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

Automatische Erkennung von Ironie in der Kommunikation von Softwareentwicklungsteams

Automated Irony Detection in Communication of Software Development Teams

Masterarbeit

im Studiengang Technische Informatik

von

Marnie Celina Schierholz

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Ann-Christin Klünder
Betreuer: Dr. rer. nat. Jil Ann-Christin Klünder

Hannover, 11.04.2022

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 11.04.2022

Marnie Celina Schierholz

Kurzfassung

Der Erfolg von Softwareprojekten hängt zu einem nicht unwesentlichen Anteil von der Zufriedenheit der Softwareentwickelnden ab. Deshalb ist es wichtig, die Stimmung innerhalb eines Teams während eines Projektes nicht zu vernachlässigen. Durch Sentiment-Analyse-Tools kann anhand von Meetingtranskripten, GitHub-Kommentaren oder Chat-Verläufen eine automatische Klassifizierung von Stimmung innerhalb eines Softwareprojekts detektiert werden. Ein oft berichtetes Problem sind dabei jedoch Fehlklassifizierungen aufgrund der Schwierigkeit Ironie zu erkennen. Deshalb untersucht diese Arbeit die computergestützte Erkennung von Ironie in Konversationen innerhalb des Software Engineerings. Dafür wurde ein Datensatz erhoben, welcher Audiomitschnitte und deren transkribierte Aussagen enthält sowie eine Klassifizierung der Aussagen in *Ironie* und *Keine Ironie*. Für die computergestützte Analyse wurden aus jeder der Aussagen 22 Merkmale extrahiert. Davon ergeben sich 14 Merkmale aus den Audiomitschnitten und die restlichen acht sind verbale Merkmale aus den Transkriptionen. Bei den verbalen Merkmalen handelt es sich um Stimmungsmerkmale, Merkmale zur Häufigkeit der Anwendung bestimmter Wörter im ironischen Kontext, sowie Wortarten, die in ironischen Aussagen gehäuft auftreten können. Mit unterschiedlichen Kombinationen dieser Merkmale wurden durch eine Support Vector Machine mehrere Klassifizierungsmodelle erstellt und anschließend analysiert. Die besten Ergebnisse ergaben sich aus einer Kombination aller Merkmale, durch die durchschnittlich 75% der ironischen Aussagen detektiert werden konnten, wobei allerdings auch durchschnittlich 19% der nicht ironischen Aussagen fälschlicherweise als ironisch klassifiziert wurden. Die Ergebnisse der Arbeit zeigen, dass es durchaus möglich ist, relativ zuverlässig *potenzielle Ironie* automatisch zu detektieren. Zukünftige Forschungen sollten daher den Nutzen der Ironieerkennung für Sentiment-Analyse-Tools näher untersuchen.

Abstract

Automated Irony Detection in Communication of Software Development Teams

The success of software projects is influenced by the mood of the software developers. Therefore, it is important not to ignore the mood within a team during a project. Sentiment analysis tools can detect an automatic classification of mood within a software project based on meeting transcripts, GitHub comments or chat histories. However, an often reported problem with this is misclassification due to the difficulty in detecting irony. However, a frequently reported problem with this is the misclassification of ironic statements. Therefore, this thesis investigates the computer-aided detection of irony in conversations within software engineering. For this purpose, a dataset was collected containing audio recordings, and their transcribed statements as well as a classification of the statements into *irony* and *no irony*. For the computer-assisted analysis, 22 features were extracted from each of the statements. Of these, 14 features are derived from the audio recordings and the remaining eight are verbal features from the transcriptions. The verbal features are mood features, features related to the frequency of use of certain words in the ironic context, and word types that may be clustered in ironic statements. Using different combinations of these features, several classification models were created by a Support Vector Machine and then analyzed. The best results came from a combination of all features, which detected an average of 75% of ironic statements, although an average of 19% of non-ironic statements were also misclassified as ironic. The results of this work show, that it is quite possible to automatically detect *potential irony* relatively reliable. Future research should therefore further investigate the utility of irony detection for sentiment analysis tools.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation . . . 1
1.2 Problemstellung und Ziele . . . 2
1.3 Lösungsansatz . . . 3
1.4 Struktur der Arbeit . . . 3

2 Grundlagen **5**
2.1 Ironie . . . 5
2.1.1 Geschichte des Ironiebegriffs . . . 5
2.1.2 Definition des Ironiebegriffs in dieser Arbeit . . . 7
2.1.3 Sarkasmus . . . 8
2.1.4 Ironie in der Anwendung . . . 9
2.1.5 Motive für die Verwendung von Ironie . . . 10
2.1.6 Merkmale der Ironie . . . 11
2.2 Maschinelles Lernen mit Support Vector Machines . . . 13
2.3 Natural Language Processing . . . 15
2.4 Mel Frequency Cepstral Coefficent . . . 17

3 Verwandte Arbeiten **21**

4 Konzept und prototypische Umsetzung **25**
4.1 Vorüberlegungen . . . 25
4.2 Aktueller Stand der Forschung . . . 26
4.3 Datenbasis . . . 26
4.3.1 Datenerhebung . . . 27
4.3.2 Analyse des Datensatzes . . . 30
4.4 Verarbeitungsprozess . . . 36
4.4.1 Erstellung des Trainings- und Testdatensatzes . . . 37
4.4.2 Preprocessing . . . 38
4.4.3 Datensatz ausgleichen . . . 38
4.4.4 Erstellung der Lexika . . . 39
4.4.5 Merkmalsextraktion . . . 40
4.4.6 Klassifizierungsmodell . . . 43

4.5 Metriken 44
4.6 Prototypische Umsetzung 46

5 Validierung **49**
5.1 Ziele der Validierung 49
5.2 Durchführung 49
5.3 Ergebnisse 51
5.3.1 Zusammenfassung der Ergebnisse 57
5.3.2 Beispiele aus einer Klassifikation 58
5.4 Zusätzliche Erkenntnisse 59

6 Diskussion **63**
6.1 Interpretation 63
6.2 Grenzen der Arbeit 65
6.3 Implikation für künftige Forschung 66

7 Zusammenfassung **69**

A Dokumente der Datenerhebung **71**
A.1 Aufgabenbeschreibung zur Gesprächsaufzeichnung 71
A.2 Aufgabenbeschreibung zur Annotation der Daten 72
A.3 Ausschnitt aus dem Datensatz 73

Literaturverzeichnis **74**

Abbildungsverzeichnis

2.1 Theoretisches Kontextmodell zur Beschreibung von Ironie (nach Tetzner [50] und Wallace [53]) . . . 9
2.2 Vektorraum einer Support Vector Machine mit Elementen aus zwei Klassen, die durch eine Hyperebene getrennt werden (nach Frochte [11]) . 13
2.3 Prozess zur Extraktion von MFCCs aus Audiomitschnitten (nach Arslan et al. [2]) . . . 18

4.1 Sprech- und Ironieanteile der einzelnen Teilnehmer in den drei Gruppen, wobei die Anteile an der Anzahl der getroffenen Äußerungen gemessen sind 32
4.2 Anteil der ironischen und nicht ironische Äußerungen der einzelnen Gruppen auf den gesamten Datensatz (1153 Aussagen) bezogen . . . 32
4.3 Anteile der kategorisierten Ironiemerkmale von den insgesamt 231 ironischen Aussagen . . . 33
4.4 Wortwolke aus ironisch gemeinten Aussagen, die verbale Ironiesignale enthalten (ohne Stoppwörter) . . . 34
4.5 Wortwolke aus ironisch gemeinten Aussagen, die **keine** verbale Ironiesignale enthalten (ohne Stoppwörter) . . . 35
4.6 Übergangsgraph für das Aufeinanderfolgen von ironischen (231 Aussagen) oder nicht ironischen Aussagen (922 Aussagen) . . . 36
4.7 Verarbeitungsprozess zur Klassifizierung ironischer und potenziell ironischer Aussagen . . . 37
4.8 Aufbau einer Konfusionsmatrix . . . 44

5.1 Prozess zur Berechnung der durchschnittlichen Kennzahlen einer Kombination von Merkmalen mittels Kreuzvalidierung . . . 51
5.2 Koeffizienzmatrizen für akustische Merkmale . . . 52
5.3 Koeffizienzmatrizen für Stimmungsmerkmale . . . 54
5.4 Koeffizienzmatrizen für Merkmale aus der Worthäufigkeit . . . 55
5.5 Koeffizienzmatrizen für Merkmale aus der Wortart . . . 56
5.6 Koeffizienzmatrizen für die Gesamtheit der Merkmale . . . 57
5.7 Koeffizienzmatrizen für Stimmungsmerkmalen welche auf einem angepassten, sowie unangepassten Sentiment-Lexikon basieren . . . 61

Tabellenverzeichnis

2.1 Beispiele für die Worstammreduktion und Lemmatisierung 16
2.2 Part-of-Speech-Tagging eines Beispielsatzes 17

4.1 Sprachliche Bewertungen von Ironie anhand von Merkmalen im Text . . . 29
4.2 Sprachliche Bewertungen von Ironie anhand von Merkmalen in der Stimme 29
4.3 Interpretation der κ-Werte nach Cohen [31] 30
4.4 Reliabilität in den einzelnen Gruppen 31
4.5 Verteilung der ironischen Aussagen auf die Wertungen verbaler und akustischer Merkmale 33
4.6 Auszug aus ironischem und nicht ironischem Lexikon 39
4.7 Berechnung der Stimmungsmerkmale 41
4.8 Berechnung des Ironiegrads 42
4.9 Berechnung der Häufigkeit bestimmter Wortarten oder Folgen von Wortarten 43

5.1 Kennzahlen für Akustische Merkmale 52
5.2 Kennzahlen für Stimmnugsmerkmale 53
5.3 Kennzahlen für Merkmale aus der Worthäufigkeit 54
5.4 Kennzahlen für Merkmale aus den Wortarten 56
5.5 Kennzahlen für die Gesamtheit der Merkmale 57
5.6 Zusammengefasste Ergebnisse aus den fünf Kreuzvalidierungen. 58
5.7 Kennzahlen für Stimmungsmerkmale welche auf einem angepassten, sowie unangepassten Sentiment-Lexikon basieren 60

Kapitel 1

Einleitung

Die Entwicklung von Software im Rahmen von Projekten wird von vielen Faktoren beeinflusst. Maßgeblich für einen erfolgreichen und zufriedenstellenden Projektabschluss ist das Team und die Kommunikation innerhalb des Teams [20]. Schlechte Kommunikation kann neben Missverständnissen auch die Stimmung der Teammitglieder negativ beeinflussen, welches durchaus auch den Projekterfolg gefährden kann [4].

1.1 Motivation

Die Zufriedenheit von Softwareentwickelnden hat einen Einfluss auf die Fähigkeiten Probleme in der Entwicklung von Software analytisch zu lösen [13]. Deshalb sollte die Stimmung der Teammitglieder innerhalb eines Projektes nicht vernachlässigt werden. Die Stimmung einer Person kann von vielen Faktoren abhängen. Unabhängig von den Einflussfaktoren, die sich ändern können, ist es von Vorteil, die Stimmung des Teams zu erkennen, damit eingegriffen werden kann, sobald die Stimmung das Arbeitsklima gefährdet. Aus diesem Grund entstanden viele Forschungsarbeiten, die sich mit der Sentiment-Analyse im Bereich des Software Engineerings befassen [35, 7, 8]. In den Forschungsarbeiten werden Stimmungen aus unterschiedlichen Kommunikationskanälen wie GitHub-Kommentaren, Chat-Verläufen oder transkribierten Meetings ermittelt, wobei durch die Vielfältigkeit der natürlichen Sprache eine Reihe von anspruchsvollen Problemstellungen entstehen [50]. Eine besondere Herausforderung besteht in der Erkennung von Ironie [17, 34, 1, 50]. Durch sie kann sich die Wahrnehmung einer Aussage komplett ändern, weshalb die bisherigen Tools zur Sentiment-Analyse Schwierigkeiten haben, die richtige Stimmung bei ironisch gemeinten Aussagen zu detektieren. Nimmt man die Aussage *„Das hast du ja mal wieder echt gut gemacht.“*, kann man diese entweder als Lob, oder beispielsweise als Tadel interpretieren. Ein automatisiertes Tool wird die Aussage durch die Schlüsselwörter *„gut gemacht“* wahrscheinlich als eine positive Aussage klassifizieren und die potenzielle Ironie, die vor allem durch die Wörter *„ja mal wieder echt“* erzeugt wird, ignorieren. Neben der wörtlichen Sprache kann der Tonfall eine große Aussagekraft über die Stimmung und auch über Ironie haben [15] und sollte deshalb nicht außer Acht gelassen werden.

Neben der Verbesserung von Sentiment-Analyse Tools kann eine automatische Erkennung von Ironie auch für die Verminderung von Missverständnissen eingesetzt werden. Wenn Rezipienten einer Aussage wissen, dass eine Äußerung möglicherweise ironisch gemeint ist, können sie dieser eine andere Bedeutung zuweisen. Zudem können Sprechende, wenn sie merken, dass sie keine eindeutige Aussage gemacht haben, diese verdeutlichen. Die Sprache kann von Team zu Team unterschiedlich sein. So kann es sein, dass in einem Team relativ viel Ironie genutzt wird, sodass neue Teammitglieder ironische Aussagen falsch interpretieren. In Videokonferenzen könnte ein kleines Hinweissymbol bei erkannter oder möglicher Ironie auf Ironie aufmerksam machen, sodass Sprechende und Rezipienten Aussagen besser einordnen können.

1.2 Problemstellung und Ziele

Das Ziel dieser Arbeit ist es, Ironie in Dialogen von Softwareentwicklungsteams automatisiert zu erkennen, bzw. die Frage zu beantworten, wie zuverlässig dies möglich ist. Durch die automatische Erkennung soll eine Optimierung von Sentiment-Analyse Tools ermöglicht werden. Zudem könnte eine Echtzeit-Erkennung von Ironie entwickelt werden, um Missverständnisse in der Kommunikation zu vermindern.

Die Erkennung von Ironie ist nicht nur in der computergestützten Analyse eine große Herausforderung. Die menschliche Interpretation von Aussagen kann auch erheblich voneinander abweichen. Bei Ironie handelt es sich um eine figurative Sprache, bei der sich die Intention der Sprechenden hinter einer wörtlichen Äußerung verbirgt. Um die eigentlich intendierte Äußerung hinter der wörtlich formulierten Aussage zu verstehen, sind ausgeprägte kognitive Fähigkeiten erforderlich [44]. Zusätzlich können semantische Informationen wie kulturelle bzw. soziale Hintergrundinformationen oder Kontextwissen zum Verstehen notwendig sein [44]. Es gibt zahlreiche Merkmale, die zum Verdeutlichen von Ironie eingesetzt werden können. Diese können in der Wortwahl, der Tonhöhe, der Mimik, oder nur im Kontext liegen. Ein großes Problem ist, dass jedes Merkmal, welches für Ironie genutzt wird, auch in nicht ironischen Aussagen verwendet werden kann [15]. Ein überschwängliches Lob wie *„Das hast du aber mal wieder richtig toll gemacht.“* würde in vielen Situationen als ironisch eingestuft werden, jedoch kann die Bemerkung vor allem in Konversationen von Erwachsenen zu Kindern durchaus ernst gemeint sein. Diese Mehrdeutigkeit macht die ironische Wirkung erst aus, erschwert jedoch die Identifikation erheblich [15]. Daraus ergibt sich, dass es keine festen ironischen Wörter und keine konkrete Syntax gibt, an denen Ironie eindeutig festgemacht werden kann [30]. Somit ist es ein weiteres Ziel dieser Arbeit, wichtige Merkmale für die Ironieerkennung herauszuarbeiten, sodass die automatische Ironieerkennung erleichtert wird.

Neben dem Problem, Merkmale von Ironie zu erkennen, liegt ein weiteres Problem darin, dass es keine eindeutige wissenschaftliche Definition von Ironie gibt. Dies erschwert die menschliche Klassifizierung von Ironie zusätzlich. Dem folgt daraus, dass Personen auch wenn sie die selben Ironiemerkmale einer Aussage bemerken, durch eine unterschiedliche Vorstellung von Ironie eine unterschiedlich Klassifizierung vornehmen würden.

Zusätzlich gibt es in jeder Sprache syntaktische und semantische Besonderheiten, sodass jede Sprache gesondert behandelt werden muss. Bestehende Forschungsarbeiten zur Ironieerkennung in der Sentiment-Analyse beziehen sich weitestgehend auf Texte in englischer, chinesischer oder italienischer Sprache [50].

Die Zielsetzung dieser Arbeit ist die Untersuchung und Optimierung des Einsatzes computergestützter Verfahren zur Identifikation von Ironie in deutschen Texten aus dem Bereich des Software Engineerings.

1.3 Lösungsansatz

Zunächst wird ein deutscher Datensatz aus dem Bereich des Software Engineerings mit ironischen und nicht ironischen Aussagen erstellt. Darauf folgt die Entwicklung eines Prototypen, der anhand von ausgewählten Merkmalen ironische Aussagen innerhalb des Datensatzes identifizieren soll. Die Merkmale sollen sich auf zwei Säulen stützen. Zum einen werden akustische Merkmale und zum anderen verbale Merkmale herausgearbeitet. Der Prototyp soll dafür auf etablierte Methoden der computergestützten Text- und Sprachanalyse zurückgreifen. Ein maschinelles Klassifizierungsverfahren soll aus den Merkmalen ein Klassifizierungsmodell erstellen, welches Aussagen in ironisch und nicht-ironisch unterteilt. Im Anschluss soll eine Aussage darüber getroffen werden können, ob eine automatische Ironieerkennung anhand der ausgewählten Merkmale möglich ist.

1.4 Struktur der Arbeit

Die Masterarbeit gliedert sich in acht Kapitel. Im zweiten Kapitel werden die grundlegenden Aspekte, die für das Verständnis der Arbeit notwendig sind, erklärt. Dabei handelt es sich vor allem um die Grundlagen zum Begriff Ironie und die für die Umsetzung eines Prototypen notwendigen technischen Voraussetzungen. Nachdem im dritten Kapitel auf verwandte Arbeiten eingegangen wird, wird im vierten Kapitel das Konzept der Arbeit vorgestellt. Dabei wird der im Rahmen dieser Arbeit erstellte Datensatz, sowie der Verarbeitungsprozess, der für eine Klassifizierung benötigt wird, behandelt. Außerdem wird auf die prototypische Umsetzung und die damit verbundenen Frameworks eingegangen. Die Bewertung des Prototypen erfolgt im fünften Kapitel. Anschließend folgt die Interpretation der Ergebnisse, sowie das aufzeigen der Grenzen und die Implikation für künftige Forschungen im sechsten Kapitel. Zuletzt werden im siebten Kapitel die Haupterkenntnisse der Arbeit zusammengefasst.

Kapitel 2

Grundlagen

Diese Arbeit behandelt die automatische Erkennung von Ironie in Softwareentwicklungsteams. Um den Begriff der Ironie, wie er in dieser Arbeit verwendet wird, richtig einordnen zu können, werden in Abschnitt 2.1 die Geschichte, sowie die Anwendung und Merkmale von Ironie behandelt. Um eine Klassifikation von ironischen und nicht ironischen Ausdrücken zu erzeugen, wird in dieser Arbeit ein maschinelles Lernverfahren durch eine Support Vector Machine (SVM) angewendet. Dieses wird in Abschnitt 2.2 konkret erläutert. Der maschinelle Lernprozess benötigt Anhaltspunkte, durch die eine Klassifikation getroffen werden kann. Diese Anhaltspunkte basieren zum einen auf textbasierten und zum anderen auf akustischen Merkmalen. Die textbasierten Merkmale werden durch das in Abschnitt 2.3 beschriebenen Natural language processing (NLP) gewonnen und die akustische Merkmale durch das in Abschnitt 2.4 beschriebenen Mel Frequency Cepstral Coefficents (MFCC).

2.1 Ironie

Ironie hat eine umfangreiche Geschichte, unterschiedliche Erscheinungsformen, Eigenschaften, Verwendungsweisen und Funktionen, sodass Ironie ein sehr komplexes Phänomen darstellt [30]. Aus verschiedenen Teilen der Linguistik, der Psychologie und auch der Soziologie gibt es Versuche Ironie zu definieren, sie beleuchten dabei jedoch nur einzelne Aspekte, ohne die Ironie in ihrer enormen Komplexität vollständig erfassen zu können. Einige Definitionsversuchen ergänzen sich und manche widersprechen sich sogar. [39]. Im Folgenden werden die Grundlagen zur Ironie beschrieben.

2.1.1 Geschichte des Ironiebegriffs

Zwischen dem heutigen Ironieverständnis und der frühsten Geschichte der Ironie gibt es viele Parallelen. Somit lassen sich auch die heutige Definitionen aus dem historischen Ironiebegriff ableiten.

Ironie lässt sich bis ca. 400 v. Chr. zurückverfolgen, wobei die Ursprünge des Ironiebegriffs aus dem antiken Griechenland von dem Begriff ειρωνεια (eironeia) stammen.

Damals war dies eine Bezeichnung für ein Verhalten, dass im Verleugnen bzw. im Sich-selbst-kleiner-machen bestand [15]. Der griechische Philosoph Sokrates (469 v. Chr. - 399 v. Chr.), der zu dieser Zeit lebte, prägt noch heute das moderne Ironieverständins. In Dialogen täuscht er Unwissenheit vor und bittet sein Gegenüber um Belehrung. Das Ziel von Sokrates ist es dabei, sein Gegenüber zu eigenen Erkenntnissen zu verhelfen. Sokrates selbst spricht bei dieser Methode nie von Ironie, da der Ironiebegriff zu seinen Lebzeiten für Betrüger, die durch Untertreibung jemanden täuschen wollen, genutzt wird und damit sehr negativ behaftet ist [39, 15]. Dass der Begriff Ironie mit den Methoden der Dialogführung von Sokrates verbunden wird, liegt eher an seinen Zeitgenossen, die seine Dialogführung als unmoralisch und hochmütig betrachteten, da ihnen möglicherweise der tiefere Sinn seines Verhaltens nicht bewusst war. Durch die Verbreitung von Sokrates' Lehre über seine Schüler etabliert sich der Fachausdruck *Sokratische Ironie* und zwar ohne die damit verbundene Abwertung [15]. Die erste schriftlich überlieferte Definition findet sich in der *Rhetorik für Alexander* wieder, die etwa um 330 v. Chr. entsteht. Sie ist nicht besonders ausführlich, jedoch wird klargestellt, dass zur Ironie eine Verstellung in der Redeweise und dem Verhalten zählt, wobei immer klar erkennbar sein muss, dass es sich um ein Verstellen handelt. Ironie wird als eine niveauvolle Strategie Heiterkeit zu erzeugen beschrieben, die durch gegensätzliche Ausdrücke hervorgerufen wird [15].

Erweiterungen zu dieser Definition erfolgen durch Cicero (106 v. Chr. - 43 v. Chr.) und Quintilian (35 - 96 n. Chr.). Beide vermitteln griechisches philosophisches Gedankengut an die lateinischsprachige Welt [39]. Cicero teilt den Ironiebegriff in *inversio* (Umgekehrt) und *dissimulatio* (Verstellung). Bei der *inversio* wird nur ein einzelnes Wort in der Bedeutung umgekehrt, während es sich bei *dissimulatio* um eine Verstellung für eine ganze Äußerung oder Sequenz handelt. Quintilian nutzt zur Beschreibung der Ironie die Begriffe *simulatio* (sich stellen) und *dissimulatio* (Verstellung). Bei der *simulatio* wird eine Überzeugung übernommen, die erfunden ist oder von jemand anderem kommt, oder es wird etwas beschrieben, das nicht geschehen kann. Bei der *dissimulatio* geht es Quintilian, anders als Cicero um ein bewusstes *Dummstellen* oder absichtliches *Missverstehen*. Eine große Übereinstimmung der beiden ist, dass für beide das Lachen bzw. sich lustig machen ein wichtiger Bestandteil der Ironie ist [15].

Erst ab dem 17. und frühen 18. Jahrhundert finden sich neue Theorieansätze zur Ironie. In der Romantik wurde die Ironie als Erzähltechnik und poetische Grundhaltung aufgefasst [30]. Da sich diese Arbeit einzig auf ironische Äußerungen bezieht, verzichte ich hier auf eine weitere Beschreibung der Konzepte der *Tragischen Ironie* oder der *Ironie des Schicksals*, die sich in der Romantik gebildet haben, da sie sich hauptsächlich auf Kunstwerke und literarische Werke beziehen.

In zahlreichen aktuellen Arbeiten wird vereinfacht eine Aussage als ironisch deklariert, wenn das Gegenteil des Gesagten gemeint ist [48, 16, 50]. Diese Herangehensweise wird wiederum in anderen Arbeiten kritisch Hinterfragt [39]. Eine einvernehmliche, fachübergreifende Kategorisierung von Ironie hat sich bis heute nicht durchsetzen können, da die Forschungsausrichtungen zu unterschiedlich sind [30].

2.1.2 Definition des Ironiebegriffs in dieser Arbeit

Aufgrund der Komplexität von Ironie gibt es bis heute keine einheitliche Definition von ihr. Es gibt eine Vielzahl von Arbeiten [48, 16, 50], die sich als Definition von Ironie auf das Gegenteil vom Gemeinten, oder einer Diskrepanz zwischen dem Gesagten und der Realität berufen. Diese sehr einfach gehaltene Definition kann die Komplexität von Ironie nicht vollständig abdecken und wird deshalb in einigen Arbeiten kritisch betrachtet [15, 27, 25, 37]. Auch in dieser Arbeit soll eine umfangreichere Definition von Ironie angewendet werden. Wenn man die Definition darauf eingrenzt, dass die Aussage das Gegenteil des Gemeinten ausdrückt, ergibt sich das Problem, dass nicht von jeder Aussage eine gleichwertige Umkehrung existiert, bzw. diese nicht zwangsweise gemeint ist. Folgende ironisch gemeinte Aussage soll die Problematik verdeutlichen:

> „Dann können wir den tollen Code ja auch gleich löschen.“

Eine komplette Umkehrung des Satzes ist schwierig. Das Adjektiv von toll in schlecht umzukehren reicht nicht aus und eine Verneinung vor dem löschen, würde die gemeinte Aussage nicht widerspiegeln. Wenn man Ironie nur in der Weise abgrenzt, dass es eine Diskrepanz zwischen dem Gemeinten und dem Gesagten gibt, dann trifft diese Definition wohl auf so gut wie jede Aussage zu und ist damit zu offen formuliert. Weiterhin muss sich die Ironie von Metaphern oder Lügen abgrenzen.

