibt, wie viele der Bewertungsobjekte insgesamt bewertet worden sind. Die Farbe der Statusbar ändert sich in 20%-Schritten von rot, über gelbtönen, hin zu unterschiedlichen Grüntönen.
Rechts in der Ansicht werden die zuvor festgelegten Kategorien in Form von Radiobuttons dargestellt, worüber der SW-User die Bewertung vornehmen kann. Für jedes Bewertungsobjekt wird, neben der Bewertung, der Profilname der aktuell angemeldeten Nutzers gespeichert. Somit ist es möglich während einer Bewertungssession das Profil zu wechseln, damit eine andere Person die Session fortsetzen kann.
Auf der linken Seite werden die Bewertungsobjekte blockweise in einer Navigationsübersicht dargestellt. In jedem Block werden jeweils 10 Bewertungsobjekte gruppiert und durch einen Doppelklick auf eines der Elemente, ist es möglich zu dem angeklickten Bewertungsobjekt zu springen. Darüber hinaus werden bereits bewertete Objekte mit einem Haken versehen und sobald alle Objekte des Blockes bewertet worden sind, wird der

gesamte Block mit einem Haken versehen. Neben der Navigation über die Navigationsübersicht, hat der SW-User die Möglichkeit unten den Pfeil nach links, bzw. den Pfeil nach rechts, zu drücken, um zum vorherigen, bzw. zum nächsten, Bewertungsobjekt zu gelangen. Ferner wurden die Pfeiltasten mit den entsprechenden Pfeiltasten auf der Tastatur belegt und die Radiobuttons zur Auswahl der Kategorien wurden mit numerischen Tasten belegt. So können bis zu 9 Kategorien mit den Hotkeys 1 - 9 belegt werden.
Um die Bewertungssession gemäß [R13] zu exportieren, kann der SW-User den Speichern-Button oben rechts drücken. Zusätzlich ermöglicht es der Verwerfen-Button, rechts daneben, die aktuelle Bewertungssession zu verwerfen.

4.2.3 Herausforderungen bei der Implementierung

In den in Kapitel 3.2 vorgestellten Arbeiten, wurden die Reliabilitätsmetriken typischerweise auf vollständig gelabelten Daten definiert. Wenn beispielsweise mit der Cohen's Kappa Metrik die Interrater-Reliabilität von zwei Bewertern gemessen werden soll, wurde die Annahme getroffen, dass beide Bewerter genau n Bewertungsobjekte bewertet haben. In der Praxis kommt es allerdings häufig vor, dass unterschiedliche Bewerter eine unterschiedliche Anzahl an Objekten bewertet haben.
In der vorliegenden Bachelorarbeit wurde der von Gwet [20] vorgeschlagene Ansatz verfolgt, fehlende Label bei der Berechnung der Gesamtübereinstimmungs- und der Zufallsübereinstimmungswahrscheinlichkeit zu berücksichtigen. Es sind allerdings andere Ansätze denkbar. Beispielsweise könnte eine Reliabilitätsuntersuchungen nur mit den $m < n$ Bewertungsobjekten erstellt werden, die von allen Bewertern bewertet worden sind. Da auf diese Weise allerdings die Informationen der Bewerterr verloren gehen, die mehr Objekte bewertet haben, wurde in der vorliegenden Arbeit der Ansatz von Gwet [20] verfolgt.

4.3 Systematisches Testen

Um die Korrektheit der Software zu testen, wurden Black-Box-Tests durchgeführt. Aus den Anforderungen wurden mithilfe der Äquivalenzklassenmethode [40] Testfälle abgeleitet. In der Tabelle 4.2 wurde exemplarisch dargestellt wie der Dateiimport getestet wurde.

ID	Eingabe	Sollresultat	Kommentar
T01.1	.xlsx-Datei	Akzeptiert	
T01.2	.ods-Datei	Akzeptiert	
T01.3	.csv-Datei	Akzeptiert	
T01.4	anderes Dateiformat	Nicht möglich	Nur die oberen drei Formate sollen auswählbar sein

Tabelle 4.2: Testfälle für den Dateiimport

Teilweise war es erforderlich die Anforderungen in atomare Anforderungen zu überführen, bevor die Testfälle abgeleitet werden konnten. So wurde beispielsweise die Anforderung [R06], aus dem Kapitel 4.1.2, in die beiden folgenden atomaren Anforderungen zerlegt:

[R06.1] Der SW-User kann festlegen von welchem Bewertern die Intrarater-Reliabilitätsanalysen vorgenommen werden sollen.
Erforderlich, um [R02] und [R03] ausführen zu können.

[R06.2] Der SW-User kann festlegen von welchem Bewertern die Interrater-Reliabilitätsanalysen vorgenommen werden sollen.
Erforderlich, um [R02] und [R03] ausführen zu können.

Schließlich wurden die in Tabelle 4.3 dargestellten Testfälle formuliert.

ID	Eingabe	Sollresultat	Kommentar
R06.1.1	Auswahl von keinem Intrarater	Akzeptiert	Es wird keine Intrarater-Analyse durchgeführt.
R06.1.2	Auswahl von einem Intrarater	Akzeptiert	
R06.1.3	Auswahl von mehreren Intraratern	Akzeptiert	
R06.2.1	Auswahl von keinem Interrater	Akzeptiert	Es wird keine Interrater-Analyse durchgeführt.
R06.2.2	Auswahl von einem Intrarater	Fehleingabe	Für die Analyse müssen mindestens zwei Bewerter ausgewählt werden.
R06.2.3	Auswahl von mehreren Interratern	Akzeptiert	

Tabelle 4.3: Testfälle für die Auswahl der Bewerter

Kapitel 5

Intrarater-Reliabilitätsuntersuchungen

Für die Bachelorarbeit wurden zwei Datensästze zur Verfügung gestellt, auf denen Intrarater-Reliabilitätsuntersuchungen mit IIRA vorgenommen werden können. In dem Kapitel werden die beiden Datensätze zunächst vorgestellt und anschließend die Ergebnisse der Untersuchungen präsentiert.

5.1 Vorstellung der Datensätze

Die beiden Datensätze stammen aus den Arbeiten von Hermann et al. [23] und aus der Bachelorarbeit von Martensen [31]. Hermann et al. [23] haben in ihrer Arbeit, unter anderem, eine Interrater-Untersuchung vorgenommen, die in dieser Bachelorarbeit um eine Intrarater-Untersuchung ergänzt werden soll. Martensen [31] hat in seiner Arbeit Einflüße bei der Sentimentvergabe analysiert und dabei eine Intrarater-Reliabilitätsuntersuchung, unter Anwendung der Fleiss' Kappa Metrik, vorgenommen.

5.1.1 Hermann et al.

Hermann et al. [23] haben in ihrer Arbeit einen Datensatz aus insgesamt 100 Textelementen zusammengestellt, der sich aus aus einem StackOverflow- und einem GitHub-Datensatz zusammensetzt. Der GitHub-Datensatz ist ein Gold-Standard-Datensatz, der von Novielli et al. [34] im Jahr 2020 veröffentlich wurde und der in der Sentimentanalyse mehrfach als Trainings- und Testdatensatz eingesetzt wurde [47], [46]. Er besteht aus insgesamt 7122 Textelementen, wovon Hermann et al. zunächst 48 Elemente zufällig ausgewählt haben. Dabei wurde darauf geachtet, dass im GitHub-Datensatz jeweils 16 Textelemente als positiv, negativ, bzw. als neutral bewertet worden sind.
Der Vollständigkeit halber sei erwähnt, dass Hermann et al. [23] zusätzlich

48 Textelemente aus einem Datensatz von Lin et al. [29] zufällig ausgewählt haben, der ebenfalls in der Sentimentanalyse eingesetzt wurde [47], [5]. Dabei wurde ebenfalls auf eine ausgeglichene Verteilung der Polaritätsklassen geachtet.
Zusätzlich haben Hermann et al. [23] vier zufällige Elemente aus dem GitHub-Datensatz als Duplikate hinzugefügt, so dass ein Datensatz aus insgesamt 100 Textelementen entsteht. Unter den vier Duplikaten sind zwei Textelemente, die als positiv bewertet worden sind und jeweils ein Textelement das als negativ, bzw. als neutral, bewertet wurde.
In der Arbeit von Hermann et al. [23] nahmen insgesamt 180 Probanden an einer Umfrage teil, um die 100 Textelemente zu bewerten und um zu überprüfen, inwieweit die Teilnehmer mit den vordefinierten Labeln übereinstimmen. Nachdem Probanden aussortiert worden sind, die kein einziges Label in der Umfrage gesetzt haben, oder die keinen Informatikhintergrund haben, blieben 94 Teilnehmer übrig, die bei den Auswertungen betrachtet worden sind [23].
In der vorliegenden Bachelorarbeit wird eine Intrarater-Reliabilitätsuntersuchung mit den Antworten der 94 Probanden erstellt. Dabei wird überprüft in welchem Ausmaß die Teilnehmer mit sich selbst, bei der Labelvergabe der vier Duplikate aus dem GitHub-Datensatz, übereinstimmen.

