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

Automatische Sentiment-Analyse und Emotionserkennung in Software Feedback-Videos

Automatic Sentiment Analysis and Emotion Recognition in Software Feedback Videos

Bachelorarbeit

im Studiengang Informatik

von

Benjamin Riebe

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Prof. Dr. Ralph Ewerth
Betreuer: M. Sc. Jianwei Shi

Hannover, 08.04.2024

Erklärung der Selbstständigkeit

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

Hannover, den 08.04.2024

Benjamin Riebe

Zusammenfassung

Automatische Sentiment-Analyse und Emotionserkennung in Software Feedback-Videos

Die Bedeutung von Kunden- und Benutzer-Feedback in der Softwareentwicklung liegt darin, Verbesserungsmöglichkeiten aufzuzeigen, sei es durch Vorschläge für neue Features oder die Identifizierung von Bedienungsschwierigkeiten. Sentiment-Analyse und Emotionserkennung gewinnt in den letzten Jahren zunehmend an Beliebtheit, insbesondere seitdem die Rechenleistung von Grafikkarten einen bestimmten Wert überschritten hat, was den Einsatz von maschinellem Lernen im Privatsektor ermöglicht, wo das Budget begrenzt ist.
In dieser Arbeit werden mehrere Deep Learning Modelle entwickelt, die Text- und Audio-Modalitäten als Eingabe erhalten, welche für die Sentiment-Analyse und Emotionserkennung verwendet werden.
Die verwendeten Datensätze bestehen aus YouTube Videos von hauptsächlich Film Reviews, die bereits mit Sentiment- und Emotionswerten annotiert wurden. Diese Videos werden in Shots unterteilt und aus diesen Shots wiederum werden Sätze extrahiert. Die Texte aus den Sätzen spiegeln die Modalität „Sprache“ wider, während die Tonlage, Lautstärke und andere Faktoren die Audiomodalität darstellen.
Nachdem das multimodale Modell trainiert wurde, kann es mit Hilfe der Metriken **Mean Absolute Error** (MAE) und **Mean Squared Error** (MSE) mit anderen Modellen verglichen werden. Es wird gezeigt, dass das Modell mit **später Fusion** den anderen Modellen mit **früher** und **mittelspäter Fusion** leicht überlegen ist im Bezug zur multimodalen Sentiment-Analyse und Emotionserkennung.
Die erstellten Modelle können behilflich sein, den Zeit- und Kostenaufwand zu reduzieren, indem sie den Prozess zum Auswerten von Kunden-Feedback unterstützen und erleichtern, wenn dieses Feedback in Form von Videos vorliegt.

Abstract

Automatic Sentiment Analysis and Emotion Recognition in Software Feedback Videos

Feedback by customers or users plays a central role in software development as they are useful to show possible ways of improvement, either by ideas for new features or displaying difficulties in usage of the software. Sentiment analysis and emotion recognition has become increasingly popular in recent years. Because the computing power of graphics cards has exceeded a certain value, this enabled machine learning to be used in the private sector where the budget is limited.
Multiple deep learning models, with the modalities text and audio, will be developed in this work which receive text and audio as input to execute sentiment analysis and emotion recognition.
The data set used for training is made up of YouTube videos which are mostly movie reviews which are already labeled with sentiment and emotion values. These videos get split into shots, and sentences are extracted from those. The text from the sentences represent the text modality and the speaker's pitch, loudness and other factors fulfil the audio modality.
After training such model it can be evaluated with the help of metrics such as **Mean Absolute Error** (MAE) and **Mean Squared Error** (MSE). These metrics help in comparing different models. It is shown that the **late fusion** model is slightly superior to the **early** and **mid** fusion models in the context of multimodal sentiment analysis and emotion recognition.
The created models may be of help by reducing the cost and time consumption by aiding the process of analysing customer feedback if they exist in the shape of videos.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Kritik ist notwendig, um Stärken und Schwächen, gute und schlechte Seiten aufzuzeigen. Sie wird benötigt für die Weiterentwicklung eines Produkts, um gute Qualitäten erweitern zu können oder Makel zu finden. Die Kategorisierung und Auswertung solcher Kritik ist zeit- und kostenaufwändig und kann zum Großteil automatisiert werden, wenn sie digital vorliegt.
Die Bedeutung von Kritik erstreckt sich über viele verschiedene Branchen, von Großhändlern wie Amazon bis hin zur Film- und Spiele-Industrie. In diesen Bereichen beschränkt sich die Kritik in der Regel auf die Bewertung von Produkten, wobei hier positive und negative Aspekte gleichermaßen betrachtet werden. Analyse solcher Kritik fällt unter den Begriff der Sentiment-Analyse, die es möglich macht, die Positionen und Meinungen der Verbraucher oder Nutzer zu erfassen.
In den sozialen Medien, wie Twitter, und Video Plattformen, wie YouTube, werden täglich zahlreiche Texte und Videos erstellt, welche sich anbieten, automatisch auf Sentiment und Emotionen analysiert zu werden. Hier teilen Konsumenten und Nutzer ihre Gedanken und Erfahrungen in Form von Texten oder Videos mit, welche unmittelbar Feedback für die Produzenten und Entwickler liefern. Vor allem dann, wenn man selbst für ein Produkt verantwortlich ist, sollte man sich dieses Feedback in Bezug auf das Produkt stark zu Herzen nehmen. Bei der automatischen Auswertung lassen sich nicht nur die Texte analysieren, sondern, wenn sie in Form von Videos vorliegen, auch Aussprache, Betonung, Mimik und Gestik.
Bei der automatischen Sentiment Analyse erzielen zwar die aktuellen state-of-the-art Ergebnisse aus der Forschung noch Genauigkeiten (Accuracy) unter denen von Menschen [1, 28], sind jedoch nicht mehr weit davon entfernt. Die Genauigkeit der Ergebnisse kann erhöht werden, wenn man mehrere Modalitäten gleichzeitig betrachtet, z.B. Text und Video anstatt nur Text [3, 7, 16, 17, 23]. Einige Wörter oder Sätze können nämlich mehrere Bedeutungen bei verschiedenen Betonungen oder Kontexten besitzen, wobei Gestik und Mimik die entscheidende Rolle spielen können, um diese richtig

einzuordnen. So kann der Satz „Ich hatte Tränen in den Augen.“ bei positiver Emotion als Freudentränen verstanden werden, und bei negativer Emotion als Tränen der Trauer.
Speziell kann man mit solch automatisierten Verfahren das Verhalten und die ausgesprochenen Gedanken analysieren, während ein Nutzer eine Software zum ersten Mal verwendet. Der Umgang mit der Software kann den Benutzer in schwierigen Situationen überfordern und dadurch Frust und Ungeduld erzeugen, was in schlechter Laune münden kann. Dies zeigt auf, dass die Software an dieser Stelle für diesen Benutzer nicht intuitiv sondern verbesserungswürdig ist.
Aus Review- oder Feedback-Videos kann man positive Kritik verwenden, um eine häufig verwendete Software oder Feature weiter auszubauen. So ist es wünschenswert auf Twitter in kurzer Zeitspanne Tweets oder Leute zu finden, die einen interessieren. Twitter hat auf solche Kritik reagiert und seitdem die Navigationssysteme weiter ausgebaut.
Bei negativer Kritik hingegen sollte man schnell reagieren, also die Software oder das Feature vereinfachen, oder im Notfall sogar entfernen. So hat Google z.B. jahrelang versucht Google+ durchzusetzen, mussten es jedoch aufgrund von geringer Aktivität im Jahr 2019 einstellen.
In diesem Zusammenhang ergibt sich die zentrale Forschungsfrage:

Zentrale Forschungsfrage

Welche Arten von Deep Learning Modellen eignen sich am besten für die Sentiment-Analyse und Emotionserkennung von Software Feedback-Videos?

1.1 Problemstellung

In der Universität Hannover im Fachgebiet Software Engineering sind wir interessiert an der Analyse von Benutzer-Feedback. Dieses Feedback wird bereitgestellt in Form von Videos, in denen die Benutzer Kritik bezüglich einer Anwendung hinterlassen. Hierbei soll automatisch ausgelesen werden, ob die Kritik der Nutzer positiv oder negativ ist und ob sie Emotionen enthält.
Die folgenden Probleme treten dabei auf:

- Es muss untersucht werden, ob es sich bei einem gegebenen Video tatsächlich um ein Feedback-Video handelt.
- Der gesprochene Text muss extrahiert werden.
- Ein Deep Learning Modell muss trainiert werden, welches Text und Audio als Eingabe erhält und anschließend das analysierte Sentiment oder Emotion ausgibt.

- Für das Trainieren des Modells muss es Daten geben, die im richtigen Format vorliegen.

Sobald das Sentiment und die Emotionen dem Softwareentwickler bekannt sind, muss dies noch mit der Stelle in der Software verknüpft werden. Dabei sind neutrale Sentiment- und Emotionswerte eher uninteressant, denn die stark positiven oder stark negativen Werte sind ausschlaggebend dafür, ob ein Teil der Software sehr gelungen oder eher problematisch ist.

1.2 Lösungsansatz

Nach der Bereitstellung von Feedback-Videos wird zuerst der Text mit Hilfe von OpenAIs Whisper aus einem Video ausgelesen. Nun wird mit OpenAIs ChatGPT-4 herausgefunden, ob es sich tatsächlich um ein Feedback-Video handelt. Anschließend kann das Audio mit der Software FFmpeg aus der Video-Datei extrahiert werden, um es anschließend mit der Software openSMILE einzulesen, womit die Audio-Merkmale extrahiert werden.
Abschließend wird jeder extrahierte Satz, also Text und Audio-Merkmale, für ein Deep Learning Modell als Eingabe bereitgestellt. Die Ausgabe des Modells ist ein Wert, welcher das Sentiment-Level (positiv, neutral, negativ) oder die Emotion (love, joy, sadness, anger, disgust, surprise, fear) des Sprechers widerspiegelt.
Es werden also zwei Deep Learning Modelle mit der Bibliothek PyTorch trainiert. Eins für die Ausgabe des Sentiment-Wertes und ein weiteres für die Ausgabe der Emotion. Beide Modelle erhalten jeweils dieselbe Eingabe.
Für das Trainieren des Sentiment-Modells stehen die folgenden Datensätze zur Verfügung:

- CMU-MOSEI (Multimodal Opinion Sentiment and Emotion Intensity)
- CMU-MOSI (Multimodal Corpus of Sentiment Intensity)
- MELD (Multimodal EmotionLines Dataset)
- ACL2020Challenge

Für das Training des Emotion-Modells können die folgenden Datensätze verwendet werden:

- CMU-MOSEI
- IEMOCAP (Interactive Emotional Dyadic Motion Capture)
- MELD
- ACL2020Challenge

Die Emotionen sind bei den Datensätzen nicht identisch gekennzeichnet, sodass diese Beschriftungen (Labels) in eine einheitliche überführt werden müssen. In Kapitel 3.2.1 wird näher darauf eingegangen.

1.3 Struktur der Arbeit

Im Kapitel Verwandte Arbeiten werden Methoden und Techniken vorgestellt, die in anderen wissenschaftlichen Arbeiten bereits erfolgreich verwendet wurden. Es folgt das Kapitel Konzeption, in der die Lösung des Problems begründet erarbeitet und beschrieben wird, um mehrere Deep Learning Modelle mit PyTorch trainieren zu können. Am Ende des Kapitels wird dargestellt, wie sich die Modell Vorhersagen repräsentieren lassen. Die trainierten Modelle werden in Kapitel Evaluation auf ihre Genauigkeit analysiert und untereinander verglichen. Abschließend werden die Ergebnisse ausgewertet und interpretiert. Im Kapitel Diskussion wird besprochen, welche Software problematisch oder ungenau arbeiten kann, Verbesserungsmöglichkeiten werden aufgezeigt, und es wird ein Ausschnitt davon vorgestellt, in welchen Arbeitsbereichen die Modelle Anwendung finden können. Das Kapitel Zusammenfassung fasst die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf ausstehende Aufgaben.