Eine gute Grundlage für eine Definition bieten die viel zitierten Arbeiten von Knox [24, 23, 22], in denen er den Ironiebegriff und den historischen Hintergrund und dessen Einfluss untersucht. Knox fasst die antike Definition dabei in folgenden vier Aussagen zusammen, die ich mit eigenen Beispielen ergänzt habe, Knox [22, S. 25]:

- Ironie als Gegenteil zwischen Sagen und Meinen
  - Beispiel: „Du bist ja pünktlich.“, wenn eigentlich gemeint ist „Du bist ja unpünktlich.“
- Ironie als etwas Anderes zu sagen als man meint
  - Beispiel: „Das ist voll gut, wie du es programmiert hast.“, wenn eigentlich gemeint ist „Das ist ganz in Ordnung programmiert.“
- Ironie als Lob durch Tadel und Tadel durch Lob
  - Beispiel: „Das hast du ja mal wieder toll gemacht.“, wenn eigentlich gemeint ist „Das hast du ja mal wieder nicht gut gemacht.“
  - Beispiel: „Eine Gehalterhöhung um 15% ist ja so wenig.“ wenn eigentlich gemeint ist „Eine Gehaltserhöhung um 15% ist richtig viel.“
- Ironie umfasst jede Art des Sich-Lustigmachens und Spottens

Auch Hartung [15] greift diese Aussagen auf und zeigt durch eine umfangreiche empirische Studie, dass sich die moderne Vorstellung von Ironie zu großen Teilen mit den antiken

Bedeutungszusammenhängen deckt. Um ein umfassendes Bild des Ironiebegriffs zu bekommen erweitert Hartung [15] die Aussagen von Knox [22] mit den folgenden Aussagen, [15, S. 29]:

- Ironie ist eine kurzfristige Verstellung in einer einzelnen Äußerung, die immer als Verstellung erkennbar gemacht werden muss, wenn sie ihre Wirkung erreichen soll
- ein wichtiger Teil dieser Wirkung kann je nach Situation in der Erheiterung bestehen, die dadurch ausgelöst wird, dass die Verstellung nicht den allgemeinen Erwartungen entspricht
- Ironie zielt darauf, den Gegner und seinen Standpunkt abzuwerten, und oft wird dazu ihr komisches Potential eingesetzt
- Ironie muss nicht explizit markiert sein, sie wird in den meisten Fällen über gemeinsame Wissensbestände erkennbar, oft über die aktuelle Situation und den Gesprächsverlauf
- aus diesen Wissensbeständen lässt sich für den Zuhörer auch die Bedeutung und die Funktion der ironischen Äußerung ableiten

Bei der aufgeführten Verstellung kann es sich um eine Verstellung der Redeweise oder des Verhaltens handeln. Außerdem ist Ironie immer eine Aussage, die eine Person oder ein Objekt bewertet [15].
Die Definition lässt sich an folgendem Beispiel erläutern:

> „Die Kundin ist ja mal wieder richtig pünktlich. Wenn sie in den nächsten zehn Minuten nicht auftaucht werden wir den Termin verschieben müssen.“

Bei dem ersten Satz *„Die Kundin ist ja mal wieder richtig pünktlich.“* handelt es sich um eine kurzfristige Verstellung. Der *Gegner* ist die Kundin, die mit dieser Aussage abgewertet werden soll. Um die Aussage richtig zu interpretieren, muss der gemeinsame Wissensbestand da sein, dass sich die Kundin schon häufiger verspätet hat. Das im darauffolgenden Satz angesprochene Warten steht in einer Diskrepanz mit der Aussage *„richtig pünktlich“*. Wenn diese Wissensbestände nicht da wären, wäre die Verstellung nicht als solche erkennbar und würde ihre Wirkung nicht erreichen.

Die in diesem Abschnitt herausgestellten Aussagen von Knox [24] und Hartung [15] werden in dieser Arbeit als Grundlage für die Klassifizierung von Ironie genutzt.

2.1.3 Sarkasmus

In einigen Arbeiten [16, 50, 1], die sich mit der Klassifizierung von Ironie auseinandersetzen, werden zusätzlich sarkastische Aussagen klassifiziert, oder mit ironischen Aussagen zusammengefasst. Deshalb soll an dieser Stelle kurz auf dieses Thema eingegangen werden. Sarkasmus ist eine beißende, scharfe, spottende Ausdrucksweise, die genau das bedeuten soll, was sie sagt [38]. Umgangssprachlich wird Sarkasmus oft als Form der Ironie

betrachtet, die etwas gemeiner ist [27]. Sarkastische Äußerungen müssen jedoch nicht zwingend ironisch sein und ironische Äußerungen auch nicht unbedingt sarkastisch. Daher kann der eine Begriff nicht wirklich als Form für den anderen dienen [25]. Dass Ironie und Sarkasmus zu differenzieren sind, zeigen auch einige Forschungsarbeiten, bei der diverse Unterschiede zwischen sakrastisch und ironisch klassifizierten Twitterbeiträgen ausgemacht werden konnten [50, 28, 49, 21].

In dieser Arbeit soll der Fokus auf ironischen Aussagen, wie sie in Abschnitt 2.1.2 erläutert wurde, liegen, weshalb im Folgenden nicht weiter auf Sarkasmus eingegangen wird.

2.1.4 Ironie in der Anwendung

Bei Ironie handelt es sich um eine figurative und nicht um eine literale (wörtliche) Sprache. Figurative Sprachen sind deutlich schwerer zu verstehen als literale Sprachen, da sich bei ihnen die Intention der Sprechenden hinter einer wörtlichen Äußerung verbirgt. Um die eigentlich intendierte Äußerung hinter der wörtlich formulierten Aussage zu verstehen, sind ausgeprägte kognitive Fähigkeiten erforderlich. Bei ironischen Äußerungen sind häufig zusätzliche semantische Informationen, wie kulturelle bzw. soziale Hintergrundinformationen oder Kontextwissen notwendig, um sie korrekt interpretieren zu können [44]. Das Erkennen und das Interpretieren von Ironie sind zwei getrennte Prozesse. Das bedeutet, eine Person die erkannt hat, dass eine Aussage ironisch ist, hat nicht automatisch die Ironie der Aussage richtig interpretiert [3].

In Abbildung 2.1 wird schematisch dargestellt, wie von einem Sprecher S, eine Äußerung A, an einen Rezipient R gerichtet wird. Der Rezipient besitzt ein internes

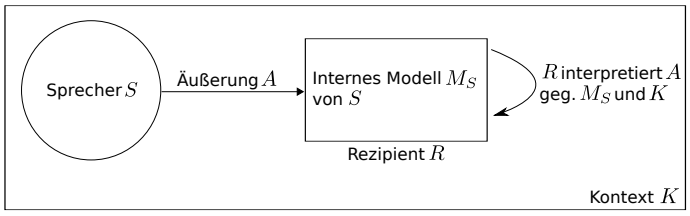

Abbildung 2.1: Theoretisches Kontextmodell zur Beschreibung von Ironie (nach Tetzner [50] und Wallace [53])

Modell M_S von dem Sprecher. Das interne Modell ist dabei das Wissen, welches der Rezipient über den Sprecher hat. Das können zum Beispiel Vorlieben, oder auch die politische Einstellung desjenigen sein. Je größer das interne Modell über den Sprecher ist, desto eher ist der Rezipient in der Lage eine ironische Aussage des Sprechers richtig einzuordnen. Zur richtigen Interpretation der Aussage gehört neben dem internen Modell auch der Kontext K, in dem die Aussage gestellt wurde. Bei dem Kontext kann es sich beispielsweise um das vorangegangene Gespräch oder das aktuelle Wetter halten. Der Sprecher muss wiederum wissen, dass der Rezipient über das nötige Wissen verfügt, um seine Aussage als ironisch einordnen zu können, da es sonst zu Fehlinterpretationen

seitens des Rezipienten kommen kann [15].

Beispiel zur Erläuterung des Kontextmodells:

> Kontext K: Zwei Personen stehen draußen und es regnet in Strömen.
> Sprecher S: „Das Wetter ist ja herrlich.“

Im Regen zu stehen, ist für die meisten Menschen eher negativ behaftet. Der Rezipient erkennt also in dieser Situation eine Diskrepanz zwischen schlechtem Wetter und der Aussage von schönem Wetter und ordnet die Aussage des Sprechers somit als ironisch ein. Wenn der Rezipient nun das Wissen über den Sprecher hat, dass dieser ein Landwirt ist, der schon lange auf Regen wartet, ändert sich der Kontext, sodass die Aussage von dem Rezipient wahrscheinlich nicht mehr als ironisch interpretiert wird.

Es wird deutlich, dass das Erkennen von Ironie nicht allein auf Basis von syntaktischen Merkmalen, wie der Satzstellung, oder statistischer Informationen wie der Häufigkeit von Wörtern, gelingen kann. Aussagen können jedoch bestimmte Merkmale enthalten, welche die Identifikation von Ironie erleichtern können [50]. Diese Merkmale werden in Abschnitt 2.1.6 weiter erläutert.

2.1.5 Motive für die Verwendung von Ironie

Ironie kann sehr vielfältig eingesetzt werden und enthält damit eine Reihe an kommunikativen Funktionen. Dabei bleibt sie eine Form von Bewertung [15]. Zu dieser Bewertung gehört immer eine positive oder negative Polarität und ermöglicht dem Sprechenden die Vermittlung von Emotionen und Empfindungen [52]. Ironie ist eine Ausdrucksweise, zu deren Wirkungen die Erheiterung zählen kann, aber nicht muss [15].

Die ironische Indirektheit kann dazu beitragen, bei kritischen Aussagen das Konfliktpotential zu reduzieren [15]. Es können negative Empfindungen geäußert werden, ohne das Gegenüber direkt verbal anzugreifen. Alle Parteien verstehen zwar die negativ intendierte Aussage, durch eine positive Formulierung wird der direkte Angriff aber abgeschwächt [3]. Der Kritisierte kann in einer Antwort auf die Kritik so tun, als ob er die Kritik nicht wahrgenommen hat und sich so vor einer Rechtfertigung drücken. Zugleich können ironische Äußerungen Kritik auch verschärfen, vor allem wenn Sarkasmus mit Ironie verknüpft wird [15].

Weiter kann Ironie, wie in Abschnitt 2.1.2 schon kurz beschrieben, als Lob genutzt werden. Empirische Beobachtungen zeigen jedoch, dass Ironie fast ausschließlich zur negativen Bewertung verwendet wird [15].

Zudem kann Ironie als Vermittlung eines Zugehörigkeitsgefühls dienen. Mit einer ironischen Äußerung ist eine gezielte Ansprache eines spezifischen Personenkreises möglich, wenn nur dieser Personenkreis das notwendige Kontextwissen besitzt, das zur Interpretation und Identifikation der Aussage notwendig ist. Andere Personen, die die Äußerung hören, werden ausgeschlossen, da sie aufgrund des fehlenden Kontextwissens nicht in der Lage sind, die intendierte Aussage des Sprechers hinter der wörtlichen zu erkennen. Gegebenenfalls wird die Aussage als faktisch wahr hingenommen [52].

Damit die typische Wirkung von Ironie nicht zerstört wird, muss die Mehrdeutigkeit einer Aussage erhalten bleiben. Das bedeutet, dass die Funktion von Ironie nur dann erreicht wird, wenn die Ironie nicht erklärt wird [15].

2.1.6 Merkmale der Ironie

Jede ironische Aussage benötigt Ironiesignale bzw. Merkmale, damit diese als ironisch eingeordnet werden kann [15]. Es besteht eine große Schwierigkeit darin, alle ironischen Merkmale in einem Modell zu erfassen. Die hohe Kontextabhängigkeit, das vorauszusetzende Weltwissen und die impliziten Ausdrucksmöglichkeiten lassen Ironie zu einem schwer erfassbaren Phänomen werden. Es gibt keine festen ironischen Wörter und keine konkrete Syntax, an denen Ironie eindeutig festgemacht werden kann [30]. Jedes Merkmal, das für Ironie genutzt wird, kann in nicht ironischen Aussagen verwendet werden. Diese Mehrdeutigkeit macht die ironische Wirkung erst aus, erschwert jedoch die Identifikation erheblich. Erst auf den Kontext bezogen, wird die Bedeutung der Aussage ersichtlich [15]. Trotz der Vielfalt an Ironiesignalen ist eine erhöhte Frequenz von einigen Merkmalen bei ironischen Aussagen erkennbar [30]. Ironische Merkmale lassen sich in drei Kategorien einordnen: kontextabhängige, verbale und paralinguistische Merkmale [14].

Die **kontextabhängige** Kategorie beinhaltet die Diskrepanz zwischen der Aussage und den gegebenen Umständen. Je größer die Diskrepanz, desto wahrscheinlicher wird eine Aussage als Ironie erkannt. Die Merkmale dieser Kategorie sind am vielfältigsten, aber dafür in jeder ironischen Aussage zu finden. Zugleich sind sie am schwierigsten zu identifizieren, da ein gemeinsamer Wissensbestand zwischen den Sprechenden besteht, der eventuell nicht nur aus dem vorherigen Gespräch, sondern schon aus einer lang vorliegenden Beziehung zwischen ihnen abgeleitet werden kann. Für kontextabhängige Merkmale gilt, dass ein gemeinsamer Wissensbestand vorhanden sein muss, damit diese Ironiesignale genutzt und erkannt werden können [14]. Folgend einige Beispiele zur kontextabhängigen Kategorie:

1. „Die Schrift ist glaube ich noch ein bisschen zu klein, kannst du sie noch ein bisschen größer machen? “ - Bei Schriftgröße 30.

2. „Das ist ein hochkomplexes Spiel, es ist nicht leicht zu gewinnen.“ - Das Spiel ist Tic-Tac-Toe.

3. A: „Welche Ehre, dass wir die erste Gruppe sind.“
   B: „Ja, kann mir nichts Besseres vorstellen. “

Bei allen drei Beispielen kann man die alleinstehenden Aussagen als Außenstehender nicht eindeutig Ironie zuordnen, da der gegebene Kontext nicht groß genug ist. Es gibt immer vorstellbare Situationen, in denen diese Aussagen vollkommen ernst gemeint sein können.

Zur **verbalen** Kategorie zählen Ironiesignale, die auf semantischer, lexikalischer oder syntaktischer Ebene vorkommen. Häufige Signale zu dieser Kategorie sind Über- und Untertreibungen, Stilbrüche (z.B. Umgangssprache zu gehobener Sprache), altmodische

Wörter, der Einsatz von Zitaten, Nachahmungen, Wiederholungen oder Redewendungen, die fast nur ironisch aufgefasst werden [30]. Ein Hinweis auf Übertreibungen können Adverbien und Adjektive sein [14]. Modalpartikel wie *ja* und *doch* werden in ironischen Äußerungen genutzt, um Wissen als allgemein bekannt und unbestritten gültig darzustellen [15]. Im Folgenden wieder ein paar Beispiele, um die Ironiesignale zu verdeutlichen:

1. „Das wird auch so garantiert richtig schön aussehen.“
2. „Das ist ein hübsches Sümmchen.“ - Bei dem *Sümmchen* handelt es sich um mehrere Millionen.
3. „Das ist ja eine schöne Bescherung.“

Auch wenn in Aussagen Ironiesignale auf der verbalen Ebene vorhanden sind, wird an den Beispielen deutlich, dass ein Kontextbezug notwendig ist, um die Aussagen eindeutig ironisch einzuordnen. Bei der ersten Aussage handelt es sich um eine starke Übertreibung, die je nach Situation schnell als ironisch aufgefasst werden kann. Im zweiten Beispiel untertreibt der Sprechende, da eine Verniedlichung durch die Endung *chen* bei dem Wort Summe entsteht und dies eine Diskrepanz zu der gemeinten Summe bildet. Das dritte Beispiel ist eine Redewendung, die im üblichen ironisch genutzt wird und daher häufig direkt als Ironie eingeordnet wird. Sie wird mit dem Modalpartikel *ja* noch verstärkt.

Zu der **paralinguistischen** Kategorie zählen die nonverbalen Signale. Dazu gehören Prosodie und Mimik [14]. Dass es einen bestimmten Tonfall gibt, der nur Ironie zuzusprechen ist, schließen mehrere Forschende aus [5, 47]. Trotzdem werden allgemein bei der Artikulation zur Verfügung stehende Parameter genutzt, um eine Aussage auffällig zu machen [47]. Ironische Aussagen können durch die Intensität oder Betonung und eine verringerte Sprechgeschwindigkeit gekennzeichnet werden, aber auch durch andere akustische Merkmale und verschiedene nonverbale stimmliche Ereignisse, wie z.B. Lachen. In der Mimik sind das Lächeln oder Hochziehen der Augenbraue gängige Mittel, um Ironie zu verdeutlichen [14].

1. „Das ist ja so schwer.“ - Betonung auf dem „o“ von „so“.
2. „Ist das Englisch?“ - Lachen folgt.
3. „Wir sind doch Profis.“ - Durchgehendes Lächeln/Schmunzeln.

Durch die besondere Betonung des Wortes *so* wird im ersten Beispiel das Wort *schwer* hervorgehoben und kann damit eine ironische Wirkung erhalten. Durch das Lachen und Lächeln in den beiden anderen Beispielen bekommen die Aussagen eine neue Wirkung, die den Zuhörenden dazu verleiten, die Äußerungen gegebenenfalls als ironisch wahrzunehmen.

Wenn Sprechende bemerken, dass die Ironie nicht verstanden wurde, kann das Anwenden von paralinguistischen Merkmalen sie verdeutlichen [14]. Zudem kann durch das Nachschieben einer weiteren deutlicher markierten ironische Äußerung die Ironie der ersten Aussage erkennbar gemacht werden [15].

Ironie findet nicht nur in gesprochener Kommunikation, sondern auch in der schriftlichen Kommunikation Anwendung. Um Ironie in schriftlicher Kommunikation zu markieren, können keine paralinguistischen Signale genutzt werden. Als Ersatz können unter anderem Satzzeichen und Emoticons eingesetzt werden [14]. Diese Arbeit analysiert die gesprochene Kommunikation, weshalb nicht weiter auf die Ironiesignale in geschriebener Kommunikation eingegangen wird.

Jede Aussage kann durch angewendete Ironiesignale ironisch wirken, wobei auf eine extrem umfangreiche Menge an Ironiesignalen zurückgegriffen werden kann [39]. Diese Komplexität stellt die automatische Identifikation von Ironie vor eine besondere Herausforderung.

2.2 Maschinelles Lernen mit Support Vector Machines

In den letzten Jahren sind maschinelle Lernverfahren zu einem wichtigen Werkzeug in der Analyse und Interpretation von Daten geworden. Es gibt eine Vielzahl an Algorithmen, die zu den maschinellen Lernverfahren zählen. Für die Analyse von Stimmungen in Texten haben vor allem Naïve-Bayes-Klassifikatoren, Maximum-Entropy-Klassifikatoren und insbesondere Support Vector Machines (SVMs) häufige Anwendung gefunden. In dieser Arbeit soll der Fokus auf die Analyse mit Support Vector Machines gelegt werden, da diese sich in ähnlichen Anwendungsfällen schon bewährt haben [33].

Anhand der Abbildung 2.2 soll die Arbeitsweise einer Support Vector Machine (SVM) anschaulich erklärt werden. Gegeben sind Elemente zweier Klassen A und B, die als Vektoren v_i für $(i = 1...n)$ mit n als Anzahl der Elemente, in einem Vektorraum dargestellt werden. Eine Hyperebene g trennt die beiden Klassen. Wenn Elemente der Klasse A in g eingesetzt werden, erhalten sie ein positives Vorzeichen, wenn Elemente der Klasse B in g eingesetzt werden, erhalten sie ein negatives Vorzeichen. Eine Hyperebene verallgemeinert den Begriff der Ebene, wobei sie hier immer ein Unterraum mit einem Freiheitsgrad weniger als der des gesamten Raums ist. Bei einem zweidimensionalen Raum impliziert der Ausdruck Hyperebene, dass damit auch eine Gerade gemeint sein kann, wie es in unserem Beispiel mit $g(x)$ der Fall ist [11].

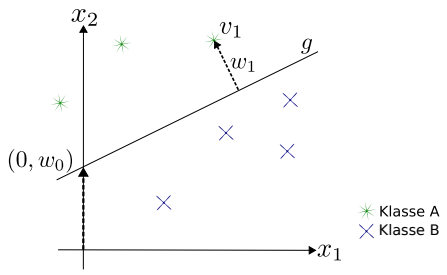

Abbildung 2.2: Vektorraum einer Support Vector Machine mit Elementen aus zwei Klassen, die durch eine Hyperebene getrennt werden (nach Frochte [11])

Das Gewicht w_1 steht senkrecht auf g und die Verschiebung von g ist durch w_0 definiert. Die Hyperebene soll zum einen die beiden Mengen trennen, sodass alle Elemente zur richtigen Klasse zugeordnet werden. Zum anderen soll der Abstand der Elemente zur Hyperebene möglichst groß sein. Wenn Elemente sehr nah an g liegen, kann es bei einem leicht veränderten Datensatz schnell zu falschen Klassifikationen kommen. Somit ergibt sich das Ziel, eine Hyperebene zu finden, bei der die Abstände zwischen Vektoren und Hyperebene maximal sind. Die Punkte, die der finalen Hyperebene am nächsten liegen, definieren ihre Lage und werden als Support Vectors bezeichnet. Der Abstand zwischen der Hyperebene und den Support Vectors wird als Margin bezeichnet [11].

Die Gleichung $g(x) = 0$ wird genutzt um die beiden Mengen zu trennen. Die Gewichte w_i bestimmen dabei die Lage der Hyperebene.

$$0 = g(x) = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n \tag{2.1}$$

x_i und w_i mit $(i = 1...n)$ können zu einem Vektor zusammengefasst werden und man erhält:

$$g(x) = w_0 + w \cdot x \tag{2.2}$$

Kodiert man nun Klasse A mit $y = +1$ und Klasse B mit $y = -1$ soll folgende Bedingung gelten:

$$w_0 + w^T x \geq +1 \quad \text{für } y = +1 \tag{2.3}$$

$$w_0 + w^T x \leq -1 \quad \text{für } y = -1 \tag{2.4}$$

Fasst man 2.3 und 2.4 zusammen, ergibt sich folgende Formel:

$$y(w_0 + w^T x) \geq +1 \tag{2.5}$$

Um eine bestmögliche Hyperebene zu finden resultiert das folgende Optimiertungsproblem für alle Elemente aus den Klassen:

$$\min\{\frac{1}{2}||w||^2\} \text{ unter der Nebenbedingung } y(w_0 + w^T x) \geq +1 \tag{2.6}$$

Der Aufwand für die Lösung des Problems hängt stark mit der Anzahl der tatsächlichen Support Vectors zusammen. Bei einer geringen Anzahl kann das Problem relativ effizient gelöst werden, da nur ein Bruchteil der Daten für die Berechnungen benötigt wird.

Für Klassifizierungsaufgaben, die nicht linear separierbar sind, können Kombinationen aus einem Soft-Margin-Ansatz und anderen Kernel-Ansätze zu guten Ergebnissen verhelfen. Das Soft-Margin vergrößert den Trennbereich, der auch mal verletzt werden darf und erweitert damit die Einsatzfähigkeit des Verfahrens. Die Anwendung eines Soft-Margin bewirkt trotzdem selten eine große Verbesserung, weshalb für bestimmte Anwendungen Kernel-Ansätze zum Einsatz kommen. Hierbei findet der *Kernel Trick* Anwendung, bei dem zusätzliche Dimensionen hinzugefügt werden, um eine passende Hyperebene zu finden. Standardmäßig wird der lineare Kernel verwendet (keine zusätzlichen Dimensionen), wobei polynomielle und radial basierte Kernel für nicht linear separierbare Probleme Anwendung finden [11].

2.3 Natural Language Processing

Das Ziel von Natural Language Processing (NLP), auch Computerlinguistik genannt, ist die Entwicklung computergestützter Methoden und Algorithmen zum Verstehen und Generieren von natürlicher Sprache. NLP kann in viele Teilbereiche, die auf den zahlreichen Aspekten der menschlichen Sprache basieren, unterteilt werden. Zu ihnen zählen morphologische, syntaktische und semantische Analysen [46], aber auch die Spracherkennung [26]. Um die Komplexität von textbasierten Aussagen zu verringern, können unterschiedliche Techniken angewendet werden.

Tokenisierung

Eine Textsequenz wird durch die Tokenisierung (*engl. tokenization*) in einzelne Bestandteile, die sogenannten Token zerlegt. Ein Token ist eine Folge von Zeichen in einer Textsequenz, die zu einer sinnvollen semantischen Einheit für die Verarbeitung zusammengefasst werden [29]. Beispiel einer möglichen Tokenisierung:

„Du bist sehr schlau.“

['Du', 'bist', 'sehr', 'schlau', '.']

Wie die Tokenisierung durchgeführt werden soll, hängt stark von der Sprache ab, da jede Sprache ihre eigene Grammatik sowie Regeln und Muster in der Schreibweise einzelner Wörter hat. Verfälschte Ergebnisse können bei fehlerhaftem Vorgehen bei Apostrophs und Bindestrichen auftreten, vor allem im Zusammenhang mit Eigennamen [29].

Stoppwortreduktion

Stoppwörter (*engl. stop words*) sind Wörter, die häufig in Texten vorkommen, jedoch eine geringe Relevanz für die Analyse des Textes haben. Bei der Stoppwortreduktion erfolgt die Filterung der Worte über Stoppwortlisten. Eine solche Liste wird aus Worten, die am häufigsten im Textkorpus vorkommen, bzw. deren Häufigkeit über einen zuvor definierten Grenzwert liegt, erstellt. Durch die Entfernung von Stoppwörtern wird die Anzahl der Daten, die gespeichert und verarbeitet werden, signifikant reduziert. Ein Nachteil dieser Methode ist, dass bestimmte Redewendungen nicht mehr ausfindig gemacht werden können. Zudem muss abgewogen werden, ob eine Entfernung von Negationen ebenfalls erlaubt werden soll, da Negationen Auswirkungen auf die Polarität einer Aussage haben. Deshalb wird der Umfang einer Stoppwortliste teilweise im Nachhinein wieder reduziert [29].

Auszüge aus einer deutschen Stoppwortliste aus dem Natural Language Toolkit:

'deines', 'ein', 'eines', 'habe', 'nun', 'eurer', 'wieder', 'aller', 'werde', 'keiner', 'gegen', 'mir', 'derer', 'einigem', 'eine', 'allem', 'seinem', 'dazu', 'haben'

Normalsierung

Es gibt einige Fälle, in denen zwei Zeichenfolgen nicht exakt gleich sind, obwohl eine gleiche Zuordnung sinnvoll wäre, wie beispielsweise bei *Polarität* und *Polaritaet*. Durch eine Normalisierung (*engl. normalization*) der aus einem Text extrahierten Token kann eine Übereinstimmung über die vollständige Deckungsgleichheit der einzelnen Zeichen hinausgehen. Zur Normalisierung von Token existieren verschiedene linguistische Ansatzpunkte. Ein Ansatz besteht in der Entfernung bestimmter Zeichen, ein anderer in der Reduktion aller Zeichen eines Textes auf Kleinbuchstaben. Weitere Ansätze berücksichtigen sprachspezifische Aspekte wie zum Beispiel die deutschen Umlaute ä, ö, ü, die zu ae, oe, ue werden.

Bei einer zu umfangreichen Normalisierung besteht die Gefahr, dass Terme als äquivalent erkannt werden, die inhaltlich nicht zusammen gehören. Eine zu geringe Normalisierung kann dazu führen, dass das Erkennen gesuchter Informationen in einem Text trotz inhaltlicher Übereinstimmung nicht möglich ist [29].

Wortstammreduktion und Lemmatisierung

Aus grammatikalischen Gründen werden in Texten verschiedene Formen eines Wortes verwendet, bei denen es jedoch sinnvoll ist, wenn sie gleich indexiert und interpretiert werden. Das Ziel der Worstammreduktion (*engl. stemming*) ist es deshalb, die Wörter auf ihren gemeinsamen morphologischen Wortstamm zu reduzieren. Für Verben bedeutet das die Rückführung auf ihr Infinitiv, für Nomen die Ermittlung des Singulars bei einem gegebenen Plural. Dies geschieht bei der Wortstammreduktion durch die Entfernung von Suffixen und Präfixen [19].

Die Lemmatisierung (*engl. lematization*) hat dasselbe Ziel wie die Worstammreduktion, nutzt jedoch eine andere Methoden. Durch vokabularische und morphologische Analysen soll die Basis bzw. Wörterbuchform eines Wortes erhalten bleiben. Dies erfordert mehr Rechenleistung, führt aber häufig zu besseren Ergebnissen als die Worstammreduktion [29]. Tabelle 2.1 veranschaulicht die unterschiedlichen Ergebnisse der beiden Verfahren.