5.1.2 Martensen

In der Bachelorarbeit von Martensen [31] wurde eine Umfrage erstellt, an der Informatikstudierende der Leibniz Universität Hannover teilnehmen konnten, die das Software-Projekt absolvierten. Die Einflüße bei der Sentimentvergabe wurden untersucht, indem die Studierenden unter Anderem nach ihrer derzeitigen Stimmung und den allgemeinen Lebensumständen befragt worden sind. Anschließend sollten die Teilnehmer 30 Textelemente labeln. Die Textelemente stammen aus gleichen den Datensätzen, die im vorherigen Unterkapitel 5.1.1 vorgestellt worden sind. Diesmal wurden aus dem GitHub-Datensatz [34], sowie aus dem StackOverflow-Datensatz [29], jeweils 15 Textelemente zufällig ausgewählt. Dabei wurde darauf geachtet, dass aus beiden Datensätzen jeweils 5 positiv, 5 negativ und 5 neutral gelabelte Textelemente zufällig ausgewählt worden sind.
Die Umfrage fand an vier unterschiedlichen Zeitpunkten während des Softwareprojektes statt. An der ersten Umfrage haben 55 Studierende teilgenommen, an der zweiten und vierten Umfrage nahmen jeweils 33 Bewerter teil und bei der dritten Umfrage gab es 26 Teilnehmer.
Im Rahmen der vorliegenden Bachelorarbeit wird die Intrarater-Reliabilität bei der Labelvergabe von den Studierenden untersucht, die mehrfach an den Umfragen teilgenommen haben.

5.2 Ergebnisse

Die Ergebnisse der Intrarater-Reliabilitätsuntersuchung wurden zunächst mit IIRA für jeden Bewerter einzeln berechnet. Anschließend wurden Gruppen gebildet, die im jeweiligen Unterkapitel näher beschrieben werden, um eine bessere Vergleichbarkeit der Ergebnisse zu ermöglichen.

5.2.1 Hermann et al.

Ein Ausschnit von der Exportdatei der Reliabilitätsuntersuchung mit IIRA ist im Anhang A.15 dargestellt. Von den 94 Probanden, die an der Umfrage von Hermann et al. [23] teilgenommen haben, haben insgesamt 22 Teilnehmer keines der vier Duplikate mehrfach bewertet. Somit bleiben 72 Bewerter übrig, von denen Intrarater-Übereinstimmungen berechnet werden können. Die Teilnehmer wurden in insgesamt vier Gruppen eingeteilt. Die erste Gruppe besteht aus 59 Bewertern, die jedes der vier Duplikate doppelt bewertet haben. Die zweite Gruppe besteht aus sechs Bewertern, die drei der Duplikate doppelt bewertet haben. In der dritten Gruppe gibt es drei Bewerter, die zwei der Duplikate doppelt bewertet haben und schließlich besteht die vierte Gruppe aus vier Probanden, die nur eines der Duplikate doppelt bewertet haben.
Damit die Ergebnisse mit denen aus der Arbeit von Hermann et al. [23] vergleichbar sind, wurden in der Tabelle 5.1 die Cohen's-Kappa-Werte der vier Gruppen dargestellt. Dabei wurde jeweils das Minimum und das Maximum der Cohen's-Kappa-Werte, in der jeweiligen Gruppe betrachtet, und zusätzlich der Durchschnittswert jeder Gruppe berechnet.

#Bewerter	#Duplikate	Min	Max	Mean
59	4	-0.3333	1.0	0.5469
6	3	0.0	1.0	0.5905
3	2	0.0	1.0	0.3333
4	1	1.0	1.0	1.0

Tabelle 5.1: Cohen's Kappa Werte der Intrarater-Reliabilitätsuntersuchung

Es fällt auf, dass es in jeder Gruppe Teilnehmer gibt, die vollständig mit ihren Bewertungen übereinstimmen. In diesen Fällen liefert Cohen's-Kappa einen Wert von 1.0. Ein Grund dafür kann die relativ kleine Anzahl an Textelementen sein, die miteinander verglichen worden sind.
Der niedrigste Cohen's Kappa Wert, in Höne von -0.3333, tritt in der ersten Gruppe auf. Der Bewerter hat ein Duplikat zweimal als neutral bewertet und in drei Fällen lag bei der Bewertung eine leichte Uneinigkeit vor. Ein zuvor als negativ bewertetes Duplikat wurde bei der zweiten Bewertung als neutral bewertet, sowie ein zuvor als neutral gelabeltes Duplikat jeweils als negativ

und eines als positiv.
Nach der Interpretation von Landis und Koch liefert die erste Gruppe, mit 59 Bewertern, moderate Übereinstimmungswerte. Die durchschnittlichen Übereinstimmungswerte in den restlichen Gruppen schwanken zwischen mäßigen, bis hin zu fast perfekten Übereinstimmungswerten in der letzten Gruppe. Allerdings bestehen die letzten drei Gruppen lediglich aus bis zu sechs Bewertern, was die Aussagekraft der Ergebnisse einschränkt.

5.2.2 Martensen

Um die Intrarater-Reliabilität bei der Labelvergabe zu überprüfen wurden die Bewerter, analog zu Martensen's Arbeit [31], in drei Gruppen eingeteilt. Die erste Gruppe besteht aus Personen, die an allen vier Umfragen teilgenommen und Label vergeben haben. Analog bestehen die zweite, bzw. die dritte Gruppe, aus Personen die an drei, bzw. an zwei, Umfragen teilgenommen und Label vergeben haben.
In Martensen's Arbeit [31] wurde die Zugehörigkeit zur jeweiligen Gruppe alleine durch die Teilnahme an den entsprechenden Umfragen bestimmt. Falls ein Bewerter an vier Umfragen teilgenommen hat, aber beispielsweise bei der letztem Umfrage lediglich Angaben zu der derzeitigen Lebenssituation gemacht und keine Label vergeben hat, so wäre er in Martensen's Arbeit [31] dennoch ein Mitglied in der ersten Gruppe.
Da in der vorliegenden Bachelorarbeit der Fokus auf der Auswertung der Intrarater-Reliabilität liegt, wird die Gruppenzugehörigkeit strenger definiert und zusätzlich die Vergabe von Labeln, bei den Umfragen, gefordert. Die Intrarater-Übereinstimmungen wurden erneut für jeden Bewerter mithilfe von IIRA berechnet. Ein Ausschnitt von der Exportdatei der Auswertungen ist im Anhang A.16 zu finden. Nach dem obigen Schema konnten anschließend die Gruppen gebildet werden, wobei insgesamt 9 Bewerter an vier Umfragen teilgenommen und Label vergeben haben, 14 Bewerter haben an drei Umfragen teilgenommen und 8 Bewerter haben an zwei Umfragen teilgenommen. Die Ergebnisse der Reliabilitätsuntersuchung wurden in der Tabelle 5.2 zusammengefasst.