Kapitel 2

Verwandte Arbeiten

In diesem Kapitel werden weitere wissenschaftliche Arbeiten und Publikationen vorgestellt, die sich ebenfalls mit Sentiment- oder Emotions-Analyse beschäftigt haben, wobei sich deren Ansätze und Methoden von dieser Arbeit meistens unterscheiden.
Das **Sentiment** spiegelt die grundsätzliche Haltung, Meinung oder Stimmung eines Textes wider und kann negativ, neutral oder positiv sein. Die **Emotion** ist dabei eine komplexe psychophysiologische Reaktion auf einen bestimmten Reiz oder eine bestimmte Situation, die in Kapitel 3.2.1 näher erläutert werden. Sie manifestieren sich mit unterschiedlicher Intensität und Dauer und sind oft mit persönlichen Erfahrungen und Wahrnehmung einer Person verbunden.

2.1 Vorbearbeitung von Text und Audio

Die Texte und Label der Sentiment- und Emotionswerte liegen in den Datensätzen von CMU-MOSI, CMU-MOSEI und IEMOCAP bereits vor, sodass weder die Extraktion des Textes noch die Beschriftung per Hand durchgeführt werden muss.
Allerdings müssen diese Texte zum Trainieren der Deep Learning Modelle in Tensoren umgewandelt werden, damit PyTorch diese einlesen kann. Yang et al. [27] konnten mit Hilfe von dem bereits existierenden „Bidirectional Encoder Representations from Transformers“ (**BERT**) pre-trained language representation Modell ein neues Modell erstellen mit Namen „Cross-Modal BERT“ (CM-BERT), welches Text- und Audio-Daten als Eingabe erhält, und dynamisch Wörtern bestimmte Gewichte zuordnet. In dieser Arbeit wurde ebenfalls **BERT** verwendet, um die Umwandlung von Text in Tensoren durchzuführen, und zwar mit Hilfe der Transformers Bibliothek von HuggingFace in Python, bei der pro Wort ein Tensor der Länge 768 zurückgegeben wird.
Eine Alternative zur Verwendung von **BERT** ist die Verwendung von GloVe

(Global Vectors for Word Representation), bei denen jedes Wort in einen Vektor der Länge 300 umgewandelt wird. Hierbei ist die Distanz zwischen zwei Vektoren klein, wenn die dazugehörigen Wörter oft im selben Kontext verwendet werden, wie beispielsweise „König“ und „Königin“. Cai et al. [2], Liang et al. [10], Tsai et al. [23] und Zadeh et al. [28] verwendeten GloVe zum Extrahieren der textuellen Merkmale, um jeweils ein oder mehrere multimodale Modelle zu erstellen.
Mit Hilfe der Zeitstempel (Start- und Endzeitpunkt der einzelnen Sätze) können die Audio Merkmale mit der Software **openSMILE** aus den Originalvideos extrahiert werden. Diese Software haben Poria et al. [17] ebenfalls verwendet. In deren wissenschaftlichen Arbeit wurde die **openSMILE** Konfiguration „IS13-ComParE“ verwendet. In dieser Arbeit wurde sich jedoch für eine andere openSMILE Konfiguration entschieden, da „IS13-ComParE“ zu viele Merkmale aus einer Audio-Datei extrahiert.
An Stelle von openSMILE haben Chen et al. [3], Liang et al. [10] und Zadeh et al.[28] die Software COVAREP verwendet, um Audio Merkmale zu extrahieren.
Anschließend können die Text- und Audio-Tensoren zu einem eindimensionalen Vektor aneinandergehängt werden, welches zu einem Vektor der Länge 7141 (= 768 + 6373) mit der Konfiguration „IS13-ComParE“ führt. Diese Methode wird als **frühe Fusion** bezeichnet, da die verschiedenen Modalitäten früh zusammengeführt werden, welche Poria et al. [18] ebenso verwendeten. Zadeh et al. [28] stellen fest, dass bei **früher Fusion** die Modalitäten in Abhängigkeit gebracht werden, was von Vorteil sein kann, aber auch wegen der Komplexität und Größe des Eingabetensors leicht zu Überanpassung des Modells führen kann.
Im Gegensatz dazu steht die **späte Fusion**, bei der zunächst mehrere unimodale Modelle trainiert werden, bevor sie am Ende zu einem Modell zusammengeführt werden. Beschrieben wurde die Technik der **späten Fusion** von Ghorbanali und Sohrabi [6], Sun et al. [20] und Yadav und Vishwakarma [26].

2.2 Sentiment-Analyse und Emotionserkennung

Die Sentiment-Analyse und Emotionserkennung kann auf verschiedenen Ebenen geschehen. Wankhade et al. [24] haben mehrere Methoden und Applikationen untersucht, bei der sie Folgendes festgestellt haben. Das Sentiment kann entweder auf **Dokument-Level**, auf **Satz-Level**, oder sogar auf **Phrasen-Level** untersucht werden. Bei der Analyse eines Dokuments auf Sentiment können Sätze gemischte Sentimentwerte enthalten, letztendlich wird jedoch nur ein Gesamtsentiment erhalten. Bei der Analyse auf **Satz-Level** wird jeder Satz nach dessen Sentiment untersucht - also wird jedem Satz ein Sentimentwert zugeordnet. Hierbei kann ebenfalls ein Teilsatz ein

positives Sentiment, und ein weiterer Nebensatz ein negatives Sentiment aufweisen, wie in dem Beispiel „Obwohl das Wetter heute wunderschön ist, konnte ich nicht den geplanten Ausflug genießen, da ich erkältet bin.“. Die Sentimentanalyse auf **Phrasen-Level** kann dies genauer untersuchen.
In dieser Arbeit wird sich auf die Untersuchung auf **Satz-Level** beschränkt. Die Datensätze liegen ebenfalls satzweise vor, sodass sich dies anbietet.
Tsai et al. [23] haben ein Multimodal Transformer entworfen, welches sowohl auf **angeordnete** (aligned) als auch **unangeordnete** (unaligned) Modalitäten angewendet wurde. Hier bedeutet „angeordnet“, dass die Modalitäten Text, Akustik und Vision auf Wort-Level angeordnet werden, also dass hier für jedes Wort die Merkmale der drei Modalitäten extrahiert und verwendet werden. Ebenfalls haben sie **unangeordnete** Daten vorliegen und zu beiden Verfahren ein Modell entwickelt und diese verglichen. Bei den **angeordneten** Daten konnten sie eine leichte Verbesserung feststellen. Für diese Arbeit ist der Aufwand zur Aufbereitung von angeordneten Daten zu groß, sodass mit **unangeordneten** Modalitäten gearbeitet wurde.

2.3 Abgrenzung der Arbeit

Die vorgestellten Arbeiten, wie an den meisten Stellen markiert, haben in diesen Bereichen dieselben Techniken oder Methoden verwendet wie diese Arbeit. Kurz gefasst hat diese Arbeit die Datensätze **CMU-MOSI** und **CMU-MOSEI** verwendet, um drei Arten (**frühe**, **mittelspäte** und **späte Fusion**) von Deep Learning Modellen mit PyTorch zu trainieren, die als Eingabe Text-Modalität kodiert mit **BERT** und Audio-Modalität durch **openSMILE** extrahierte Audio-Merkmale erhält. Texte aus den nicht-annotierten Test-Videos wurden mit Hilfe von OpenAIs **Whisper** extrahiert. Die Kombination dieser Methoden ist neuartig und hat bisher nicht stattgefunden. Die Arbeit wäre ohne die vorherigen Forschungsarbeiten nicht möglich gewesen, da sie die Basis für die entwickelten Erkenntnisse und Ideen darstellen.

Kapitel 3

Konzeption

Dieses Kapitel versucht Fragen zu beantworten, wie die Eingabedateien in Form von Videos in Tensoren transformiert werden können, sodass die **Text**- und **Audiomodalitäten** verwendet werden können um mehrere Modelle zu trainieren, welche es möglich machen, eine Sentiment- oder Emotionsvorhersage zu tätigen. In den ersten Unterkapiteln werden zunächst die Datensätze genauer betrachtet, für welche sich in dieser Arbeit entschieden wurde und warum andere wiederum entfallen, und welche Schritte notwendig sind, um sie in ein einheitliches Format zu bringen. Danach wird zunächst die Vorbereitung der Daten aus Datensätzen und Videos mit Hilfe von Software untersucht, sodass diese in einem Format vorliegen, sodass ein Modell trainiert werden kann, damit es Vorhersagen für Sentiment- und Emotionswerte erzeugen kann. Abschließend wird der Trainingsprozess des PyTorch Modells vorgestellt, sowie eine Übersicht über mögliche Ausgaben der Vorhersagen.

3.1 Zielsetzung

Das Hauptziel dieser Bachelorarbeit besteht darin, drei Deep Learning Modelle für die Sentiment-Analyse und Emotionserkennung in Feedback-Videos zu entwickeln und zu trainieren. Dabei wird besonderes Augenmerk auf die Integration der Modalitäten **Text** und **Audio** gelegt, während der Modalität „Video“ und damit der „Gestik und Mimik“ keine Beachtung geschenkt wird.

3.1.1 Initiale Fragen

Um ein Deep Learning Modell effektiv trainieren zu können, ist ein umfangreicher und vielfältiger Datensatz erforderlich. Wird sich auf wenige

und monotone Daten beschränkt, besteht die Gefahr, dass das Modell überangepasst wird und nur für die spezifischen Daten der Domäne akkurat ist. Um diesem Problem entgegenzuwirken, ist es wichtig, einen breit gefächerten Datensatz zu verwenden, der eine Vielzahl von Variationen enthält. Nur so kann das Modell ein umfassendes Verständnis für die zugrunde liegenden Muster und Merkmale entwickeln.
Im Allgemeinen ist das richtige Datenformat für ein Deep Learning Modell eine ein- oder mehrdimensionale Matrix, die auch als Tensoren bezeichnet werden. Diese Eingabetensoren liegen als Gleitkommazahlen vor und nehmen in der Regel nur positive Werte ($x \in \mathbb{R}_0^+$) an.

3.2 Datensätze

Im Kapitel Einleitung wurde bereits beschrieben, welche Datensätze sich für das Trainieren der PyTorch-Modelle anbieten. Der **CMU-MOSI** Datensatz, publiziert im Jahr 2016, bietet 2199 gesprochene Sätze in Form von Film Review Videos von YouTube, bei denen der Text und das Sentiment (Wertebereich -3 bis +3) bekannt sind. In den Videos spricht der einzelne Sprecher frontal in die Kamera und dabei sind Mimik und Gestik ersichtlich. Der **CMU-MOSEI** Datensatz, publiziert im Jahr 2018, bietet dies ebenfalls, jedoch ist der Datensatz weitaus größer mit 23453 gesprochenen Sätzen und bietet zusätzlich einen Wert zwischen 0 und 3 für die Emotionen: **Happiness**, **Sadness**, **Anger**, **Disgust**, **Surprise**, **Fear**. Hierbei muss beachtet werden, dass sich die Emotionen nicht unbedingt gegenseitig ausschließen müssen. So hat beispielsweise der Satz „For some reason, as soon as they started teaching me German, I couldn't stop learning the language!“ und „Maybe I can say it was one of the worst days in my life, but it changed to one of the best days in my life later.“ einen Wert für „Happiness“ von +1 und „Sadness“ von +1, sodass also beide Emotionen vertreten sind. Ebenso wäre es ein Trugschluss zu behaupten, dass der Sentimentwert immer negativ ist, wenn der Wert von „Sadness“ positiv ist. Ein Beispiel aus dem Datensatz lautet „I have never had a student in my class as intelligent as your daughter.“, welches einen maximal positiven Sentimentwert von +3 und einen „Sadness“-Wert von +1 besitzt, was dadurch begründet werden kann, dass dies als Kompliment an die Tochter dient, jedoch auch Traurigkeit erweckt, weil sie die Erste mit solch einem Intelligenzlevel ist.
In dieser Arbeit wurde sich entschieden, mit diese Datensätzen zu arbeiten, denn **CMU-MOSI** [3, 5, 7, 8, 10, 11, 12, 13, 14, 15, 17, 21, 23, 22, 25, 28] und **CMU-MOSEI** [4, 5, 7, 8, 9, 12, 13, 14, 21, 22, 23, 29] wurden bereits in anderen wissenschaftlichen Arbeiten verwendet. Der Vorteil dieser beiden Datensätze ist, dass beide die Multimodalitäten von **Text**- und **Audiodaten** umfassen. Die Datensätze sind sehr groß und weit gefächert, enthalten viele verschiedene Sprecher, und verschiedene Meinungen und Themen vertreten

