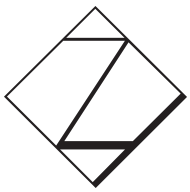


Westsächsische Hochschule Zwickau
Studiengang Informatik

Masterarbeit

Domänenübergreifende Sentiment Analysis auf Twitter-Tweets und Film-Reviews

Falk Puschner
Matrikel: 36973

Sommersemester 2021

Betreuer	Einrichtung
Prof. Dr. Sven Hellbach	Westsächsische Hochschule Zwickau
Prof. Dr. Tina Geweniger	Westsächsische Hochschule Zwickau

30. September 2021

Danksagung

Mit dieser Seite möchte ich mich bei allen Personen bedanken, die auf unterschiedliche Art und Weise zum Gelingen dieser Arbeit beigetragen haben.

Zuerst gebührt mein Dank Professor Sven Hellbach, der meine Masterarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Letztlich richte ich auch ein Dankeschön an die Korrekturleser meiner Arbeit, sowie an meine Familie, die mich unterstützt hat.

Abkürzungsverzeichnis

AUC Area Under Curve

BART Bidirectional and Auto-Regressive Transformer

BERT Bidirectional Encoder Representations from Transformers

BoW Bag of Words

CART Classification and Regression Tree

FN False Negative

FP False Positive

FPR False Positive Rate

GloVe Global Vectors for Word Representation

GLUE General Language Understanding Evaluation

GPT Generative Pre-trained Transformer

KI Künstliche Intelligenz

LSTM Long Short-Term Memory

ML Machine Learning

MLNI Multi-Genre Natural Language Inference

NLP Natural Language Processing

POS Part of Speech

RBF Radial Basis Function

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

SQuAD Stanford Question Answering Dataset

SVM Support Vector Machine

TF Term Frequency

TF-IDF Term Frequency-Inverse Document Frequency

TN True Negative

TP True Positive

TPR True Positive Rate

Abbildungsverzeichnis

Tabellenverzeichnis

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Seit den frühen 2000er Jahren ist das Internet im Wandel. Aus einer statischen Sammlung von Informationen entwickelte sich eine dynamische und interaktive Plattform, wodurch ein Austausch von Inhalten über die ganze Welt möglich ist. Der Wandel wurde durch das Aufkommen von sozialen Netzwerken wie bspw. Twitter oder Instagram beschleunigt. Durch die vielen Daten ist eine manuelle Analyse nicht mehr möglich und die Automatisierung ist notwendig, wodurch neue technische und rechnerische Herausforderungen gelöst werden müssen (Whitehead & Yaeger, 2010).

Die Sentiment Analysis ist ein Teilgebiet des Natural Language Processing (NLP) und des Machine Learning (ML). Sie beschäftigt sich mit der Aufgabe der Verarbeitung natürlicher Sprache und der Informationsextraktion mit dem Ziel die Gefühle der Verfasser durch die Untersuchung von Texten zu ermitteln.
Es gibt verschiedene Anwendungsfälle für die Analyse aus unterschiedlichen Domänen, welches eine Sammlung von Texten bezeichnet, die ein gleiches semantisches Konzept beschreiben und aus ähnlichen Quellen stammen wie bspw. Produktbewertungen oder Tweets (Pan et al., 2010). Die Sentiment Analysis kann durch ein Unternehmen genutzt werden, um die Meinungen von Kunden über ein bestimmtes Produkt oder einen angebotenen Service zu erhalten. Die Firmen können interne Daten wie z. B. Kundenfeedback aus E-Mails oder Umfragen verwenden, um die Analyse umsetzen zu können.
In Regierungskreisen kann es als Hilfe zum Treffen von Entscheidungen basierend auf den sich ändernden sozialen, wirtschaftlichen und politischen Klima genutzt werden. Ein Extrem zeigt der Staat China, in welchem die Behörden die sozialen Medien überwachen, um öffentliche Meinungen z. B. zur Regierungspolitik und Sorgen der Bürger zu erfahren. Die Äußerungen über die soziale Plattform „Weibo“ können verwendet werden, um Korruption und Skandale aufzudecken. Nicht nur Regierungen und Unternehmen profitieren von der Analyse. Der Verbraucher kann die Meinung vieler Nutzer verwenden, um eine Kaufentscheidung für ein Produkt oder eine Dienstleistung zu treffen (Liu, 2015, 2012).