Beispiel	Wortstammreduktion	Lemmatisierung
laufend	laufen	laufen
gelaufen	laufen	laufen
lief	lief	laufen
läuft	läuft	laufen

Tabelle 2.1: Beispiele für die Worstammreduktion und Lemmatisierung

Part-of-Speech-Tagging

Das Part-of-Speech-Tagging (POS-Tagging) markiert Tokens mit der Wortfamilie, aus welcher das Wort stammt. Sie zeigt also die syntaktische Rolle eines Wortes im Text [29].

Wie in Abschnitt 2.1.6 angedeutet, kann die Wortfamilie womöglich Aufschluss über vorhandene Ironie geben. Adjektive, Adverbien und Modalpartikel können Hinweise auf Ironie sein. Zum Teil gibt es unterschiedliche Tags zu einzelnen Wörtern, die etwas spezifischer sein können. Die Tabelle 2.2 zeigt einen Beispielsatz, mit angegebenen Tags. Für die richtige Bestimmung der Tags ist eine korrekte Rechtschreibung sowie vollständige Interpunktion wichtig. Die meisten Part-of-Speech-Tagger basieren auf Verfahren des maschinellen Lernens, die mit per Hand gelabelte Textdatenbestände trainiert werden [29].

Du	bist	sehr	schlau	.
nicht-reflexives Personalpronomen	Hilfsverb	Adverb	Adjektiv, Adverb oder Prädikat	Satzendezeichen

Tabelle 2.2: Part-of-Speech-Tagging eines Beispielsatzes

Lexikonbasierte Ansätze

Die Ergebnisse der zuvor beschriebenen Techniken, können mit den Inhalten von speziellen Lexika verglichen werden. Aus den Vergleichen können aufschlussreiche Informationen ermittelt werden. Lexikonbasierte Ansätze nutzen Sentiment-Lexika, welche je nach Anwendungsbereich unterschiedlich aufgebaut sind. Einige enthalten nur Wörter, andere auch N-Gramme und Phrasen. Jedem Lexikoneintrag ist eine Polarität (positiv, negativ, ggf. mit Abstufungen, oder neutral), oder ein Emotionstyp wie beispielsweise Freude, Angst, Trauer etc. zugewiesen [32]. Es ist sinnvoll, domänenspezifische Lexika zu nutzen, da in jedem Bereich Wörter anders gedeutet werden. Im Software Engineering Kontext tragen Wörter wie *Support* oder *Error* keine bestimmte Stimmung, wogegen sie in einem anderen Kontext positiv oder negativ gedeutet werden können [18].

Es gibt unterschiedliche Möglichkeiten Sentiment-Lexika zu erstellen. Diese können zum Beispiel anhand der Vorkommenshäufigkeiten von Wörtern Polaritäten bestimmen. Tritt ein Wort häufig in positiv gelabelten Aussagen auf, ist davon auszugehen, dass das Wort selbst eine positive Polarität besitzt. Kommt das Wort annähernd mit der gleichen Häufigkeit in positiven und negativen Aussagen vor, ist es als neutral zu bewerten [32]. Diesen Ansatz kann man auch auf die Ironieerkennung transferieren, um ein Ironie-Lexikon zu erstellen. Statt einer positiven und negativen Polarität ergibt sich ein ironischer und nicht ironischer *Grad* für die jeweiligen Wörter.

2.4 Mel Frequency Cepstral Coefficent

Bei den Mel Frequency Cepstral Coefficients (MFCCs) handelt es sich um Parameter, die sich als besonders vorteilhaft für die Repräsentation von sprachspezifischen Merkmalen erwiesen haben. MFCCs dienen somit bei der Merkmalextraktion aus akustischen

Sprachmaterial und reduzieren dabei zugleich die Daten auf wesentliche Merkmale [10]. Die Abbildung 2.3 zeigt den Ablauf wie aus dem Sprachsignal die MFCCs gebildet werden.

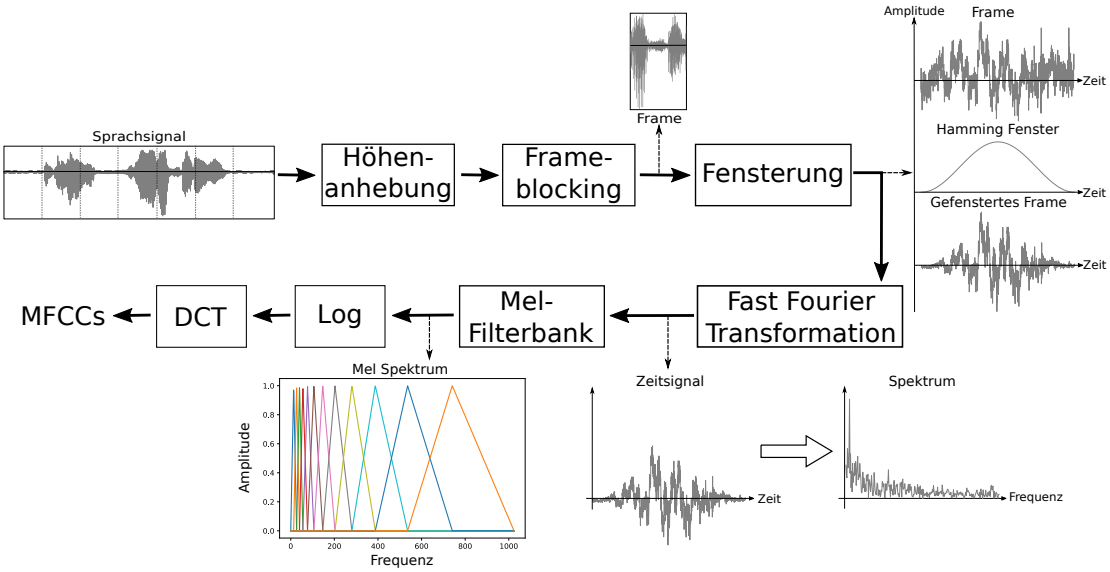

Abbildung 2.3: Prozess zur Extraktion von MFCCs aus Audiomitschnitten (nach Arslan et al. [2])

Die einzelnen Verarbeitungsschritte werden im folgenden Beschrieben:

1. **Höhenanhebung**
   Zuerst wird mit dem Sprachsignal eine Höhenanhebung (Preemphase) durchgeführt. Die Signalenergie von vielen Sprechlauten, vor allem Vokalen, nimmt in Richtung höherer Frequenzen um etwa sechs Dezibel pro Oktave ab. Die leistungsschwachen und gleichzeitig höherfrequenten Anteile werden somit oft von den stets vorhandenen Störungen überdeckt. Die Höhenanhebung sorgt durch eine spezielle Filterung für eine zusätzliche Verstärkung dieser Frequenzen [10].

2. **Frame-Blocking**
   Mit dem Frame-blocking wird das höhenagehobene Eingangsignal in kurze Zeitintervalle (Frames) unterteilt [2]. Die einzelnen Frames überlappen sich dabei, um Informationsverluste zu vermeiden [9]. Die folgenden Schritte werden für jedes Frame einzeln durchgeführt.

3. **Fensterung**
   Die Fourier Transformation ist für periodische Signale ausgelegt, weshalb Diskontinuitäten entstehen, wenn die ersten Ableitungen am Anfangs- und Endpunkt eines jeden Frames, nicht übereinstimmen. Durch eine Fensterung können diese Diskontinuitäten minimiert werden. Dafür wird ein Frame mit einer Fensterfunktion

multipliziert. Die Hamming-Fensterfunktion ist eine der am häufigsten verwendeten Fensterfunktionen [2].

4. **Fast Fourier Transformation**
   Die Fast-Fourier-Transformation (FFT) wird verwendet, um die Signale vom Zeitbereich in den Frequenzbereich zu transformieren [2].

5. **Mel-Filterbank**
   Naheliegende Frequenzen kann das menschliche Gehör schlecht auseinander halten. Dieser Effekt verstärkt sich bei höheren Frequenzen. Bei der Melfilterung wird das Signal mit einer Filterbank, die aus dreieckförmigen Filtern besteht, gefiltert. Die Bandbreite der Filter nimmt in Richtung höherer Frequenzen entsprechend zu, da Schwankungen in dem Bereich unwichtiger werden [10].

6. **Log**
   Das menschliche Gehör nimmt Lautstärke nicht auf einer linearen Skala wahr, weshalb das Signal weiter angepasst sollte. Somit folgt die Logarithmierung des gefilterten Spektrums [10].

7. **DCT**
   Da sich die Filterbänke alle überlappen, sind die Filterbankenergien stark miteinander korreliert. Die diskrete Kosinustransformation (*engl. discrete cosine transform, DCT*) dekorelliert die logarithmierten Filterbankenergien, wodurch die Mel Frequency Cepstral Coefficients entstehen [2].

Ironie kann, wie in Abschnitt 2.1.6 beschrieben, durch bestimmten akustische Signale markiert werden. Beispielsweise durch eine starke Betonung eines Wortes oder durch ein Lachen. Mithilfe von MFCCs der jeweiligen Audiodaten einer Aussage, ist es möglich diese Signale zu detektieren.

Kapitel 3

Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die sich mit der automatischen Ironieerkennung, oder dem Stellenwert von Ironie in Softwareentwicklungsteams beschäftigen. Vorab ist festzustellen, dass sich diese Arbeit von den vorgestellten Arbeiten in vielen Punkten abgrenzt, da die automatische Erkennung von Ironie in mündlichen Dialogen bisher wenig Aufmerksamkeit erhalten hat.

Im Rahmen des International Workshop on Semantic Evaluation 2018 (SamEval 2018)[1] wurden verschiedene Aufgaben zur maschinellen Verarbeitung natürlicher Sprache an die Teilnehmenden gestellt. Die eingereichten Ergebnisse wurden zum Ende hin miteinander verglichen und bewertet. Eine Aufgabe war es in englischen Tweets Ironie zu erkennen. Der bereitgestellte Datensatz umfasste einen Korpus von 4.792 englischsprachigen Tweets (2.396 ironisch, 2.396 nicht-ironisch). Die ironischen Tweets sind mittels Twitter-Suche nach den Hashtags #irony, #sarcasm und #not gesammelt und zur Kontrolle zusätzlich manuell annotiert worden. Die nicht-ironischen Tweets stammten aus dem selben Nutzerkreis. Die Aufgabe bestand aus zwei Teilaufgaben, zum einen aus einer binäre Einteilung von Tweets in die Klassen ironisch und nicht-ironisch, und zum anderen in eine feingranularere Unterteilung. Innerhalb des SamEval 2018 enstanden viele Forschungsbeiträge.

Rohanian et al. [45] nahmen mit einer Klassifikation mittels logistischer Regression (LR) und Support Vector Machine (SVM) an der SamEval 2018 teil und erreichten damit den dritten Platz bei der Teilaufgabe der binären Klassifikation. Sie suchten Merkmale zu Stimmungsgegensätzen zwischen Elementen im Tweet, Stimmungsintensitäten und wortbasierten Merkmale wie Großschreibung. Diese Merkmale kombinierten sie mit vortrainierten Worteinbettungen (*engl. word embedding*).

Wu et al. [54] erreichten mit einem künstlichen neuronalen Netz, mit einem Accuracy-Wert von 73,5%, die besten Ergebnisse für die binäre Klassifikation bei der SamEval 2018. Zum Vergleich, Rohanian et al. [45] erreichten einen Accuracy-Wert von 64,3%. In das Netz wurden vortrainierte Worteinbettungen, Stimmungsmerkmale und syntaktische Merkmale (z.B. POS-Tag-Merkmale) gegeben. Zusätzlich zur Klassifizierung bestimmt

[1] https://alt.qcri.org/semeval2018/

das Netz den fehlenden Ironie-Hashtags.

Tetzner [50] erstellt im Rahmen ihrer Doktorarbeit einen umfangreichen Datensatz, ähnlich dem aus der SamEval 2018, jedoch in deutscher Sprache. Basierend aus Erkenntnissen von bisherigen Forschungsarbeiten entwickelte sie ein neuronales Netz, welches in der Lage ist, Ironie in Textdaten zu identifizieren. Neben der Erkennung von unterschiedlichen Arten von Ironie, soll das Netz durch eine hohe Generalisierbarkeit mit unterschiedlichen Daten zurecht kommen. Um die Generalisierbarkeit zu überprüfen, wurde das Netz neben den von ihr erstellten Datensatz, mit weiteren schon existierenden Datensätzen untersucht. Ihr neuronales Netz weißt für einige der getesteten Datenbestände gute bis sehr gute Ergebnisse auf, jedoch ist insgesamt die Schwankung der Ergebnisse sehr stark. Der Einsatz in weiteren Domänen, wie zum Beispiel dem Software-Engineering, ist denkbar.

Burckard et al. [6] klassifizieren ironische Äußerungen auf Grundlage der Stimmung und des Tonfalls der Stimme. Dafür wurde ein Datensatz mit 937 Äußerungen in deutscher Sprache gesammelt. Die Äußerungen wurden in einem ironischen oder verärgerten Tonfall von zwölf Probanden erzeugt. Sechs Personen markierten diese anschließend auf Ironie und Wut. Ihre Ergebnisse zeigen, dass eine ironische Stimme allein auf der Basis akustischer Merkmale erkannt werden kann.

Die bisher vorgestellten Arbeiten sind nicht aus dem Bereich des Software Engineering, aber es gibt erste Ansätze aus dieser Domäne die im Folgenden vorgestellt werden.

Rakow [40] untersuchte in seiner Bachelorarbeit die Häufigkeit und Art des Auftretens von Ironie in der Kommunikation zwischen Softwareentwickelnden, sowie deren Einfluss auf die Stimmungslage, anhand einer Studie. Insgesamt wurden 31 Teilnehmende über die in ihrem Arbeitsumfeld genutzte Ironie und deren Wirkung befragt. Rakow [40] stellt fest, dass die Anwendungsart von Ironie einen Einfluss auf die Polarität einer Äußerung hat. In der textbasierten Kommunikation wird Ironie, die durch Emoticons und Interpunktion dargestellt wird, eher positiv aufgenommen, wobei Ironie in mündlichen Äußerungen dahingehend eher negativ aufgefasst wird. Zudem stellt Rakow [40] heraus, dass Personen mit Erfahrung im Bereich der Softwareentwicklung, in ihrem Arbeitsumfeld weniger Ironie nutzen und eine negativere Einstellung zu Ironie haben, als Personen ohne Erfahrungen in diesem Bereich.

Islam und Zibran [17] erstellten im Rahmen ihrer Arbeit ein Tool zur automatischen Stimmungsanalyse für Texte speziell aus dem Bereich des Software Engineering. Das entwickelte Tool bestimmt nicht nur die Polariät von Texten, sondern ordnet sie auch emotionalen Zustände wie Stress und Entspannung zu. Es nutzt dabei einen lexikalischen Ansatz mit Wörterbüchern die aus dem Bereich des Software Engineering entstanden sind, sowie diverse Heuristiken. Bei den Heuristiken handelt es sich zum Beispiel darum, dass Emoticons emotionale Zustände ausdrücken, oder das Wörter, die nur groß geschrieben sind eine hohe Aufgeregtheit darstellen. Die Trainings- und Testdaten

kamen vor allem aus Texte von dem Projektmanagment-Tool JIRA[2]. Auch wenn die automatische Stimmungsanalyse in vielen Fällen gute Ergebnisse liefert, kann es in Texten, die Ironie oder Sarkasmus enthalten, die Emotionen nicht korrekt identifizieren.

Obaidi und Klünder [35] führten eine systematische Literaturanalyse zu Sentiment-Analyse-Tools von Texten aus dem Software Engineering durch. Diese soll einen Überblick über die Entwicklung, Anwendung und die Probleme dieser Tools geben. Zu den identifizierten Problemen der 80 genauer untersuchten Publikationen gehörte unter anderem der Umgang mit Ironie und Sarkasmus. Dieser wurde in elf Forschungsarbeiten gefunden und nach dem Problem der Anpassung an die Domäne des SE, am zweit häufigsten angegeben.

Obeidi [36] untersuchte im Rahmen ihrer Masterarbeit destruktive Äußerungen in Meetings von Softwareprojekten. Solche Äußerungen können den Erfolg eines Meetings gefährden und zu einer negativen und pessimistischen Haltung von Entwicklern führen. Für eine schnelle Erkennung destruktiver Äußerungen erstellte Obeidi ein automatisiertes Tool. Auch ironische Aussagen können destruktiv sein, müssen jedoch gesondert betrachtet werden, da sie schnell zu falschen Interpretationen führen können.

Arbeiten zur Sentiment-Analyse im Software Engineering sehen zwar ein Problem bei der richtigen Einordnung von ironischen Aussagen, jedoch löst keine Arbeit dieses Problem. Diese Arbeit grenzt sich mit der Zielsetzung und Vorgehensweise sowie der Datenquelle eindeutig von den hier vorgestellten Publikationen ab. Dennoch ist das Konzept des Analyseverfahrens durch die aufgeführten Arbeiten beeinflusst worden.

[2] https://www.atlassian.com/de/software/jira

Kapitel 4

Konzept und prototypische Umsetzung

Der Erfolg eines Softwareprojektes hängt unter anderem von der Kommunikation innerhalb des Teams ab [20]. Dabei können ironische Kommentare einen positiven oder negativen Effekt auf die Kommunikation haben. Zusätzlich kann es zu großen Missverständnissen kommen, wenn die Ironie einer Person nicht erkannt wird [12]. In diesem Kapitel wird das Konzept zur automatischen Identifikation von Ironie in der Kommunikation von Softwareentwicklungsteams vorgestellt.

4.1 Vorüberlegungen

Wie im Grundlagenkapitel 2.1 bereits beschrieben, ist Ironie ein sehr komplexes Phänomen. Durch die nicht vorhandene klare Definition in der Wissenschaft, ist es nur logisch, dass die Wahrnehmung von Ironie für jede Person sehr unterschiedlich sein kann. Ob eine Person eine Aussage als ironisch wahrnimmt, hängt zum einen davon ab, ob sie die Ironiesignale erkennt und zum anderen von ihrer Vorstellung von Ironie. Es ist also interessant zu betrachten, ob zwei Personen, die einen Dialog führen, dieselben Aussagen als ironisch identifizieren. Ein Datensatz aus einem transkribierten Gespräch zweier Personen, bei dem beide Personen im Nachhinein ironische Aussagen markieren, kann die Subjektivität bei der Bewertung von Ironie aufzeigen. Wenn dieser Datensatz nach einer gewissen Zeit noch einmal von den Personen bewertet wird, kann zusätzlich die Schwierigkeit der Identifikation von Ironie mit weniger Kontextwissen herausgestellt werden.

Eine eindeutige Klassifikation von Ironie ist bei einer starken Subjektivität in der Identifizierung nicht möglich. Deshalb stellt die Klassifizierung von möglicher Ironie eine Alternative dar. Wenn Rezipienten einer Aussage wissen, dass die Äußerung möglicherweise ironisch gemeint ist, können sie dieser eine andere Bedeutung zuweisen. Zudem können Sprechende, wenn sie merken, dass sie keine eindeutige Aussage gemacht haben, diese verdeutlichen.

Ironie wird durch vielfältige Merkmale gekennzeichnet, wodurch einfache Regeln

für eine Klassifikation nicht hinreichend sind. In der Sentiment-Analyse werden unterschiedliche Methoden zur Bestimmung von Polaritäten eines Textes eingesetzt. Gängige Methoden sind Naïve-Bayes-Klassifikatoren [7], Random Forest [7] und Support Vector Machines (SVMs) [8], sowie neuronale Netzwerke [55]. Diese Methoden kommen für die Ironiedetektion ebenfalls infrage. Um eine Detektion von ironischen Aussagen innerhalb von Dialogen in Softwareentwicklungsteams zu ermöglichen, werden unabhängig von der Methode Trainings- und Testdaten mit dementsprechenden Labeln benötigt. Neuronale Netzwerke benötigen für adäquate Ergebnisse einen großen Trainingsdatensatz. Andere Verfahren wie zum Beispiel die SVM können bereits mit weniger Trainingsdaten gute Ergebnisse erzielen.

Es gibt keinen öffentlich zugänglichen deutschen Datensatz, der Dialoge von Softwareentwicklungsteams enthält. Zudem müssen die Label (*Ironie* oder *Keine Ironie*) für einen Datensatz nachträglich hinzugefügt werden, wobei das nötige Kontextwissen vorhanden sein muss, um Aussagen als ironisch klassifizieren zu können. Aus diesem Grund muss ein neuer Datensatz erstellt werden. Das genaue Vorgehen für die Erstellung des Datensatzes wird in Abschnitt 4.3 erläutert. Der Datensatz soll zum einen Transkripte von Aussagen enthalten und zusätzlich die akustische Tonspur. Zudem werden die Aussagen von den Teilnehmenden selbst klassifiziert. In dem zeitlich begrenzten Rahmen einer Masterarbeit ist es nicht möglich, genug der eben beschriebenen Daten zum Trainieren eines neuronalen Netzes zu erfassen. Da die SVM in der Vergangenheit gute Ergebnisse erzielt hat, dient sie in dieser Arbeit als grundlegendes maschinelles Lernverfahren für die Klassifikation.

4.2 Aktueller Stand der Forschung

Eine Klassifikation von Ironie auf Grundlage von Audiodaten, sowie transkribierter Texte, gibt es bisher nicht. Das Forschungsfeld ist allerdings dem Forschungsfeld der Sentiment-Analyse ähnlich, weshalb hier einige Aspekte übernommen werden können. In der Sentiment-Analyse werden aus Textdaten Merkmale extrahiert, welche auch in der Ironieerkennung genutzt werden können. Zudem haben Support Vector Machines auch in diesem Forschungsfeld gute Ergebnisse erzielen können.

4.3 Datenbasis

Die Grundlage der Klassifikation von Ironie bildet die Datenbasis. Das Ziel dieser Arbeit ist es, Ironie in Dialogen von Softwareentwicklungsteams zu erkennen. Zur Bildung eines Klassifikationsmodells und zu der Verifikation dieses Modells werden Daten von Gesprächen aus dem Bereich des Software Engineerings benötigt. Aus den Zielen der Arbeit ergeben sich folgenden Anforderungen an den zu nutzenden Datensatz:

1. *Der Datensatz muss Transkripte von Dialogen in Softwareentwicklungsteams bereitstellen.*
   Wie auch schon in Abschnitt 2.3 beschrieben, unterscheidet sich die Kommunikation

in verschiedenen Domänen. Klassifizierer die mit Daten aus einer Domäne trainiert werden erreichen schlechtere Ergebnisse bei Daten aus anderen Domänen.

2. *Der Datensatz muss einen Audiomitschnitt jeder Aussage aus den Dialogen enthalten.*
   Wie in Abschnitt 2.1.6 beschrieben, können für die Markierung von Ironie auch paralinguistische Ironiesignale verwendet werden. Dazu gehört beispielsweise die Betonung bestimmter Wörter, oder eine Veränderung der Sprechgeschwindigkeit. Die Mimik, die auch paralinguistische Ironiesignale beinhalten kann, wird in diesem Datensatz nicht beachtet.

3. *Die Transkripte und Audioaufnahmen in dem Datensatz müssen in deutscher Sprache vorliegen.*
   In jeder Sprache gibt es syntaktische und semantische Besonderheiten, die gesondert behandelt werden müssen. Zudem ist das Erkennen von Ironie für Nichtmuttersprachler um einiges schwieriger, was eine nachträgliches Labeln von Ironie erschwert.

4. Die Aussagen innerhalb des Datensatzes müssen der Klasse *ironisch* oder *nicht ironisch* zugeordnet sein.
   Für eine Klassifizierung mit der SVM werden klassifizierte Daten benötigt.

5. *Der Datensatz soll möglichst ausgeglichen zwischen ironischen und nicht ironischen Aussagen sein.*
   Für eine automatische Erkennung von Ironie müssen generalisierbare Regeln, die für die Bildung und Verwendung von Ironie genutzt werden, in den ironischen Aussagen gefunden werden. Viele Klassifizierer neigen dazu, sich hauptsächlich mit den großen Klassen eines Datensatzes zu beschäftigen und die kleinen zu ignorieren. Das führt dazu, dass die kleinen Klassen nie detektiert werden [41]. Zudem werden die generalisierbaren Regeln überhaupt erst erkennbar, wenn sie in mehreren Äußerungen auftauchen.

Es gibt keinen Datensatz, der diesen Anforderungen genügt. Daraus folgt, dass ein neuer Datensatz erhoben werden muss.

4.3.1 Datenerhebung

Damit die im vorherigen Abschnitt gegebenen Anforderungen an den Datensatz erfüllt werden können, ist eine Aufnahme von Dialogen im Bereich des Software Engineering notwendig. In Softwareentwicklungsteams wird auf verschiedene Art und Weise kommuniziert. Mündliche Kommunikation gibt es vor allem in Meetings oder beispielsweise beim Pair Programming. Im Folgenden werden die Schritte zur Erstellung des Datensatzes beschrieben. Die Erstellung gliedert sich in die Erhebungsphase und die manuelle Annotation.

Erhebungsphase

Für die Erstellung des Datensatzes werden die Dialoge dreier Gruppen mit jeweils zwei Personen beim Pair Programming aufgenommen. Die Aufnahme der Kommunikation beim Pair Programming eignet sich für diese Untersuchung besonders gut, da keine Nebengespräche entstehen können. Die Gruppen bekommen einen Quellcode den sie analysieren, bewerten und verbessern sollen. Es ist notwendig das die Teilnehmenden Python Quellcode lesen und verstehen können und sich somit mit anderen darüber unterhalten können. Außerdem sollte ihnen das Prinzip von Pair Programming bekannt sein. Die freiwilligen Teilnehmenden sind alle männlich, zwischen 24 und 27 Jahre alt und studieren (Technische-) Informatik oder haben das Studium schon abgeschlossen. Alle haben in ihrem Studium schon Erfahrungen mit Python und Pair Programming gesammelt.

Die Probanden bekommen dreißig Minuten Zeit, um den Quellcode zu ihrer Zufriedenheit zu verbessern. Zwei Gruppen erfüllen die Aufgabe in Präsenz, die dritte Gruppe bearbeitet die Aufgabe an getrennten Computern, über ein Onlinekommunikationstool. Die Tonaufzeichnung erfolgt bei jeder der Gruppen somit über unterschiedliche Aufnahmegeräte. In der Aufgabenbeschreibung, in Anhang A.1 einsehbar, werden die Probanden explizit darum gebeten auch ironische Aussagen als Bewertung zu nutzen. Dies soll sicherstellen, dass die Anforderung an den Datensatz im Bezug zum Ironieanteil erfüllt werden. Bei dem ausgehändigten Python Quellcode handelt es sich um das Spiel Tic-Tac-Toe. Die Grundlage für den Code entstammt aus dem GitHub-Projekt von Aqeel Anwar[1]. Um Bewertungen des Codes zu provozieren, wurden Teile des Codes verändert. Beispielsweise wurden Rechtschreibfehler hinzugefügt, hilfreiche Kommentare entfernt, unpassende Farbkombinationen eingebaut und Variablennamen verändert.

Die Aufzeichnung der Gespräche bildet die Basis des Datensatzes. Es folgte die händische Transkription der Gespräche. Jede Aussage wird zusätzlich mit einer Start- und Endzeit markiert. Die Teilnehmenden werden durch die Buchstaben A und B anonymisiert.