sind, welches einen Vorteil beim Trainieren des Modells bietet. Ein Nachteil ist, dass die Qualität der Annotation aus Gründen der Subjektivität variieren kann, ein weiterer dass der Datensatz nur die englische Sprache enthält und somit ein weiterer Datensatz erforderlich ist, um die Modelle für andere Sprachen trainieren zu können.
Im Kontrast zu diesen beiden Datensätzen sprechen mehrere verschiedene Sprecher in den Videos des **IEMOCAP** Datensatzes, und teilweise überlappen diese Sätze, sodass die Extraktion der Audio-Merkmale gestört sein kann. Es wird erwartet, dass in der Domäne der Software Feedback-Videos nur ein Sprecher zu sehen und zu hören ist. Hintergrundgeräusche sind jedoch nicht auszuschließen.
Der **MELD** Datensatz enthält Videos aus der Fernsehserie „Friends“, welches Lachkonserven verwendet, die in Szenen im Hintergrund abgespielt werden, wenn diese als lustig empfunden werden sollen. Hier kann es auch zu Überlappungen der Texte kommen, wenn ein Sprecher dem anderen in das Wort fällt.
Im **ACL20Challenge** Datensatz sind keine Rohdaten auffindbar gewesen, sondern nur Daten, bei denen bereits eine Vorbearbeitung ausgeführt wurde. Jedoch wurde nicht genau dokumentiert, in welchen Schritten sie zu diese Daten gekommen sind, sodass es im Kapitel Evaluation schwer sein wird, dieselbe Verarbeitung auf die Testdaten anzuwenden, um das selbe Format zu erreichen. Angeblich wurden die textuellen Merkmale mit Hilfe von **GloVe** umgewandelt, und die Audio-Merkmale mit Hilfe von **openSMILE** extrahiert, jedoch steht nicht dabei, welche **openSMILE** Konfigurationsdatei verwendet wurde. Da in dieser Arbeit **BERT** und nicht **GloVe** verwendet wird, ist dieser Datensatz nicht geeignet.
Aus diesen Gründen wurden die Datensätze **IEMOCAP**, **MELD** und **ACL20Challenge** nicht für das Trainieren der Modelle verwendet.

3.2.1 Strategie zur Emotionszuordnung

Bei Verwendung von zwei unterschiedlichen Datensätzen muss mindestens ein Datensatz so verändert werden, dass dieser sich dem anderen anpasst. Beispielsweise kann ein Datensatz, welches das Label „Happy“ verwendet, in das Label „Joy“ von dem anderen ersetzt werden, da ansonsten die zwei Labels von dem Modell als verschieden interpretiert werden, obwohl sie inhaltlich dasselbe aussagen.
Zur Umwandlung in einheitliche Emotionen kann das Modell von Shaver et al. [19] als Vorbild genommen werden, denn sie haben die Emotionen vielschichtig untersucht. Es wurde bereits in zahlreichen Studien und Forschungsprojekten verwendet und ist bekannt in den Geisteswissenschaften. Jedoch haben die Datensätze **CMU-MOSEI** und **IEMOCAP** jeweils ihr eigenes System geschaffen, welches in das Shaver Modell übertragen werden kann. Hier bietet es sich an, diese Werte in einheitliche Werte zu übertragen

wie in Tabelle 3.1 dargestellt.
Die Datensätze enthalten verschieden lange Sätze und es muss überlegt

Shaver	**CMU-MOSEI**	**IEMOCAP**
Love	-	-
Joy	Happiness	Happy Excited
Sadness	Sadness	Sadness
Anger	Anger	Anger Frustrated
Disgust	Disgust	Disgust
Surprise	Surprise	Surprise
Fear	Fear	Fear

Tabelle 3.1: Umwandlung der Emotionen der Datensätze in das Shaver Modell

werden, ob es Sinn hat, zu kurze oder zu lange Sätze herauszufiltern. So liegt die Vermutung nahe, dass ein langer, komplizierter und gut formulierter Satz eines Politikers eher selten in Feedback Videos auftauchen wird. Zu lange Sätze können schwer zu verstehen sein, oder eine Überfülle an Informationen enthalten, die es schwierig machen können, das Gesamtsentiment zu identifizieren. Die Wahrscheinlichkeit, dass ein Satz sowohl positive als auch negative Sentimentwerte wie auch verschiedene Emotionen enthält, steigt, je länger der Satz ist. Je nach Wahl des Transformators, welcher Wörter in Tensoren umwandelt, können auch längere Sätze Probleme bereiten, falls jedes Wort in ein Token umgewandelt wird und nur eine begrenzte Anzahl an Platz zur Verfügung steht. Die Maximallänge des Tensors ist begrenzt.
Bei zu kurzen Sätzen wiederum kann es sein, dass die Zeitstempel so gesetzt sind, sodass Ausschnitte von vorherigen oder nachfolgenden Sätzen inkludiert sind. Dadurch beziehen sich die Audio-Merkmale nicht nur auf den gesprochenen Satz, sondern beziehen zu viel Kontext mit ein, welcher gar nicht zum Gesprochenen gehört.
Weitere Emotionen aus dem IEMOCAP Datensatz, die in der Tabelle 3.1 nicht gelistet sind, sind mit „other“ und „xxx“ annotiert, bei denen sich die Emotionen nicht eindeutig zuordnen lassen. Diese werden beim Training des Emotionsmodells nicht verwendet, da sie nicht ausreichend spezifisch sind und somit keine klare Zuordnung ermöglichen.

3.3 Vorverarbeitung der Datensätze

Nachdem ein Datensatz gefunden wurde, muss er aufbereitet werden. Liegt er in Form von Videos (MP4-Dateien) vor, wird der gesprochene Text, idealerweise satzweise, extrahiert. Dies kann manuell erfolgen, welches

jedoch eine sehr zeitaufwändige Aufgabe ist. Eine Alternative bietet das Modell „Whisper“ von OpenAI, das Texte automatisch extrahieren und die Zeitstempel der Sätze bereitstellen kann. Die Zeitstempel sind wichtig für die Extraktion der Audio-Merkmale im nächsten Schritt.
Die extrahierten Texte müssen nun noch in Tensorform umgewandelt werden. Im Kapitel Verwandte Arbeiten wurde dazu bereits die Modelle **BERT** und **GloVe** vorgestellt. Diese können beide verwendet werden, um ein Wort in einen Vektor umzuwandeln. Dies kann dazu führen, dass ein Satz mit mehreren Wörtern einen Vektor der Länge 300 erzeugt, während ein Satz mit weniger Wörtern einen Vektor der Länge 100 produziert. Da das Modell nicht mit unterschiedlich langen Vektoren umgehen kann und die Größe des Tensors einmalig festgelegt wird, bleibt als beste Lösung, immer einen Vektor der Länge 300 anzunehmen. Bei kurzen Sätzen werden dann die fehlenden Stellen mit Nullen aufgefüllt. Diese Technik wird als **Zero-Padding** bezeichnet.
Nach Abschluss der Textmodalität wenden wir uns nun der Audio-Modalität zu. Wie in Kapitel Verwandte Arbeiten bereits angesprochen, stehen mehrere Tools zur Verfügung. Verschiedene wissenschaftliche Arbeiten haben bereits **openSMILE** verwendet, um Audio-Merkmale zu extrahieren. Die Software bietet eine große Anzahl von vorgefertigten Konfigurationsdateien, die den Einstieg erleichtern. In dieser Arbeit wurde ebenfalls **openSMILE** verwendet, da es benutzerfreundlich ist und auch in einem Docker-Image verwendet werden kann. Nach der Installation von **openSMILE** können die Audio-Merkmale einer gesamten Audio-Datei mit einem einfachen Befehl extrahiert werden. Es ist sogar möglich, nur einen Teil der Datei zu betrachten, indem die Parameter „-start <t in seconds>“ und „-end <t in seconds>“ verwendet werden. Alternativ kann die Software **FFmpeg** genutzt werden, um einen Teil aus einer WAV-Datei auszuschneiden und dann **openSMILE** auf die ausgeschnittene Datei anzuwenden. Nach Ausführung von **SMILExtract** wird eine CSV-Datei erhalten, die die extrahierten Audio-Merkmale enthält. Hierbei liegt jedes Mermal als Fließkommazahl vor. Die Auflistung der exakten Befehle befindet sich im Anhang openSMILE und FFmpeg Befehle.
Auf diese Weise liegen uns nun die Merkmale beider Modalitäten als Vektoren bzw. Tensoren vor. Diese werden gesammelt in eine **SQLite Datenbank** geschrieben, sodass die Daten in einem Format vorliegen, in welchem sie gefiltert und sortiert abrufbar sind.

3.4 Vorverarbeitung von Feedback-Videos

Das Ziel der Vorbearbeitung besteht darin, wie in der Abbildung 3.1 dargestellt, eine Eingabedatei in der Form eines Videos so zuzubereiten, dass sie in einem Format vorliegen um PyTorch Modelle trainieren zu können.

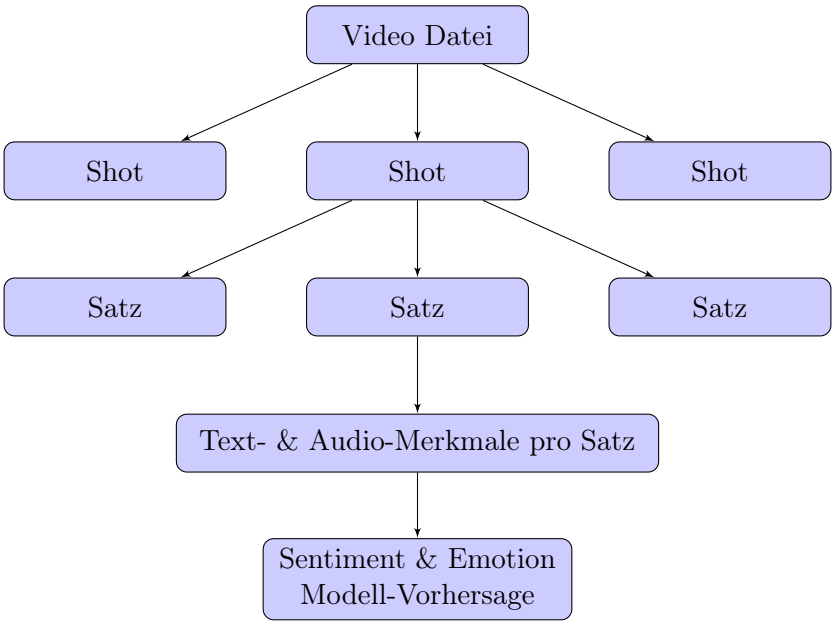

Abbildung 3.1: Prozessflussdiagramm zur Annotation der Sentiment- und Emotionswerte einer Video Datei

Diese Eingabedatei wird nun mit Hilfe von **PySceneDetect** auf Szenenwechsel analysiert, um die Datei in mehrere kleine Shots zu unterteilen. Hierbei wird der „ContentDetector“ mit einer Mindestszenenlänge verwendet, um zu verhindern, dass zu kurze Videoabschnitte extrahiert werden, die keine vollständigen Sätze enthalten. Dieser Prozess wird mehrere Male mit verschiedenen Schwellenwerten (Thresholds) wiederholt, denn es werden nur 1 Shot pro 3-4 Minuten Videolaufzeit angestrebt. Als Beispiel soll also ein Video mit einer Laufzeit von 20 Minuten in maximal 5 bis 6 Shots aufgeteilt werden.
Nun werden die erkannten Shots mit **FFmpeg** extrahiert, welche einige Sekunden bis einige Minuten lang sein können. Weil die Genauigkeit von **Whisper** ab etwa 10 Minuten stark nachlassen kann, findet diese Unterteilung statt, um dem Problem vorzubeugen. Außerdem soll jeder Satz einzeln auf Sentiment und Emotion analysiert werden.
Mit Hilfe von **Whisper** werden die Texte und deren Zeitstempel für jeden Satz extrahiert. Auf diese Weise liegen die Daten in dem Format vor, das den Datensätzen entspricht, und es kann mit den Methoden aus Kapitel 3.3 weitergearbeitet werden.