		Fleiss' Kappa			**Gwet's AC_1**		
#Bewerter	**#Umfragen**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**
9	4	0.0853	0.7808	0.5090	0.5077	0.9541	0.7113
14	3	0.2354	0.7393	0.5418	0.2813	0.7724	0.6111
8	2	-0.0989	0.777	0.35538	0.0431	0.9311	0.6223

Tabelle 5.2: Fleiss' Kappa- und Gwet's AC_1 Werte der Intrarater-Reliabilitätsuntersuchung

Martensen [31] kam in seiner Argbeit zu dem Ergebnis, dass der durchschnittliche Fleiss' Kappa Wert bei der vierfachen Umfragenteilnahme bei 0.51, bei der dreifachen Teilnahme bei 0.53 und bei zweifacher Teilnahme bei 0.47 liegt. Die Abweichungen lassen sich durch die unterschiedlichen Definitionen der Gruppenzugehörigkeit erklären.
In der vorliegenden Bachelorarbeit wurden den Fleiss' Kappa Werten zusätzlich die Gwet's AC_1 Werte gegenübergestellt. Beim Vergleich der Werte lässt sich feststellen, dass es teilweise große Abweichungen voneinander gibt. So liegt das Minimum der Fleiss' Kappa Werte bei der vierfachen Teilnahme bei 0.0853, während das Minimum der Gwet's AC_1 Werte bei 0.5090 liegt.
In der Tabelle 5.3 wurden die Fleiss' Kappa-, bzw. die Gwet's AC_1 Werte, der drei Bewerter dargestellt, bei denen es die größten Abweichungen gab. Die Abweichungen bei den anderen Bewertern waren wesentlich geringer.

Bewerter ID	**Fleiss' Kappa**	**Gwet's AC_1**
25JrB	0.0853	0.6878
30NgV	0.3103	0.9541
08PvU	-0.0345	0.9311

Tabelle 5.3: Gegenüberstellung der Fleiss' Kappa- und Gwet's AC_1 Werte mit den größten Abweichungen

Möglicherweise handelt es sich bei den Fleiss' Kappa Werten um Kappa-Paradoxien [21]. Der Bewerter 08PvU hat an insgesamt zwei Umfragen teilgenommen. Die Auswertung der Labelvergabe hat ergeben, dass er bei der ersten Umfrage alle 30 Textelemente als neutral klassifiziert hat. Bei der zweiten Umfrage wurden 2 positive Label vergeben und die restlichen 28 Textelemente wurden als neutral klassifiziert. Intuitiv würde man in dem Fall von einer fast perfekten Intrarater-Reliabilität ausgehen, weshalb der Gwet's AC_1 Wert, in Höhe von 0.9311, als der geeignetere Wert erscheint. Dem gegenüber steht ein Fleiss' Kappa Wert von -0.0345, der nach der Interpretation von Landis und Koch, eine schlechte Übereinstimmung ausdrückt.
Die Auswertung der Labelvergabe von dem Bewerter 30NgV lässt ähnliche Rückschlüsse zu. Der Bewerter hat an allen vier Umfragen teilgenommen und in allen Umfragen jeweils 29 Textelemente als neutral und jeweils eines als negativ klassifiziert. Dabei wurden in allen vier Umfragen, die gleichen 28 Textelemente, übereinstimmend als neutral bewertet. Dennoch liefert der Fleiss' Kappa Wert von 0.3103 nur eine mäßige Übereinstimmung.
Der Bewerter 25JrB hat ebenfalls an vier Umfragen teilgenommen und übereinstimmend, in jeder Umfrage, 17 Textelemente als neutral bewertet. Die Labelvergabe der drei Bewerter wurde zusätzlich im Anhang A.17 abgebildet.

Kapitel 6

Zusammenfassung und Ausblick

Dieses Kapitel fasst die wesentlichen Aspekte der Bachelorarbeit zusammen. Anschließend wird ein Ausblick über mögliche Fortsetzungen der Arbeit gegeben.

6.1 Zusammenfassung

Um Reliabilitätsuntersuchungen vorzunehmen, formatieren die Untersucher typischerweise Excel-Tabellen, auf eine geeignete Art und Weise, und berechnen die Ergebnisse anschließend manuell. IIRA bietet die Möglichkeit, mit geringem Formatierungsaufwand, Intra- und Interrater-Reliabilitätsuntersuchungen zu erstellen. Die SW-User haben die Auswahl zwischen bis zu sechs Metriken, wobei die Anwendung Erklärungen zu den Metriken liefert und eine geeignete Vorauswahl der Metriken trifft. So werden dem Nutzer nur die Metriken angezeigt, die für die Analyse der zugrundeliegenden Daten geeignet sind. Schließlich können die Untersuchungen als Excel-, LibreOffice-Calc-, oder CSV-Datei exportiert werden, wobei zusätzlich Informationen zu den p-Werten und den 95%-Konfidenzintervallen der Untersuchungen, in der Exportdatei, zu finden sind.
Darüber hinaus bietet IIRA die Möglichkeit Bewertungsobjekte zu labeln. Um äußere Einflüße bei der Bewertung zu minimieren, wird der Fokus in der Bewerten-Ansicht auf den zu bewertenden Text gelegt. Es kann nicht eingesehen werden, wie andere Bewerter die Objekte klassifiziert haben, wie es möglicherweise in einer Excel-Tabelle der Fäll wäre. Zudem bietet IIRA die Möglichkeit, die Reihenfolge der Bewertungsobjekte zu randomisieren. So wird vermieden, dass sich die Bewerter an auftretende Patterns bei der Bewertung erinnern, was die Bewertung beeinflußen könnte.
Die Korrektheit der Software wurde im Rahmen der Bachelorarbeit

systematisch getestet und zum Abschluss wurden mithilfe von IIRA zwei Intrarater-Reliabilitätsuntersuchungen vorgenommen.

6.2 Ausblick

Im Rahmen der Bachelorarbeit konnten nur einige Reliabilitätsmetriken selektiv implementiert werden. Die Einbindung zusätzlicher Metriken, wie dem G-Index [22], würden eine sinnvolle Erweiterung der Software darstellen. Zusätzlich können weitere statistische Maße eingebunden werden, wie dem Berechnen des Mittelwertes, oder des Medians. So könnte IIRA von einem Tool zur Berechnung von Intra- und Interrater-Reliabilitäten zu einem Tool zur Berechnung von statistischen Zusammenhängen verallgemeinert werden. Außerdem kann Natural Language Processing betrieben werden, um die Software benutzerfreundlicher zu gestalten. Beim Import der Datensätze werden beispielsweise bestimmte Header-Namen vorgegeben. Durch geeignete Weiterentwicklungen ließe sich realisieren, dass einige Tippfehler erkannt und behoben werden können. Auch bei der Auswahl der Bewerter wird derzeitig auf Groß- und Kleinschreibung geachtet. Die Bewerter „Alice“ und „alice“ würden also als zwei unterschiedliche Bewerter aufgefasst werden. Da dieser Sachverhalt bei den Analysen möglicherweise nicht erwünscht ist, wäre es sinnvoll IIRA um einen Modus zu ergänzen, in dem der SW-User die Möglichkeit hat zu entscheiden, ob unterschiedliche Groß- und Kleinschreibungen beachetet werden sollen, oder nicht.