Manuelle Annotation

Innerhalb einer Woche bekommen die Teilnehmenden die jeweilige Transkription und Audioaufnahme ausgehändigt. Sie werden gebeten die Aussagen ihres Gespräches auf Ironie zu markieren. Da es eine große Unsicherheit bei der Definition von Ironie gibt, wird neben der Möglichkeit, eine Aussage als Ironie zu markieren, die Möglichkeit gegeben, eine Aussage als eventuell ironisch gemeint zu markieren. Zwei der sechs Teilnehmenden haben die Audioaufnahme nicht zur Hilfe genommen. In Anhang A.2 ist die Aufgabenbeschreibung für die Klassifikation des Datensatzes abgelegt. Eine Klassifizierung des Datensatzes wird von mir zusätzlich durchgeführt. Nach drei Monaten klassifizieren fünf der sechs Teilnehmer die Transkripte erneut. Dieses Mal stehen ihnen die Audioaufnahmen nicht zur Verfügung.

[1] https://github.com/aqeelanwar/Tic-Tac-Toe

Es stellte sich im Laufe der Arbeit heraus, dass neben der Klassifizierung in ironische und nicht ironische Aussagen, die ironischen Aussagen weiter unterteilt werden müssen. Es ist notwendig herauszustellen, ob die Aussagen verbale, paralinguistische oder nur kontextabhängige Ironiesignale beinhalten. Anhand der in Tabelle 4.1 aufgezeigten Beschreibung bekommt jede ironische Aussage eine weitere Wertung auf Grundlage ihrer verbalen Ironiesignale.

Da eine feingranulare Klassifikation nach dieser Wertung nicht möglich ist, wird die Wertung 0 als *Keine Ironie* und die Wertung $1-3$ im Weiteren als Ironie angenommen.

Wertung	Beschreibung
0	Kein verbaler Hinweis auf Ironie bzw. nur Hinweise im Kontext.
1	Meinungsäußerung oder Bewertungen. Floskeln wie: Ich glaube, ich vermute, ich denke...
2	Leichte Überspitzung in der Wortwahl erkennbar, wobei die Äußerung auch ernst gemeint sein könnte. Hinweise geben Modalpartikel wie beispielsweise *ja* oder *doch*.
3	Starke Übertreibung und viele Modalpartikel.

Tabelle 4.1: Sprachliche Bewertungen von Ironie anhand von Merkmalen im Text

Neben den verbalen Merkmalen kann die Stimme auch Hinweise auf Ironie geben. Jedoch ist in vielen ironischen Aussagen gar kein Hinweis in der Stimme enthalten. Somit müssen die Äußerungen, in denen Ironie in der Stimme erkennbar ist, herausgearbeitet werden. Es stellte sich als besonders schwierig heraus, Ironiesignale in der Stimme zu erkennen. Eine feinere Klassifizierung der ironischen Aussagen, mit Hinsicht auf akustische Merkmale, wird mit Hilfe der Wertung aus Tabelle 4.2 gemacht.

Wertung	Beschreibung
0	Kein erkennbarer Unterschied in der Stimme
1	Stimmung der Worte passt nicht zu akustischer Stimmung. Entweder extrem neutral, oder freudig statt ärgerlich.
2	Leichte Betonung eines Wortes. Veränderung der Stimme, kann nicht genau eingeordnet werden
3	Betonung von bestimmten Worten, lang gezogen (Hyperbel). Verstellung in einem einzigen Wort
4	Verstellung der Stimme in ganzen Phrasen

Tabelle 4.2: Sprachliche Bewertungen von Ironie anhand von Merkmalen in der Stimme

Da auch hier eine feingranulare Klassifikation nach der gegebenen Wertung nicht

möglich ist, wird im Weiteren die Wertungen $0 - 1$ als *Keine Ironie* angesehen und die Wertung $2 - 4$ als Ironie.

4.3.2 Analyse des Datensatzes

Der Umfang des Datensatzes beträgt 1153 Aussagen. Diese Aussagen stammen von sechs Personen bzw. drei Gruppen. Im Anhang A.3 ist ein Ausschnitt des Datensatzes abgebildet.

Reliabilität der Bewertungen

Um die Objektivität der durch die Teilnehmer klassifizierten Daten zu beurteilen, wird das Maß der Reliabilität angewendet. Sie gibt das Ausmaß der Übereinstimmungen von Einschätzungsergebnissen an. Bei der Interrater-Reliabilität handelt es sich um die Reliabilität zwischen zwei oder mehr unterschiedlichen Ratern (Bewertern), bei der Intrarater-Reliabilität zwischen dem selben Rater zu unterschiedlichen Zeitpunkten. Die Kappa-Statistik wird häufig zur Prüfung der Reliabilität verwendet. Das Cohens Kappa wird bei zwei Ratern genutzt, bei Interrater- als auch der Intrarater-Reliabilitäten. Das Fleiss Kappa ist eine Anpassung des Cohens Kappa, sodass es auch für mehr als zwei Rater angewendet werden kann [31]. Zur Berechnung von κ wir der gemessene Übereinstimmungswert $Pr(a)$ und die zufällig erwartete Übereinstimmung $Pr(e)$ benötigt. Der Unterschied in der Berechnung von Cohens und Fleiss Kappa liegt in der Berechnung von $Pr(e)$ und $Pr(a)$. Das Cohens, sowie Fleiss Kappa wird dann folgendermaßen berechnet:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

Das κ kann zwischen -1 und +1 liegen, wobei 0 für eine zufällige Übereinstimmung, 1 für die perfekte Übereinstimmung und -1 für eine systematische Nichtübereinstimmung zwischen den Ratern steht. Kappa-Werte unter 0 sind zwar möglich, aber nach Cohen in der Praxis unwahrscheinlich. Der Kappa-Wert ist ein standardisierter Wert und wird daher über mehrere Studien hinweg gleich interpretiert. Cohen schlägt vor, das Kappa-Ergebnis wie in Tabelle 4.3 zu interpretieren [31].

Kappa-Wert	Übereinstimmung
0-0,20	Keine
0,21-0,39	Minimale
0,40-0,59	Schwache
0,60-0,79	Moderate
0,80-0,90	Starke
über 0,90	Fast Perfekte

Tabelle 4.3: Interpretation der κ-Werte nach Cohen [31]

Bezüglich des erstellten Datensatzes kann die Reliabilität zwischen den beiden Teilnehmenden und der von mir erstellten Klassifikation geprüft werden. Äußerungen die mit einer eventuellen Ironie bewertet wurden, werden als erkannte Ironie gedeutet, sodass nur zwei Klassen (*Ironie* und *Keine Ironie*) zur Auswertung verglichen werden. Dies ist sinnvoll, da vor allem die potenzielle Ironie untersucht werden soll. In Tabelle 4.4 wird die Interrater-Reliabilität zwischen den einzelnen Teilnehmern der Gruppen, Intrarater-Reliabilität der einzelnen Teilnehmer und das Fleiss Kappa der Teilnehmer und der von mir erstellten Klassifikation angegeben.

	Gruppe 1	Gruppe 2	Gruppe 3
Cohens Kappa zwischen A und B	0.61	0.53	0.46
Cohens Kappa zwischen A und A_3	0.61	0.70	0.57
Cohens Kappa zwischen B und B_3	-	0.56	0.60
Fleiss Kappa	0.68	0.62	0.60

Das A_3 und B_3 steht für die Klassifizierung drei Monate nach der ursprünglichen Aufnahme. Diese wurde von fünf der sechs Teilnehmer durchgeführt.

Tabelle 4.4: Reliabilität in den einzelnen Gruppen

Die Cohens Kappa-Werte zwischen den einzelnen Teilnehmern liegen zwischen 0.46 und 0.61. Diese Werte können als eine schwache bis moderate Übereinstimmung interpretiert werden und lassen somit auf eine hohe Subjektivität schließen. Höher sind die κ-Werte zwischen den Klassifikationen der selben Teilnehmer zu unterschiedlichen Zeiten, also die Intrarater-Reliabilität von A zu A_3 und B zu B_3. Sie liegt zwischen 0.56 und 0.7, womit eine moderate Übereinstimmung vorliegt. Daraus lässt sich schließen, dass ein gewisser Verlust des Kontextwissens bei den Teilnehmern vorhanden ist, die meisten Aussagen jedoch ähnlich wie drei Monate zuvor eingeschätzt werden. Das Fleiss Kappa, bei dem neben den Klassifikationen der Teilnehmer noch die eigene Klassifizierung hinzugenommen wird, hat eine moderate Übereinstimmung. Alle Fleiss Kappa-Werte liegen über den jeweiligen Cohens Kappa-Werten zwischen den Teilnehmern. Dies kann nur der Fall sein, wenn die Übereinstimmung zwischen der eigenen Klassifizierung mit mindestens einem der Teilnehmer höher ist, als die zwischen den Teilnehmern.

Für weitere Bearbeitungen wird die eigene Klassifizierung genutzt. Damit wird sichergestellt, dass es sich um die gleiche Definition von Ironie handelt. Zudem erhöht sich die Reliabilität in allen drei Gruppen, wenn diese Klassifizierung hinzugenommen wird.

Verteilung der Aussagen

Jede Person hat innerhalb der dreißig Minuten unterschiedlich viele Äußerungen gemacht, wodurch der Einfluss auf den Datensatz unterschiedlich hoch ist. In Abbildung 4.1 werden die prozentualen Sprechanteile, gemessen an der Anzahl der Äußerungen innerhalb der eigenen Gruppe, aufgezeigt. Zusätzlich ist der Anteil der ironische markierten Äußerungen farblich abgesetzt. In der Gruppe zwei bestehen die größten Unterschiede

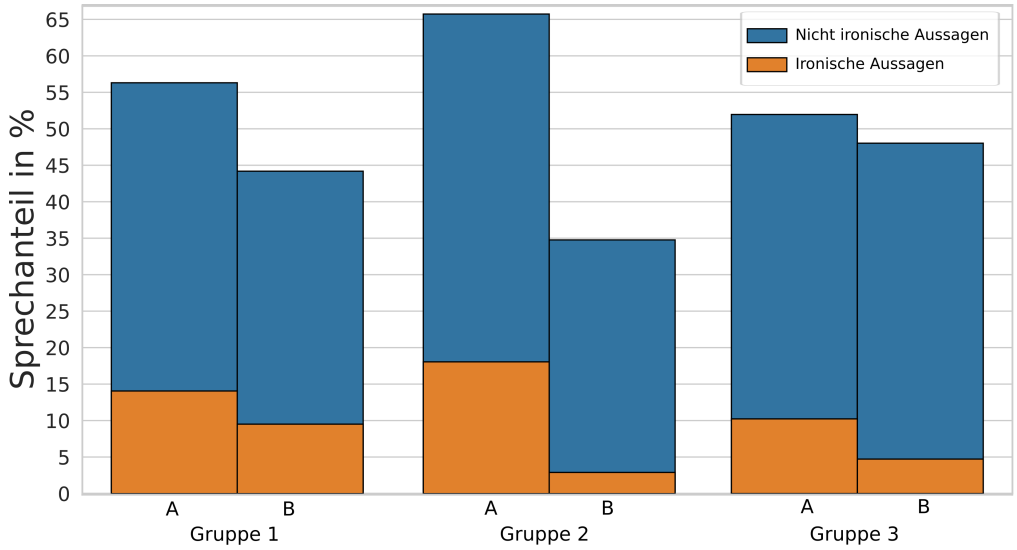

Abbildung 4.1: Sprech- und Ironieanteile der einzelnen Teilnehmer in den drei Gruppen, wobei die Anteile an der Anzahl der getroffenen Äußerungen gemessen sind

zwischen den Sprechanteilen. A hat fast doppelt so viele Äußerungen wie B gemacht und der Ironieanteil von A ist sehr viel höher als der von B. In Gruppe drei sind die Sprechanteile relativ ausgeglichen, trotzdem ist der Anteil an genutzter Ironie von A doppelt so groß wie der Anteil von B. Allgemein besteht die Regel, je größer die Sprechanteile sind, desto mehr Ironie wird von den Teilnehmern genutzt. Zudem schwanken, abhängig des jeweiligen Teilnehmers, die Sprech- und Ironieanteile stark.

In Abbildung 4.2 werden die Sprech- und Ironieanteile der einzelnen Teilnehmer einer Gruppe zusammengefasst dargestellt. Die Angaben beziehen sich dementsprechend auf den gesamten Datensatz.

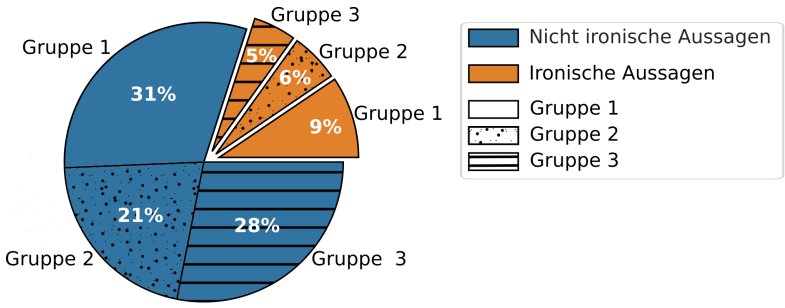

Abbildung 4.2: Anteil der ironischen und nicht ironische Äußerungen der einzelnen Gruppen auf den gesamten Datensatz (1153 Aussagen) bezogen

Der Ironieanteil im Datensatz beträgt 20% und resultiert aus den einzelnen Ironieanteilen der Gruppen. Der größte Anteil an Äußerungen bezogen auf den gesamten

Datensatz kommt mit 40% aus der ersten Gruppe. Sie trägt mit 9% auch am Meisten zu den ironischen Äußerungen bei. Obwohl die zweite Gruppe einen geringeren Gesamtanteil (27%) als die dritte Gruppe (33%) beiträgt, ist der Anteil an Ironie mit 6% um ein Prozent höher als die der dritten Gruppe (5%).

Analyse verbaler und akustischer Merkmale

Ironische Äußerungen können auf unterschiedlichen Ebenen Ironiesignale enthalten. Diese Ironiesignale sind für die automatische Klassifizierung sehr wichtig. Jede ironische Äußerung wurde auf die Signale nach den Tabellen 4.1 und 4.2 untersucht und zugeordnet. In Tabelle 4.5 wird die Verteilung der bewerteten ironischen Aussagen angegeben. Der größte Anteil enthält jeweils keine akustischen oder verbalen Merkmale. Für eine mögliche Klassifizierung werden die verbalen Wertungen 1-3 und die akustischen Wertungen 2-4 als ironisch geltend zusammengefasst.

Wertung	Verbale Merkmale	Akustische Merkmale
0	106	129
1	33	27
2	58	24
3	34	48
4	-	3

Tabelle 4.5: Verteilung der ironischen Aussagen auf die Wertungen verbaler und akustischer Merkmale

Daraus folgt in Abbildung 4.3 die Darstellung, in welchem Maße verbale oder akustische Merkmale in ironischen Äußerungen enthalten sind. Insgesamt fallen 37% der ironischen Aussagen für die automatische Klassifizierung weg, da sie keine verbalen oder akustischen Merkmale enthalten.

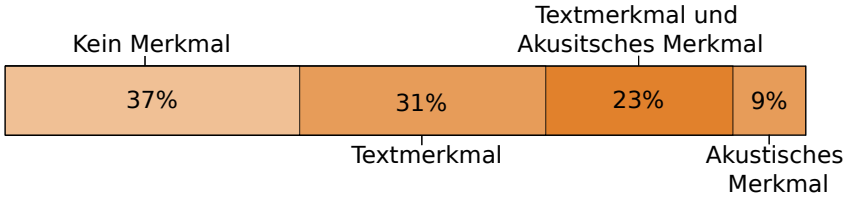

Abbildung 4.3: Anteile der kategorisierten Ironiemerkmale von den insgesamt 231 ironischen Aussagen

Anstatt verbaler oder akustische Merkmale enthalten die wegfallenden Aussagen dafür häufig ausschließlich Merkmale der Kontextebene. Dazu gehören beispielsweise folgende Äußerungen, die alle neutral ausgesprochen wurden:

- „Es fehlt glaube ich ein o.“

- „Dieser Kreis hier ist kreisrund.“
- „Ja, es war grün vorhanden.“

All diese Aussagen zeigen erst in ihrem Kontext, dass es sich um Ironie handeln muss, da die Äußerungen an sich nicht der Wahrheit entsprechen. Solche Äußerungen sind auch für Außenstehende schwer als Ironie zu klassifizieren.

Insgesamt enthalten 63% der ironischen Äußerungen ein verbales oder akustisches Ironiesignal, wobei sogar 23% beide Formen der Ironiesignale enthalten. Beispiele aus den 54% ironischen Aussagen mit Textmerkmal sind:

- „Das richtig schöne fancy Windows Teil.“
- „Das ist so gut dokumentiert, das wird schon richtig sein. “
- „Du Intelligenzbestie, alter, ein Kreis.“

32% der ironischen Aussagen beinhalten akustische Merkmale. Beispiele für die akustischen Ironiesignale sind schwer in Textform wiederzugeben. Alle Beschreibungen aus der Tabelle 4.2 sind im Datensatz gegeben.

Wortanalyse

Einige Worte werden häufiger in ironischen Aussagen genutzt als andere. Mittels der Python Bibliothek *wordcloud* [2] wird dieser Aspekt in der Abbildung 4.4 visualisiert dargestellt. Je größer ein Wort abgebildet ist, desto häufiger ist es in ironischen Äußerungen, die ein verbales Merkmal enthalten, aufgetaucht.

Abbildung 4.4: Wortwolke aus ironisch gemeinten Aussagen, die verbale Ironiesignale enthalten (ohne Stoppwörter)

Um die häufig auftretenden Wörter, die gewöhnlich keine Relevanz haben, nicht abzubilden, wurden vor der Erstellung der Wortwolke Stoppwörter aus der deutschen

[2] https://amueller.github.io/word_cloud/

Stoppwortliste der Python Library NLTK[3] (Natural Language Toolkit) entfernt. Da sich in der Stoppwortliste auch Modalpartikel befinden, diese jedoch ein Hinweis auf Ironie sein können, wurden Modalpartikel aus der Stoppwortliste entfernt. Es handelt sich speziell um die Modalpartikel *aber*, *auch*, *denn*, *doch*, *nur*. Besonders auffällig in Abbildung 4.4 ist das Wort *ja* welches nicht nur als zustimmende Antwort genutzt werden kann, sondern auch als Bekräftigung für eine Aussage dient. Weitere häufige Wörter sind *natürlich*, *super*, *gut* und *richtig*. Nur allein anhand dieser Wörter kann nicht auf Ironie geschlossen werden, da sie auch sehr häufig in nicht ironischen Aussagen vorkommen. Wenn man die Wörter jedoch kombiniert, wie zum Beispiel zu *„Das finde ich ja wunderschön.“*, oder *„Das ist garantiert eine super Idee.“*, steigt das Potenzial, dass es sich hierbei um Ironie handelt. Insgesamt befinden sich in der Wortwolke viele Adjektive und Modalpartikel wieder. Die Adjektive haben vor allem eine positive Polarität (*schön, gut, super, wunderschön*). Bei den Modalpartikeln handelt es sich vor allem um *auch*, *doch*, *ja* und *mal*. Die Abbildung 4.5 zeigt zum Vergleich Wörter, die häufig in nicht ironischen Aussagen enthalten sind.

Abbildung 4.5: Wortwolke aus ironisch gemeinten Aussagen, die **keine** verbale Ironiesignale enthalten (ohne Stoppwörter)

Vergleicht man die Abbildungen 4.4 und 4.5 sind einige Übereinstimmungen erkennbar. In beiden Abbildungen ist das *Ja* sehr groß abgebildet, wobei es in der zweiten Abbildung groß geschrieben wird. Dies deutet daraufhin, dass das *Ja* am Anfang eines Satzes steht, und für eine Zustimmung genutzt wird. Auffällig ist zudem, dass die Wortwolke in Abbildung 4.5 sehr viel mehr Wörter enthält. Dies liegt vor allem an dem großen Anteil nicht ironischer Aussagen, wodurch mehr Wörter in mehreren Aussagen auftauchen.

Weitere Zusammenhänge in den Aussagen

Es ist auffällig, dass innerhalb des Datensatzes ironische Äußerungen häufig nacheinander stehen. Auf eine ironische Aussage folgt vom Gegenüber eine ironische Antwort. Mittels

[3] https://www.nltk.org/index.html

einer ironischen Antwort kann gezeigt werden, dass die Ironie der ersten Aussage verstanden wurde. Ein Beispiel für einen solchen Dialog ist:

> A: „Ja, fällt dir gerade so ein spontaner Hexadezimalfarbton ein den du unbedingt haben willst? “
> B: „Ähm, ich glaube FFB007.“
> A: „Ja aber, ich fand FC98 irgendwie besser. Der gibt mir ein Gefühl von Wärme.“
> B: „Ja der hat ein bisschen sattere Farbe ne.“
> A: „Ja ja, genau genau genau. “

Um die Ironie der Aussagen zu verstehen, wird das Kontextwissen benötigt, dass A sowie B keine Ahnung haben welche Farben sich hinter welchen Hexadezimal-Codes befinden.

Abbildung 4.6 zeigt einen Übergangsgraphen für die Wahrscheinlichkeiten, ob ironische oder nicht ironische Aussagen aufeinander folgen. Dabei handelt es sich um alle ironischen Aussagen, nicht nur die, welche verbale oder akustische Merkmale beinhalten. Es gibt knapp viermal so viele nicht ironische Äußerungen in dem Datensatz wie ironische.

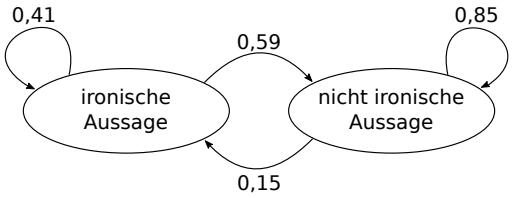

Abbildung 4.6: Übergangsgraph für das Aufeinanderfolgen von ironischen (231 Aussagen) oder nicht ironischen Aussagen (922 Aussagen)

Das bedeutet, die Wahrscheinlichkeit für das Auftreten einer nicht ironischen Äußerung ist höher als die Wahrscheinlichkeit für das Auftreten einer ironischen Äußerung. Demnach fällt auf, dass 41% der ironischen Aussagen nach einer ironischen Aussage erfolgt sind. Nach nicht ironischen Aussagen folgen im Gegensatz dazu nur 15% ironische Äußerungen. Dieser Aspekt könnte für die Klassifizierung hilfreich sein, jedoch ändern sich die Werte, wenn nur die ironischen Aussagen mit verbalen und akustischen Signalen betrachtet werden. Dann folgt eine ironische Aussage auf eine ironische Aussage mit der gleichen Wahrscheinlichkeit, wie eine ironische Aussage auf eine nicht ironische Aussage. Dies liegt vor allem daran, dass das Ironiesignal für aufeinanderfolgende ironische Aussagen im Kontext der vorherigen Aussage liegt. Da in dieser Arbeit die Merkmale der Kontextebene nicht näher betrachtet werden, kann dieser Zusammenhang der ironischen Aussagen nicht genutzt werden.

4.4 Verarbeitungsprozess

In dem Verarbeitungsprozess werden aus Rohdaten einer Klassifikation numerische Merkmale gebildet, welche zu einer Klassifikation der Daten in *Keine Ironie* und

Potenzielle Ironie führt. Der Prozess teilt sich in eine Trainings- und eine Testphase, für die unterschiedliche Komponenten des Datensatzes genutzt werden. Die einzelnen Schritte bis zur Klassifizierung werden in Abbildung 4.7 dargestellt. Die einzelnen Teilschritte der Trainings- und Testphase unterscheiden sich in drei wesentlichen Punkten. In der Testphase wird der Datensatz nicht ausgeglichen, es werden keine Lexika erstellt und das Klassifizierungsmodell, das aus der Trainingsphase entstanden ist, wird hier angewendet. In beiden Phasen werden die Daten in einem Preprocessing-Schritt vorverarbeitet und aus diesen vorverarbeiteten Daten werden Merkmale extrahiert. Um das Klassifizierungsmodell zu erstellen werden zu den Merkmalen die zugehörigen Label (*Ironie*, *Keine Ironie*) in die SVM gegeben. Im Folgenden werden die Teilschritte des Verarbeitungsprozesses genauer aufgeführt.

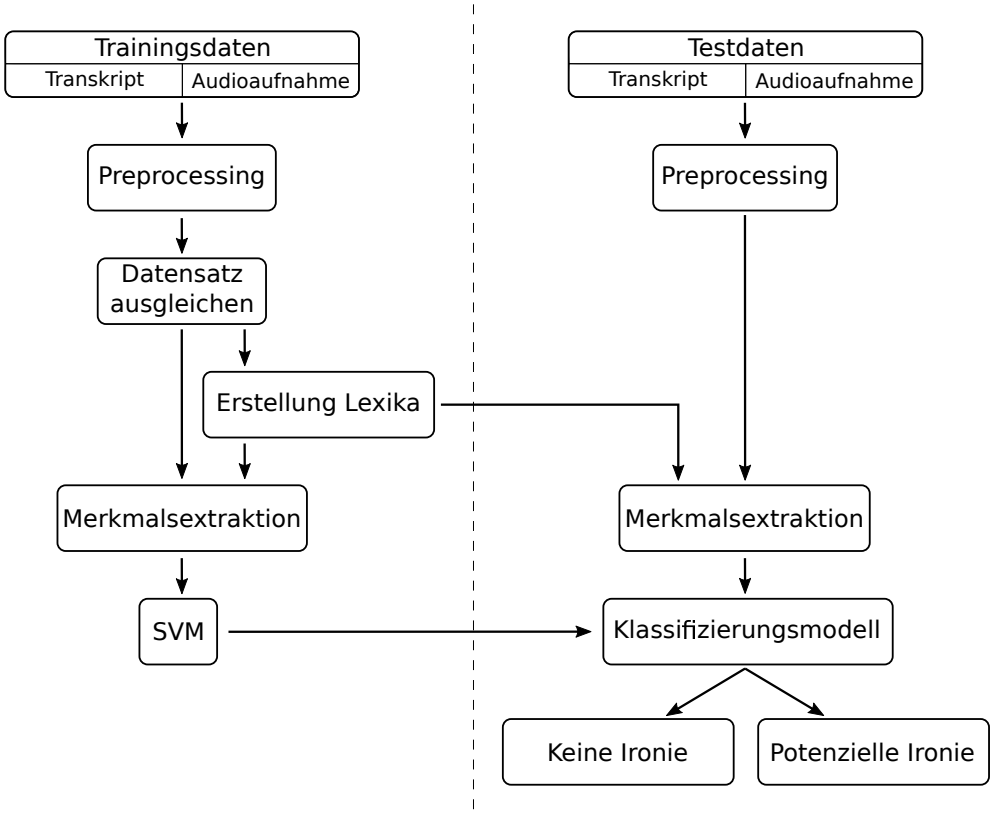

Abbildung 4.7: Verarbeitungsprozess zur Klassifizierung ironischer und potenziell ironischer Aussagen

4.4.1 Erstellung des Trainings- und Testdatensatzes

Bevor der Verarbeitungsprozess beginnen kann, muss der Datensatz in einen Trainings- und einen Testdatensatz aufgeteilt werden. Der Trainingsdatensatz wird, wie der Name bereits andeutet, zum Trainieren genutzt. Anhand der Eingangsdaten erlernt die SVM

Muster und Zusammenhänge. Die Testdaten werden innerhalb des Trainings nicht genutzt. Mit ihnen wird die Qualität des erstellten Klassifizierungsmodells überprüft. In dieser Arbeit wird die k-fache Kreuzvalidierung (*engl. k-fold cross-validation*) für die Aufteilung und Validierung des Datensatzes verwendet. Dafür wird der Datensatz in k gleich große Segmente aufgeteilt. Es werden $k - 1$ Segmente als Trainingsdaten genutzt. Das übrig gebliebene Segment wird als Testdatensatz angewendet. Dieses Vorgehen wird k-Mal wiederholt, sodass insgesamt k Verarbeitungsprozesse durchlaufen werden und k unterschiedliche Klassifizierungsmodelle entstehen. Ein Vorteil der k-fachen Kreuzvalidierung ist, dass die Performanz eines Modells genauer approximiert werden kann, als wenn der Datensatz nur einmal in einen Trainings- und Testdatensatz geteilt wird. Zudem wird mit diesem Vorgehen sichergestellt, dass alle Daten sowohl zum Trainieren als auch zum Testen verwendet werden [42]. Für k wird in dieser Arbeit 5 eingesetzt, sodass jeder Testdatensatz aus 20% des gesamten Datensatzes besteht. Daraus folgt, dass der Testdatensatz etwa 230 Aussagen enthält, von denen etwa 29 Aussagen ironisch sind. Wenn das k zu groß gewählt wird, befinden sich zu wenige ironische Aussagen im Testdatensatz, um aussagekräftige Informationen aus der Validierung zu ziehen.