3.5 PyTorch Modell

In dieser Arbeit wurde sich entschieden, die Python Bibliothek **FastAI** zu verwenden, welche auf PyTorch aufbaut. Sie erleichtert die Erstellung und das Training eines Deep Learning Modells, indem sie im Hintergrund Tricks und Techniken anwendet.
Beispielsweise implementiert **FastAI** einen „Learning Rate Scheduler“, welcher das Training mit einer hohen Lernrate beginnen lässt, um diese nach und nach abzusenken, entweder bei jeder Epoche, oder wenn bestimmte Kriterien erreicht wurden. Eine konstante Lernrate kann suboptimal sein, denn eine zu hohe Lernrate kann zu instabilem Training führen, während eine zu niedrige Lernrate zu langsamer Konvergenz führen kann, welches mehr Epochen Training in Anspruch nimmt. Dies ist in dieser Arbeit von Vorteil, da damit das bestmögliche Modell in kurzer Zeit gefunden werden kann und auch in Zukunft das Training der Modelle beschleunigt.
Des Weiteren bietet **FastAI** einen „SaveModelCallback“-Hook, welcher automatisch das bisher beste erkannte Modell abspeichert. Während des Trainings kann man epochenweise den Trainingsverlust und Validierungsverlust beobachten. Der Trainingsverlust sagt aus, wie gut sich das Modell an die Trainingsdaten angepasst hat, während der Validierungsverlust bedeutet, wie gut das Modell Vorhersagen für Daten treffen kann, die es vorher noch nicht gesehen hat. Das beste bisher gefundene Modell ist daher nicht unbedingt solches mit dem geringsten Trainingsverlust, sondern solches mit dem geringsten Validierungsverlust, sofern eine gute Wahl beim Unterteilen des Datensatzes in Trainings- und Validierungsdaten getroffen wurde. Dies wird benötigt um sich weitere Arbeit zu ersparen, indem auf vorherige Evaluation verzichtet werden kann um schlechte Modelle herauszufiltern.
Nachdem die Eingabedaten, wie in Kapitel 3.3 beschrieben, in Tensoren umgewandelt wurden, kann mit dem Training des Modells begonnen werden. Der Eingabetensor nimmt eine Dimension von (1, 2351) an. Es bietet sich an mit einem kleinen Modell zu beginnen, welches nur ein verstecktes Layer besitzt um zu überprüfen, ob es überhaupt zu einem Resultat führt. Nach und nach werden dann weitere Layer hinzugefügt, bis man zu einem akzeptablen Ergebnis kommt.
Die Daten werden aus der in Kapitel 3.3 angelegten **Datenbank** ausgelesen. Falls es bei der Vorverarbeitung zu Fehlern kam (z.B. fehlende Texte, Sentimentwerte oder Audio-Merkmale), können diese mit einer **SELECT-Query** via **WHERE-Filter** übersprungen werden. Anschließend werden sie in ein **Pandas Dataframe** überführt, welche von der **FastAI** Bibliothek unterstützt wird.
Die Daten werden in den **Trainings**- und **Validierungs-Satz** aufgeteilt. Hierbei erfordert es gewisses Fingerspitzengefühl, wie sich entschieden wird die Aufteilung durchzuführen. Im Datensatz gibt es mehrere gesprochene Sätze, die zu einem Video bzw. zu einem Sprecher gehören. Es ist wahr-

scheinlich, dass das Modell einen Sentimentwert besser vorhersagen kann, wenn derselbe Sprecher bereits im Trainings-Satz von dem Modell gesehen wurde. Dies sollte vermieden werden, denn der **Validierungs-Satz** soll Daten darstellen, die dem Modell unbekannt sind. Daher wird in dieser Arbeit der Datensatz in Videos vom selben Sprecher gruppiert, sodass derselbe Sprecher nur in einem Datensatz vorkommt. Deswegen wird sich entschieden den Datensatz so aufzuteilen, dass dem **Trainings-Satz** 70% und dem **Validierungs-Satz** 30% der Daten zugeordnet werden. Dabei wird darauf geachtet, dass ein Sprecher nur in einem Satz vorkommt, damit er bei Verwendung des **Validierungs-Satzes** dem Modell unbekannt ist.
Für die Auswahl der Loss-Funktion, die im Training verwendet wird, kommen nur solche in Frage, die die Differenz zwischen dem vorhergesagten Sentimentwert und dem annotierten Sentimentwert berechnen. Zwei solcher Loss-Funktionen bieten sich an: das **Mean Squared Error** (MSE) und das **Mean Absolute Error** (MAE). Beide werden häufig in wissenschaftlichen Arbeiten verwendet. **MSE** wird genutzt, um stark auf Ausreißer zu reagieren, während **MAE** stärker auf Werte in der unmittelbaren Umgebung fokussiert. In dieser Arbeit wurde sich für **MSE** entschieden, da es bevorzugt wurde, dass das Modell auch für Ausreißer relativ genaue Vorhersagen liefert.

3.5.1 Modell Design

In den ersten Iterationen der Modelle wurde mit der openSMILE Konfiguration „IS13-ComParE“, welches einen Vektor mit einer Größe von 6373 extrahiert, für die Audio-Merkmale gearbeitet, jedoch musste schnell festgestellt werden, dass trotz variierender Lernrate nur in seltenen Fällen ein akzeptables Modell trainiert werden konnte. Erst nach dem Umstieg auf die Konfiguration „emobase2010“ mit einer Vektorgröße von 1583 können Modelle trainiert werden, die regelmäßig gute Resultate erbrachten, also wird damit das Training des Modells deterministisch.
Um herauszufinden und in Kapitel Evaluation vergleichen zu können, welches Fusionsmodell für diese Aufgabe am besten geeignet sei, werden Modelle mit **früher**, **mittelspäter** und **später Fusion** erstellt und trainiert. Diese Modellarten wurden bereits in Kapitel Verwandte Arbeiten vorgestellt. Bei dem Modell mit **später Fusion** wurde früh erkannt, dass schlechte **MSE** Werte zustande kommen, wenn die unimodalen Modelle erst im letzten Schritt zu einem Layer zusammengeführt werden, denn dann entsteht ein Mehrheitsvotum, bei dem eine Modalität eine höhere Gewichtung zugesprochen wird als der anderen. Es wurde also ein weiteres Layers vor dem Letzten eingefügt, um das Problem zu beheben. Mit diesem extra Layer kann das Modell selber entscheiden, mit welcher Gewichtung die Ergebnisse in das letzte Layer einfließen.
Die Abbildung 3.2 stellt vereinfacht dar, wie die Modelle vom Prinzip her aussehen. Hierbei liegen immer die Eingabedaten, also Text- und Audio-

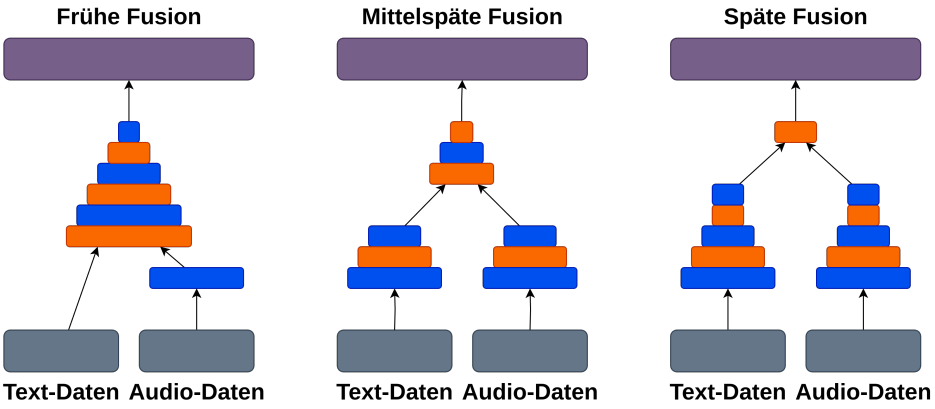

Abbildung 3.2: Modelle mit verschiedenen Arten von Fusionen

Daten, anfangs getrennt vor. Die Zusammenführung der Daten findet beim Modell mit **früher Fusion** in den ersten Layern statt, hingegen erst in den letzten Layern beim Modell mit **später Fusion**. Die Anzahl und Größe der Layers in der Abbildung entspricht dabei nicht der Realität, sondern soll nur die Fusion darstellen. Bei den Modellen dieser Arbeit handelt es sich dabei immer um Fully-connected Layers, bei denen sich die Zahl der Neuronen um den Faktor 2 bis 4 schrittweise verkleinert. Für diese Art der Konnektivität wurde sich entschieden, um dem Modell freie Wahl zu lassen, wie es sich über die Epochen hinweg entwickelt. Des Weiteren wurde sich dafür entschieden, die Layers der unterschiedlichen Modalitäten auf die gleiche Größe zu reduzieren, bevor diese zusammengeführt werden, damit jede Modalität etwa dieselbe Gewichtung auf das Endergebnis einfließen lässt.

3.5.2 Modell Vorhersage und Repräsentation

Nachdem das Modell trainiert wurde, kann über den Evaluationsmodus eine Vorhersage für einen Eingabetensor getroffen werden. Hierbei muss darauf geachtet werden, dass die Eingabedaten demselben Format entsprechen, welches zum Trainieren des Modells verwendet wurde. Es lohnt sich Helferfunktionen bereitzustellen, die einen ganzen Shot oder sogar ein komplettes Video als Eingabe erhalten um mehrere Vorhersagen zu treffen, welche anschließend in ein oder mehrere **Histogrammen** oder **Kuchendiagrammen** zusammengefasst werden. Anhand der Diagramme können die Sentimentwerte schnell abgelesen werden, sodass diese interpretiert und Aussagen über das Video getroffen werden können, ohne das komplette Video ansehen zu müssen.
Zum Trainieren der Modelle, wie in Kapitel 3.2 bereits besprochen, wurde der CMU-MOSI und CMU-MOSEI Datensatz verwendet. Beide enthalten

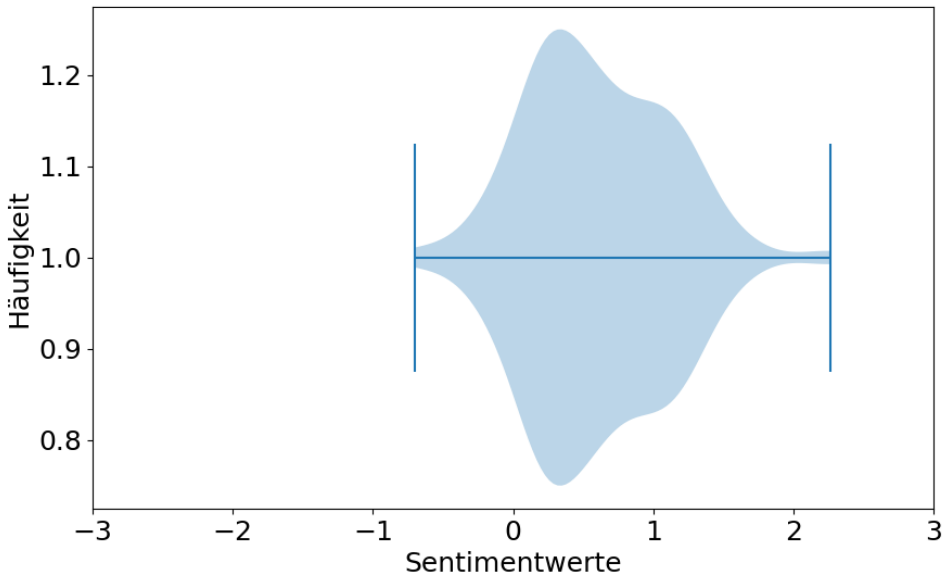

Abbildung 3.3: Violinplot mit Sentimentwerten eines Videos

Sentimentwerte von -3 bis +3, und somit befindet sich der vorhergesagte Wert des Modells ebenfalls zwischen -3 und +3. Ein Video mit von dem Modell vorhergesagten Sentimentwerten kann, wie in Abbildung 3.3 dargestellt, durch einen **Violinplot** dargestellt werden. Es können die Grenzen der Sentimentwerte der einzelnen Sätze anhand der senkrechten **blauen Striche** abgelesen werden, und anhand der **Dichte** lässt sich die Häufigkeit der Sentimentwerte auslesen. Da sich der Großteil auf der rechten Seite vom **Null-Wert** befindet, kann ein positives Gesamtsentiment erkannt werden.
In Kapitel 4.2 wird näher darauf eingegangen, wie diese Zahlenwerte Klassen zugeordnet werden können. Da ein **Violinplot** nicht sehr geläufig ist, können die Sentimentwerte eines Videos auch in anderen Diagrammen dargestellt werden, die kompakter und einfacher abzulesen sind. Abbildung 3.4 zeigt ein **Histogramm**, welches auf der linken Seite, ähnlich zum **Violinplot**, die negativen Sentimentwerte aufzeigt und auf der rechten Seite die positiven Sentimentwerte. Mit einer farblichen Kodierung lässt es sich noch schneller ablesen, ohne dass die Namen der Klassen gelesen werden müssen. Auch wenn Informationen im Gegensatz zum **Violinplot** verloren gegangen sind, so kann das Gesamtsentiment dennoch ebenso gut erkannt werden. Daneben in Abbildung 3.5 ist ein **Kuchendiagramm** dargestellt, in welcher der Anteil der Sentimentwerte zueinander dargestellt werden, während das **Histogramm** die absolute Anzahl an gesprochenen Sätzen an der Y-Achse aufzeigt.
Es kann geschlussfolgert beziehungsweise interpretiert werden, dass dieses