Anhang A

Anhang

A.1 Softwarearchitektur

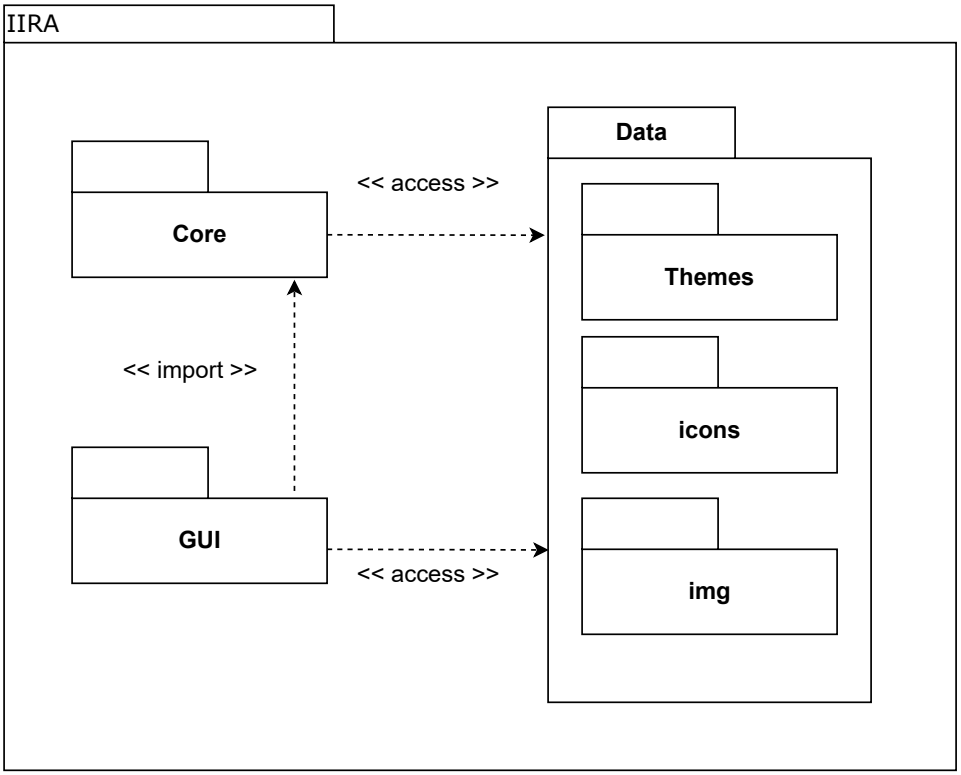

Abbildung A.1: Package-Struktur von IIRA

A.2 GUI

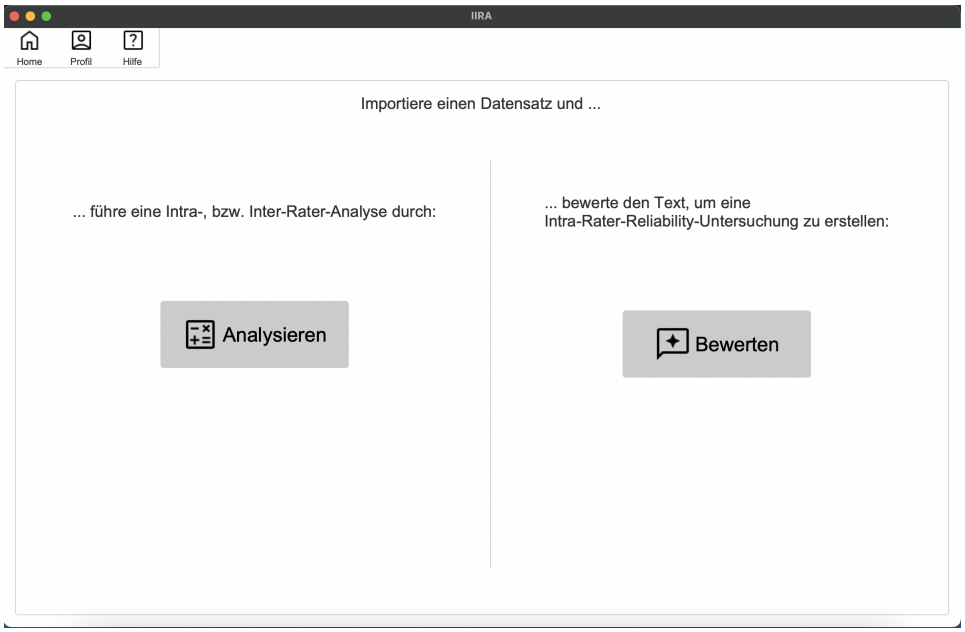

Abbildung A.2: Hauptfenster

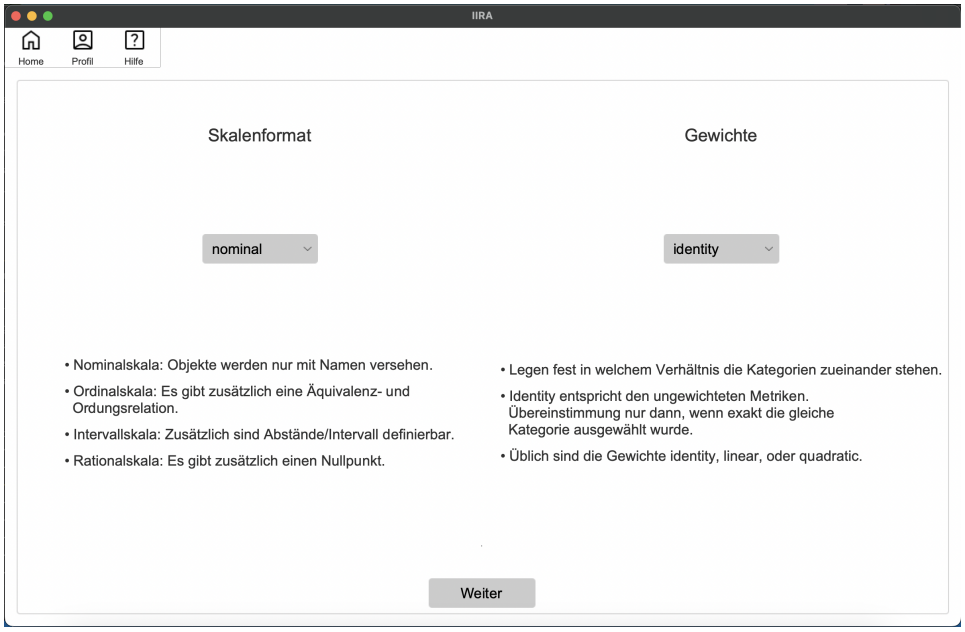

Abbildung A.3: Skalenformat auswählen - Analysieren

Abbildung A.4: Skalenformat auswählen - Bewerten

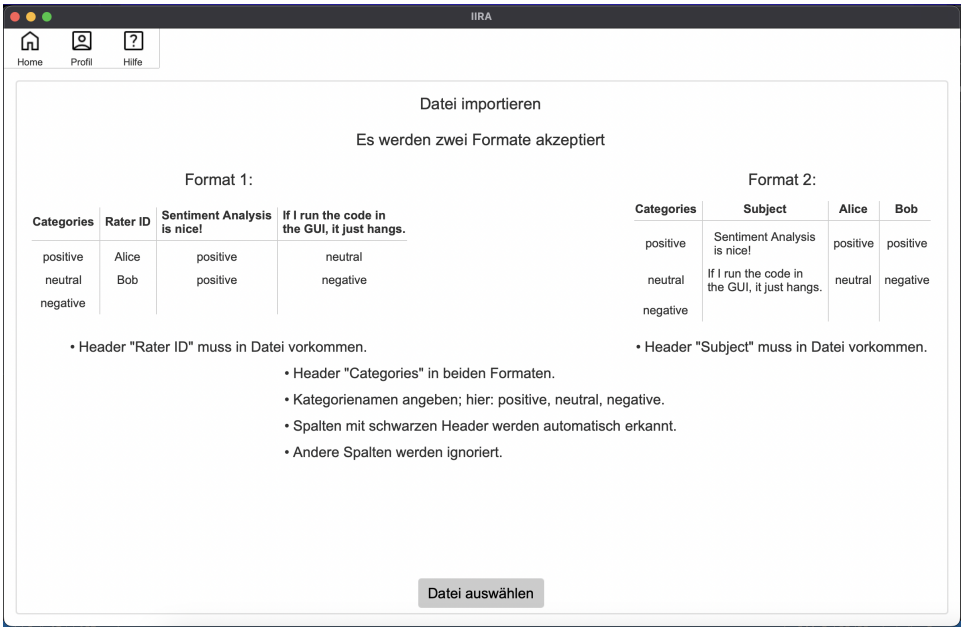

Abbildung A.5: Datei importieren - diskretes Skalenformat

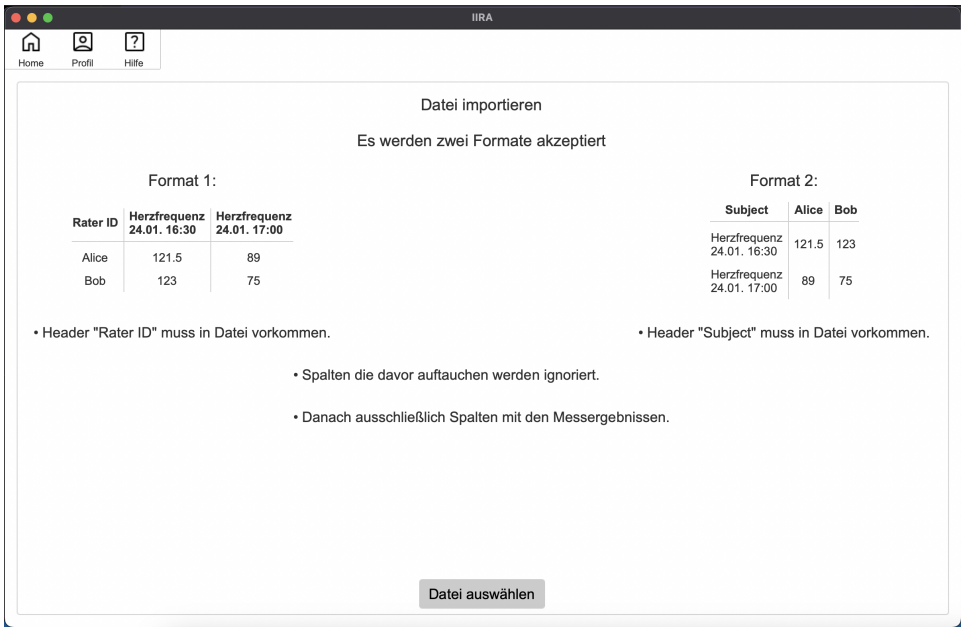

Abbildung A.6: Datei importieren - kontinuierliches Skalenformat