Durch die verschiedenen Anwendungsfälle zeigt sich, dass für jede Domäne ein eigenes Modell zur Klassifikation der Stimmung erstellt werden muss. Das hat zur Folge, dass viel Zeit und Daten benötigt werden. Zur Lösung des Problems sind in den letzten Jahren Ansätze zum Transfer Learning in der natürlichen Sprachverarbeitung auf dem Vormarsch (Devlin et al., 2018; Tsakalidis et al., 2014). Das Ziel ist ein trainiertes Modell bspw. auf Review-Daten zu nutzen, um die Klassifikation auf einem anderen Datensatz bspw. Twitter-Daten durch die Anpassung des Modells mit wenigen Daten umzusetzen.

Die Nutzung mehrerer Domänen nennt man domänenübergreifende Sentiment Analysis und ermöglicht die Nutzung eines gut trainierten Modells, um in der Praxis mit Domänen eine zufriedenstellende Performance zu erreichen, bei denen nur kleine Datensätze vorhanden sind (Pan et al., 2010). Der Einsatz von ähnlichen Domänen wurde bereits mehrfach untersucht (Peddinti & Chintalapoodi, 2011; Tsakalidis et al., 2014), der Einsatz von unterschiedlichen Domänen in bspw. Wortwahl und Textlänge sind weitgehend unerforscht.

In der Arbeit sollen zwei Methoden untersucht werden, um eine domänenübergreifende Sentiment Analysis zu realisieren. Die erste Methode sind die Ensemble-Methoden (Aue & Gamon, 2005) als ein Zusammenschluss von verschiedenen klassischen Lernalgorithmen. Die andere Methode sind die Transformer-Methoden (Vaswani et al., 2017), welches große vortrainierte Sprachmodelle für viele Aufgaben der NLP wie z. B. Text Summarization sind.

1.2 Zielstellung

Im Rahmen der Arbeit soll der Frage nachgegangen werden, ob die Nutzung eines trainierten Modells auf Review-Daten zur Klassifikation von Twitter-Daten möglich ist. Das Ziel der Arbeit ist es herauszufinden, ob die domänenübergreifende Sentiment Analysis umsetzbar ist und ob die Transformer- oder die Ensemble-Methoden die besseren Ergebnisse liefern.

1.3 Vorgehensweise

Die vorliegende Arbeit beschäftigt sich mit der Sentiment Analysis und ist folgendermaßen aufgebaut: Es beginnt in Kapitel 2 mit der Analyse der grundlegenden Literatur zum Thema (domänenübergreifende) Sentiment Analysis. Anschließend werden im theoretischen Rahmen u.a. relevante Begriffe wie „Machine Learning“ erläutert. Im Anschluss wird in Kapitel 3 der empirische Teil der Arbeit in Form eines Experimentes beschrieben. Es werden die Methoden implementiert, um quantitative Ergebnisse durch die Qualitätsmaße zu erhalten und einen Vergleich ausführen zu können. Die Resultate der Untersuchungen zu den verschiedenen Methoden werden in Kapitel 4 zusammengefasst und mit Abbildungen visualisiert. In Kapitel 5 werden die Ergebnisse der Methoden evaluiert und interpretiert. Es werden Vergleiche zwischen den unterschiedlichen Methoden gezogen. Am Ende der Arbeit wird in Kapitel 6 das Ergebnis der Arbeit zusammengefasst und ein Ausblick auf die mögliche zukünftige Forschung gegeben.