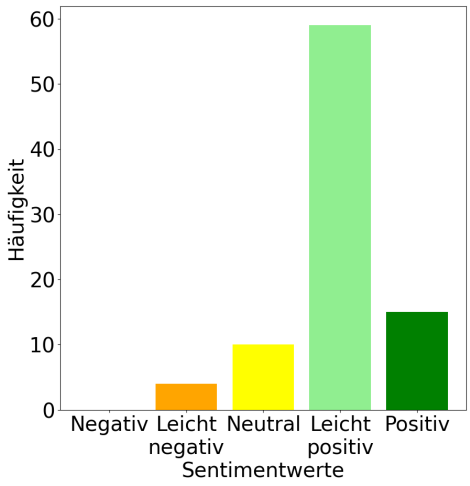

Abbildung 3.4: Histogramm mit Sentimentwerten eines Videos

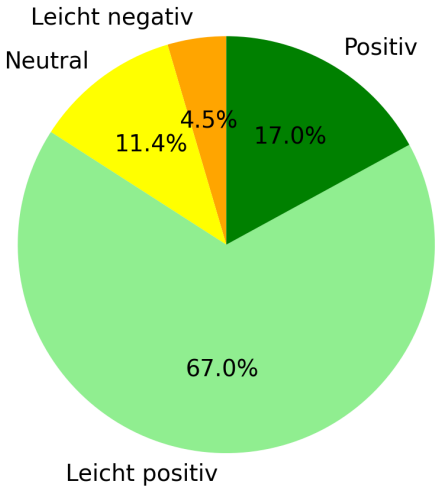

Abbildung 3.5: Kuchendiagramm mit Sentimentwerten eines Videos

Video für Softwareentwickler sehenswert ist, da es Informationen enthält, bei dem der Sprecher im allgemeinen sehr zufrieden mit dem Produkt ist und eventuell Vorschläge zur Verbesserung anbietet.
Falls ein **Diagramm** ausgewogene Sentiment- und Emotionswerte enthält, ist die Schlussfolgerung nicht eindeutig. Möglich ist, dass der Sprecher sehr vertraut mit dem Produkt ist und relativ neutral davon berichtet, jedoch keine starken Vorlieben oder Abneigungen aufzeigt. Für den Entwickler ist solch ein Video eher nicht sehenswert, da es keine starke Kritik an das Produkt zu enthalten scheint.
Ein weiteres hypothetisches **Histogramm**, gespiegelt zu Abbildung 3.4, enthält überwiegend negative Sentimentwerte. Mögliche Interpretationen von solchen Werten reichen von negativer Stimmung oder Unzufriedenheit bis hin zu Problemen, Beschwerden, Frustration oder gar Enttäuschung eines Produktes. Allgemein lässt sich hierbei sagen, dass schlechte Erfahrungen gemacht wurden oder bestimmte Erwartungen nicht erfüllt wurden. Es kann aber auch sein, dass sich diese Person noch nicht viel mit dem Produkt auseinandergesetzt hat und einen Ersteindruck des Produkts zeigt. Solch ein Feedback ist für das Entwicklerteam sehr sehenswert, da es aufzeigt, an welchen Stellen die Bedienung des Produkts nicht reibungslos oder verständlich ist.

Kapitel 4

Evaluation

Dieses Kapitel dient als Stütze für die Beantwortung der in Kapitel Einleitung vorgestellten Forschungsfrage und zur Untersuchung, wie akkurat und zuverlässig die trainierten PyTorch Modelle aus Kapitel Konzeption sind. Mit Hilfe von verschiedenen Metriken lässt sich ein Modell evaluieren, indem vom Modell vorhergesagte Werte mit echten Werten verglichen werden. Zuerst wird darauf eingegangen, wie die Testdaten von Menschen annotiert wurden. Anschließend werden Metriken vorgestellt, die in Frage kommen und es wird begründet für welche Metriken sich entschieden wurde. Danach werden die Ergebnisse der Modelle vorgestellt, indem die trainierten Modelle Werte vorhersagen und diese Werte mit von Menschen annotierten Werten verglichen werden. Im Anschluss werden die Modelle mit Hilfe der Metriken verglichen. Im letzten Abschnitt werden die Ergebnisse interpretiert.

4.1 Manuelles Labeln der Testdaten

Zur Bewertung des Modells werden weder der **Trainings-** noch der **Validierungs-Satz** aus Kapitel Konzeption verwendet, denn diese wurden bereits zum Trainieren des Modells benutzt. Zur Bewertung des Modells werden dazu weitere Daten verwendet, der sogenannte **Testdaten-Satz**, welcher zu Beginn beiseite gelegt wurde. Diese **Testdaten** können beliebige Videos sein, die sich in der Domäne von Software Feedback-Videos befinden. Hierbei wird das Video zuerst in Szenen oder Shots unterteilt, um anschließend mit Whisper jeden Shot in einzelne Sätze zu unterteilen. Dank Whisper wird nicht nur der Text extrahiert, sondern auch Zeitstempel, sodass für jeden Satz ein Minivideo aus dem Originalvideo extrahiert werden kann. Ausgewählte Personen können diese Minivideos anschauen und anschließend mit einem Sentiment- und Emotionswert versehen. Je mehr Menschen diese einzelnen Sätze mit diesen Werten annotieren, desto akkurater ist der Wert und desto besser kann überprüft werden, wie gut die

Vorhersagen des trainierten Modells sind. Gibt es einen kleinen Unterschied in der Annotation von Menschen, wird der Durchschnitt genommen. Falls es einen größeren Unterschied bei der Bewertung eines Satzes und somit Klärungsbedarf gibt, wird hierüber diskutiert und sich hinterher geeinigt. Wenn es zu keiner Einigung kommt, wird der Satz verworfen.

4.2 Überlegung zu den Metriken

Es ist überlegenswert, mit welchen Metriken die Ergebnisse verglichen werden können. Es bietet sich an die Metriken **Mean Squared Error** (**MSE**) und **Mean Absolute Error** (**MAE**) zu benutzen, welche verwendet werden um die durchschnittliche Abweichung zwischen den echten Zahlenwerten und denen vom Modell vorhergesagten Zahlenwerten berechnen zu können.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.1}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{4.2}$$

In Gleichung 4.1 kann man erkennen, dass bei **MSE** Ausreißer stärker ins Gewicht fallen als bei **MAE**, welche in Gleichung 4.2 beschrieben wird und Abweichungen eine lineare statt quadratische Gewichtung erhalten, je nach Betrag der Differenz zum echten Wert. Bei beiden Metriken ist ein kleiner Wert erstrebenswert, denn dieser sagt aus, wie groß die Fehler der Modelle gegenüber den echten Sentimentwerten sind.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{All Predictions}} \tag{4.3}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.4}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.5}$$

Des Weiteren bieten sich die Metriken **Genauigkeit**, **Precision**, **Recall** und **F1-Score** an, welche häufig Verwendung finden bei Vorhersagen mit Binärklassen. Beispielsweise bei medizinischen Tests, welche ein positives oder negatives Ergebnis anzeigen können, wobei ein positiver Schwangerschaftstest eines biologischen Mannes ein **False Positive** darstellt, und ein negativer Schwangerschaftstest einer eindeutig schwangeren Frau als **False Negative** gilt. Ein **True Positive** ist ein positives Testergebnis und ein **True Negative** ein negatives Testergebnis bei jeweils korrekter Vorhersage. Mit der **Genauigkeit**, beschrieben in Gleichung 4.3, werden die Anzahl aller richtig vorhergesagten Fälle beschrieben, während **Precision** von der Anzahl

der **True Positives** und **False Positives** abhängt, wie in Gleichung 4.4 beschrieben. Somit ist der **Precision**-Wert niedrig, wenn es weniger **True Positives** als **False Positives** gibt. Der Recall-Wert aus Gleichung 4.5 hängt von der Anzahl der **True Positives** und **False Negatives** ab und ist niedrig, wenn weniger **True Positives** als **False Negatives** vorhanden sind.
Anhand des Beispiels eines Email Spamfilters bedeutet ein hoher **Precision**-Wert, dass viele Spam-Emails korrekt als solche markiert werden, genannt **True Positives**, und nur wenig legitime Emails, genannt **False Positives**, als Spam markiert werden, jedoch bei einem niedrigen **Precision**-Wert werden viele legitime Emails als Spam markiert.
Bei einem Spamfilter mit hohem **Recall**-Wert werden viele Spam-Emails als solche markiert und nur wenige bleiben unerkannt, die **False Negatives**. Währenddessen werden bei einem niedrigen **Recall**-Wert viele Spam-Emails als legitime Emails eingestuft werden.
Ob ein hoher **Precision** oder hoher **Recall** Wert im Beispiel des Email Spamfilters angestrebt wird, so ist in diesem Fall ein hoher **Precision**-Wert bevorzugt, denn lieber wird manuell im Posteingang Spam aussortiert, als dass im Spam-Ordner nach legitimen Emails gesucht werden muss.
Die verwendeten Datensätze CMU-MOSI und CMU-MOSEI enthalten Sentimentwerte im Wertebereich -3 bis +3 und diese Werte springen in $\frac{1}{3}$-Abständen. Die Autoren der Datensätze haben in ihrer wissenschaftlichen Arbeit zu CMU-MOSEI beschrieben, dass der 0-Wert der Klasse **neutral** zugeordnet wird, Werte größer als 0 und kleiner-gleich 1 werden der Klasse **leicht positiv** zugeordnet, und alle Werte größer als 1 werden der Klasse **positiv** zugeordnet. Daraus kann abgelesen werden, dass die Klassen unterschiedliche Größen besitzen, denn die Klasse **neutral** enthält einen Wert (0), **leicht positiv** enthält drei Werte $(\frac{1}{3}, \frac{2}{3}, 1)$ und **positiv** enthält 6 Werte $(1\frac{1}{3}, 1\frac{2}{3}, 2, 2\frac{1}{3}, 2\frac{2}{3}, 3)$.
Eine Umwandlung von der Intervallskala mit kontinuierlichen Werten in diskrete Klassen auf einer Ordinalskala führt also zu Informationsverlust. Bei einem Text mit echtem Sentimentwert von 1.2 und vom Modell vorhergesagten Wert von 3 gibt es eine starke Abweichung mit einer Differenz von 1.8, jedoch befinden sich beide Werte in der selben Klasse **positiv**. Im Gegensatz dazu kann ein Text einen echten Sentimentwert von -0.2 besitzen und vom Modell den Wert 1.2 vorhergesagt bekommen. Diese Differenz von 1.4 ist zwar kleiner als im Beispiel davor, jedoch in Form von Klassen ist dies ein Unterschied von drei Klassen, denn der echte Wert bekommt die Klasse **leicht negativ** und der vorhergesagte Wert die Klasse **positiv** zugeordnet, dazwischen liegen die Klassen **neutral** und **leicht positiv**.
Aus diesem Grund sollte an dieser Stelle die Auswertung und Vergleiche der Modelle untereinander mit Hilfe von Klassen vermieden werden, und stattdessen werden die Metriken **MSE** und **MAE** benutzt, denn sonst entsteht bei der Umwandlung eine erhebliche Verzerrung der Daten, da feine

Unterschiede zwischen den tatsächlichen Sentimentwerten und den vorhergesagten Werten durch die Klassenbildung nicht angemessen abgebildet werden können. Die Unterteilung der Sentimentwerte in die fünf Klassen **negativ**, **leicht negativ**, **neutral**, **leicht positiv** und **positiv** hilft dennoch bei der Darstellung der Sentimentwerte in Diagrammen, wie in den Abbildungen 3.4 und 3.5 zu sehen ist.
Durch die Vermeidung der Verwendung von Klassen für die Analyse kann eine präzisere Bewertung der Modelle erreicht und potenzielle Verzerrungen minimiert werden, die durch die Diskretisierung der kontinuierlichen Sentimentwerte entstehen könnten.