4.4.2 Preprocessing

Damit die Merkmale aus den Text- und Audiodaten extrahiert werden können, müssen die Daten zuvor das Preprocessing durchlaufen. Die Audiodaten sind mit einer Frequenz von 42 kHz aufgenommen worden. Da das menschliche Gehör Schallwellen im Frequenzbereich zwischen 20 Hz (untere Grenze) und 16 kHz (obere Grenze) wahrnehmen kann [10], werden die Audiodaten ohne einen großen Informationsverlust auf $16kHz$ unterabgetastet. Zusätzlich wäre ein Anpassen der Audiolänge der einzelnen Aussagen denkbar, da dadurch die MFCCs der Audioabschnitte besser vergleichen werden können. Dies ist bei den großen Unterschieden zwischen den Längen der Audioabschnitte jedoch für nicht sinnvoll erachtet worden.

Die Textdaten werden innerhalb des Preprocessing in Tokens aufgeteilt. Jedes Token erhält ein POS-Tag, welches die Wortart des Tokens angibt. Zusätzlich ist in jedem Token der morphologische Wortstamm, der anhand Lemmatisierung bestimmt wurde, gespeichert. Ein Beispiel für POS-Tags und Lemmatisierung findet sich in Abschnitt 2.3.

4.4.3 Datensatz ausgleichen

Standardklassifizierer neigen dazu, sich hauptsächlich mit den großen Klassen eines Datensatzes zu beschäftigen und die kleinen zu ignorieren. Das führt dazu, dass die kleinen Klassen nie detektiert werden [41]. Der in dieser Arbeit angewendete Datensatz enthält 20% Ironie, wobei nicht der gesamte Anteil genutzt werden kann. 34% von den als ironisch klassifizierten Aussagen weisen keine verbalen oder akustischen Ironiemerkmale auf, sodass nur 13% des Datensatzes für die Klassifikation als Ironie deklariert ist. Dies ist im Verhältnis zu der Klasse von *Keiner Ironie* (87%) ein sehr geringer Anteil, weshalb der Datensatz für das Training ausgeglichen werden muss. Zum Ausgleichen können

entweder Methoden der Unterabtastung (*engl. undersampling*) oder Überabtastung (*engl. oversampling*) angewendet werden. Das Ziel der Methoden ist es, ein Gleichgewicht der Klassen herzustellen. Beim Undersampling wird die Mehrheitsklasse durch das Entfernen von Beobachtungen verkleinert, bis der Datensatz ausgeglichen ist. Beim Oversampling wird dagegen die Minderheitsklasse durch Hinzufügen von Beobachtungen vergrößert. In dieser Arbeit wird eine einfache Methode des Oversampling genutzt, um den Datensatz auszugleichen. Aus der unterrepräsentierten Klasse, die Klasse der ironischen Aussagen, werden zufällige Aussagen gezogen und verdoppelt. Dieser Vorgang wird so häufig wiederholt, bis in der unterrepräsentierten Klasse genauso viele Aussagen vorliegen, wie in der überräprestierten Klasse der nicht ironischen Aussagen. Das Ausgleichen des Datensatzes wird nur für den Trainingsdatensatz durchgeführt. In einem Trainingsdatensatz der aus 80% des gesamten Datensatzes besteht, gibt es etwa 100 ironische und 820 nicht ironische Aussagen. Somit erhöht sich durch die Verdopplung von ironischen Aussagen die Anzahl der ironischen Aussagen von 100 auf 820.

4.4.4 Erstellung der Lexika

Mithilfe eines Lexikons auf Basis von ironischen und eines Lexikons auf Basis von nicht ironischen Aussagen, soll ein *Grad* von *Ironie* oder *Keiner Ironie* bestimmt werden. Für die Erstellung eines Lexikons wird das Lemma der aus dem Preprocessing entstandenen Tokens normalisiert. Für die Normalisierung werden alle Zeichen der Tokens auf Kleinbuchstaben reduziert und Zahlen sowie Satzzeichen entfernt. Zuletzt werden Stoppwörter, wie in der Wortanalyse in Abschnitt 4.3.2 beschrieben, mithilfe einer Stoppwortliste entfernt. Die Modalpartikel, die sich in der Stoppwortliste befinden werden nicht aus den Textdaten entfernt.

Nachdem diese Vorverarbeitung der Aussagen stattgefunden hat, wird die Häufigkeit der normalisierten Token in einem Lexikon vermerkt. Die Anzahl des Vorkommens wird zuletzt durch den Wert des am häufigsten vorkommenden Token geteilt. So liegen alle Einträge zwischen 0 und 1. Es wird ein Lexikon für Wörter aus ironischen Aussagen und ein Lexikon für Wörter aus nicht ironischen Aussagen erstellt. Die Tabelle 4.6 zeigt die ersten sieben Einträge der beiden Lexika, die anhand des gesamten Datensatzes erstellt wurden.

Wörter aus ironischen Aussagen		Wörter aus nicht ironischen Aussagen	
ja	1,0	ja	1,0
gut	0,75	auch	0,44
auch	0,67	ne	0,42
richtig	0,58	mal	0,38
super	0,33	ok	0,31
glauben	0,33	aber	0,28
natürlich	0,33	code	0,24

Tabelle 4.6: Auszug aus ironischem und nicht ironischem Lexikon

Es lassen sich die gleichen Schlüsse wie aus den Wortwolken der Wortanalyse in Abschnitt 4.3.2 ziehen, da beide Methoden auf die Worthäufigkeit zurückzuführen sind. Das *ja* ist das Wort, welches am Häufigsten genutzt wurde, wobei es im Verhältnis zum zweithäufigsten Wort einen größeren Abstand bei den nicht ironischen Aussagen aufweist.

4.4.5 Merkmalsextraktion

Die Merkmalsextraktion (*engl. feature extraction*) ist der Prozess, in der Rohdaten in numerische Merkmale umgewandelt werden. Diese numerischen Merkmale können anschließend von einem Klassifizierer verarbeitet werden. Aus den vorliegenden Daten werden 22 Merkmale für jede Aussage herausgearbeitet. Davon bilden sich 14 Merkmalsvektoren aus den Audiomitschnitten und acht aus den Textdaten. Um welche Merkmale es sich spezifisch handelt, wird in den folgenden Unterabschnitten erläutert.

Akustische Merkmale

Audiodaten sind sehr umfangreich. Allein eine Sekunde, die mit $16kHz$ abgetastet wird, enthält 16000 Werte. Diese umfangreichen Daten können nicht direkt zur Klassifikation genutzt werden. Um die wichtigen Merkmale aus den Audiodaten zu erhalten, werden in dieser Arbeit MFCCs, die wesentlich kompakter als die Eingangsdaten sind, berechnet. Das Verfahren, um MFCCs eines Audioabschnitts zu generieren, wird in Abschnitt 2.4 genauer erläutert. Für die Framebreite und Schrittweite werden die Standardwerte $25ms$ und $10ms$ übernommen. Dadurch, dass die Schrittweite geringer ist als die Framebreite, findet eine gewisse Überlappung der Frames statt. Dies soll einen Informationsverlust verringern. Die Anzahl an genutzten Koeffizienten für die Spracherkennung beträgt in den meisten Fällen 12 [10]. Da in dieser Arbeit jedoch nicht die Sprache, sondern vor allem Betonungen erkannt werden sollen, werden wie in Burckard et al. [6] die ersten 14 Koeffizienten berechnet. Durch die unterschiedlichen Längen der Audioausschnitte ergibt sich für jeden Audioausschnitt eine andere Anzahl an Frames. Für jedes Frame werden die ersten 14 MFCCs berechnet. Um die MFCCs unterschiedlicher Audioabschnitte vergleichen zu können, wird das arithmetische Mittel jedes MFCCs (1.-14.) aller Frames eines Audioabschnittes berechnet. Somit erhält jede Aussage 14 Merkmalswerte zugehörig zu dem Audiomitschnitt.

Stimmungsmerkmale

Die Stimmung einer Aussage wird in vielen Arbeiten für die Ironieerkennung extrahiert [6, 16, 48]. Dazu kann beispielsweise die Polarität einer Aussage von negativ, positiv oder neutral bestimmt werden. Die Polarität lässt sich an verschiedenen Merkmalen ausmachen. Ein Anhaltspunkt ist die Polarität einzelner Wörter oder Emoticons. Wie in Abschnitt 2.3 erläutert, werden in Sentiment-Lexika Polaritätswerte für stimmungstragende Wörter notiert. Aus diesen lässt sich die Polarität einer vollständigen Aussage bestimmen, sodass sie als Merkmal für die Klassifikation dienen können. Bei der Auswahl eines Sentiment-Lexikons ist es wichtig, die thematische Domäne zu

beachten, denn je nach Domäne können Wörter unterschiedlich bewertet werden [48]. Das Lexikon SentiStrength-SE [51] ist speziell auf die Software Engineering Domäne angepasst, jedoch liegt es nur in englischer Sprache vor. Somit wird auf das Lexikon sentiWS [43] zurückgegriffen, wobei einige wenige Anpassung aufgrund der Domäne und des Datensatzes vorgenommen werden. Zum Beispiel wurde das Wort *Fehler*, sowie *Fehlermeldung* aus dem Lexikon entfernt, da in der Softwareentwicklung häufig neutral über Fehler gesprochen werden muss. Hinzugefügt wurden die Wörter *Profi*, sowie *Joke* mit den Sentiment-Werten der Synonyme *Experte* und *Witz*. Neben dem Polaritätswert, der zwischen -1 und 1 liegt, ist die Wortart in dem Sentiment-Lexikon vermerkt. Damit die genutzten Wörter einer Aussage in dem Lexikon gefunden werden, muss nicht nur die Schreibweise, sondern auch die Wortart übereinstimmen. Adverbial genutzte Adjektive erhalten das POS-Tag Adverb, in dem Sentiment-Lexikon sind sie jedoch nur als Adjektiv vermerkt. Damit auch diese Adjektive mit in die Polarität einer Aussage fließen, wurde jeder Eintrag, der als Adjektiv vermerkt ist, zusätzlich als Adverb vermerkt. Ohne diese Anpassung hätte in der Äußerung *„Das wird auch so garantiert richtig schön aussehen.“* kein Wort eine Polarität. Mit der Anpassung erhalten die Adjektive *garantiert*, *richti* und *schön* eine Polarität. Bei Verneinungen durch Wörter wie *nicht* oder *kein* werden die Polaritäten einer Aussage umgekehrt.

Für die Berechnung der Polarität einer Aussage werden die Polaritätswerte der einzelnen Wörter der Aussage summiert. Die Summe wird anschließend durch die Anzahl der Werte, die in die Berechnung mit einfließen, geteilt. Neben der Polarität einer Aussage werden noch die Anteile positiver sowie negativer Wörter in einer Aussage als Merkmale für die Klassifizierung genutzt. In den folgenden Beispielen sind die Polaritätswerte aus dem angepassten sentiWs Lexikon in Klammern hinter den Wörtern notiert. Wenn kein Wert angegeben ist, dann befindet sich das Wort nicht in dem Lexikon. Die ersten beiden Aussagen sind ironisch gemeint worden und die anderen beiden nicht.

(1) Das wird auch so garantiert (0,004) richtig (0,004) schön (0,0081) aussehen.
(2) Ja ich habe immer, immer gute (0,3716) Ideen.
(3) Ich will diese Kante loswerden(-0,0048).
(4) Oh, das ist immer noch kein gutes (0,3716) Formatting hier.

Die Tabelle 4.7 zeigt die Berechnungen der Stimmungsmerkmale für die Beispiele.

Beispiel	(1)	(2)	(3)	(4)
Polarität	$\frac{0{,}004+0{,}004+0{,}0081}{3} = 0,03$	$\frac{0{,}3716}{1} = 0,3716$	$\frac{-0{,}0048}{1} = -0,0048$	$\frac{0{,}3716}{1} \cdot (-1) = -0,3716$
Positiver Anteil	$\frac{3}{3} = 1$	$\frac{1}{1} = 1$	$\frac{0}{1} = 0$	$\frac{0}{1} = 0$
Negativer Anteil	$\frac{0}{3} = 0$	$\frac{0}{1} = 0$	$\frac{1}{1} = 1$	$\frac{1}{1} = 1$

Tabelle 4.7: Berechnung der Stimmungsmerkmale

Da keine der Aussagen unterschiedliche Polaritäten enthält, ist der negative sowie positive Anteil entweder null oder eins. Das *kein* in Beispiel vier weist keine Polarität auf, hat aber trotzdem einen großen Einfluss auf die Polarität der Aussage. Durch das *kein* kehren sich alle Polaritäten in dem Satz um, sodass die Aussage von einer positiven Polarität zu einer negativen wird.

Merkmale aus der Worthäufigkeit

Um einen weiteren Anhaltspunkt für ironische Aussagen zu haben, wurden Lexika aus den Wörtern ironischer und nicht ironischer Aussagen gebildet. Das Vorgehen wird in Unterabschnitt 4.4.4 erläutert. Um mithilfe der Lexika einen numerischen Merkmalswert zu berechnen, wird jede zu überprüfende Aussage auf die selbe Art normalisiert, wie es bei der Erstellung der Lexika der Fall war. Nach der Normalisierung besteht die Aussage nur noch aus Token. Für jedes Token aus der zu untersuchenden Aussage wird die jeweilige Wertung aus dem Lexikon summiert. Die Summe wird anschließend durch die Anzahl der Token der Aussage geteilt. Für jede Aussage ergeben sich somit zwei neue Merkmale.

Der eine Merkmalswert steht für einen wörtlichen Grad an Ironie, der andere für einen Grad von *Keiner Ironie*. Um das Vorgehen zu verdeutlichen, werden anhand der Beispiele des letzten Abschnitts die Werte berechnet. Die Lexika wurden wie in Abschnitt 4.4.4 mittels des gesamten Datensatzes erstellt. Normalerweise werden die Lexika nur anhand des Trainingsdatensatzes erstellt. Zuerst werden die Aussagen aus den Beispielen normalisiert. Hinter den Token stehen in Klammern die Werte, die in den Lexika vermerkt sind (Wert aus Ironie-Lexikon|Wert aus Keine-Ironie-Lexikon).

(1) auch (0,67|0,44) garantieren (0,13|0,0) richtig (0,58|0,06) schön (0,17|0,06) aussehen (0,08|0,0)

(2) ja (1,0|1,0) immer (0,21|0,06) immer (0,21|0,06) gut (0,75|0,21) idee (0,25|0,01)

(3) kante (0,04|0,02) loswerden (0,0|0,01)

(4) oh (0,17|0,1) immer (0,2|0,06) gutes (0,0|0,01) formatting (0,0|0,01)

Die Tabelle 4.8 zeigt die Berechnungen für die jeweiligen Merkmale.

Beispiel	(1)	(2)	(3)	(4)
Ironie	$\frac{0{,}67+0{,}13+0{,}58+0{,}17+0{,}8}{5} = 0,47$	$\frac{1+0{,}21+0{,}21+0{,}75+0{,}25}{5} = 0,484$	$\frac{0{,}04}{1} = 0,04$	$\frac{0{,}17+0{,}2}{2} = 0,185$
Keine Ironie	$\frac{0{,}44+0{,}06+0{,}06}{3} = 0,187$	$\frac{1+0{,}06+0{,}06+0{,}21+0{,}01}{5} = 0,268$	$\frac{0{,}02+0{,}01}{2} = 0,015$	$\frac{0{,}1+0{,}06+0{,}01+0{,}01}{4} = 0,045$

Tabelle 4.8: Berechnung des Ironiegrads

Auffällig ist, dass die Werte für *Keine Ironie* immer kleiner sind als die für *Ironie*. Dies liegt vor allem daran, dass auf Grundlage der Skalierung mit dem Wort welches

am Häufigsten vorkommt, die Werte in dem Lexikon für *Keine Ironie* generell kleiner sind. Allgemein müssen diese Werte deshalb separat, als im Zusammenhang betrachtet werden.

Merkmale aus den Wortarten

Wie in den Grundlagen in Abschnitt 2.1 erwähnt, kann die Verwendung von vielen Adjektiven oder Adverbien auf Ironie hinweisen. Außerdem treten gehäuft Personalpronomen in Bewertungen und ironischen Aussagen auf. Diese wurden bei den bisherigen Merkmalen nicht betrachtet, da es sich um Stoppwörter handelt und sie keine Polarität aufweisen. Aus diesem Grund wird mittels POS-Tags die Wortart der Wörter auf Adjektiven, Adverbien und Personalpronomen untersucht. Bei den genutzten Merkmalen handelt es sich um das Verhältnis der Anzahl von Adjektiven oder Adverbien, das Verhältnis von hintereinander stehenden Adjektiven oder Adverbien und dem Verhältnis von Personalpronomen, im Bezug zu allen Wörtern der Aussage.

Für die Veranschaulichung der Berechnungen wurden wieder die Beispielaussgen aus den vorherigen Abschnitten übernommen. Die betrachteten Wortarten wurden in den Beispielen folgendermaßen abgegkürzt: Adverbien mit ADV, Adjektive mit ADJ und Personalpronomen mit PPER.

(1) Das wird auch (ADV) so (ADV) garantiert (ADV) richtig (ADV) schön (ADV) aussehen.
(2) Ja ich (PPER) habe immer (ADV), immer (ADV) gute (ADJ) Ideen.
(3) Ich (PPER) will diese Kante loswerden.
(4) Oh, das ist immer (ADV) noch (ADV) kein gutes (ADJ) Formatting hier (ADV).

In Tabelle 4.9 werden die Werte für die drei Merkmale angegeben. Die Anzahl an Wörtern, durch die geteilt wird, wird anhand der separierbaren Wörter durch Leerzeichen berechnet. Bei diesen Beispielen sind die Werte der ironischen Aussagen (1 und 2) tendenziell größer.

Beispiel	(1)	(2)	(3)	(4)
Adjektive	$\frac{5}{8} = 0,625$	$\frac{3}{7} = 0,429$	$\frac{0}{5} = 0$	$\frac{4}{8} = 0,5$
Doppelte Adjektive	$\frac{3}{8} = 0,375$	$\frac{2}{7} = 0,286$	$\frac{0}{5} = 0$	$\frac{1}{8} = 0,125$
Personalpronomen	$\frac{0}{8} = 0$	$\frac{1}{7} = 0,143$	$\frac{1}{5} = 0,2$	$\frac{0}{8} = 0$

Tabelle 4.9: Berechnung der Häufigkeit bestimmter Wortarten oder Folgen von Wortarten

4.4.6 Klassifizierungsmodell

Die SVM, deren Verfahren in Abschnitt 2.2 beschrieben wird, berechnet anhand eines Merkmalsvektors und der dazugehörigen Label (*Ironie*, *Keine Ironie*) ein Klassifi-

zierungsmodell. Dabei werden der SVM ausschließlich Trainingsdaten zur Verfügung gestellt, damit das Klassifizierungsmodell anschließend mit den Testdaten evaluiert werden kann. Jeder Merkmalsvektor beinhaltet insgesamt 22 Merkmale, wobei 14 aus der Analyse der Audiodaten stammen und 8 aus der Analyse der Textdaten. Um die Testdaten mit dem erstellten Klassifizierungsmodell zu klassifizieren, werden mit ihnen die selben Preprocessing Schritte, wie sie mit den Trainingsdaten in der Trainingsphase vorgenommen wurden, durchgeführt. Die Merkmalsextraktion findet mithilfe der erstellten Lexika aus der Trainingsphase statt. Anhand des Merkmalsvektors einer Aussage bestimmt das Klassifizierungsmodell anschließend, ob eine Aussage möglicherweise ironisch ist oder nicht.

4.5 Metriken

Um die entstandenen Klassifizierungsmodelle zu interpretieren bzw. zu evaluieren, werden geeignete Kennzahlen für die Bewertung von Klassifikationsalgorithmen benötigt. Dafür werden die sehr verbreiteten Metriken Accuracy, Precision, Recall und F_1-Score genutzt.

Die Berechnungen der Kennzahlen beruhen auf vier möglichen Fällen, die ein Ergebnis einer Klassifikation annehmen kann, wenn eine ironische Aussage als positiv und eine nicht ironische Aussage als negativ angesehen wird:

- True Positive (TP): Eine ironische Aussage wurde als ironisch klassifiziert.
- True Negative (TN): Eine nicht ironische Aussage wurde als nicht ironische klassifiziert.
- False Positive (FP): Eine nicht ironische Aussage wurde als ironische klassifiziert
- False Negative (FN): Eine ironische Aussage wurde als nicht ironische Aussage klassifiziert

Eine Konfusionsmatrix, wie in Abbildung 4.8 dargestellt, fasst die vier Fälle in einem Diagramm zusammen.

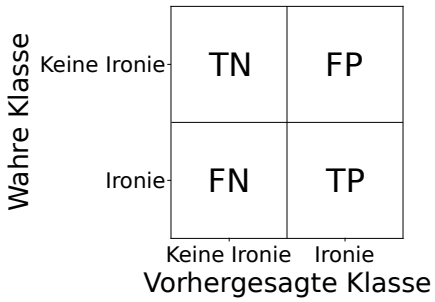

Abbildung 4.8: Aufbau einer Konfusionsmatrix

Die Summe aller Fälle (TP+TN+FP+FN) ergibt die gesamte Anzahl an Aussagen die in der Klassifikation ausgewertet wurden.

Die **Accuracy**, oder auch als Genauigkeitsgrad bezeichnet, definiert die Anzahl an korrekt bestimmten Aussagen. Sie wird anhand folgender Formel bestimmt:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Wenn die Klassen innerhalb eines Datensatzes unausgeglichen verteilt sind, eignet sich diese Kennzahl weniger. Angenommen 90% der Daten eines Datensatzes sind der Klasse *Keine Ironie* und die verbleibenden 10% der Klasse *Ironie* zugeordnet. Ein Klassifizierer, der jeder Aussage die Klasse *Keine Ironie* zuordnet, hätte einen Accuracy-Wert von 90%, obwohl er über keine korrekte Abbildung der Eingabedaten auf die Ausagebdaten verfügt. Deshalb ist es wichtig auch auf andere Kennzahlen zu schauen.

Die Kennzahl **Precision**, oder auch als Präzision bezeichnet, berechnet das Verhältnis von den positiv richtig klassifizierten Aussagen, zur Gesamtheit der positiv klassifizierten Aussagen mit folgender Formel:

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

Der Precision-Wert kann sehr hoch sein, wenn die Aussagen die als Ironie vorhergesagt wurden auch wirklich ironisch sind. Wenn ironisch vorhergesagte Aussagen, in Wahrheit nicht ironisch sind, verringert sich der Wert.

Der **Recall** wird auch als Sensitivität oder Trefferquote bezeichnet. Er repräsentiert den Anteil der korrekt als positiv erkannten Aussagen, zu allen positiven Aussagen und wird folgendermaßen berechnet:

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

Wenn alle Aussagen als ironisch vorhergesagt werden, wäre der Recall-Wert auf 1, auch wenn darunter viele falsch positiven Bewertungen wären. Sobald ironische Aussagen nicht als ironisch vorhergesagt werden, sinkt der Wert. Für eine Detektion von möglicher Ironie, sollte dieser Wert hoch sein, damit die Ironie, die vorhanden ist auch erkannt wird.

Der **F_1-Score** gilt als harmonisches Mittel zwischen Precission und Recall.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Da alleine ein hoher Recall-Wert oder Precision-Wert nicht aussagt, ob ein Modell besonders gut ist, werden sie im F_1-Score zusammengesetzt. Es ist ein Wert bezüglich der allgemeinen Genauigkeit eines Klassifizierungsmodell.

Precsion, Recall und F_1-Score vernachlässigen die korrekte Klassifizierung negativer Aussagen und konzentrieren sich stattdessen auf die als positiv prognostizierten Datensätze. Aus diesem Grund werden bei der Validierung der Klassifizierungsmodelle die Kennzahlen auch für die als negativ angesehen Klasse angegeben.

4.6 Prototypische Umsetzung

In diesem Abschnitt soll die Implementierung des Prototypen vorgestellt werden. Das Ziel des Protoypen ist es zu zeigen, ob Ironie anhand von den ausgewählten Merkmalen erkannt werden kann, bzw. ob potenzielle Ironie detektiert werden kann. Für die Umsetzung des Prototypen wird die Programmiersprache Python (Version 3.9.7) verwendet. Python bietet eine kompakte einfache Syntax, sowie die Einsatzmöglichkeit auf verschiedenen Plattformen. Die Programmiersprache wird häufig im Bereich des NLP und maschinellem Lernens eingesetzt [1, 11, 48, 50], da sie über eine Reihe von Frameworks verfügt, welche die Umsetzung spezifischer Problemstellungen unterstützen.

Die Entwicklung erfolgt über die web-basierte interaktive Umgebung JupyterLab[4]. Sie eignet sich besonders gut, da sie durch den modularen Aufbau des Programmcodes in Zellen, sehr flexibel in der Bearbeitung und Ausführung ist. Jede Zelle die Programmcode enthält wird einzeln ausgeführt und die Ergebnisdarstellung folgt unmittelbar im Anschluss an eine Zelle.

Verwendete Frameworks

Die Python-Bibliothek **pandas**[5] bietet hilfreiche Funktionen zur Verwaltung und Analyse von Datensätzen. Es können externe Daten aus CSV-Dateien, oder aus ähnlichen Datenformaten, eingelesen und in DataFrame Objekten gespeichert werden. Diese DataFrame Objekte können anschließend beliebig verändert werden. Zudem können sehr einfach statistische Auswertungen sowie mathematische Operationen auf diesen Objekten ausgeführt werden. In dieser Arbeit werden die Datensätze der drei Gruppen, die als CSV-Datei vorliegen, mittels pandas eingelesen und weiter verarbeitet. In ihnen sind die Transkripte, sowie deren Label zu *Ironie* und *Keine Ironie* enthalten.

Das **librosa**[6] Python-Paket bietet eine Vielzahl an Funktionen für die Audioanalyse. Mithilfe des Paketes werden die Audioauschnitte der Gespräche, die im WAV-Datei Format vorliegen, unterabgetastet geladen. Aus den ursprünglichen 42 kHz werden 16 kHz. Zusätzlich wird eine librosa Funktion genutzt, um die MFCCs der Audiodaten zu berechnen.