Kapitel 2

Theoretische Grundlagen

In den folgenden Abschnitten wird die wesentliche Literatur zur (domänenübergreifenden) Sentiment Analysis dargestellt. Anschließend werden die zwei Teilbereiche NLP und ML sowie die Sentiment Analysis vorgestellt. Es werden die zu nutzenden Datensätze beschrieben und am Ende des Abschnittes die zwei zu untersuchenden Methoden sowie eine spezielle Lerntechnik zum Transfer Learning erläutert.

2.1 Stand der Forschung

In der Forschung wird zwischen zwei Varianten zur Durchführung der Sentiment Analysis unterschieden (Müller et al., 2020; Pang et al., 2002; Whitehead & Yaeger, 2010). In der folgenden Abbildung werden sie schematisch dargestellt.

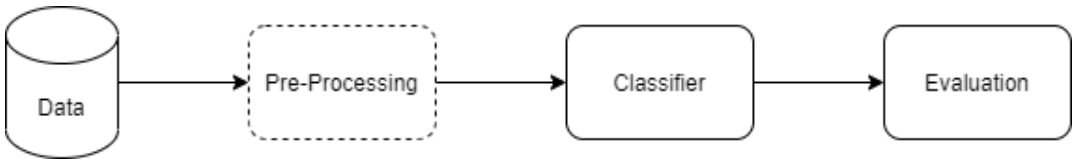


Abbildung 2.1: Überblick über Sentiment Analysis

Es ist zu erkennen, dass die Sentiment Analysis eine Datengrundlage benötigt, um die Klassifikation umsetzen zu können. Anschließend können die Daten wahlweise bereinigt werden, wie es bspw. bei den Ensemble-Methoden gemacht wird. Zum Schluss können die Daten mithilfe des Klassifikators nach deren Stimmung klassifiziert und die Ergebnisse evaluiert werden. In den folgenden Absätzen soll mithilfe von drei Tabellen ein aktueller Überblick über genutzte Datensätze, mögliche Vorverarbeitungsschritte und Verfahren zur Klassifikation beschrieben werden.

In Tabelle 2.1 wird ein Überblick über die genutzten Datensätze abgebildet. Es ist zu sehen, dass die Rezensionen zu Filmen, Bewertungen zu Amazon-Produkten und Twitter-Nachrichten häufig verwendet werden. Im Gegensatz werden bspw. Bewertungen zu Büchern oder Restaurants vereinzelt genutzt. Aufgrund der Ähnlichkeit zwischen den Bewertungen von Amazon-Produkten und den Rezensionen von Filmen in Bezug auf das Bewertungssystem und die ausführlicheren Beschreibungen soll in der Arbeit der Standarddatensatz von Film-Reviews und Twitter-Nachrichten verwendet werden.

	Pang et al., 2002	Whitehead und Yaeger, 2010	Peddinti und Chintalapoodi, 2011	Glorot et al., 2011	Aue und Gamon, 2005	Xia und Zong, 2011	Peng et al., 2018	Korovkinas et al., 2019	Vishal und Sonawane, 2016	Khalid et al., 2020	Müller et al., 2020	Hoang et al., 2019
Filme	✓		✓		✓							
Tweets			✓					✓	✓		✓	✓
Amazon		✓		✓		✓	✓	✓				
Google Apps										✓		
Bücher					✓							
Befragungen					✓							
Restaurant		✓										✓
Anwalt		✓										
TV		✓										
Telefon												✓
Kamera												✓
Hotel												✓
Museum												✓