4.3 Ergebnisse der Modelle

In der in Kapitel 4.1 vorgestellten Vorgehensweise werden 137 Sätze mit Sentimentwerten von Menschen versehen. Anschließend werden für diese Sätze die Sentimentwerte von allen drei trainierten Modellen vorhergesagt. Für jeden Satz liegt nun der echte Wert und drei von den Modellen vorhergesagte Werte vor, sodass hiermit die Metriken **MAE** und **MSE** für alle drei Modelle berechnet werden können. Diese Metriken werden mit den Formeln aus Kapitel 4.2 berechnet und sind in Tabelle 4.1 ersichtlich.

Tabelle 4.1: Klassifizierungsreport für das Testset

Modell	MAE	MSE
Frühe Fusion	0.888	1.457
Mittelspäte Fusion	0.908	1.468
Späte Fusion	0.845	1.409

Jeder dieser Werte kann, unter Informationsverlust wie bereits in Kapitel 4.2 genauer erläutert, einer der fünf Klassen zugeordnet werden, sodass diese in einem Histogramm dargestellt werden können.
Die Sentimentwerte von allen 137 Sätzen sind in Abbildung 4.1 dargestellt. Die X-Achse verläuft von links nach rechts vom negativen zum positiven Sentiment, während auf der Y-Achse die Häufigkeit der annotierten Sentimentwerte aufgetragen ist. Jede Klasse enthält vier Säulen, bei der die erste Säule die Häufigkeit der von Menschen annotierten Werte widerspiegelt, und anschließend die Häufigkeit der vorhergesagten Sentimentwerte von dem Modell der **frühen Fusion**, der **mittelspäten Fusion** und ganz rechts von der **späten Fusion**.
Aus dem Histogramm ist ersichtlich, dass es starke Übereinstimmung der Häufigkeiten bei den Klassen **leicht negativem** und **negativem** Sentiment von allen vier Annotationen gibt.

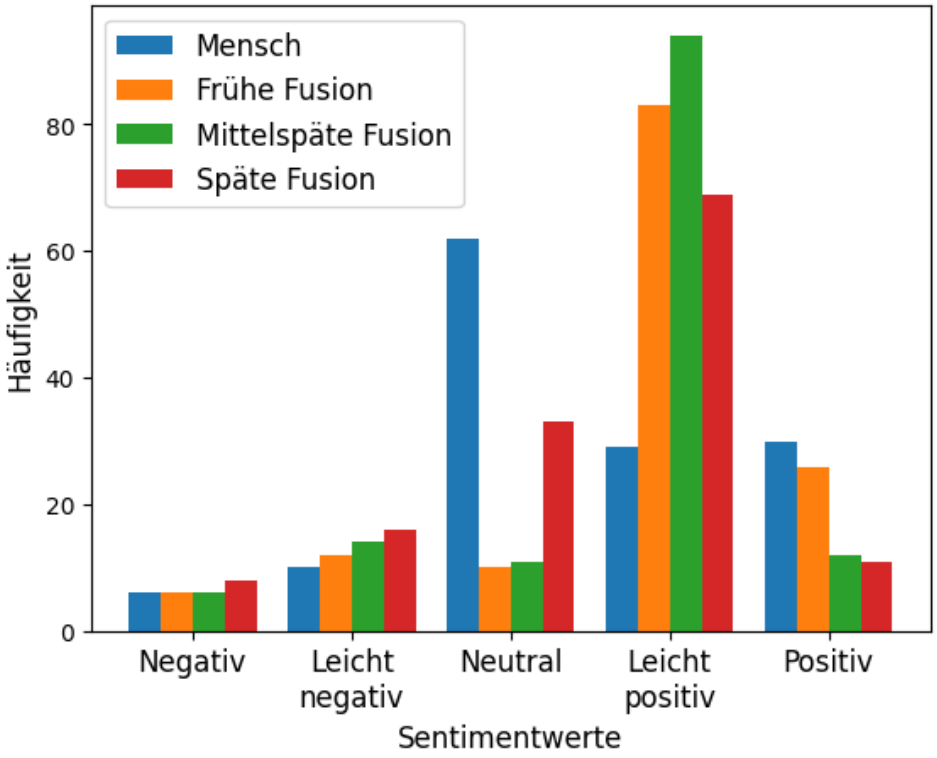

Abbildung 4.1: Darstellung der Sentimentwerte aus den Testvideos

4.4 Vergleich der Modelle untereinander

Die Ergebnisse zeigen, dass das Modell mit **später Fusion** die beste Leistung erzielt und mit den Werten **MAE** von 0.845 und **MSE** von 1.409, da diese am kleinsten sind. Da die Ergebnisse von allen Modellen nicht stark voneinander abweichen, lässt sich nicht eindeutig schlussfolgern, dass ein Modell signifikant überlegen ist gegenüber dem anderen. Es ist jedoch bemerkenswert, dass die **späte Fusion** eine leicht bessere Leistung erzielt als die anderen Modelle in Bezug auf den **MAE** und den **MSE**. Dies kann bedeuten, dass die späte Fusion eine leicht bessere Wahl für die Sentimentanalyse von Software Feedback-Videos mit Text- und Audio-Modalitäten ist. Weitere Untersuchungen sind jedoch erforderlich, um die Robustheit und Generalisierungsfähigkeit der Modelle zu validieren.
Anhand der Abbildung 4.1 lässt sich erkennen, dass die annotierten Sentimentwerte von Menschen übermäßig neutral sind, während die Modelle eher leicht positive Sentimentwerte voraussagen. Die Anzahl der **leicht negativen** und **negativen** Sentimentwerte ist sowohl bei den von Menschen annotierten als auch bei den von den Modellen vorhergesagten Werten fast identisch, wobei hier nicht ersichtlich ist, ob auch dieselben Sätze von den

Modellen in diesen beiden Klassen korrekt annotiert wurden. Hier müssen die Sätze einzeln untersucht werden.

4.5 Interpretation der Ergebnisse

Das Modell mit **früher Fusion** hat etwa gleich viele Sätze wie Menschen mit Sentiment **positiv**, also einem Sentimentwert von über 1, eingeordnet, während die anderen beiden Modelle nur etwa halb so viele Sätze diesem Wertebereich zugeordnet haben. Daraus kann geschlussfolgert werden, dass in den äußeren Sentiment-Klassen, also **negativ** und **positiv**, das Modell mit **früher Fusion** am besten die Sentimentwerte vorhersagen kann, zumindest nach der Häufigkeit der vorhergesagten Sentimentwerte.
Dabei kommt das Modell mit **später Fusion** der Häufigkeit von Menschen am nächsten bezüglich der Klasse **neutral**, ist jedoch immer noch weit entfernt. Wie in Kapitel 4.2 angesprochen kann es daran liegen, dass der Wertebereich für die Klasse **neutral** sehr klein ist und eine kleine Abweichung bereits einer benachbarten Klasse zugeordnet wird. Während des Trainings wurde die Loss-Funktion verwendet, die den **MSE** so gut wie möglich verringert. Es wurde beim Training also darauf geachtet, dass die Fehler der vorhergesagten Sentimentwerte so klein wie möglich werden und dabei wurde verstärkt auf die Ausreißer geachtet.
Obwohl die Trainingsdaten eine andere Domäne besaßen als die Testdaten, konnte dennoch eine relativ gute Vorhersage für die Sentimentwerte getroffen werden.

Kapitel 5

Diskussion

In diesem Kapitel geht es um die Beantwortung der in Kapitel Einleitung vorgestellten Forschungsfrage. Anschließend werden Möglichkeiten aufgezeigt, welche weiteren Methoden oder Verfahren sich anbieten, um ein besseres Ergebnis erzielen zu können. Im Anschluss werden Probleme mit den Werkzeugen aufgezeigt, die während der Ausarbeitung aufgetreten sind. Darauf folgt ein Abschnitt über einige mögliche Faktoren, die den Sentimentwert beeinflussen. Zum Abschluss werden die Anwendungsmöglichkeiten der Automatischen Sentiment-Analyse und Emotionserkennung aufgezeigt.
Die in Kapitel Einleitung gestellte Forschungsfrage

Zentrale Forschungsfrage

Welche Arten von Deep Learning Modellen eignen sich am besten für die Sentiment-Analyse und Emotionserkennung von Software Feedback-Videos?

kann nur bedingt beantwortet werden, da in dieser Arbeit nicht alle Möglichkeiten ausgeschöpft wurden. Im Kapitel Evaluation wurden bereits die Ergebnisse vorgestellt und interpretiert, welche helfen, eine Antwort auf die Forschungsfrage zu formulieren:

Beantwortung der zentralen Forschungsfrage

Es konnte gezeigt werden, dass von den drei verschiedenen Arten von Modellen das Modell mit **später Fusion** die besten Ergebnisse in Bezug zu Sentiment-Analyse erbringen konnte.

Jede Art von Modellen wurde mehrmals trainiert, wobei ab etwa acht Epochen keine bemerkenswerte Verbesserung beobachtet werden konnte. Stattdessen wurden die drei Modellarten **frühe Fusion**, **mittelspäte Fusion** und **später Fusion** etwa 30 mal mit unterschiedlichen Startbedingungen trainiert. Hiermit wurde dem Zufall so wenig wie möglich Spielraum gelassen,

falls dieser ein Faktor darin gespielt hat, dass das Modell mit **später Fusion** leicht bessere Leistung erzielt hat.

5.1 Reflexion

Um bessere Resultate zu erzielen, gibt es eine Vielzahl an Möglichkeiten, die man ausschöpfen kann, bis der Rat von Experten erforderlich ist. In Kapitel Verwandte Arbeiten werden bereits weitere Methoden und Verfahren aufgelistet. Bevor der Datensatz noch weiter aufbereitet wird, kann die Anzahl und Größe der Modell Layers verändert werden, eine andere Aufteilung von Trainings- und Validierungs-Satz vorgenommen werden, weitere Text-Transformationen wie zum Beispiel GloVe benutzt werden, oder andere openSMILE Konfigurationen zur Extraktion der Audio-Merkmale eingesetzt werden.
Erst wenn dies keinen Erfolg erzielt, können die Eingabedaten in andere Formate gebracht werden, da dies einen erhöhten Zeitaufwand erfordert. So können einzelne Wörter betrachtet und deren Audio-Merkmale wortweise extrahiert werden, welches den Vorteil besitzt, dass jedes Wort auf Lautstärke und andere Eigenschaften untersucht wird im Gegensatz zum Durchschnitt oder Maximum der Lautstärke des gesamten Satzes. Allerdings kann dies zu einer Vergrößerung des Eingabetensors führen, welches die Trainingszeit erhöht.
Statt **MSE** oder **MAE** als Loss-Funktion beim Training zu verwenden, kann auch Genauigkeit verwendet werden. Dies kann getan werden, wenn nicht das Ziel ist, das Modell mit den geringsten Fehlern in Form von Differenzen zu den echten Sentimentwerten zu finden, sondern das Modell mit der besten Genauigkeit für die Voraussage der Sentimentklassen **negativ**, **leicht negativ**, **neutral**, **leicht positiv** und **positiv**. Dazu müssen vor dem Training die Label der Datensätze von kontinuierlichen Zahlenwerten in diese Klassen umgewandelt werden.