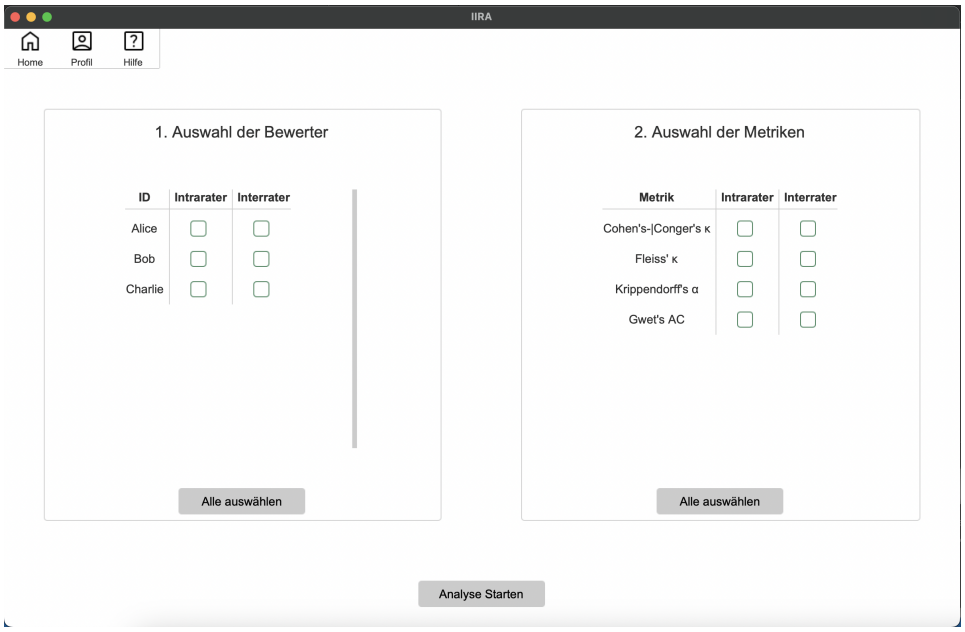

Abbildung A.7: Reliabilitätsuntersuchung vorbereiten

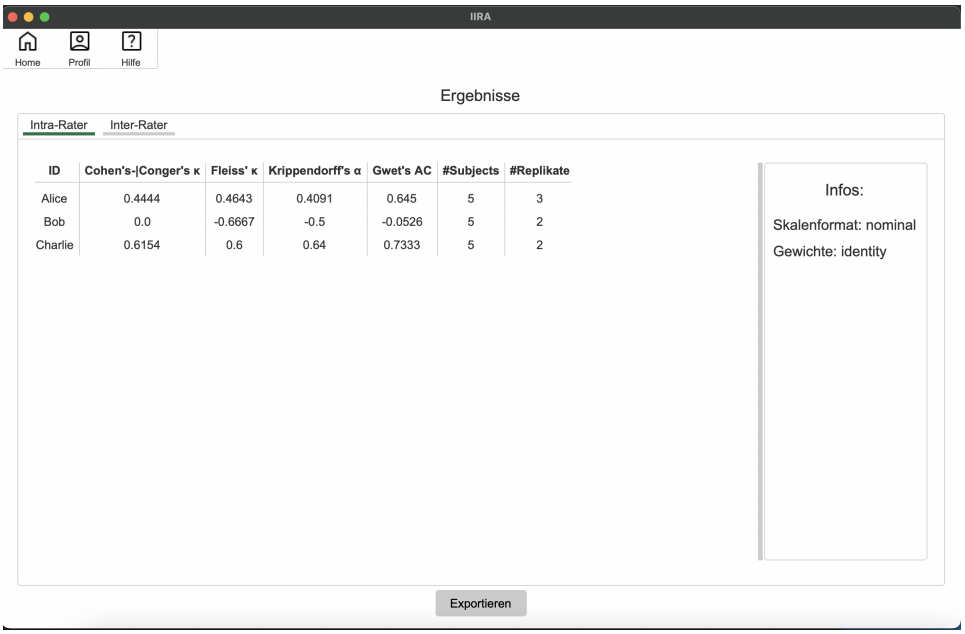

Abbildung A.8: Intrarater-Ergebnisse

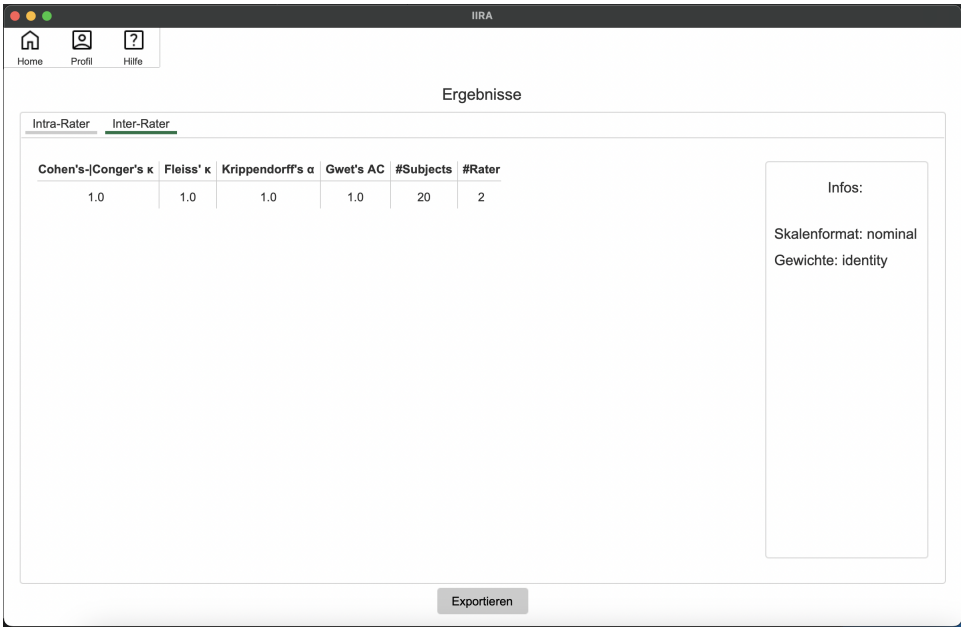

Abbildung A.9: Interrater-Ergebnisse

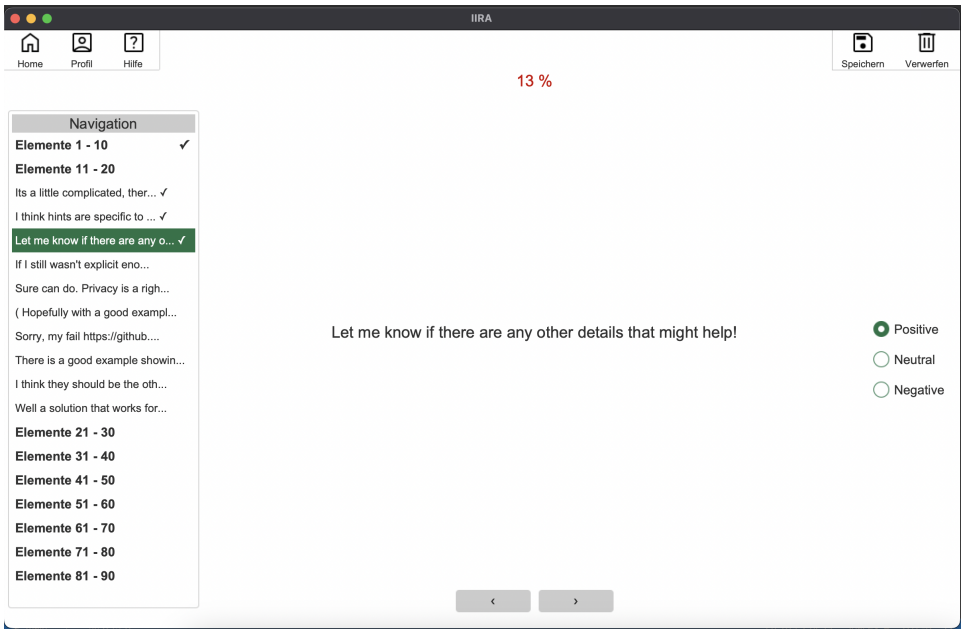

Abbildung A.10: Bewerten-Fenster

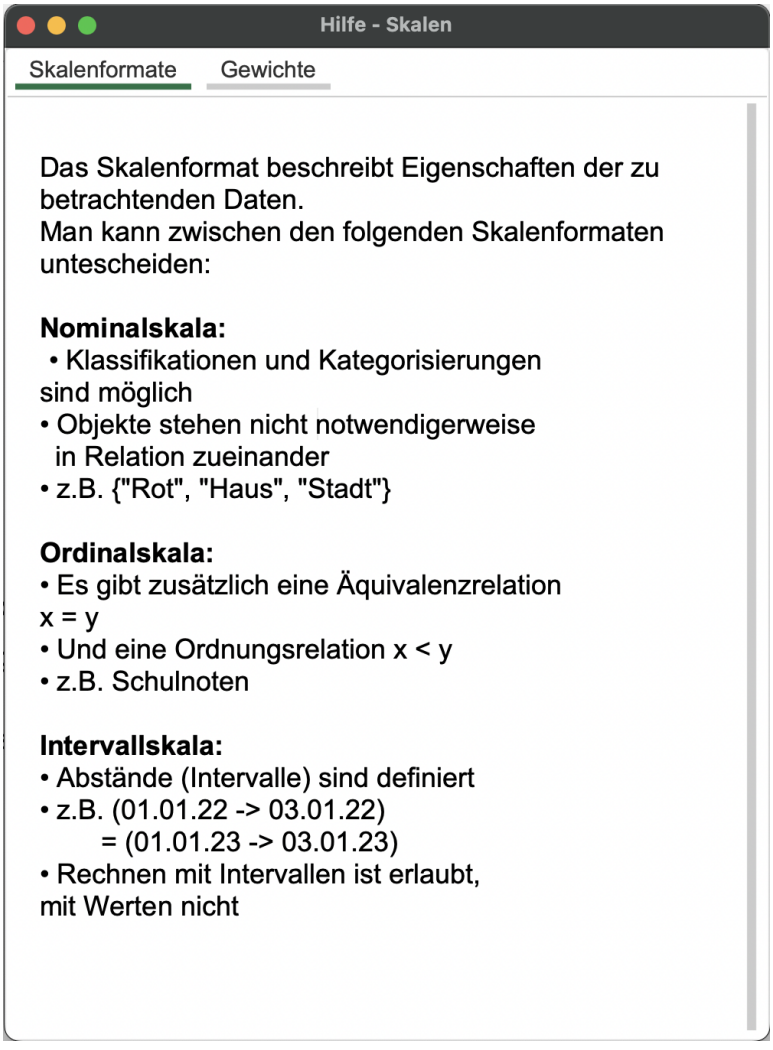

Abbildung A.11: Hilfe-Fenster

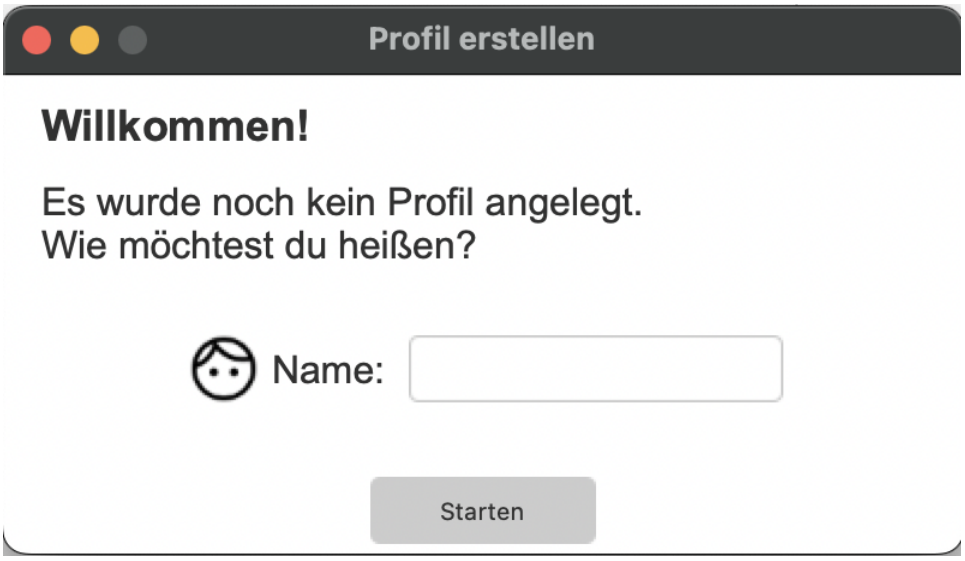

Abbildung A.12: Willkommen-Fenster

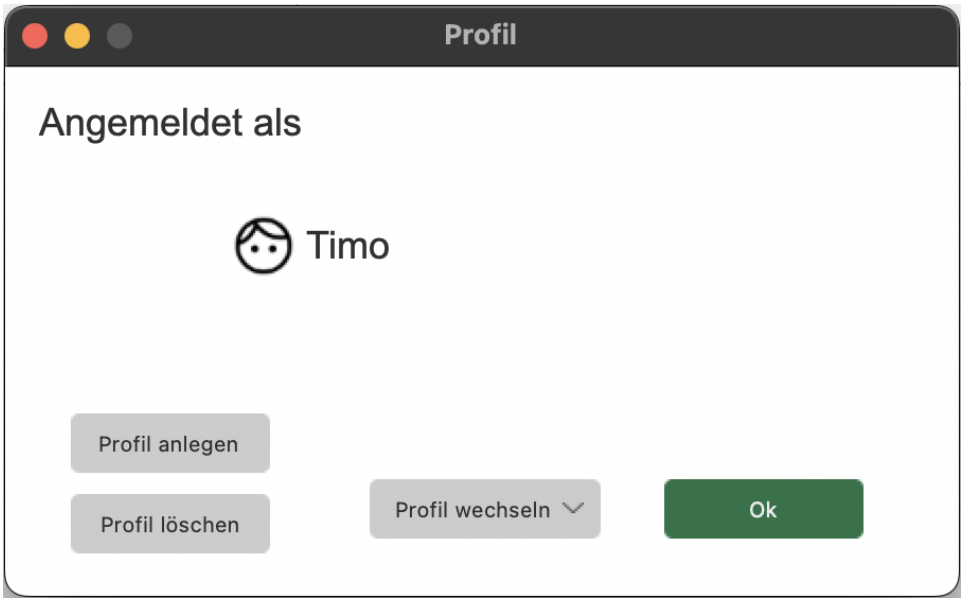

Abbildung A.13: Profil-Fenster

A.3 Dateiexport

	A	B	C
1	**Intra-Rater-Analyse**		
2			
3	**Gewichte**		
4	identity		
5			
6			
7	**Rater ID**	**#Subjects**	**#Replikate**
8	Alice	5	3
9			
10	**Cohen's-|Conger's κ**	**p-Wert**	**95% Konfidenzintervall**
11	0.4444	0.24325195974229485	(-0.4579, 1)
12			
13	**Fleiss' κ**	**p-Wert**	**95% Konfidenzintervall**
14	0.4643	0.27301900158833026	(-0.5509, 1)
15			
16	**Krippendorff's α**	**p-Wert**	**95% Konfidenzintervall**
17	0.4091	0.35868025361316125	(-0.6872, 1)
18			
19	**Gwet's AC**	**p-Wert**	**95% Konfidenzintervall**
20	0.645	0.04483340063951813	(0.024, 1)
21			