Tabelle 2.1: Überblick der genutzten Datensätze

Die Tabelle 2.2 stellt genutzte Schritte zur Vorverarbeitung der Daten dar. Damit werden die Datensätze bereinigt und die Komplexität reduziert. Es beginnt mit der Tokenization, indem der Text in einzelne Wörter (Token) aufgeteilt wird (Jianqiang & Xiaolin, 2017; Zin et al., 2017). Die Schritte Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF) (siehe Abschnitt 2.4.3) und die Feature Selection werden häufig verwendet. Außerdem werden einzelne Wörter wie z. B. Nummern, Satzzeichen und Nutzernamen oder bedeutungslose Wörter wie z. B. Stoppwörter und Links entfernt. Es wird vereinzelt die Rechtschreibung überprüft, die Wortart (engl. Part of Speech (POS)) eingebunden und Parameter der Klassifikatoren optimiert (Parameter-Tuning). Aus den genutzten Schritten geht hervor, dass TF und TF-IDF für die Vorverarbeitung genutzt werden können. Die Feature Selection wird nicht betrachtet, weil die Arbeit die Machbarkeit der domänenübergreifenden Sentiment Analysis zwischen zwei unterschiedlichen Datensätzen zeigen und keine Optimierung der Klassifikation durchgeführt werden soll.

	Pang et al., 2002	Whitehead und Yaeger, 2010	Peddinti und Chintalapoodi, 2011	Glorot et al., 2011	Aue und Gamon, 2005	Xia und Zong, 2011	Peng et al., 2018	Korovkinas et al., 2019	Vishal und Sonawane, 2016	Khalid et al., 2020	Müller et al., 2020	Hoang et al., 2019
Negation	✓											
POS	✓					✓						
Position des Wortes	✓											
Stoppwörter		✓							✓	✓		
TF	✓	✓	✓	✓	✓	✓			✓	✓		
TF-IDF							✓	✓		✓		
Nummern			✓					✓	✓	✓		
Links			✓					✓	✓		✓	
Symbole								✓	✓	✓		
Stemming										✓		
Emoticons			✓						✓		✓	
Nutzername			✓					✓	✓		✓	
Hashtag			✓					✓	✓			
Feature Selection		✓	✓	✓	✓		✓	✓				
Satzzeichen								✓	✓	✓		
Rechtschreibung									✓			
Abkürzungen									✓			
Parameter-Tuning								✓				
Autoencoder				✓			✓					

Tabelle 2.2: Überblick der genutzten Vorverarbeitung

Auf Grundlage der Arbeiten von Jianqiang und Xiaolin (2017), Vishal und Sonawane (2016) und Zin et al. (2017) können weitere Bereinigungsschritte für die Film-Rezensionen und die Tweets verwendet werden. Für die unstrukturierten Reviews kann das Stemming genutzt werden, um verschiedene Varianten eines Wortes auf eine gemeinsame Stammform zurückzuführen und Präfixe sowie Suffixe zu entfernen bzw. zu kürzen. Ein einfaches Beispiel ist das Wort „pictures“, dass zu „picture“ reduziert wird. Das Resultat ist die Minimierung von redundanten Wörtern in dem Text. Durch das Entfernen von Stoppwörtern werden Wörter gelöscht, die zur Bestimmung des Sentiment nicht gebraucht werden und häufig im Text auftreten. Dazu gehören bspw. die Artikel „a“, „an“ und „the“. Die Entfernung von bedeutungslosen Wörtern ermöglicht das Löschen von Wörtern, die keine Auswirkungen auf die Analyse haben. Es werden spezielle Symbole wie „@“ und „#“ oder Bewertungen wie „a+“ und „b-“ gelöscht. Es können Wörter mit weniger als drei Zeichen aus dem Text entfernt werden. Ein Beispiel sind Satzzeichen wie „.“ oder „!“.

Die Verbesserung der Genauigkeit der Klassifikation der kurzen Twitter-Nachrichten wird mit dem Entfernen von Negationen und der Erweiterung von Akronymen erreicht. Das Löschen von Verneinungen ist ein wichtiger Schritt, indem z.B. aus „won't“ die Wörter „will not“ werden. Die Abkürzungen bzw. der Slang sind schlecht gebildete Wörter und werden oft in Tweets verwendet. Mithilfe eines Wörterbuchs kann die ursprüngliche Form bestimmt werden. Ein Beispiel ist die Abkürzung „*4u“, die für den Satz „Kiss for you“ steht. Die folgenden Schritte haben einen geringen Einfluss auf die Sentiment Analysis, jedoch werden nicht nützliche Informationen entfernt, die keinen Einfluss auf das Sentiment haben. Dazu zählt das Entfernen von URLs, Nummern und Satzzeichen, die keine weiteren Informationen über die Stimmung eines Tweets enthalten. Es werden Stoppwörter gelöscht, um einen negativen Einfluss auf die Analyse zu vermeiden. Der nächste Schritt ist die Entfernung des Hashtag-Symbols und der genannten Nutzernamen in einem Tweet.