5.2 Zuverlässigkeit der benutzten Werkzeuge

PySceneDetect wird zum Unterteilen der Videos in kleinere Shots verwendet. Es kann passieren, dass zu kurze Shots entdeckt werden, die **keine** gesprochenen Sätze enthalten, zum Beispiel beim **Intro** oder **Outro**, bei denen es zu starken farblichen Übergängen führt und diese als einzelne Shots erkannt werden. Dies ist ein Problem, da letztendlich jeder Shot eine Zusammenfassung an Sentiment- und Emotionswerten in einem Histogramm dargestellt bekommen soll. Jedoch kann kein Wert erkannt werden, da das Modell sowohl Text- als auch Audiomodalitäten erfordet, um eine Vorhersage treffen zu können, jedoch fehlt in diesem Fall beides.
Falls PySceneDetect zu **lange** Shots entdeckt kann es zu Problemen

mit **Whisper** führen, da es bei längeren Shots oftmals halluziniert, also bereits Gesprochenes wiederholt oder Wörter und Zeitstempelgrenzen stark verschiebt. Dies ist von Nachteil, denn es wirkt sich auf die vorhergesagten Werte des Modells aus. Diese sind dann schlichtweg unvollständig oder sogar falsch.
Wenn Sprecher „ohne Punkt und Komma“ reden, also wenn der Redefluss rasant ist und sie **keine Pause** zwischen den Sätzen machen, ist in Verwendung mit **Whisper** aufgefallen, dass diese Software keine Satzendezeichen einfügt und es damit zu extrahierten Sätzen kommt, die mehr als 100 Wörter lang sind. Dies ist falsch und somit kann es passieren, dass für einen Shot nur ein Sentimentwert erkannt wird, da in diesem Shot nur ein Satz durch Whisper erkannt wurde.
Während der Erstellung dieser Arbeit wurde bemerkt, dass die Zeitstempel von **Whisper nicht immer akkurat** sind. Dabei kann es Abweichungen von wenigen Millisekunden bis hin zu mehreren Sekunden geben. Dies hat Einfluss auf die Extraktion der Audiomerkmale, welches bei kurzen Sätzen mehr ins Gewicht fällt als bei langen Sätzen, da der Anteil der Hintergrundgeräusche größer ist.

5.3 Einflussfaktoren der Sentiment-Analyse

Sprache ist nicht leicht und nicht immer eindeutig. Man merkt es in Online-Chaträumen, in denen nur der Text des Gegenübers gelesen werden kann. **Sarkasmus**, **Ironie** und Benutzung **zweideutiger Wörter** kann zu Verwirrung führen, welches den Gesprächsfluss behindert. Mit Zuhilfenahme des Kontextes kann eine Vermutung angestellt werden, was gemeint ist, aber noch mehr würde die Betonung oder sogar Mimik und Gestik aushelfen. Dieselben Probleme wird auch ein PyTorch Modell haben und auf die eine oder andere Art behandeln müssen.
Personen von verschiedenen Kulturen können unterschiedliche Interpretationen für dieselben Wörter haben. So kann der Satz „Das ist ein interessanter Ansatz.“ als Zustimmung oder Lob für eine innovative Idee interpretiert werden, während es in anderen Personengruppen oder Umständen als abwertend oder mindestens skeptisch verstanden werden kann. Die Modalität „Video“ mit der Interpretation von Mimik und Gestik kann dabei aufschlussreich sein. Ebenfalls kann „direkte Kommunikation“ als Ehrlichkeit und Offenheit eingeschätzt werden, oder aber auch negativ im Sinne von Grobheit oder Respektlosigkeit, sodass in einigen Situationen ein oder mehrere indirekte Hinweise gegeben werden, anstatt direkt auf ein Problem hinzuweisen.
In dem Fall, dass Sprecher sich gegenseitig in das Wort des anderen Fallen, also zeitgleich sprechen, sodass die Audio-Merkmale weder dem einen noch dem anderen Text zugeordnet werden können sondern womöglich beiden, bedeutet dies, dass das Deep Learning Modell Probleme haben wird, das

Sentiment über die Audio-Modalität akkurat zu erfassen. Bei zu starken Hintergrundgeräuschen wird die Audio-Modalität ebenfalls gestört, sodass die Audio-Merkmale fälschlich den Sprechern zugeordnet wird. Wie in Kapitel 5.2 bereits angedeutet, kann Whisper die Zeitstempel bei den erkannten Wörtern oder Sätzen zu weit fassen. Bei solch Variation in der Vergabe der Zeitstempel, muss das Modell lernen damit umzugehen und kann sich als Nachteil erweisen.

5.4 Praktische Anwendung und Implikationen

In den folgenden Abschnitten werden Beispiele aufgeführt, in denen es möglich und meistens sinnvoll ist, Sentiment-Analyse und Emotionserkennung einzuführen.

5.4.1 In der Softwareentwicklung

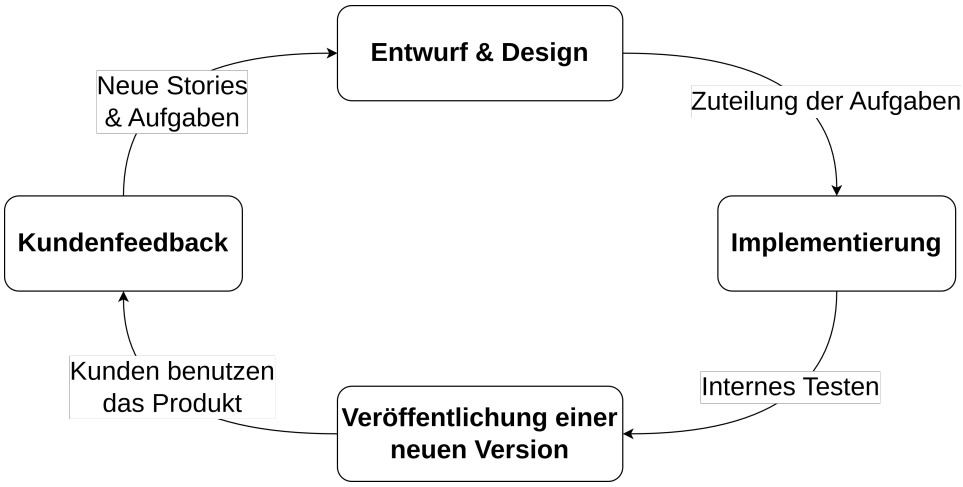

Abbildung 5.1: Ablauf von Softwareentwicklung

In der Softwareentwicklung durchläuft der Entwicklungsprozess mehrere Phasen, beginnend mit der Planung und Erstellung eines Prototyps, der von den Entwicklern überprüft und getestet wird, bevor er dem Kunden präsentiert wird. Diese frühe Phase enthält starken Bias der Entwickler gegenüber der Software aufgrund ihres Wissens und der Erfahrung bezüglich der Nutzung der Software. Im Gegensatz dazu steht der Kunde als neuer Benutzer, der sich erst mit der Software vertraut machen muss.
Die Anfangsphase der Software-Nutzung ist entscheidend für das Verständnis des Benutzerverhaltens und der Lernkurve. Hier kann bereits ermittelt werden, wie leicht der Benutzer mit der Software zurechtkommt und welche

Aspekte noch verbessert werden müssen, um eine optimale Benutzererfahrung zu gewährleisten. Das **Feedback** des Kunden spielt eine entscheidende Rolle in diesem Prozess, da es dem Entwicklungsteam Einblicke darüber gibt, welche Funktionen verbessert, hinzugefügt oder sogar entfernt werden sollten.
Durch das **Kundenfeedback** entstehen neue Aufgaben für das Entwicklerteam, meist in Form von Jira-Tasks, welche im Zusammenhang mit anderen Aufgaben weitere Funktionen hinzufügen oder verbessern sollen. Diese **Feedbackschleife** ermöglicht es, die Software kontinuierlich auf die Bedürfnisse der Benutzer anzupassen und zu verbessern.
Die automatische Sentiment-Analyse und Emotionserkennung in Software **Feedback**-Videos können diesen Prozess unterstützen, indem sie zusätzliche Einblicke in die Gefühle und Emotionen der Benutzer liefern. Abbildung 5.1 zeigt abstrakt den iterativen Prozess der Softwareentwicklung, und unter Zuhilfenahme der automatischen Sentiment-Analyse kann der Zeitaufwand zur Auswertung des **Kundenfeedback** stark reduziert werden.
Ein **Feedbackvideo** kann so aussehen, dass ein Benutzer seine Interaktion mit der Software aufzeichnet und während der Benutzung kommentiert. Anschließend wird das Video entweder dem Entwicklerteam direkt zugesendet oder sogar als **Review-Video** auf YouTube hochgeladen. In beiden Fällen kann die automatische Sentiment-Analyse und Emotionserkennung erfolgen. Die Emotionen des Benutzers können Freude, Frustration, Langeweile oder Verwirrung repräsentieren, welche es ermöglichen, das Entwicklerteam nicht nur über explizite Probleme oder Schwierigkeiten von Benutzern zu informieren, sondern auch die zugrunde liegenden emotionalen Zustände, die diese Reaktionen ausgelöst haben. Durch die zusätzliche Betrachtung von Bildschirmaufnahmen können die Entwickler den exakten Zeitpunkt ausfindig machen, an welchen Stellen der Benutzer Schwierigkeiten hat oder besonders viel Freude empfindet.
Anstelle von **Kundenfeedback**, wie „Ich mag die Software nicht.“, erhalten die Entwickler nun sehr detailliertes **Feedback** und können den Kontext mit einbeziehen, wie zum Beispiel ein Nutzer zu einer Stelle in einer Mobile App navigiert, bei der er Schwierigkeiten hat, wieder herauszukommen, also den „Zurück-Button“ zu finden. Der Benutzer wird an dieser Stelle frustriert oder verwirrt sein und im schlimmsten Fall aggressiv werden.

5.4.2 Automatische Sentiment-Analyse in Bildungseinrichtungen

Mittels Sentiment-Analyse in Schulen kann Schülerfeedback und Evaluierung von Lehrkräften automatisch ausgewertet werden, welches ermöglicht, Einblicke in die Zufriedenheit der Schüler und die Wirksamkeit der Lehrmethoden zu ermitteln.
Durch die Anwendung von Sentiment-Analyse und Emotionserkennung

lassen sich problematische Situationen, die Stress, Angst oder Frustration auslösen, frühzeitig erkennen. Durch diese Identifikation können Unterstützungssysteme Schüler bei der Bewältigung von Herausforderungen helfen und das Wohlbefinden verbessern, was wiederum das Lernumfeld insgesamt zum Positiven bewegen kann.
Dem gegenüber steht die Herausforderung solche Systeme in Bildungseinrichtungen zu integrieren, denn dafür müssen Lehrkräfte umfassend dafür geschult werden und die Administratoren diese Systeme implementieren. Wichtig ist es, die Vorteile und Potenziale dieser Technologie zu kommunizieren und sicherzustellen, dass die Beteiligten diese akzeptieren und nutzen. Die Akzeptanz dieser Technologie kann durch Misstrauen gegenüber dem Datenschutz und der Ethik verlangsamt werden. Es ist von entscheidender Bedeutung sicherzustellen, dass die Privatsphäre sowohl der Schüler als auch der Lehrkräfte geschützt wird. Es besteht die Gefahr, dass diese sensiblen Daten für unethische Zwecke missbraucht werden könnten. Aus diesem Grund sollten Maßnahmen ergriffen werden, um sicherzustellen, dass die Daten entweder schnell gelöscht werden, nur einer begrenzten Gruppe zur Verfügung stehen oder verschlüsselt auf einem Server gespeichert sind, der für Unbefugte nicht zugänglich ist.

5.4.3 Sentiment-Analyse und Emotionserkennung bei der Automatisierung von Entscheidungsprozessen

Ein Gegenbeispiel, vor dem so oft gewarnt wird und bereits in Filmen Horrorszenarien vorgestellt wurden, ist die Integration von Sentiment-Analyse und Emotionserkennung in Entscheidungsprozessen.
Diese Integration kann im Finanzsektor missbraucht werden, indem eine künstliche Intelligenz (KI) die Kreditwürdigkeit von Kunden durch Sentiment- und Emotionsanalysen bewertet. Falsche Bewertungen können zu unfairer Ablehnung von Kreditanträgen führen.
Im Strafjustizsystem kann ein KI System das Risiko von Rückfällen bewerten oder die Schwere von Straftaten bestimmen, wobei hier eine schlechte Einschätzung zu ungerechten Haftstrafen oder Diskriminierung führt.
Im Militär oder bei der Polizei können bei der Ersetzung des Personals durch KI-Roboter die Situationen ebenfalls falsch eingeschätzt werden, wodurch es zu Diskriminierung oder im schlimmsten Fall zum Einsatz von Waffen und zu lebensbedrohlichen Situation kommen kann.
Durch mangelnde Transparenz in der Analyse der Situation und im darauf folgenden Entscheidungsprozess wird es schwierig sein, rückwirkend die Entscheidung der KI begründen zu können, da Menschen und KI unterschiedliche Sichtweisen auf dieselbe Situation haben können.
Dieses Risiko wird reduziert, wenn Richtlinien und Datenschutzgesetze bei dem Entscheidungsprozess eingehalten werden, sowie Datenerfassung und -speicherung minimiert werden. Transparenz sollte in den Vordergrund

rücken, indem der Entscheidungsprozess nachvollziehbar werden durch Offenlegung von Algorithmen, Datenquellen und Entscheidungskriterien sowie die Erklärung, wie die Ergebnisse verwendet werden.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Diese Arbeit fokussierte sich auf die Entwicklung von Deep Learning Modellen, die in der Lage sind, Sentimente und Emotionen aus einzelnen Sätzen mit Hilfe der Modalitäten Text und Audio vorherzusagen, um die Auswertung von Software Feedback-Videos zu unterstützen, welche zeit- und kostenaufwändig sein kann.
Es wurden drei verschiedene Modellarten, die sich von ihrem Design voneinander unterschieden, mit den CMU-MOSI und CMU-MOSEI Datensätzen trainiert. Für die Transformation von Text in Tensoren wurde BERT [27] verwendet, und für die Extraktion der Audio-Merkmale wurde die Open Source Lösung openSMILE benutzt. Die Ausgabe des Sentiment-Modells liefert einen Wert im Wertebereich von -3 bis +3, welcher zur Auswertung und Darstellung in Diagrammen den fünf Klassen zugeordnet werden: **negativ**, **leicht negativ**, **neutral**, **leicht positiv**, **positiv**. Ein Modell für die Emotionserkennung wurde nicht trainiert und evaluiert.
Die Evaluation zeigte, dass das Modell der **späten Fusion** den anderen Modellen leicht überlegen war und die besten Ergebnisse bei der Sentiment-Analyse mit einem **Mean Absolute Error** Wert von 1.409 und **Mean Squared Error** Wert von 0.845 erzielte. Diese Abweichungen sind zwar verbesserungswürdig aber immerhin so klein, dass das Modell eine automatisierte Unterstützung zur Auswertung von in Feedback-Videos sein kann.