Im Bereich des NLP bietet das Framework **spaCy**[7] umfassende Funktionen zur Verarbeitung von Textdaten. Die Tokenisierung, sowie das Erstellen der POS-Tags und

[4] https://jupyter.org/
[5] https://pandas.pydata.org/
[6] https://doi.org/10.5281/zenodo.4792298
[7] https://spacy.io/

Lemma für die jeweiligen Token erfolgt mittels spaCy. Während einige Funktionen von spaCy unabhängig voneinander funktionieren, müssen für die Tokensierung und Lemmatisierung vorher trainierte Pipelines geladen werden. Eine trainierte Pipeline kann aus mehreren Komponenten bestehen, die ein auf gelabelte Daten trainiertes statistisches Modell verwenden. In dieser Arbeit wird die deutschsprachige Pipeline *de_core_news_sm* verwendet. Des weiteren werden die Eintragungen des externen angepassten Sentiment-Lexikon *sentiWS* über eine spaCy Pipeline-Erweiterung aufgerufen. Um Texte auf Vorkommen bestimmter Muster zu untersuchen, kann der regelbasierte Abgleich (*engl. rule-based matching*) von spaCy durchgeführt werden.

Die transkribierten Aussagen, welche im DataFrame Objekt von pandas vorliegen, werden mittels spaCy zu einem Doc Objekt umgewandelt. Das Doc Objekt enthält neben der Aussage somit die einzelnen Tokens der Aussagen mit ihren POS-Tags und Lemmas. Der regelbasierte Abgleich von spacy wird für die Analyse der Wortarten, sowie das Aufeinanderfolgen von Worarten genutzt.

Das Framework **NLTK**[8] bietet wie spacy im Bereich des NLP umfangreiche Funktionen. Es wurde nicht für die generell benötigten Funktionen genutzt, da spacy von der Performance besser abschneidet und beide solide Ergebnisse erbringen. Trotzdem wird in dieser Arbeit die Stoppwortliste des NLTK Frameworks genutzt, da sie kompakter ist und keine Verneinungen enthält, die Polaritäten von Aussagen umkehren können.

Mit der Python-Bibliothek **NumPy**[9] können sogenannte NumPy-Arrays erzeugt werden. Diese benötigen weniger Speicher als Python-Listen. Zusätzlich ist das Laufzeitverhältnis für Berechnungen besser. Die Bibliothek bietet eine Vielzahl nützlicher Funktionen die auf die NumPy-Arrays angewendet werden können. Für eine effiziente Weiterverarbeitung werden in dieser Arbeit die ermittelten Merkmalsvektoren in NumPy-Arrays abgelegt.

Das **scikit-learn**[10] Paket stellt eine große Anzahl an Funktionen für das maschinelle Lernen bereit. In dieser Arbeit wird sie für das Aufteilen des Datensatzes mittels Kreuzvalidierung sowie für das Berechnen eines Klassifizierungsmodells durch eine SVM genutzt.

Der Datensatz wird mittels der scikit-learn Klasse KFold in k Trainings- und Testdatensätze aufgeteilt. Dabei wird der Datensatz zuvor durchmischt, damit in jedem erstellten Segment Aussagen von jeder Gruppe auftauchen.

Für die Berechnung des Klassifizierungsmodells wird die Klasse SVM aus scikit-learn angewendet. Diese bekommt das numPy-Array der Merkmalsvektoren inklusive eines numPy-Arrays mit den dazugehörigen Labeln übergeben.

Zusätzlich bietet scikit-learn die Erstellung eines Berichts zur Auswertung des Klassifizierungsmodells (Precision, Recall, F1-Wert und Accuracy) und das Darstellen

[8] https://www.nltk.org/
[9] https://numpy.org/
[10] https://scikit-learn.org/stable/

sowie ausgeben von Konfusionsmatrizen.

Die Python-Bibliothek **imblearn**[11], die auf scikit-learn aufbaut, bietet Funktionen um unausgewogene Datensätze anzugleichen. Für das ausgleichen des Datensatzes dieser Arbeit, wurde der *RandomOverSampler* (zufälliger Überabtaster) angewendet. Dabei wird die Minderheitsklasse im Trainingsdatensatz an die Mehrheitsklasse angeglichen.

Seaborn[12] ist eine Python-Bibliothek, die für die Datenvisualisierung eingesetzt wird und auf **matplotlib**[13] basiert. Beide Bibliotheken dienen in dieser Arbeit zur Erstellung von Konfusionsmatrizen. Die matplotlib Bibliothek wird genutzt um die Darstellung der ausgegebenen Konfusionsmatrizen der scikit-learn Funktion anzupassen, wobei mit Hilfe der Seaborn Bibliothek aus einem Array eine eigene Darstellung der Konfusionsmatrix erstellt wird.

[11] https://imbalanced-learn.org/stable/
[12] https://seaborn.pydata.org/
[13] https://matplotlib.org/

Kapitel 5

Validierung

In diesem Kapitel wird der implementierte Prototyp und damit das erarbeitete Konzept validiert. Dafür werden Klassifizierungsmodelle für eine Auswahl an ausgewählten Merkmalen erstellt und die Performanz von diesen gemessen. Zur Messung der Performanz werden die in Abschnitt 4.5 vorgestellten Metriken genutzt.

5.1 Ziele der Validierung

Ziel der Validierung ist es herauszufinden, ob die als ironisch und nicht ironisch klassifizierten Aussagen des genutzten Datensatzes durch den erstellten Prototypen richtig detektiert werden. Entsprechend der in den Grundlagen aufgeführten Problematik der Identifikation von Ironie, soll anhand der Ergebnisse eine Aussage darüber getroffen werden können, in wie weit eine automatische Ironieerkennung anhand der ausgewählten Merkmale möglich ist. Zudem soll eingeschätzt werden, welchen Einfluss die einzelnen Merkmale im Verhältnis zu den anderen Merkmalen auf die Ergebnisse haben beziehungsweise ob überhaupt ein Einfluss besteht.

5.2 Durchführung

Für die Durchführung wird der Datensatz, der in Abschnitt 4.3 beschrieben wird, genutzt. Dieser enthält als ironisch klassifizierte Aussagen, die verbale und akustische Merkmale enthalten. Dabei wird nicht beachtet, dass die Klassifizierungsmodelle, die auf Basis von verbalen Merkmalen erstellt wurden, Schwierigkeiten bei ironisch klassifizierten Aussagen mit ausschließlich akustischen Merkmalen haben könnten und umgekehrt. Somit kommen in 86% der als ironisch klassifizierten Aussagen verbale Merkmale vor und nur in 51% akustische Merkmale. Dies ist notwendig, damit dieselben Training- und Testdaten für jedes Klassifizierungsmodell genutzt werden und ein darauffolgender Vergleich möglich bleibt.

Zu Anfang wird mittels der Kreuzvalidierung ein Set aus k Trainings- und Testdatensätzen erstellt. Damit in jedem Trainings- und Testdatensatz Aussagen aus jeder

Gruppe enthalten sind, wird der Datensatz zuvor durchmischt. Für k wird 5 eingesetzt. Daraus resultiert, dass jeder Testdatensatz aus 20% des gesamten Datensatzes besteht. Jeder Testdatensatz enthält damit im Durchschnitt 230 Aussagen, von denen im Schnitt 29 Aussagen als ironisch klassifiziert sind. Wird k zu groß gewählt, befinden sich zu wenige ironische Aussagen im Testdatensatz, um aussagekräftige Informationen aus der Validierung zu ziehen. Wird k zu klein gewählt, reichen die Trainingsdaten zum Erlernen eines repräsentativen Modells nicht aus. Es sollen insgesamt fünf Kombinationen aus Merkmalsvektoren, die in die Erstellung eines Klassifizierungsmodells fließen, verglichen und bewertet werden. Dabei wurden die Merkmale zusammengefasst, für die dieselben oder sehr ähnliche Techniken zur Merkmalsextraktion angewendet wurden. Es handelt sich um die folgenden Zusammensetzungen:

1. *Akustische Merkmale* (14 Merkmalsvektoren)
   Die ersten 14 MFCCs der Audiomitschnitte einer Aussage

2. *Stimmungsmerkmale* (3 Merkmalsvektoren)
   Polarität und Anteil positiver sowie negativer Wörter einer Aussage

3. *Merkmale aus der Worthäufigkeit* (2 Merkmalsvektoren)
   Ironie und *Keine Ironie* bemessen an Ironie-/ Keine Ironie-Lexika

4. *Merkmalen aus den Wortarten* (3 Merkmalsvektoren)
   Anteile von Adjektiven, doppelter Adjektive und Personalpronomen in einer Aussage

5. *Gesamtheit der Merkmale* (22 Merkmalsvektoren)

Das Vorgehen zur Extraktion der Merkmale wird in Abschnitt 4.4.5 beschrieben. Die folgenden Schritte der Kreuzvalidierung werden für jede oben aufgeführte Kombination aus Merkmalsvektoren durchgeführt und mittels der Abbildung 5.1 verdeutlicht. Es werden fünf Klassifizierungsmodelle aus jeweils vier Teildatensätze mittels SVM erstellt. Für jedes der fünf Klassifizierungsmodelle wird eine Vorhersage für die Aussagen des jeweiligen Testdatensatzes berechnet. Aus dieser Vorhersage und der wahren Klassifikation werden die Kennzahlen Accuracy, Precision, Recall und F1-Score bestimmt. Wie es für die Kreuzvalidierung üblich ist, wird das arithmetische Mittel für die einzelnen Kennzahlen gebildet. Für jedes Modell bestehen somit sieben Kennzahlen, da Precision, Recall und F1-Score für beide Klassen (*Ironie* und *Keine Ironie*) gebildet werden und der Accuracy-Wert für das gesamte Modell einmal bestimmt wird.

Zusätzlich werden Konfusionsmatrizen für die Klassifizierungsmodelle gebildet. Die Konfusionsmatrizen von den Klassifizierungsmodellen mit dem höchsten und niedrigsten F1-Score, sowie eine Konfusionsmatrix, die sich aus dem Mittel aller fünf entstanden Konfusionsmatrizen ergibt, werden jeweils dargestellt. Es wird sich am F1-Score orientiert, da dieser durch das hohe Klassenungleichgewicht eine größere Aussagekraft hat als der Accuracy-Wert. Der Accuracy-Wert kann sehr hoch sein, obwohl die Minderheitenklasse nie detektiert wird. Der F1-Score bildet das harmonische Mittel zwischen Precision und Recall. Dieser Wert ist besonders hoch, wenn viele ironischen Aussagen richtig detektiert

und zugleich wenige nicht ironische Aussagen fälschlicherweise als ironisch detektiert wurden. Die Konfusionsmatrizen werden nach der Anzahl der Elemente in jeder Klasse normalisiert. Im Falle eines Klassenungleichgewichts, wie es hier besteht, bietet diese Art der Normalisierung eine gute visuelle Interpretation der falsch klassifizierten Klasse. Die Recall-Wert der Klasse *Ironie* ist im Quadranten rechts unten und der Recall-Wert der Klasse *Keine Ironie* ist im Quadranten links oben zu finden.

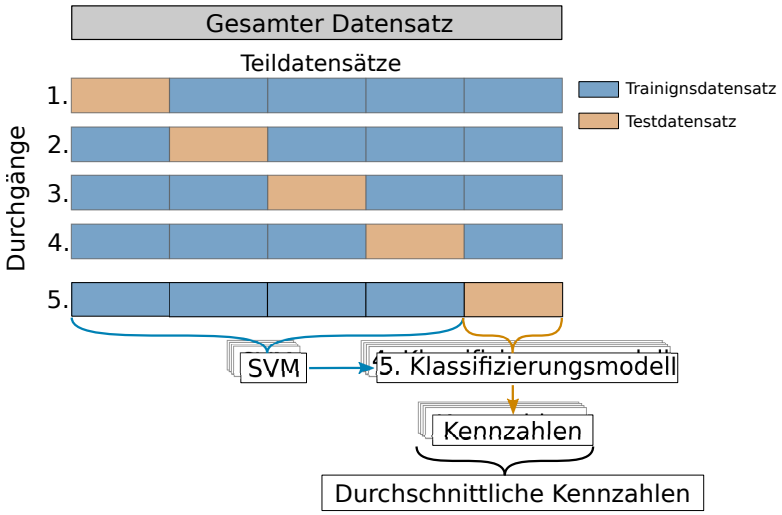

Abbildung 5.1: Prozess zur Berechnung der durchschnittlichen Kennzahlen einer Kombination von Merkmalen mittels Kreuzvalidierung

5.3 Ergebnisse

In den folgenden Teilabschnitten werden die Ergebnisse für die im vorherigen Abschnitt vorgestellten fünf unterschiedlichen Kombinationen aus Merkmalen dargestellt und bewertet. Für jede Kombination aus Merkmalen wird eine Tabelle mit den Kennzahlen und drei Konfusionsmatrizen angegeben. Bei den Kennzahlen handelt es sich um die Durchschnittswerte der durchgeführten Kreuzvalidierung. Es werden dabei beide Klassen (*Ironie* und *Keine Ironie*) betrachtet. Bei den drei Konfusionsmatrizen handelt es sich um die normalisierten Matrizen der Modelle mit dem höchsten und niedrigsten F1-Score der Ironieklasse sowie einer Konfusionsmatrix, die sich aus dem Mittel aller fünf entstanden Konfusionsmatrizen ergibt. Zum Ende werden alle Ergebnisse in der Tabelle 5.6 zum direkten Vergleich zusammengefasst.

1. Akustische Merkmale

Die *akustischen Merkmale* werden aus den Mittelwerten für die ersten 14 MFFCs eines Audiomitschnittes erzeugt. Sie sind eine kompakte Darstellung des Frequenzspektrums und enthalten wichtige Merkmale zur Spracherkennung.

Tabelle 5.1 zeigt die durchschnittlichen Kennzahlen aus der Kreuzvalidierung. Die fünf Klassifizierungsmodelle die innerhalb der Kreuzvalidierung entstanden sind, sind ausschließlich auf Grundlage der 14 MFCCs erzeugt wurden. Da sich der Datensatz aus 1153 Aussagen nicht in fünf genau gleich große Teildatensätze einteilen lässt, besteht ein Teildatensatz entweder aus 230 oder 231 Aussagen. Im Durchschnitt kommen so 29,2 Aussagen der Klasse *Ironie* 29,2 und 201,4 Aussagen der Klasse *Keine Ironie* in einem Teildatensatz vor. Aus der Anzahl der Elemente in den beiden Klassen lässt sich schließen, dass der Ironieanteil des Testdatensatzes knapp 12,5% beträgt. Ein Klassifizierer, der jede Aussage als *Keine Ironie* detektiert, würde somit einen Accuracy-Wert von 87,5% erreichen, wogegen Precision, Recall und F1-Score der Klasse *Ironie* 0% betragen würden. Somit ist der hier erreichte Accuracy-Wert von 65% zwar geringer, Precision, Recall und F1-Score sind dafür größer als 0.

Accuracy	Klasse	Precision	Recall	F1-Score	Anzahl
0,65	Ironie	0,22	0,68	0,33	29,2
	Keine Ironie	0,94	0,65	0,76	201,4

Tabelle 5.1: Kennzahlen für Akustische Merkmale

Ein Recall-Wert von 68% und ein Precision-Wert von 22% der Klasse *Ironie*, weisen darauf hin, dass neben vielen ironisch klassifizierten Aussagen, auch viele nicht ironisch klassifizierte Aussagen als ironisch klassifiziert wurden. Die Koeffizienzmatrizen der Abbildung 5.2 verdeutlichen dies.

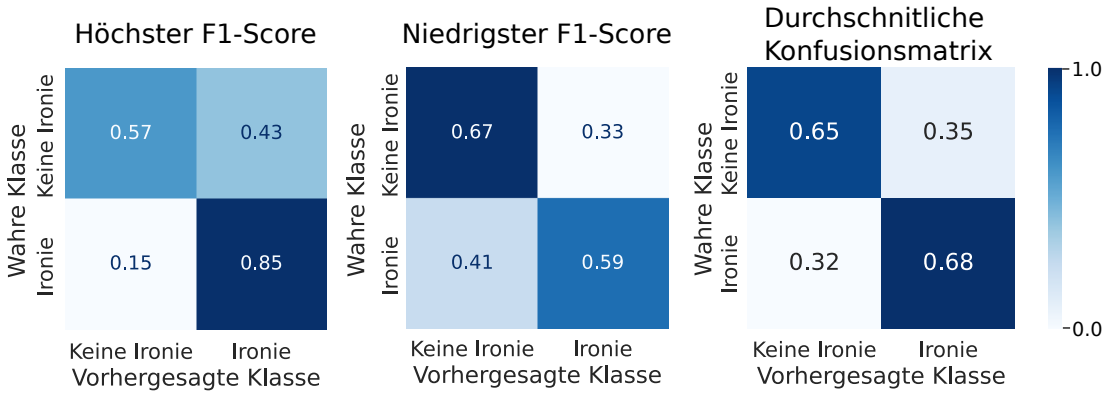

Abbildung 5.2: Koeffizienzmatrizen für akustische Merkmale

Die blaue Färbung der Quadranten hebt den angegebenen prozentualen Anteil grafisch hervor. Je dunkler ein Feld, desto größer ist der prozentuale Anteil. Im Durchschnitt werden 35% der nicht ironischen Aussagen als Ironie klassifiziert. In der Konfusionsmatrix des höchsten F1-Scores sind es sogar 43% der Aussagen, die fälschlicherweise der Klasse *Ironie* zugeordnet werden. Dadurch, dass dieses Modell gleichzeitig 85% ironischer Aussagen richtig detektiert, kann überhaupt erst der

höchste F1-Score erreicht werden. Im Gegensatz dazu hat die Konfusionsmatrix des niedrigsten F1-Scores sogar 10% weniger Aussagen fälschlicherweise als Ironie detektiert (33%), aber dafür auch 26% weniger ironisch Aussagen gefunden, welches zum niedrigsten F1-Score führt. Diese Varianz zwischen den Konfusionsmatrizen des höchsten und niedrigsten F1-Scores in den falsch klassifizierten Klassen wird in der durchschnittlichen Konfusionsmatrix ausgeglichen, wobei die Werte der durchschnittlichen Konfusionsmatrix näher an den Werten der Konfusionsmatrix mit dem niedrigsten F1-Score liegen. Aus diesen Ergebnissen ergibt sich folgende Erkenntnis:

Erkenntnis:
Beim Vergleich der ironischen und nicht ironischen Aussagen ergeben sich bezüglich der akustischen Merkmale sichtbare Unterschiede. Dabei sind die Unterschiede nicht immer eindeutig abgrenzbar, sodass dies noch zu vielen Fehlklassifikationen führt.

Weiterhin ist auffällig, dass nur etwa die Hälfte der ironisch klassifizierten Aussagen ein akustisches Merkmal enthalten. Trotzdem wird durchschnittlich mehr als die Hälfte der ironischen Merkmale detektiert. Daraus ergibt sich folgende Erkenntnis:

Erkenntnis:
Entweder wurden in der manuellen Annotation akustische Merkmale in den ironischen Aussagen übersehen, oder es existieren akustische Merkmale, die nicht in der manuellen Annotation beachtet wurden.

2. Stimmungsmerkmale

Bei den drei ausgewerteten *Stimmungsmerkmalen* handelt es sich um die Polarität einer Aussage und den Anteil an positiven sowie negativen Wörtern in einer Aussage.

Die durchschnittlichen Kennzahlen aus der Kreuzvalidierung für die Klassifizierungsmodelle ausschließlich auf Grundlage der drei *Stimmungsmerkmalen* finden sich in Tabelle 5.2. Auffällig ist, dass alle Kennzahlen über denen der ersten Auswertung mit den *akustischen Merkmalen* liegen. Dies deutet darauf hin, dass die Stimmung einer Aussage einen größeren Einfluss auf die Ironiedetektion hat, als es die *akustischen Merkmale* haben. Dabei muss man jedoch beachtet, dass 85% der ironisch klassifizierten Aussagen ein verbales Merkmal und nur 51% ein akustisches Merkmal beinhalten.

Accuracy	Klasse	Precision	Recall	F1-Score	Anzahl
0,71	Ironie	0,26	0,69	0,37	29,2
	Keine Ironie	0,94	0,71	0,81	201,4

Tabelle 5.2: Kennzahlen für Stimmnugsmerkmale

Die Abbildung 5.3 zeigt die drei Konfusionsmatrizen zur Kreuzvalidierung. In der Konfusionsmatrix des niedrigsten und höchsten F1-Scores liegen die größten Unterschiede

in der Anzahl der richtig detektierten Ironie, wobei die durchschnittlichen Werte etwas näher an der Konfusionsmatrix des höchsten F1-Scores liegen. Somit werden, je nachdem welche Trainings- bzw. Testdaten verwendet werden, mehr oder weniger Aussagen übersehen. Der Grund dafür könnte darin liegen, dass ironischen Aussagen nicht immer von übermittelten Stimmungen abhängen und dass einige Aussagen keine *Stimmungsmerkmale* enthalten.

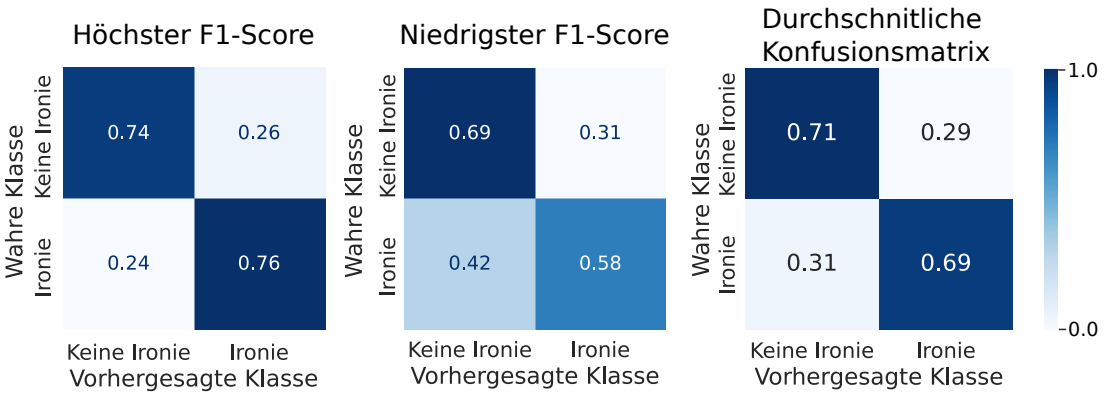

Abbildung 5.3: Koeffizienzmatrizen für Stimmungsmerkmale

Insgesamt kann aus den Ergebnissen die folgende Erkenntnis gezogen werden:

Erkenntnis:
Anhand der Stimmungsmerkmale ist es möglich, einen Großteil der ironischen Aussagen zu detektieren, jedoch kommt es auch hier zu einigen Fehlklassifikationen.

3. Merkmale aus der Worthäufigkeit

Der Grad von *Ironie* und der Grad von *Keiner Ironie* sind zwei Merkmale, die sich aus der Worthäufigkeit ergeben. Aus normalisierten Aussagen des Trainingsdatensatzes wird die Auftrittshäufigkeit eines Wortes in einem Lexikon gespeichert. Dabei hängt es davon ab, ob es sich um eine ironische oder nicht ironische Aussage handelt, in welchem Lexikon die Anzahl des auftretenden Wortes erhöht wird.

Die Tabelle 5.3 zeigt die durchschnittlichen Kennzahlen aus der Kreuzvalidierung für die Klassifizierungsmodelle ausschließlich auf Grundlage der des Grads von *Ironie* und *Keiner Ironie*.

Accuracy	Klasse	Precision	Recall	F1-Score	Anzahl
0,86	Ironie	0,45	0,54	0,48	29,2
	Keine Ironie	0,93	0,9	0,92	201,4

Tabelle 5.3: Kennzahlen für Merkmale aus der Worthäufigkeit

Auffällig ist ein Precision-Wert von 45%. Damit liegt dieser 20% höher als bei den beiden zuvor betrachteten Kreuzvalidierungen (22% und 26%). Zudem liegen alle Kennzahlen für die Klasse *Keine Ironie* über 90%. Dies hat zufolge, dass der durchschnittliche Accuracy-Wert bei 86% liegt. Der Recall-Wert liegt mit 54% jedoch unter den vorherigen Ergebnisse. Diese Werte spiegeln sich auch in den drei Konfusionsmatrizen in Abbildung 5.4 wieder. In allen drei Konfusionsmatrizen ergibt sich ein sehr geringer Wert für die falsche Klassifikation von *Keiner Ironie* (kleiner gleich 10%). Da es sich hierbei um die Mehrheitsklasse handelt, lässt sich daraus der hohe Precision-Wert, sowie der hohe Accuracy-Wert erschließen. Durchschnittlich wurde die Hälfte der wahren Ironie detektiert, im Falle des niedrigsten F1-Scores jedoch nur 36%.

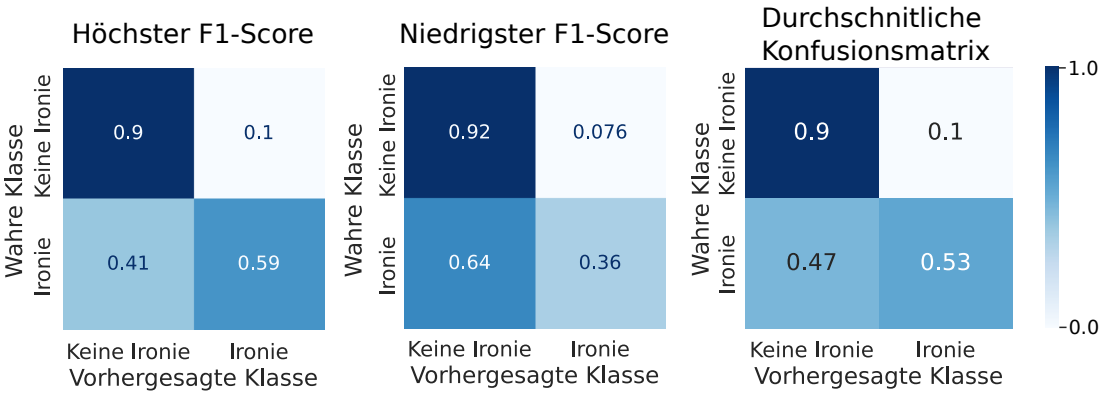

Abbildung 5.4: Koeffizienzmatrizen für Merkmale aus der Worthäufigkeit

Daraus lässst sich folgendes schließen:

Erkenntnis:
Einige Wörter oder Wortkombinationen geben einen starken Hinweis auf Ironie, sodass es bei diesen nur selten zu Fehlklassifikationen kommt. Viele ironischen Aussagen beinhalten allerdings nicht diese Wörter oder Wortkombinationen, sodass gleichzeitig viel Ironie übersehen wird.

4. Merkmale aus den Wortarten

Die anhand der Wortarten herausgearbeiteten Merkmale beziehen sich auf das Verhältnis der Anzahl von Adjektiven oder Adverbien, dem Verhältnis von hintereinander stehenden Adjektiven oder Adverbien und dem Verhältnis von Personalpronomen, in Bezug zu allen Wörtern einer Aussage.

Die durchschnittlichen Kennzahlen für die gebildeten Klassifikationsmodelle anhand dieser drei Merkmale sind in Tabelle 5.4 dargestellt. Der Precision-Wert der Ironie-Klasse ist in diesem Fall mit 17% sehr gering. Die Recall-Werte beider Klassen liegen sogar bei beiden Klassen nur bei 58%, wodurch der zudem recht niedrige Accuracy-Wert von 58% hinzukommt.