Abbildung A.14: Ausschnit aus der Exportdatei der Reliabilitätsuntersuchung

A.4 Intrarater-Reliabilitätsuntersuchungen

	A	B	C	D	E	F	G
1	Intra-Rater-Analyse						
2							
3	Gewichte						
4	identity						
5							
6							
7	Rater ID	#Subjects	#Replicates	Cohen's-|Conger's κ	Fleiss' κ	Krippendorff's α	Gwet's AC
8	5	4	2	0.6	0.5789	0.6316	0.6444
9	6	4	2	1.0	1.0	1.0	1.0
10	7	3	2	0.4	0.3333	0.4444	0.5556
11	8	4	2	0.6364	0.619	0.6667	0.6279
12	9	2	2	1.0	1.0	1.0	1.0
13	11	4	2	1.0	1.0	1.0	1.0
14	12	4	2	1.0	1.0	1.0	1.0
15	15	4	2	0.3333	0.2	0.3	0.2727
16	16	4	2	0.5	0.4667	0.5333	0.6735
17	17	4	2	1.0	1.0	1.0	1.0
18	19	4	2	-0.3333	-0.4118	-0.2353	-0.0213
19	22	4	2	1.0	1.0	1.0	1.0
20	23	3	2	0.0	0.0	0.1667	0.0
21	24	4	2	1.0	1.0	1.0	1.0
22	29	4	2	1.0	1.0	1.0	1.0
23	32	4	2	0.5556	0.5294	0.5882	0.6596
24	33	4	2	0.2	0.2	0.3	0.2727
25	35	4	2	-0.0909	-0.2632	-0.1053	-0.0667
26	39	4	2	0.0	-0.1429	0.0	0.7193
27	41	4	2	1.0	1.0	1.0	1.0
28	44	4	2	0.6364	0.619	0.6667	0.6279
29	45	1	2	1.0	1.0	1.0	1.0
30	46	4	2	1.0	1.0	1.0	1.0
31	47	4	2	1.0	1.0	1.0	1.0
32	48	4	2	0.5556	0.5294	0.5882	0.6596
33	50	1	2	1.0	1.0	1.0	1.0

Abbildung A.15: Ausschnitt vom IIRA-Export der Reliabilitätsuntersuchung des Datensatzes von Hermann et al.

	A	B	C	D	E	F
1	Intra-Rater-Analyse					
2						
3	Gewichte					
4	identity					
5						
6						
7	Rater ID	#Subjects	#Replicat	Fleiss' κ	Gwet's AC	
8	15MeK	30	3	0.5102	0.6123	
9	25JrB	30	4	0.0853	0.6878	
10	01LeG	30	4	0.6268	0.6365	
11	07MiK	30	4	0.5312	0.5825	
12	15KhL	30	4	0.7808	0.8197	
13	07MäW	30	3	0.5382	0.6676	
14	16OzG	30	4	0.6288	0.7942	
15	09JaE	30	4	0.3947	0.5077	
16	16TrA	30	3	0.5288	0.5834	
17	12MüD	10	2	0.8058	0.8653	
18	19JrE	30	2	0.3382	0.4268	
19	07NaW	30	4	0.6094	0.7881	
20	15YeN	30	2	0.5161	0.5652	
21	02MnI	30	2	0.67	0.777	

Abbildung A.16: Ausschnitt vom IIRA-Export der Reliabilitätsuntersuchung des Datensatzes von Martensen

Intra Ratings for ID 30NgV:

	0	1	2	3
[Very good example of steady pooling readHere.]	neutral	neutral	neutral	neutral
[It does its job, but was built for our use ca...	neutral	neutral	neutral	neutral
[The following is the error I keep getting.]	neutral	neutral	neutral	neutral
[I don't know what else to do to make things t...	neutral	neutral	neutral	neutral
[Hope this helps.]	neutral	neutral	neutral	neutral
[If I still wasn't explicit enough: 9cc4976393...	neutral	neutral	neutral	neutral
[@timmywil Sounds good!]	neutral	neutral	neutral	neutral
[i was afraid it'd do unimaginable things!]	neutral	neutral	neutral	neutral
[OMG stupid me]	negativ	neutral	negativ	neutral
[final class?]	neutral	neutral	neutral	neutral
[The following code therefore, might be (as I...	neutral	neutral	neutral	neutral
[We ran into the same sort of problem with Fle...	neutral	neutral	neutral	neutral
[The data structure you are saving your data i...	neutral	neutral	neutral	neutral
[Here's an example of a JSON structure.]	neutral	neutral	neutral	neutral
[It dependes where it is exactly located.]	neutral	neutral	neutral	neutral
[Most awesome! :+1:]	neutral	neutral	neutral	neutral
[This is the only one that bothers me. If an o...	neutral	neutral	neutral	neutral
[That's what obfuscation does. We do our best ...	neutral	neutral	neutral	neutral
[oh nice find, that's been bugging the crap ou...	neutral	neutral	neutral	neutral
[md5 not good enough for you?]	neutral	neutral	neutral	neutral
[(Hopefully with a good example.)]	neutral	neutral	neutral	neutral
[Now we're getting to the good part.]	neutral	neutral	neutral	neutral
[I have looked and found that there are some p...	neutral	neutral	neutral	neutral
[help!]	neutral	neutral	neutral	neutral
[which someone converted to a JSON string as f...	neutral	neutral	neutral	neutral
[Yay for improving consistency, +1]	neutral	neutral	neutral	neutral
[lol :)]	neutral	negativ	neutral	negativ
[Why was this reverted? :(]	neutral	neutral	neutral	neutral
[I ended up using `strtotime()` as the various...	neutral	neutral	neutral	neutral
[Is this good to go then?]	neutral	neutral	neutral	neutral

Abbildung A.17: Labelvergabe der Rater ID 30NgV

Intra Ratings for ID 25JrB:

	0	1	2	3
[Very good example of steady pooling readHere.]	neutral	neutral	neutral	neutral
[It does its job, but was built for our use ca...	neutral	neutral	neutral	neutral
[The following is the error I keep getting.]	neutral	positiv	neutral	negativ
[I don't know what else to do to make things t...	neutral	neutral	negativ	neutral
[Hope this helps.]	neutral	positiv	negativ	negativ
[If I still wasn't explicit enough: 9cc4976393...	neutral	neutral	neutral	neutral
[@timmywil Sounds good!]	neutral	neutral	neutral	neutral
[i was afraid it'd do unimaginable things!]	neutral	neutral	neutral	neutral
[OMG stupid me]	neutral	neutral	neutral	negativ
[final class?]	neutral	positiv	positiv	negativ
[The following code therefore, might be (as I...	neutral	neutral	neutral	neutral
[We ran into the same sort of problem with Fle...	neutral	neutral	neutral	neutral
[The data structure you are saving your data i...	neutral	positiv	negativ	neutral
[Here's an example of a JSON structure.]	neutral	neutral	neutral	neutral
[It dependes where it is exactly located.]	neutral	negativ	neutral	neutral
[Most awesome! :+1:]	neutral	neutral	negativ	neutral
[This is the only one that bothers me. If an o...	neutral	neutral	neutral	neutral
[That's what obfuscation does. We do our best ...	neutral	neutral	neutral	neutral
[oh nice find, that's been bugging the crap ou...	neutral	neutral	neutral	neutral
[md5 not good enough for you?]	neutral	neutral	negativ	neutral
[(Hopefully with a good example.)]	neutral	neutral	neutral	neutral
[Now we're getting to the good part.]	neutral	neutral	neutral	neutral
[I have looked and found that there are some p...	neutral	neutral	neutral	neutral
[help!]	neutral	positiv	neutral	positiv
[which someone converted to a JSON string as f...	neutral	neutral	neutral	neutral
[Yay for improving consistency, +1]	neutral	neutral	positiv	neutral
[lol :)]	neutral	neutral	neutral	negativ
[Why was this reverted? :(]	neutral	neutral	neutral	positiv
[I ended up using `strtotime()` as the various...	neutral	neutral	neutral	neutral
[Is this good to go then?]	neutral	neutral	neutral	neutral

Abbildung A.18: Labelvergabe der Rater ID 25JrB

Intra Ratings for ID 08PvU:

	0	1
[Very good example of steady pooling readHere.]	neutral	neutral
[It does its job, but was built for our use ca...	neutral	neutral
[The following is the error I keep getting.]	neutral	neutral
[I don't know what else to do to make things t...	neutral	neutral
[Hope this helps.]	neutral	positiv
[If I still wasn't explicit enough: 9cc4976393...	neutral	neutral
[@timmywil Sounds good!]	neutral	neutral
[i was afraid it'd do unimaginable things!]	neutral	neutral
[OMG stupid me]	neutral	neutral
[final class?]	neutral	positiv
[The following code therefore, might be (as I...	neutral	neutral
[We ran into the same sort of problem with Fle...	neutral	neutral
[The data structure you are saving your data i...	neutral	neutral
[Here's an example of a JSON structure.]	neutral	neutral
[It dependes where it is exactly located.]	neutral	neutral
[Most awesome! :+1:]	neutral	neutral
[This is the only one that bothers me. If an o...	neutral	neutral
[That's what obfuscation does. We do our best ...	neutral	neutral
[oh nice find, that's been bugging the crap ou...	neutral	neutral
[md5 not good enough for you?]	neutral	neutral
[(Hopefully with a good example.)]	neutral	neutral
[Now we're getting to the good part.]	neutral	neutral
[I have looked and found that there are some p...	neutral	neutral
[help!]	neutral	neutral
[which someone converted to a JSON string as f...	neutral	neutral
[Yay for improving consistency, +1]	neutral	neutral
[lol :)]	neutral	neutral
[Why was this reverted? :(]	neutral	neutral
[I ended up using `strtotime()` as the various...	neutral	neutral
[Is this good to go then?]	neutral	neutral

Abbildung A.19: Labelvergabe der Rater ID 08PvU

Literaturverzeichnis

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

[2] R. L. Brennan and D. J. Prediger. Coefficient kappa: Some uses, misuses, and alternatives. *Educational and psychological measurement*, 41(3):687–699, 1981.

[3] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli. Sentiment polarity detection for software development. In *Proceedings of the 40th International Conference on Software Engineering*, pages 128–128, 2018.

[4] Z. Chen, Y. Cao, X. Lu, Q. Mei, and X. Liu. Sentimoji: an emoji-powered learning approach for sentiment analysis in software engineering. In *Proceedings of the 2019 27th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, pages 841–852, 2019.

[5] Z. Chen, Y. Cao, H. Yao, X. Lu, X. Peng, H. Mei, and X. Liu. Emoji-powered sentiment and emotion detection from software developers' communication data. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 30(2):1–48, 2021.

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

[7] J. Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

[8] A. J. Conger. Integration and generalization of kappas for multiple raters. *Psychological Bulletin*, 88(2):322, 1980.

[9] G. Destefanis, M. Ortu, S. Counsell, S. Swift, M. Marchesi, and R. Tonelli. Software development: do good manners matter? *PeerJ Computer Science*, 2:e73, 2016.

[10] R. J. Dolan. Emotion, cognition, and behavior. *science*, 298(5596):1191–1194, 2002.

[11] A. R. Feinstein and D. V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549, 1990.

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

[13] P. Fitsilis. Measuring the complexity of software projects. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 7, pages 644–648. IEEE, 2009.

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

[15] J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973.

[16] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. *Icwsm*, 7(21):219–222, 2007.

[17] P. Graham and R. Jackson. The analysis of ordinal agreement data: beyond weighted kappa. *Journal of clinical epidemiology*, 46(9):1055–1062, 1993.

[18] D. Graziotin, X. Wang, and P. Abrahamsson. Are happy developers more productive? the correlation of affective states of software developers and their self-assessed productivity. In *Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings 14*, pages 50–64. Springer, 2013.

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

[20] K. Gwet. Handbook of inter-rater reliability. *Gaithersburg, MD: STATAXIS Publishing Company*, pages 223–246, 2001.

[21] K. L. Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

[22] K. L. Gwet. Intrarater reliability. *Wiley encyclopedia of clinical trials*, 4, 2008.

[23] M. Herrmann, M. Obaidi, L. Chazette, and J. Klünder. On the subjectivity of emotions in software projects: How reliable are pre-labeled data sets for sentiment analysis? *Journal of Systems and Software*, 193:111448, 2022.

[24] M. R. Islam and M. F. Zibran. Exploration and exploitation of developers' sentimental variations in software engineering. *International Journal of Software Innovation (IJSI)*, 4(4):35–55, 2016.

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

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

[27] T. K. Koo and M. Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine*, 15(2):155–163, 2016.

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

[29] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto. Sentiment analysis for software engineering: How far can we go? In *Proceedings of the 40th international conference on software engineering*, pages 94–104, 2018.

[30] M. Maclure and W. C. Willett. Misinterpretation and misuse of the kappa statistic. *American journal of epidemiology*, 126(2):161–169, 1987.

[31] J. Martensen. Analyse von einflüssen in der sentimenterkennung von entwicklern. 2022.

[32] S. R. Munoz and S. I. Bangdiwala. Interpretation of kappa and b statistics measures of agreement. *Journal of Applied Statistics*, 24(1):105–112, 1997.

[33] A. Murgia, P. Tourani, B. Adams, and 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*, pages 262–271, 2014.

[34] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and 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*, pages 158–168, 2020.

[35] N. Novielli, D. Girardi, and F. Lanubile. A benchmark study on sentiment analysis for software engineering research. In *Proceedings of the 15th International Conference on Mining Software Repositories*, pages 364–375, 2018.

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

[37] M. Obaidi, L. Nagel, A. Specht, and J. Klünder. Sentiment analysis tools in software engineering: A systematic mapping study. *Information and Software Technology*, page 107018, 2022.

[38] B. Pang, L. Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135, 2008.

[39] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Twenty-First International Joint Conference on Artificial Intelligence*. Citeseer, 2009.

[40] K. Schneider. *Abenteuer Softwarequalität: Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement*. dpunkt. verlag, 2012.

[41] K. Schneider, J. Klünder, F. Kortum, L. Handke, J. Straube, and S. Kauffeld. Positive affect through interactions in meetings: The role of proactive and supportive statements. *Journal of Systems and Software*, 143:59–70, 2018.

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

[43] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, A. Ihara, and K. Matsumoto. The impact of mislabelling on the performance and interpretation of defect prediction models. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 812–823. IEEE, 2015.

[44] M. Thelwall. The heart and soul of the web? sentiment strength detection in the social web with sentistrength. *Cyberemotions: Collective emotions in cyberspace*, pages 119–134, 2017.

[45] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558, 2010.

[46] J. Wu, C. Ye, and H. Zhou. Bert for sentiment classification in software engineering. In *2021 International Conference on Service Science (ICSS)*, pages 115–121. IEEE, 2021.

[47] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and 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)*, pages 70–80. IEEE, 2020.