6.2 Ausblick

Wie in Kapitel Zusammenfassung erwähnt, konnte in dieser Arbeit kein Modell für Emotionserkennung trainiert werden. In dem CMU-MOSEI Datensatz gibt es für jeden Satz mehrere Emotionen, die jeweils einen Wert

von 0 bis 3 zugeordnet bekommen. Dabei kann es passieren, dass scheinbar widersprüchliche Emotionen wie „joy“ und „sadness“ stark repräsentiert sind. Dies lässt sich dadurch erklären, dass ein Teil des Satzes „joy“ und ein anderer „sadness“ widerspiegelt. Ein ähnliches Phänomen kann auch bei der Sentiment-Analyse eintreten, dass ein Teil ein negatives und ein anderer Teil ein positives Sentiment annehmen kann, wobei hier die Gewichtung der Satzteile nicht eindeutig ist, sodass man eventuell den Durchschnittswert verwendet, was aber nicht akkurat ist. Um die Emotionserkennung zu erleichtern, könnte jedem Satz nur eine Emotion zugeordnet werden, und diese können via One-Hot-Encoding kodiert werden. Im CMU-MOSEI Datensatz fehlt jedoch die neutrale Emotion, die ebenfalls repräsentiert werden sollte. Dies ist der Fall, wenn alle Emotionswerte unter einen bestimmten Schwellenwert liegen, denn dann ist keine Emotion überaus herausragend vertreten.
In Kapitel 5.1 wurden bereits weitere Möglichkeiten angesprochen, die untersucht werden können, um ein besseres Ergebnis der Deep Learning Modelle zu erzielen.
Eine solche Möglichkeit wäre es einen anderen Text Transformator zu wählen. Die Vermutung liegt nahe, dass es in Zukunft einen idealen Transformator für Sentiment und einen separaten Transformator für Emotionserkennung geben wird, sodass mit diesen ein besseres Ergebnis erzielt werden kann. Da BERT und GloVe Transformatoren häufig in aktuellen wissenschaftlichen Arbeiten verwendet werden, scheinen diese aktuell die beste Wahl für Sentiment-Analyse und Emotionserkennung zu sein.
OpenSMILE Konfigurationen bieten ein weites Spektrum an Audio-Merkmalen an, die aus Video- und Audiodateien extrahiert werden können. Je mehr Merkmale extrahiert werden, desto besser, jedoch sind einige Merkmale wichtiger als andere. Und ebenso wie im letzten Abschnitt, sind vermutlich auch einige Merkmale wichtiger für die Sentiment-Analyse und andere wiederum wichtiger für die Emotionserkennung.
Des weiteren kann zusätzlich zu den Modalitäten Text und Audio auch noch die Modalität Video zur Erkennung von Mimik und Gestik verwendet werden. Einige der im Kapitel Verwandte Arbeiten vorgestellten Arbeiten haben dies bereits mit verschiedenen Techniken erfolgreich einsetzen können und eine leichte Verbesserung in der Vorhersage von Sentimentwerten ermöglicht.
Es gibt also noch viele offene Fragen in diesem Gebiet, die beantwortet werden können, und viele weitere Möglichkeiten die erkundet werden können, welche wiederum neue Fragen aufwerfen. Diese Arbeit hat mit Hilfe weiterer Software eine Grundlage geschaffen, mit welcher Sentiment und Emotion aus Feedback-Videos extrahiert werden kann, um so einen Überblick über das Sentiment und Emotionen eines solchen Videos zu erhalten.

Anhang A

openSMILE und FFmpeg Befehle

Mit dem folgenden Befehl lassen sich die Audio-Features mit der Software openSMILE aus einer WAV-Datei extrahieren:

```
SMILExtract –C myconfig.conf –I input.wav –O output.csv
```

Mit diesem Befehl lässt sich ein Ausschnitt aus einer WAV-Datei mit Hilfe von FFmpeg extrahieren, wenn die Start- und Endzeitstempel bekannt sind. Somit lassen sich dann die Audio-Features eines Satzes aus einem Gesamtvideo extrahieren:

```
ffmpeg -ss <start> -to <end> -i input.wav output.wav
```

Zur Umwandlung einer Videodatei in eine Audio Datei, lässt sich der folgende Befehl anwenden. Dateiendungen können beliebig angepasst werden, solange diese FFmpeg bekannt ist:

```
ffmpeg -i input.mp4 output.wav
```

Anhang B

Struktur der CD

Die beigefügte CD oder DVD enthält die folgende Struktur.
Im **Hauptverzeichnis** liegen verschiedene Shell-Skripte zum Erstellen von Docker Images und zum Erstellen von Docker Containern. Mit dem **docker_process_datasets.sh** Skript kann die Vorverarbeitung der Datensätze ausgeführt werden. Mit **docker_train_model.sh** kann der Trainingsprozess begonnen werden, welcher je nach System mehrere Minuten bis mehrere Stunden pro Epoche erfordert. Mit **docker_process_test_data.sh** können die Testdaten in ein Format geladen werden, sodass mit der **interactive.ipynb** Datei Vorhersagen der Sentimentwerte getroffen werden können.
Die **Dockerfile.train** enthält die Anleitung zum Bauen eines solchen Docker Images. Die **.env** Dateien enthalten Konfigurations-Strings, um bestimmte Einstellungen vorzunehmen.
Weitere Dateien im Hauptverzeichnis sind die verwendeten Bibliotheken, die mit Python verwendet wurden.
Dies steht auch nochmals detaillierter in der **README.md** Datei.
Im Ordner **data** landen die SQLite-Datenbank Dateien und trainierten Modelle, sowie andere temporäre Dateien.
Im Ordner **preprocess_datasets** liegen Python Skripte zur Vorbearbeitung des vorliegenden CMU-MOSI, CMU-MOSEI und IEMOCAP Datensatz, und ebenso ein Skript welches die Datenbank-Struktur beschreibt, welches zum Trainieren verwendet wird.
Der Ordner **model** beinhaltet Python Dateien, die zum Trainieren des Modells benötigt werden, sowie ein weiteres Skript für eine weitere Datenbank, die zum Evaluieren verwendet wird.
Der Ordner **evaluation** enthält Helfer-Skripte zur Evaluation und zur Umwandlung der Modelle von **.pth** ins **.pkl**-Format, sodass die Modelle zur Evaluation verwendet werden können.
Im Ordner **other** liegen weitere allgemeine Helfer-Skripte, die von den anderen Skripten verwendet werden.

Literaturverzeichnis

[1] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246. Association for Computational Linguistics.

[2] L. Cai, Y. Hu, J. Dong, and S. Zhou. Audio-textual emotion recognition based on improved neural networks. 2019:1–9.

[3] M. Chen, S. Wang, P. P. Liang, T. Baltrušaitis, A. Zadeh, and L.-P. Morency. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 163–171. ACM.

[4] O. S. Chlapanis, G. Paraskevopoulos, and A. Potamianos. Adapted multimodal bert with layer-wise fusion for sentiment analysis. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

[5] Y. Fu, Z. Zhang, R. Yang, and C. Yao. Hybrid cross-modal interaction learning for multimodal sentiment analysis. 571:127201.

[6] A. Ghorbanali and M. K. Sohrabi. A comprehensive survey on deep learning-based approaches for multimodal sentiment analysis. 56:1479–1512.

[7] D. Ghosal, M. S. Akhtar, D. Chauhan, S. Poria, A. Ekbal, and P. Bhattacharyya. Contextual inter-modal attention for multi-modal sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3454–3466. Association for Computational Linguistics.

[8] D. Jiang, H. Liu, R. Wei, and G. Tu. CSAT-FTCN: A fuzzy-oriented model with contextual self-attention network for multimodal emotion recognition. 15(3):1082–1091.

[9] A. Khalane, R. Makwana, T. Shaikh, and A. Ullah. Evaluating significant features in context-aware multimodal emotion recognition with xai methods. page e13403.

[10] P. P. Liang, Z. Liu, A. Zadeh, and L.-P. Morency. Multimodal language analysis with recurrent multistage fusion. Publisher: arXiv Version Number: 1.

[11] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency. Efficient low-rank multimodal fusion with modality-specific factors. Publisher: arXiv Version Number: 1.

[12] S. Mai, Y. Zeng, and H. Hu. Multimodal information bottleneck: Learning minimal sufficient unimodal and multimodal representations. 25:4121–4134.

[13] S. Mai, Y. Zeng, S. Zheng, and H. Hu. Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis. 14(3):2276–2289.

[14] J. Mingyu, Z. Jiawei, and W. Ning. AFR-BERT: Attention-based mechanism feature relevance fusion multimodal sentiment analysis model. 17(9):e0273936.

[15] G. K. P, A. A. V. S, J. P. V, A. Paul, and A. Nayyar. A context-sensitive multi-tier deep learning framework for multimodal sentiment analysis.

[16] S. Poria, E. Cambria, R. Bajpai, and A. Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. 37:98–125.

[17] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883. Association for Computational Linguistics.

[18] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain. Convolutional MKL based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 439–448. IEEE.

[19] P. Shaver, J. Schwartz, D. Kirson, and C. O'Connor. Emotion knowledge: Further exploration of a prototype approach. 52(6):1061–1086.

[20] L. Sun, M. Xu, Z. Lian, B. Liu, J. Tao, M. Wang, and Y. Cheng. Multimodal emotion recognition and sentiment analysis via attention enhanced recurrent model. In *Proceedings of the 2nd on Multimodal Sentiment Analysis Challenge*, pages 15–20. ACM.

[21] T. Sun, J. Ni, W. Wang, L. Jing, Y. Wei, and L. Nie. General debiasing for multimodal sentiment analysis. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5861–5869. ACM.

[22] J. Tang, D. Liu, X. Jin, Y. Peng, Q. Zhao, Y. Ding, and W. Kong. BAFN: Bi-direction attention based fusion network for multimodal sentiment analysis. 33(4):1966–1978.

[23] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569. Association for Computational Linguistics.

[24] M. Wankhade, A. C. S. Rao, and C. Kulkarni. A survey on sentiment analysis methods, applications, and challenges. 55(7):5731–5780.

[25] J. Wu, T. Zhu, J. Zhu, T. Li, and C. Wang. A optimized BERT for multimodal sentiment analysis. 19(2):1–12.

[26] A. Yadav and D. K. Vishwakarma. A deep multi-level attentive network for multimodal sentiment analysis. 19(1):1–19.

[27] K. Yang, H. Xu, and K. Gao. CM-BERT: Cross-modal BERT for text-audio sentiment analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 521–528. ACM.

[28] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. Tensor fusion network for multimodal sentiment analysis. Publisher: arXiv Version Number: 1.

[29] R. Zhang, C. Xue, Q. Qi, L. Lin, J. Zhang, and L. Zhang. Bimodal fusion network with multi-head attention for multimodal sentiment analysis. 13(3):1915.