Accuracy	Klasse	Precision	Recall	F1-Score	Anzahl
0,58	Ironie	0,17	0,58	0,26	29,2
	Keine Ironie	0,9	0,58	0,7	201,4

Tabelle 5.4: Kennzahlen für Merkmale aus den Wortarten

Dies deutet auf eine sehr hohe falsch Klassifizierung hin, die in den Konfusionsmatrizen aus Abbildung 5.5 bestätigt wird. In der Konfusionsmatrix mit dem niedrigsten F1-Score werden fast die Hälfte der nicht ironischen Aussagen (44%) als ironisch klassifiziert. Trotzdem ist der Anteil an richtig klassifizierten ironischen Aussagen nur bei 45%. Allgemein sind dies sehr niedrige Kennzahlen. Trotzdem liegt der Durchschnitt der richtig klassifizierten Aussagen über 50% und in der Konfusionsmatrix mit dem höchsten F1-Score setzen sich die 65% und 67% richtig klassifizierten Aussagen von den falsch klassifizierten Aussagen ab.

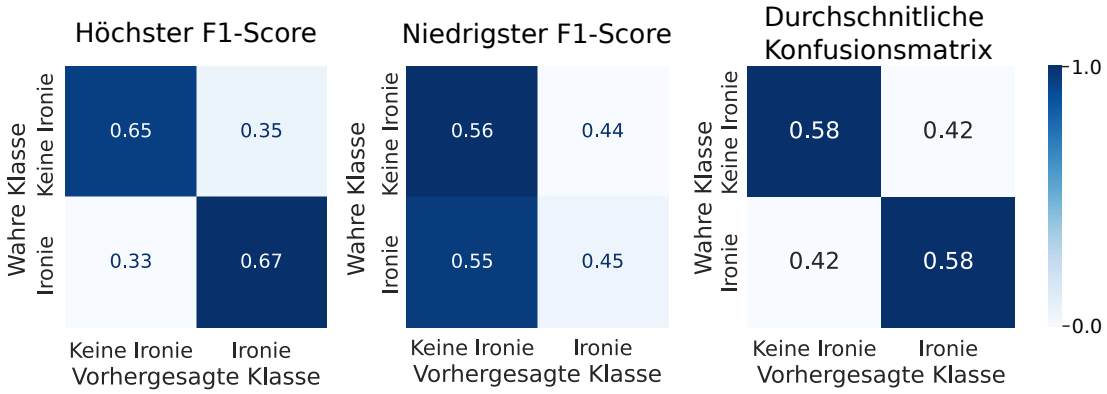

Abbildung 5.5: Koeffizienzmatrizen für Merkmale aus der Wortart

Daraus lässt sich folgende Erkenntnis ziehen:

Erkenntnis:
Die *Merkmale aus den Wortarten* haben nur eine sehr geringe Aussagekraft über den Ironiegehalt einer Aussage.

5. Gesamtheit der genutzten Merkmale

Zuletzt wird ein Klassifizierungsmodell anhand aller eben vorgestellten Merkmale gebildet. Die Ergebnisse dazu finden sich in Tabelle 5.5 wieder.

Mit einem durchschnittlichen Recall-Wert von 75% und einem Precision-Wert von 36% der Klasse *Ironie*, ergeben sich solide Ergebnisse für die gemeinsame Klassifizierung. Der Precision-Wert der Klasse *Keine Ironie* erlangt mit 96% ebenfalls Höchstwerte. Verdeutlicht wird das Ergebnis in den drei Konfusionsmatrizen in Abbildung 5.6. Die Anteile der richtig und falsch klassifizierten Klasse *Keine Ironie* sind in allen

Accuracy	Klasse	Precision	Recall	F1-Score	Anzahl
0,8	Ironie	0,36	0,75	0,48	29,2
	Keine Ironie	0,96	0,81	0,88	201,4

Tabelle 5.5: Kennzahlen für die Gesamtheit der Merkmale

drei Konfusionsmatrizen sehr stabil, dagegen variieren die Anteile der Klasse *Ironie* mehr. Die durchschnittlichen Werte sind zusammen betrachtet höher als die Werte der Klassifizierungsmodelle, in denen nur einzelne Merkmale genutzt wurden. Daraus kann geschlossen werden, dass sich die Merkmale ergänzen. Es werden jedoch immer noch durchschnittlich 25% der ironischen Aussagen nicht detektiert, sodass sich die Klassifizierungen der einzelnen Merkmale gleichzeitig auch stark überschneiden müssen.

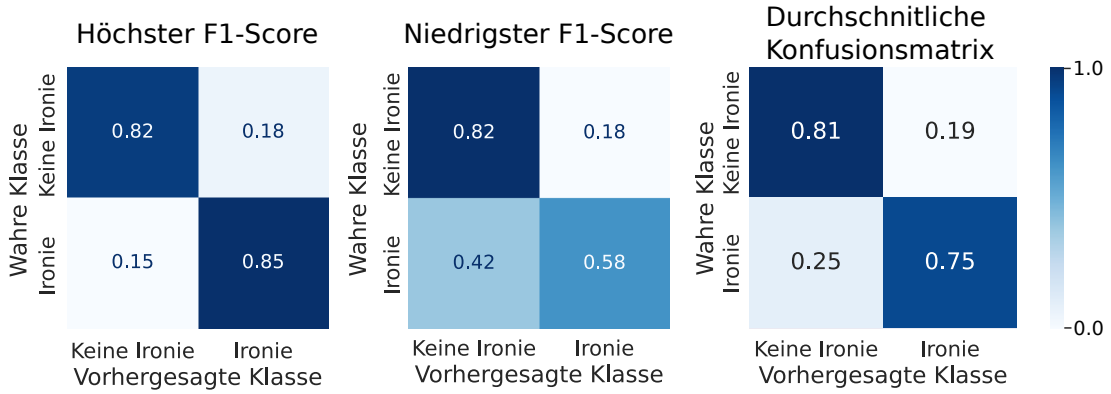

Abbildung 5.6: Koeffizienzmatrizen für die Gesamtheit der Merkmale

Insgesamt ergibt sich folgende Erkenntnis:

Erkenntnis:
Bei einer Klassifikation, in der alle Merkmale eingebunden werden, ergänzen sich die Merkmale und erzielen so bessere Ergebnisse.

5.3.1 Zusammenfassung der Ergebnisse

Um einen besseren Überblick über die Ergebnisse zu bekommen, werden die durchschnittlichen Ergebnisse der einzelnen Kreuzvalidierungen in Tabelle 5.6 zusammengefasst. Bei den Kennzahlen Precision, Recall und F1-Score handelt es sich um Werte für die Klasse *Ironie*. Die *Merkmale der Häufigkeit* haben von den verschiedenen Merkmalen die höchsten Kennzahlen-Werte, bis auf den Recall-Wert, der sogar der niedrigste von allen ist. Die niedrigsten Kennzahlen weisen die *Merkmale der Wortarten* auf, was auf einen sehr geringen Einfluss schließen lässt. Die Kennzahlen der akustischen Merkmale sind zwischen 1% bis 6% niedriger als die Kennzahlen der *Stimmungsmerkmale*. Die *Stimmungsmerkmale* sind zwar präziser in der Detektion von Ironie, die ähnlichen Werte

könnten jedoch darauf hinweisen, dass sie zusammenhängende Merkmale analysieren. Die Kennzahlen der *Merkmalen der Worthäufigkeit*, die einen 19% höheren Precision-Wert und einen 15% niedrigerem Recall-Wert, als die Kennzahlen der *Stimmungsmerkmale* haben, deuten auf einen geringeren Zusammenhang zwischen den analysierten Merkmalen hin.

	Accuracy	Precision	Recall	F1-Score
Akustische Merkmale	0,65	0,22	0,68	0,33
Stimmungsmerkmale	0,71	0,26	0,69	0,37
Merkmale der Worthäufigkeit	**0,86**	**0,45**	*0,54*	**0,48**
Merkmale der Wortarten	*0,58*	*0,17*	0,58	*0,26*
Gesamtheit der Merkmale	0,8	0,36	**0,75**	**0,48**

Fettgedruckte Zahlen geben die höchsten und kursivgedruckte Zahlen die niedrigsten Werte einer Kennzahl an.

Tabelle 5.6: Zusammengefasste Ergebnisse aus den fünf Kreuzvalidierungen.

Die Klassifizierung mit allen Merkmalen schneidet im Allgemeinen am besten ab, da sie mit einem Recall-Wert von 75% und einem F1-Score von 48% die höchsten Kennzahlen in diesem Bereich aufweisen. Es zeigt sich, dass es vorteilhaft ist, aus allen Merkmalen eine Klassifizierung zu generieren, vor allem wenn der Schwerpunkt darauf liegt, möglichst viele ironische Aussagen zu detektieren (hoher Recall-Wert), als nur tatsächlich ironische Aussagen als ironisch zu detektieren (hoher Precision-Wert). Wie die Ergebnisse weiter interpretiert und eingeordnet werden können, wird in Kapitel 6 behandelt.

5.3.2 Beispiele aus einer Klassifikation

Um einen Überblick darüber zu bekommen, welche ironischen Aussagen als Ironie detektiert werden beziehungsweise welche übersehen werden, werden in diesem Abschnitt einige Beispiele aufgelistet. Für die entsprechenden Vorhersagen wurde ein Klassifizierungsmodell genutzt, welches mit allen Merkmalen trainiert wurde. Es handelt sich dabei um ein Modell aus der Kreuzvalidierung, welches nicht den besten und nicht den schlechtesten F1-Score erreicht hat. Die Beispiele bilden nur eine kleine Menge der Klassifikationen ab und zeigen nicht unbedingt alle Merkmale, an denen sich das Modell orientiert. Weiterhin ist zu beachten, dass das Modell die Klassifikation auch anhand der akustischen Merkmale durchgeführt hat, die hier nicht mit bereitgestellt werden können.

Das Klassifizierungsmodell hat beispielsweise die folgenden Aussagen richtig als Ironie detektiert:

„Ja natürlich, du bist ja ein Profi, oder nicht?“
„Das ist glaube ich das Wichtigste in diesem Moment.“
„Der Code ist super.“
„Ja, hat sich jemand richtig Mühe gegeben.“

Hier scheinen die Aussagen im wörtlichen Sinn mit Wörtern wie, *Profi*, *Wichtigste* oder *super* vor allem eine positive Polarität zu beinhalten.

Die folgenden Aussagen wurden fälschlicherweise als Ironie markiert:

„Sehr gute Idee.“
„Ja schaut doch gut aus.“
„Nein es sieht nicht schön aus, es sieht scheiße aus.“
„Ich glaube Python wird gar nicht mehr unterstützt.“

Auch in diesen Beispielen ist eine positive Polarität erkennbar. Die dritte Aussage wurde wahrscheinlich fälschlicherweise als positiv erkannt, da sich die Polarität der Aussage durch das *nicht* umkehrt, die Polarität des Nebensatzes jedoch nicht umgekehrt werden darf. Zusätzlich scheint die letzte sehr neutrale sachliche Aussage entweder durch akustische Hinweise, oder das Schlüsselwort *glaube* als Ironie detektiert worden zu sein. Dieses ist in der Wortwolke, welche in der Abbildung 4.4 dargestellt ist, relativ groß geschrieben.

Das Klassifizierungsmodell hat beispielsweise folgende Aussagen übersehen und als *Keine Ironie* klassifiziert:

„Sieht sehr falsch aus.“
„100 passt auf jeden Fall.“
„Und dann noch der Rechtschreibfehler in den Variablen, ouoouoouou.“

Diese Aussagen weisen eher eine negative, oder neutrale Polarität auf. Hinweise auf Ironie sind nur schwer zu erkennen.

Insgesamt kann aus diesen Beispielen geschlossen werden, dass vor allem Aussagen mit besonders positiven Polaritäten als Ironie detektiert werden. Zusätzlich können bestimmte Schlüsselwörter zur Ironiedetektion hinzukommen. Akustische Merkmale werden hier nicht weiter ausgewertet. Es können weitere Merkmale vorhanden sein, die in diesem Zusammenhang nicht klar erkennbar sind.

5.4 Zusätzliche Erkenntnisse

Für die Erstellung der *Stimmungsmerkmale* wird das Sentiment-Lexikon SentiWS genutzt. Da die verwendete Pipeline von spacy zur Integration der Polariätsbestimmung von Tokens nicht nur ein passendes Wort im Lexikon sucht, sondern auch ein passendes Wort mit derselben Wortart des Tokens, wurden weitere Anpassungen am Lexikon vorgenommen. Adverbial genutzte Adjektive werden in spacy mit dem POS-Tag Adverb vermerkt, sodass diese, wenn sie nur als Adjektiv im Lexikon vermerkt sind, keine Polarität zugewiesen bekommen. Damit auch adverbiale Adjektive eine Polarität zugewiesen bekommen, wird für jedes aufgelistete Adjektiv im Lexikon ein dazu passendes Adverb hinzugefügt. Der Effekt dieses Vorgehens soll mit einer weiteren Kreuzvalidierung, in der die *Stimmungsmerkmale* ohne die Veränderung des Lexikons bestimmt werden, gezeigt werden.

Die Ergebnisse werden in Tabelle 5.7 und Abbildung 5.7 gezeigt. Der Recall-Wert steigt mit der Veränderung des Lexikons von 33% auf 69%, was darauf hinweist, dass zuvor viele Aussagen eine Polarität von null hatten, da die Adjektive als Adverbien markiert waren, und damit als *Keine Ironie* markiert wurden. In den Konfusionsmatrizen ist deutlich zu erkennen, dass durch diese Veränderung sehr viel mehr Aussagen als *Ironie* klassifiziert wurden, sodass auch die Anzahl an falsch vorhergesagter Ironie steigt. Dies hat zufolge, dass der Precision-Wert nur um 1% ansteigt. Der Precision-Wert der Klasse *Keine Ironie* stiegt auch, jedoch sinkt der Recall-Wert sogar von 86% auf 71% ab. Dies liegt vor allem daran das allgemein mehr Aussagen der Klasse *Ionie* zugewiesen werden und damit auch mehr nicht ironische Aussagen. Durch die Anpassungen des Lexikons erhalten mehr Aussagen eine Polarität ungleich null. Dadurch wird zwischen den Aussagen besser differenziert.

	Accuracy	Klasse	Precision	Recall	F1-Score
Lexika **ohne** Adjektiven als Adverbien deklariert	0,79	Ironie	0,25	0,33	0,28
		Keine Ironie	0,9	0,86	0,88
Lexika **mit** Adjektiven als Adverbien deklariert	0,72	Ironie	0,26	0,69	0,37
		Keine Ironie	0,94	0,71	0,81

Tabelle 5.7: Kennzahlen für Stimmungsmerkmale welche auf einem angepassten, sowie unangepassten Sentiment-Lexikon basieren

Folgende Erkenntnis lässt sich aus den Ergebnissen ziehen:

Erkenntnis:
Durch die Anpassung im sentiWs Lexikon, das Adjektive zusätzlich als Adverbien deklariert werden, können durchschnittlich 36% mehr ironische Aussagen detektiert werden.

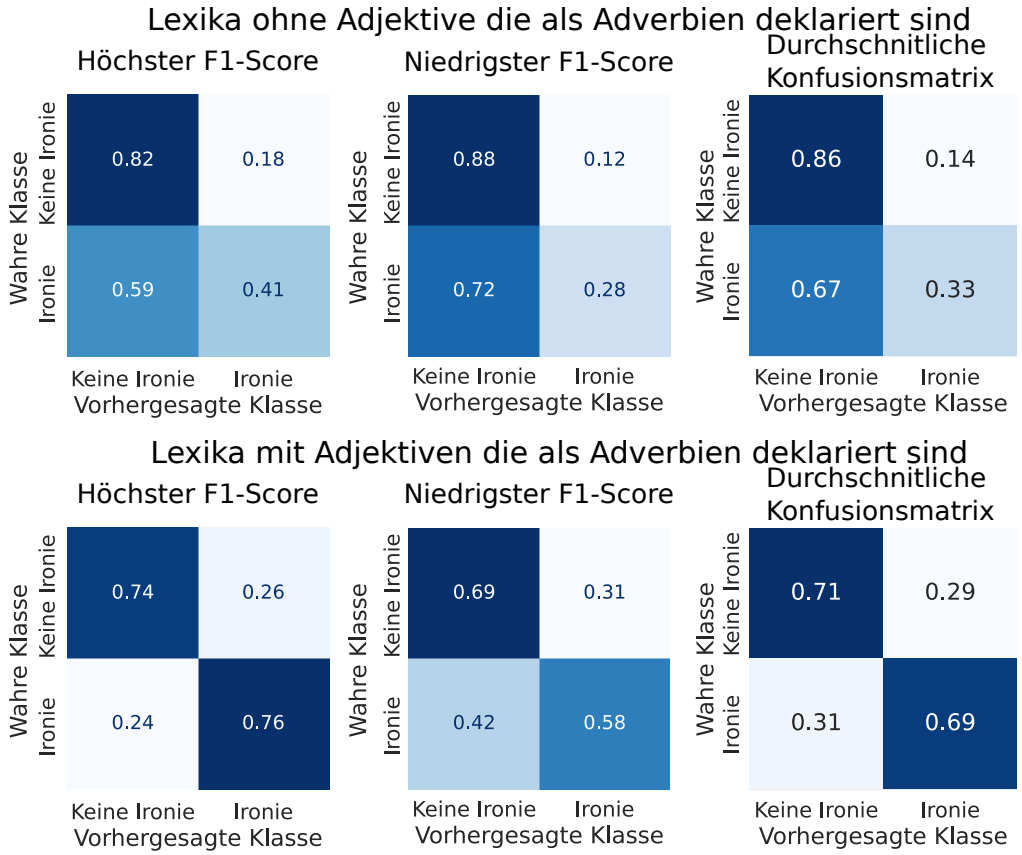

Abbildung 5.7: Koeffizienzmatrizen für Stimmungsmerkmalen welche auf einem angepassten, sowie unangepassten Sentiment-Lexikon basieren

Kapitel 6

Diskussion

Diese Arbeit untersucht den Einsatz computergestützter Verfahren zur Identifikation von Ironie im Bereich des Software Engineerings. In diesem Kapitel werden die Ergebnisse der Validierung des entwickelten Prototyps, welche im Kapitel 5 beschrieben werden, im Hinblick auf die Anwendbarkeit interpretiert und Aspekte, welche die Validität der Ergebnisse gefährden könnten, aufgezeigt. Zudem werden die möglichen Auswirkungen auf weitere Forschungen sowie weitere Anwendungsfälle behandelt.

6.1 Interpretation

Die bisherigen Tools zur Sentiment-Analyse im Software Engineering haben Schwierigkeiten, die richtige Stimmung bei ironisch gemeinten Aussagen zu detektieren [35]. Durch Ironie kann sich die Wahrnehmung einer Aussage komplett ändern, jedoch besteht eine große Herausforderung darin, Ironie mit computergestützten Verfahren zu erfassen. Einige Arbeiten gehen davon aus, dass sich die Vorhersagen eines Tools zur Sentiment-Analyse mithilfe einer Ironieerkennung verbessern können [17]. Somit wurde in dieser Arbeit ein Prototyp entwickelt, der Ironie detektiert.

Um ein Modell zur Identifikation von Ironie im Bereich des Software Engineerings zu entwickeln, wurde zunächst ein Datensatz erstellt, der ironische und nicht ironische Aussagen aus Gesprächen des Pair Programming enthält. Die Aussagen wurden jeweils von drei Ratern annotiert. Ein Fleiss-Kappa von 60% bis 68% zeigt eine moderate Übereinstimmung der Klassifizierungsergebnisse und lässt auf eine hohe Subjektivität in der Bewertung von Ironie schließen. Nachträglich wurden die ironischen Aussagen dahingehend klassifiziert, ob sie verbale oder akustische Merkmale aufweisen. Im Detail wird das Vorgehen in Kapitel 4 beschrieben. Es ist wichtig, zu beachten, dass aufgrund der nachträglichen Einordnung 37% der ironischen Aussagen im weiteren Verlauf nicht weiter betrachtet wurden, da sie keine verbalen oder akustischen Merkmale aufweisen. Sie weisen häufig nur kontextabhängige Merkmale auf, welche zum Verstehen Hintergrundwissen benötigt und daher gesondert betrachtet werden müssen.

Für die Erstellung eines Klassifizierungsmodells wurden 22 Merkmale aus den Aussagen extrahiert. Um die Aussagekraft der Merkmale zu vergleichen, wurden

insgesamt fünf Kombinationen aus Merkmalen untersucht. Bei der Validierung liegen die durchschnittlichen Recall-Werte beider Klassen für alle Kombinationen über 50%. Das zeigt, dass anhand jeder Kombination aus Merkmalen Unterschiede zwischen ironischen und nicht ironischen Aussagen gefunden werden konnten. Unter den Ergebnissen gibt es jedoch noch erhebliche Unterschiede. So zeigt die Validierung, dass die *Merkmale der Wortarten* eine geringere Aussagekraft über den Ironiegehalt einer Aussage haben, als die anderen Merkmale. Daraus lässt sich schließen, dass der Gebrauch von Adjektiven, Pronomen oder aufeinanderfolgenden Adjektiven sich in ironischen Aussagen nicht besonders zu anderen Aussagen abhebt.

Der entwickelte Prototyp benötigt einen Audiomitschnitt sowie dessen Transkription. Da die Validierung gezeigt hat, dass auch mit Kombinationen aus Merkmalen, die nur verbale Merkmale untersuchen, ein relativ hoher Anteil der ironischen Aussagen gefunden werden konnte, können in Zukunft auch Datensätze ohne Audiomitschnitte mit dem Prototyp untersucht werden. Zugleich kann das Potenzial der akustischen Analyse weiter ausgebaut werden, da schon anhand der Mittelwerte der ersten 14 MFCCs ein recht großer Unterschied zwischen den ironischen und nicht ironischen Aussagen gefunden wurde.

Aus der Validierung ist deutlich zu erkennen, dass Klassifizierungsmodelle, die mit den gesamten 22 Merkmalen trainiert werden, bessere Ergebnisse erzielen als diejenigen, die mit weniger Merkmalen trainiert werden. Dadurch, dass die Merkmale sich ergänzen, können durchschnittlich 75% der ironischen Aussagen detektiert werden. Wie im Grundlagenkapitel 2 beschrieben, können alle ironischen Signale in ironischen und auch nicht ironischen Aussagen verwendet werden. Dadurch entsteht eine Mehrdeutigkeit, welche die Ironie erst ausmacht [15]. Diese Mehrdeutigkeit spiegelt sich auch in den Validierungsergebnissen wider. Der Precision-Wert der Klasse Ironie liegt für die Klassifizierung mit allen Merkmalen bei 36%. Das zeigt, dass es noch viele Aussagen gibt, die fälschlicherweise als Ironie detektiert werden.

Die hohe Subjektivität von Ironie und die hohe Anzahl an fälschlich klassifizierter Ironie zeigt, dass es trotz der zahlreichen Merkmale nicht möglich ist, Ironie eindeutig zu detektieren. Anstatt der Detektion von wahrer Ironie detektiert der Prototyp somit eher *potenzielle Ironie*.

Es stellt sich die Frage, ob die Erkenntnisse im Bereich der Sentiment-Analyse im Software Engineering angewendet werden können. Unterschiedliche Ansätze, wie die Überarbeitung der Lexika für die Domäne oder das Anpassen von Datensätzen konnten bereits zu verbesserten Ergebnissen von Sentiment-Analyse-Tools führen [18].

Da in ironischen Aussagen immer eine Bewertung und somit auch eine Stimmung vermittelt wird, ist zu erwarten, dass sich die Vorhersagen von Sentiment-Analyse-Tools bei richtiger Anwendung einer Erweiterung zur Ironiedetektion definitiv verbessern. Zusätzlich muss geklärt werden, inwieweit die *potenzielle Ironie* bewertet werden muss. Eine fälschlicherweise erkannte Ironie sollte die Polarität in einer Sentiment-Analyse nicht verändern. Bei der richtig detektierten Ironie ist nicht klar, inwieweit sich die Polarität ändert, denn die in dieser Arbeit genutzte Definition von Ironie kehrt die Polarität einer Aussage nicht einfach um. Die Polaritäten können erhalten bleiben oder neutral eingestufte Aussage können durch einen Ironiegehalt eine Polarität erlangen.

Auch wenn Rakow [40] in seiner Arbeit eine geringe Nutzung von Ironie im Software Engineering feststellt, kann ein Merkmalsvektor, der angibt, ob eine Aussage *potenziell ironisch* ist, in die Klassifikation von Sentiment-Analyse-Tools einfließen und so die Vorhersagen möglicherweise verbessern. Weiterhin darf der Einfluss ironischer Aussagen welche ausschließlich kontextabhängigen Merkmalen haben nicht unterschätzt werden, da diese mit 37% einen beachtlichen Anteil aufweisen.

6.2 Grenzen der Arbeit

Für die Einordnung der Ergebnisse dieser Arbeit müssen die Grenzen der Arbeit berücksichtigt werden. Dabei handelt es sich um Aspekte des verwendeten Datensatzes und angewandten Konzeptes, welche die Ergebnisse beeinflussen können.

Der entwickelte Prototyp wurde nicht mit weiteren Datensätzen evaluiert, da bisher keine Datensätze mit akustischen sowie transkribierten Aussagen, welche zusätzlich auf Ironie markiert sind, vorliegen. Dadurch konnte eine Übertragbarkeit der Ergebnisse nicht sichergestellt werden.

Weiterhin gibt es mehrere Aspekte bei der Erstellung des Datensatzes, welche die Validität der Ergebnisse einschränken. In der Aufgabenbeschreibung für das Pair Programming wurde darauf hingewiesen, dass ironische Kommentare erwünscht sind. Es sollte damit sichergestellt werden, dass der Anteil an ironischen Aussagen für die Erstellung eines Klassifizierungsmodells groß genug ist. Dadurch spiegeln die Gespräche die Realität jedoch weniger gut wider. Das Verhalten der Probanden kann außerdem durch den Faktor, dass die Gespräche aufgezeichnet wurden, beeinflusst worden sein. Zudem wurden die Gespräche mit unterschiedlichen Mikrofonen aufgezeichnet, wodurch die Tonqualität der Aufzeichnungen variiert. Des Weiteren handelte es sich bei den Probanden um junge Erwachsene, die gut befreundet sind. Es ist wahrscheinlich, dass sich die Kommunikation in beruflichen Entwicklerteams von den für diese Arbeit aufgenommenen Gesprächen unterscheidet. All diese Faktoren können die Ergebnisse beeinflussen.

Für den Datensatz wurden Gespräche von sechs Personen aufgezeichnet. Da der Sprachgebrauch von Personen sehr stark variieren kann, ist eine Allgemeingültigkeit des Prototyps wahrscheinlich nicht gegeben. Zudem können sich Merkmale von männlichen zu weibliche Stimmen stark unterscheiden. Damit eine höhere Allgemeingültigkeit erreicht werden kann, ist ein Datensatz mit mehr Vielfältigkeit der Umstände empfehlenswert.

Die Annotation des Datensatzes wurde von den Teilnehmenden selber und mir durchgeführt. Externe Beobachter, die für die Annotation nur die Tonspur und das Transkript zur Verfügung gehabt hätten, hätten womöglich in vielen Fällen die Ironie nicht erkennen können. Trotzdem ergab die Klassifizierung nur eine moderate Reliabilität, was auf eine hohe Subjektivität von Ironie hinweist. Dadurch, dass jede Person Ironie unterschiedlich auffasst und keine eindeutige Definition existiert, entsteht wiederum eine geringere Allgemeingültigkeit.

Ein weiterer kritischer Punkt ist die nachträgliche Klassifizierung der ironischen Aussagen nach verbalen und akustischen Merkmalen, die ausschließlich von mir durchgeführt

wurde. Um sicherzustellen, dass die Klassifizierung objektiv getroffen wurde, muss sie von weiteren Personen validiert werden.

6.3 Implikation für künftige Forschung

Wie die vorherigen Abschnitte andeuten, birgt die Analyse von Ironie noch viele Herausforderungen, welche in Zukunft weiter erforscht und entwickelt werden können.

Um die Übertragbarkeit der Ergebnisse sicherzustellen, sollten weitere Datensätze erstellt werden, welche Audiomitschnitte und deren Transkription enthalten, die mit Ironie markiert sind. Um einen Datensatz mit größtmöglicher Vielfältigkeit und damit hoher Aussagekraft erstellen zu können, sollten viele unterschiedliche Personen an der Erhebung teilnehmen. Zudem wäre es günstig, nicht nur Konversationen aus Pair Programming, sondern auch aus Meetings zu nutzen. Dabei ist zu beachten, dass sich die Anwendung von Ironie in Meetings von der in Einzelgesprächen stark unterscheiden könnte. Mit weiteren Datensätzen, die eine solche Vielfältigkeit der Umstände aufweisen, müssen die Ergebnisse der Validierung dieser Arbeite evaluiert werden.

In dieser Arbeit wurde festgestellt, dass eine eindeutige Detektion von Ironie durch ihre hohe Subjektivität nicht möglich ist. Trotzdem ist es möglich, *potenzielle Ironie* zu detektieren. Weiterhin wäre es interessant, ob unter Einbeziehung nicht betrachteter Merkmale bessere Ergebnisse erzielt werden können. Für die akustische Analyse werden nur die Mittelwerte der ersten 14 MFCCs eines Audiomitschnittes genutzt. Anhand weiterer Merkmale, wie beispielsweise der psychoakusitschen Schärfe oder Harmonie, der spektralen Varianz oder vielen weiteren, wie sie von Burkhardt et al. [6] für die Detektion von Ironie genutzt wurden, könnte die Klassifizierung weiter optimiert werden. Neben verbalen und akustischen Merkmalen gibt es auch Ironiemerkmale in der Mimik, die in weiteren Arbeiten analysiert werden könnten. Zudem wären weitere Forschungen interessant, die sich mit der Problematik der Erkennung von kontextabhängiger Ironie in Aussagen beschäftigen. In solchen Aussagen besteht eine Diskrepanz zwischen der Aussage und den gegebenen Umständen. Dabei können die gegebenen Umstände sehr vielfältig sein und sich zum Teil aus Hintergrundwissen über vorangegangene Gespräche beziehungsweise weiterem Wissen zusammensetzen. Bei der Detektion von kontextabhängiger Ironie kann die Erkenntnis hilfreich sein, dass ironische Aussagen mit kontextabhängigen Merkmalen gehäuft aufeinander folgen.

Wie im Abschnitt 6.1 schon aufgeführt, können sich zukünftigen Arbeiten mit dem Einfluss von *potenzieller Ironie* auf die Vorhersagen von Stimmungs-Analyse-Tools beschäftigen. Dabei sollten Aussagen, die fälschlicherweise als Ironie erkannt wurden, die Polarität einer Aussage nicht verändern. Es muss jedoch geklärt werden, wie sich die Polarität auf richtig detektierte Aussagen auswirkt. Die in dieser Arbeit genutzte Definition von Ironie kehrt die Polarität einer Aussage nicht einfach um, sodass die Polaritäten einer ironischen Aussage auch erhalten bleiben oder neutral eingestufte Aussagen einen Ironiegehalt eine Polarität erlangen.

Da viele Tools ausschließlich Textdaten analysieren, wäre es interessant, wenn weitere Arbeiten sich speziell mit dem Ironiegehalt von schriftlicher Kommunikation beschäftigen

würden. Dabei fällt allerdings die akustische Analyse, welche durchaus Potenzial hat, weg. Dafür ergeben sich neue Merkmale wie Emoticons oder Zeichensetzung, die für die Ironieerkennung interessant sein können.

Um einen größeren Anwendungsbereich zu ermöglichen, ist auch eine Verbindung des Prototyps mit einem automatischen Transkriptionstool denkbar. Anhand von Aussagen, die mit einer Autotranskription transkribiert und automatisch unterteilt werden, könnte eine Echtzeiterkennung potenzieller Ironie entwickelt werden. Diese könnte für die Verbesserung von Kommunikation genutzt werden. Beispielsweise könnte in Videokonferenzen ein Hinweissymbol potenzielle Ironie anzeigen. Dies ist vor allem praktisch, wenn neue Teammitglieder den Umgangston im Team noch nicht kennen. So kann ihnen der Einstieg in Team erleichtert werden. Wenn neue Teammitglieder den Hinweis für *potenzielle Ironie* bei einer Aussage angezeigt bekommen, können sie dieser Äußerung eine andere Bedeutung zuweisen. Zudem können Sprechende, wenn sie merken, dass sie keine eindeutige Aussage gemacht haben, diese für neue Teammitglieder verdeutlichen.

Ein letzter wichtiger Punkt für weitere Forschungsarbeiten liegt in der Anwendung des SentiWs Lexikons in Kombination mit der Pipeline von spacy. In dieser Arbeit wurden die Adjektive zusätzlich als Adverbien deklariert, was zu einer erheblichen Verbesserung der Ergebnisse führte. Tools, welche dieses Lexikon nutzen, sollten überprüfen, ob eine Anpassung auch dort zu besseren Ergebnissen führt.

Kapitel 7

Zusammenfassung

Es gibt einen Zusammenhang zwischen der Zufriedenheit von Softwareentwickelnden und deren Fähigkeit Probleme in der Entwicklung von Software zu lösen. Aus diesem Grund sollte die Stimmung innerhalb eines Softwareentwicklungsteams nicht vernachlässigt werden. Bei den aktuellen Tools zur Sentiment-Analyse, welche die Stimmung anhand von Meetingtranskripten, Git-Hub-Kommentaren oder Chatverläufen innerhalb von Software-Projekten bestimmen, ergibt sich die Schwierigkeit, die richtige Polarität für ironische Aussagen zu detektieren. Wenn die Tools um eine automatische Ironieerkennung ergänzt werden würden, würden sich die Polaritätsvorhersagen verbessern. Daraus resultiert das Ziel dieser Arbeit, Ironie mittels computergestützter Verfahren in Dialogen zu untersuchen.

Grundvoraussetzung war es zunächst eine für diese Arbeit gültige Definition für den Ironiebegriff zu erarbeiten, da keine einheitliche wissenschaftliche Definition existiert. Allgemein verbirgt sich die Bedeutung einer ironischen Aussage hinter dem wörtlichen Sinn. Sie kann viele unterschiedliche Motive sowie Wirkungen haben, zudem gibt es verschiedene Möglichkeiten, Ironie durch Signale erkennbar zu machen. Die Ironiesignale wurden in dieser Arbeit in kontextabhängige, verbale und paralinguistische Signale unterteilt.

Eine weitere Herausforderung war das Nichtvorhandensein einer nutzbaren Datengrundlage auf Basis deutscher Konversationen im Bereich des Software Engineerings. Aus diesem Grund wurde ein neuer Datensatz erstellt, der Audiomitschnitte und deren transkribierte Aussagen sowie eine Klassifizierung in *Ironie* und *Keine Ironie* enthält. Für die Datenerhebung wurden die Konversationen dreier Gruppen, die aus jeweils zwei Personen bestanden, beim Bewerten und Verbessern von Programmcode aufgenommen. Damit sichergestellt werden konnte, dass genug ironische Aussagen im endgültigen Datensatz vorhanden sind, wurde darauf hingewiesen, dass bei der Bewertung ironische Aussagen erwünscht sind. Die Aussagen wurden anschließend von den Teilnehmenden und mir nach Ironie klassifiziert. Die Klassifikation der manuellen Annotation ergab eine moderate Übereinstimmung, was auf eine hohe Subjektivität bei der Bewertung von Ironie hinweist. Des Weiteren stellte sich heraus, dass nach einer ironischen Aussage, häufig eine weitere ironische Aussage folgt.

Um Aussagen zu verstehen, welche ausschließlich kontextabhängige Ironiesignale aufweisen, wird Hintergrundwissen benötigt. Um die zu detektierende Ironie einzugrenzen, sollte sich die computergestützte Analyse nur auf die verbalen und akustischen Ironiesignale beziehen. Dafür wurden die ironischen Aussagen nach der ersten Annotation auf verbale oder akustische Merkmale markiert.

Für die computergestützte Klassifizierung von Aussagen wurden jeweils 14 MFCC Merkmale aus einem Audiomitschnitt und acht weitere Merkmale aus dessen Transkriptionen extrahiert. Somit erhielt jede Aussage insgesamt 22 Merkmale. Bei den Merkmalen aus den Transkriptionen handelt es sich um Stimmungsmerkmale, Merkmale zur Häufigkeit von Worten sowie dem Verwenden von bestimmten Wortarten in ironischen und nicht ironischen Aussagen. Zum Vergleich der Merkmale wurden Klassifizierungsmodelle für fünf unterschiedlichen Kombinationen von Merkmalen erstellt. Dabei wurden die Merkmale zusammengefasst, für die dieselben oder sehr ähnliche Techniken zur Merkmalsextraktion angewendet wurden. Für die Erstellung von Klassifizierungsmodellen wurde eine Support Vector Machine genutzt. Bei jeder Kombination aus Merkmalen konnte ein Unterschied zwischen ironischen und nicht ironischen Aussagen ermittelt werden, wobei sich herausstellte, dass die Wortarten eine eher geringere Aussagekraft in Bezug auf Ironie haben. Die besten Klassifizierungsergebnisse konnten unter gleichzeitiger Anwendung aller 22 Merkmale erzielt werden. Dabei ist zu beachten, dass zwar durchschnittlich 75% der ironischen Aussagen mit akustischem oder verbalen Merkmal detektiert wurden, der Precision-Wert für die Klasse *Ironie* jedoch trotzdem nur bei 36% liegt, da noch durchschnittlich 19% der nicht ironischen Aussagen fälschlicherweise als Ironie detektiert werden. Aus diesen Ergebnissen und der moderaten Übereinstimmung von ironischen Aussagen konnte geschlossen werden, dass es nicht möglich ist, Ironie eindeutig zu identifizieren. Allerdings geht ebenfalls aus den Untersuchungen hervor, dass es durchaus möglich ist anhand einer computergestützten Analyse *potenzielle Ironie* zu detektieren. Inwieweit das Erkennen von *potenzieller Ironie* für weitere Forschungen genutzt werden kann, muss geklärt werden.

Anhang A

Dokumente der Datenerhebung

A.1 Aufgabenbeschreibung zur Gesprächsaufzeichnung

Beurteilen Sie den vorliegenden Code

Sie haben die Aufgabe den vorliegenden Code in dem Ordner zu beurteilen. Verbessern Sie die Stellen, die Sie für schlecht lesbaren Code halten. Wenn sich Fehler im Code bef nden verbessern Sie diese auch. Sie können den Code in der Konsole mit der Konsoleneingabe "python3 main.py" ausführen. Sie haben eine halbe Stunde Zeit um den Code so zu verbessern, dass dieser für Sie gut lesbar und schnell verständlich wirkt. Tauschen Sie sich die gesamte Zeit viel mit ihrem Teammitglied aus. Achten Sie dabei auf eine relativ deutliche Aussprache. Es ist ausdrücklich erwünscht auch ironische Kommentare als Bewertung zu nutzen. Ironische Aussagen sind solche, bei denen man etwas sagt, dass man nicht so meint. Dazu gehört, dass das Gegenüber merkt, das dieser Satz nicht so gemeint war.
Ein Beispiel: "Ich weiß nach dieser langen Anweisung ganz genau was zu tun ist."

A.2 Aufgabenbeschreibung zur Annotation der Daten

Labeln von Ironie in diversen Datensätzen

Bei dem ausgehändigten Datensatz müssen Aussagen die Ironie beinhalten markiert werden. Eine große Schwierigkeit besteht darin, dass Ironie nur erkannt werden kann, wenn ein gemeinsamer Wissensbestand zwischen Sprechenden und Adressaten vorhanden ist, deswegen ist es von großem Vorteil, dass sie selbst an der Datenerhebung teilgenommen haben. Es können/müssen nur Aussagen markiert werden, bei denen die Ironie ausreichend erkennbar ist. In der Spalte „Ironie" sollen die Aussagen wie folgt markiert werden:

- Mit einer „2", wenn Sie sich sicher sind, dass diese Aussage ironisch gemeint ist.
- Mit einer „1", wenn sie sich unsicher sind, ob die Aussage ironisch gemeint war
- Gar nicht, wenn Sie sich sicher sind, dass es sich um keine ironische Aussage handelt.

Die Arbeit hält sich an folgende Aussagen um Ironie zu def nieren:
Ironie als Gegenteil zwischen Sagen und Meinen (Beispiel: „Du bist ja pünktlich.", wenn eigentlich gemeint ist „Du bist ja unpünktlich.")
Ironie als etwas Anderes zu sagen als man meint (Beispiel: „Die Party war fantastisch.", wenn eigentlich gemeint ist „Die Party war ok.")
Ironie als Lob durch Tadel und Tadel durch Lob (Beispiel: „Das hast du ja mal wieder toll gemacht.", wenn eigentlich gemeint ist „Ich bin nicht begeistert, dass du das kaputt gemacht hast." oder „Eine 1,7 ist ja echt so schlecht." wenn eigentlich gemeint ist „Eine 1,7 ist voll gut.")
Ironie als Art des Sich-Lustigmachens und Spottens (Beispiel: Jemand zitiert eine Person und verstellt dabei die Stimme.)
Um es noch etwas zu verdeutlichen: Ironie muss immer erkennbar gemacht werden, wenn sie ihre Wirkung erreichen soll. Ironie muss trotzdem nicht explizit markiert werden. Häuf g wird sie über gemeinsame Wissensbestände erkennbar, oft über die aktuelle Situation und den Gesprächsverlauf. Aus diesen Wissensbeständen lässt sich für den Zuhörer die Bedeutung und die Funktion der ironischen Äußerung ableiten.

A.3 Ausschnitt aus dem Datensatz

	Nummer	Datei	Onset_Time	Offset_Time	Teilnehmer	Ironie	Ironie_A	Ironie_B	Ironie_A2	Ironie_B2
22	23	gruppe1	00:01:20,845	00:01:22,525	A	2	2	2	2	-
23	24	gruppe1	00:01:22,439	00:01:26,622	B	2	2	2	2	-
24	25	gruppe1	00:01:26,533	00:01:32,919	B	0	0	0	0	-
25	26	gruppe1	00:01:33,227	00:01:35,316	B	0	0	0	0	-

	Merkmal_Text	Merkmal_Audio	Transcript
22	3.0	2.0	Welche Ehre, das wir die erste Gruppe sind.
23	3.0	1.0	Ja, kann mir nichts besseres Vorstellen
24	NaN	NaN	Ok, sie haben die Aufgabe den vorliegenden Code in dem Ordner Gruppe1 zu beurteilen.
25	NaN	NaN	Also da ist anscheinend Code drinne.

Literatur

[1] José María Alcaide, Raquel Justo und María Inés Torres. "Combining Statistical and Semantic Knowledge for Sarcasm Detection in Online Dialogues". In: *Pattern Recognition and Image Analysis*. Hrsg. von Roberto Paredes, Jaime S. Cardoso und Xosé M. Pardo. Cham: Springer International Publishing, 2015, S. 662–671. ISBN: 978-3-319-19390-8.

[2] Ayşe Arslan und Oktay Yildiz. "Automated Auscultative Diagnosis System for Evaluation of Phonocardiogram Signals Associated with Heart Murmur Diseases". In: *Gazi University Journal of Science* 31 (März 2018), S. 112–124.

[3] Salvatore Attardo. "Irony markers and functions: Towards a goal-oriented theory of irony and its processing". In: *Rask* 12.1 (Jan. 2000), S. 3–20. DOI: 10.1145/3136273.3136298.

[4] David Baccarini. "The Logical Framework Method for Defining Project Success". In: *Project Management Journal* 30.4 (1999), S. 25–32. DOI: 10.1177/875697289903000405.

[5] Wolfgang Berg. "Uneigentliches Sprechen zur Pragmatik und Semantik von Metapher, Metonymie, Ironie, Litotes und rhetorischer Frage". In: Tübinger Beiträge zur Linguistik 102 (1978).

[6] Felix Burkhardt u. a. "Detecting Vocal Irony". In: *Language Technologies for the Challenges of the Digital Age, Lecture Notes in Artificial Intelligence*. Hrsg. von Georg Rehm und Thierry Declerck. Bd. 10713. Cham: Springer International Publishing; 2018, S. 11–22. ISBN: 978-3-319-73706-5. DOI: 10.1007/978-3-319-73706-5_2.

[7] Stefano Cagnoni u. a. "Emotion-based analysis of programming languages on Stack Overflow". In: *ICT Express* 6.3 (2020), S. 238–242. ISSN: 2405-9595. DOI: 10.1016/j.icte.2020.07.002.

[8] Fabio Calefato u. a. "Sentiment Polarity Detection for Software Development". In: 23 (Mai 2018), S. 1352–1382. DOI: 10.1145/3180155.3182519.

[9] Mohamed M. Dessouky u. a. "Computer Aided Diagnosis System Feature Extraction of Alzheimer Disease Using MFCC". In: *International Journal of Intelligent Computing in Medical Sciences & Image Processing* 6.2 (2014), S. 65–78. DOI: 10.1080/1931308X.2015.1004823.

[10] Klaus Fellbaum. *Sprachverarbeitung und Sprachübertragung.* Berlin: Springer, 2012. DOI: 10.1007/978-3-642-31503-9.

[11] Jörg Frochte. *Maschinelles Lernen : Grundlagen und Algorithmen in Python.* München: Hanser; 2018. ISBN: 978-3-446-45997-7.

[12] Raymond W. Gibbs. "Irony in Talk Among Friends". In: *Metaphor and Symbol* 15.1-2 (2000), S. 5–27. DOI: 10.1080/10926488.2000.9678862.

[13] Daniel Graziotin, Xiaofeng Wang und P. Abrahamsson. "Happy software developers solve problems better: psychological measurements in empirical software engineering". In: *PeerJ* 2 (2014). DOI: 10.7717/peerj.289.

[14] Jeffrey T. Hancock. "Verbal Irony Use in Face-To-Face and Computer-Mediated Conversations". In: *Journal of Language and Social Psychology* 23.4 (Dez. 2004), S. 447–463. DOI: 10.1177/0261927X04269587.

[15] Martin Hartung. *Ironie in der Alltagssprache : Eine gesprächsanalytische Untersuchung.* Wiesbaden: VS Verlag für Sozialwissenschaften; 1998. DOI: 10.1007/978-3-322-88985-0.

[16] Irazú Hernández-Farías, José-Miguel Benedí und Paolo Rosso. *Applying Basic Features from Sentiment Analysis for Automatic Irony Detection.* Juni 2015. DOI: 10.1007/978-3-319-19390-8_38.

[17] Md Rakibul Islam und Minhaz F. Zibran. "DEVA: Sensing Emotions in the Valence Arousal Space in Software Engineering Text". In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing.* New York, NY, USA: Association for Computing Machinery, 2018, S. 1536–1543. DOI: 10.1145/3167132.3167296.

[18] Md Rakibul Islam und Minhaz F. Zibran. "Leveraging Automated Sentiment Analysis in Software Engineering". In: *2017 IEEE/ACM 14th International Conference on Mining Software Repositories MSR.* IEEE; 2017, S. 203–214. DOI: 10.1109/MSR.2017.9.

[19] Anjali Ganesh Jivani u. a. "A comparative study of stemming algorithms". In: *Int. J. Comp. Tech. Appl* 2.6 (2011), S. 1930–1938.

[20] Simone Kauffeld und Nale Lehmann-Willenbrock. "Meetings Matter: Effects of Team Meetings on Team and Organizational Success". In: *Small Group Research* 43.2 (2012), S. 130–158. DOI: 10.1177/1046496411429599.

[21] Maria Khokhlova, Viviana Patti und Paolo Rosso. "Distinguishing between irony and sarcasm in social media texts: Linguistic observations". In: *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT).* Sep. 2016, S. 1–6. DOI: 10.1109/FRUCT.2016.7584765.

[22] Norman Knox. "Die Bedeutung von Ironie. Einführung und Zusammenfassung". In: *HASS, Hans-Egon; MOHRLÜDER, Gustav-Adolf. Ironie als literarisches Phänomenon. Köln: Kiepenheuer & Witsch* (1973), S. 21–30.

[23] Norman Knox. "On the Classification of ironies". In: *Modern Philology 70.* (1972), S. 53–62.

[24] Norman Knox. *The word irony and its context, 1500-1755.* Durham: Duke University Press, 1961.

[25] Roger J. Kreuz. *Irony and Sarcasm, MIT Press Essential Knowledge Series.* Cambridge: MIT Press, Juni 2020. ISBN: 978-0-262-53826-8.

[26] Rohit Kumar. "Natural Language Processing". In: *Machine Learning and Cognition in Enterprises: Business Intelligence Transformed.* Berkeley, CA: Apress, Nov. 2017, S. 65–73. DOI: 10.1007/978-1-4842-3069-5_5.

[27] Edgar Lapp. "Linguistik der Ironie". Diss. Tübingen: Zugl.: München, Univ., Diss., 1977, 1992. ISBN: 3-8233-4224-X.

[28] Jennifer Ling und Roman Klinger. "An Empirical, Quantitative Analysis of the Differences Between Sarcasm and Irony". In: *The Semantic Web.* Hrsg. von Harald Sack u. a. Cham: Springer International Publishing, 2016, S. 203–216. DOI: 10.1007/978-3-319-47602-5_39.

[29] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2010. DOI: 10.1017/S1351324909005129.

[30] Friedrich Markewitz. "Ironie". In: *Wörterbücher zur Sprach- und Kommunikationswissenschaft (WSK) Online.* Hrsg. von Kirstin Casemir u. a. De Gruyter, 2013. URL: https://www.degruyter.com/database/WSK/entry/wsk_idcfbf9fa4-19c9-41f8-8fe5-7eb5007f31e8/html.

[31] Mary L McHugh. "Interrater reliability: The kappa statistic". In: *Biochemia medica* 22.3 (Okt. 2012), S. 276–282. DOI: 10.11613/BM.2012.031.

[32] Walaa Medhat, Ahmed Hassan und Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". In: *Ain Shams Engineering Journal* 5.4 (2014), S. 1093–1113. DOI: 10.1016/j.asej.2014.04.011.

[33] Rodrigo Moraes, João Francisco Valiati und Wilson P. Gavião Neto. "Document-level sentiment classification: An empirical comparison between SVM and ANN". In: *Expert Systems with Applications* 40.2 (2013), S. 621–633. DOI: 10.1016/j.eswa.2012.07.059.

[34] Nicole Novielli, Daniela Girardi und Filippo Lanubile. "A benchmark study on sentiment analysis for software engineering research". In: *Proceedings of the 15th International Conference on Mining Software Repositories.* ACM, Mai 2018. DOI: 10.1145/3196398.3196403.

[35] Martin Obaidi und Jil Klünder. "Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review". In: *Evaluation and Assessment in Software Engineering.* EASE 2021. Trondheim, Norway: Association for Computing Machinery, 2021, S. 80–89. DOI: 10.1145/3463274.3463328.

[36] Zena Obeidi. *Automatisierte Erkennung von destruktiven Äußerungen in Meetings von Softwareprojekten.* Masterarbeit, Leibniz Universität Hannover. 2021.

[37] Ursula Oomen. "Ironische Äußerungen: Syntax – Semantik – Pragmatik". In: *Zeitschrift für germanistische Linguistik* 11.1 (1983), S. 22–38.

[38] Eric Partridge. *Usage and abusage. A guide to good English.* Norton, W. W. & Company, Inc., 1997. ISBN: 9780393317091.

[39] Elke Prestin. *Ironie in Printmedien.* Psycholinguistische Studien. Deutscher Universitätsverlag, 2000. DOI: 10.1007/978-3-663-08175-3. (Besucht am 08. 10. 2021).

[40] Alexander Rakow. *Analyse der Verwendung von Ironie in natürlicher Sprache in Softwareprojekten.* Bachelorarbeit, Leibniz Universität Hannover. 2021.

[41] D. Ramyachitra und P. Manikandan. "Imbalanced dataset classification and solutions: a review". In: *International Journal of Computing and Business Research (IJCBR)* 5.4 (2014), S. 1–29.

[42] Payam Refaeilzadeh, Lei Tang und Huan Liu. "Cross-validation." In: *Encyclopedia of database systems* 5 (2009), S. 532–538. DOI: 10.1007/978-0-387-39940-9_565.

[43] Robert Remus, Uwe Quasthoff und Gerhard Heyer. "SentiWS - A Publicly Available German-language Resource for Sentiment Analysis". In: *LREC.* 2010.

[44] Antonio Reyes und Paolo Rosso. "On the difficulty of automatically detecting irony: beyond a simple case of negation". In: *Knowledge and information systems : an international journal* 40.3 (2014), S. 595–614. DOI: 10.1007/s10115-013-0652-8.

[45] Omid Rohanian u. a. "WLV at SemEval-2018 Task 3: Dissecting Tweets in Search of Irony". In: *Proceedings of The 12th International Workshop on Semantic Evaluation.* New Orleans, Louisiana: Association for Computational Linguistics, 2018, S. 553–559. DOI: 10.18653/v1/S18-1090.

[46] Vasile Rus, Anne L. C. Runehov und Lluis Oviedo. "Natural Language Processing". In: *Encyclopedia of Sciences and Religions.* Dordrecht: Springer Netherlands, 2013, S. 1401–1404. DOI: 10.1007/978-1-4020-8265-8_1225.

[47] Rachel Rhoda Schaffer. "Vocal cues for irony in English". Diss. The Ohio State University, 1982.

[48] Melanie Siegel und Melpomeni Alexa. "Sentiment-Analyse deutschsprachiger Meinungsäußerungen: Grundlagen, Methoden und praktische Umsetzung". In: Wiesbaden: Springer Fachmedien Wiesbaden, 2020, S. 71–74. ISBN: 9783658296988.

[49] Emilio Sulis u. a. "Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not". In: *Knowledge-Based Systems* 108 (2016), S. 132–143. DOI: 10.1016/j.knosys.2016.05.035.

[50] Anja Tetzner. *Einsatz künstlicher neuronaler Netze zur Identifikation von Ironie im Rahmen der Sentiment-Analyse, Schriftenreihe Studien zur Wirtschaftsinformatik*. Hamburg, 2020.

[51] Mike Thelwall u. a. "Sentiment Strength Detection in Short Informal Text". In: *Journal of the American Society for Information Science and Technology* 61 (Dez. 2010), S. 2544–2558. DOI: 10.1002/asi.21416.

[52] T. Veale u. a. "Detecting Ironic Intent in Creative Comparisons". In: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. Bd. 215. NLD: IOS Press, 2010, S. 765–770. DOI: 10.5555/1860967.1861117.

[53] Byron C Wallace. "Computational irony: A survey and new perspectives". In: *Artificial intelligence review : an internat. survey and tutorial journal* 43.4 (2015), S. 467. DOI: doi.org/10.1007/s10462-012-9392-5.

[54] Chuhan Wu u. a. "THU_NGN at SemEval-2018 Task 3: Tweet Irony Detection with Densely connected LSTM and Multi-task Learning". In: **SEMEVAL*. Juni 2018.

[55] Ting Zhang u. a. "Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?" In: *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Sep. 2020, S. 70–80. DOI: 10.1109/ICSME46990.2020.00017.