Durch die beschriebenen Ansätze konnte gezeigt werden, dass für die zwei Domänen unterschiedliche Schritte gute Ergebnisse liefern. Aufgrund dessen soll ein Experiment in Abschnitt 3.1 umgesetzt werden, um gemeinsame Schritte zur Bereinigung der Texte zu finden.

Durch die Tabelle 2.3 werden die genutzten Verfahren zur Klassifikation repräsentiert. Es ist zu erkennen, dass die Verfahren Naive Bayes und Support Vector Machine (SVM) häufig verwendet werden. In der Auswahl von Arbeiten sind die Ensemble-Methoden (siehe Abschnitt 2.6) selten in Verwendung. Zudem werden in aktuellen Arbeiten Transformer-Methoden (siehe Abschnitt 2.7) verwendet, um die Stimmung eines Textes zu ermitteln. Aus den genutzten Verfahren sollen die Klassifikatoren SVM und Naive Bayes genutzt werden. Außerdem sollen die Transformer-Methoden wie bspw. Bidirectional Encoder Representations from Transformers (BERT) verwendet werden.

	Pang et al., 2002	Whitehead und Yaeger, 2010	Peddinti und Chintalapoodi, 2011	Glorot et al., 2011	Aue und Gamon, 2005	Xia und Zong, 2011	Peng et al., 2018	Korovkinas et al., 2019	Vishal und Sonawane, 2016	Khalid et al., 2020	Müller et al., 2020	Hoang et al., 2019
Naive Bayes	✓		✓		✓	✓			✓			
Logistic Regression	✓								✓			
SVM (linear)	✓	✓	✓	✓	✓			✓	✓	✓		
Bagging		✓										
Boosting		✓								✓		
Voting										✓		
MLP							✓					
BERT											✓	✓

Tabelle 2.3: Überblick der genutzten Verfahren

In der bisherigen Forschung lag der Schwerpunkt vor allem auf der Verbesserung der Sentiment-Klassifikation und der domänenübergreifenden Sentiment-Klassifikation bspw. zwischen unterschiedlichen Review-Datensätzen oder mit Klassifikatoren wie SVM. Die Sentiment Analysis zwischen unterschiedlichen Kontexten wie bspw. Reviews und Tweets von Twitter unter der Verwendung von Ensemble-Methoden wurde in den bisherigen Untersuchungen nicht betrachtet. Für die Transformer-Methoden konnten die Autoren Siebert et al. (2019) mit einer RoBERTa-basierten Methode zeigen, dass die domänenübergreifende Sentiment Analysis zwischen Tweets und Film-Reviews möglich ist. In der Arbeit soll überprüft werden, ob die domänenübergreifende Sentiment Analysis auch ohne bzw. nur mit Feinabstimmung auf Film-Reviews zur Klassifikation von Tweets umgesetzt werden kann.

2.2 Natural Language Processing

In dem Abschnitt wird der erste Teilbereich der Sentiment Analysis beschrieben. Dazu wird der Begriff Natural Language Processing definiert, die unterschiedlichen Ebenen dargestellt und weitere Anwendungsbereiche erwähnt.

2.2.1 Definition

Natural Language Processing (NLP) ist ein computergestützter Ansatz zur Analyse von Texten. In dem folgenden Zitat sollen einige Aspekte von NLP beschrieben werden.

> Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. (Liddy, 2001)

Die Definition zeigt, dass es verschiedene Methoden und Techniken zur Analyse von Sprache gibt. Die Voraussetzung für den zu analysierenden Text ist, dass er aus dem tatsächlichen Sprachgebrauch stammt und nicht für den Verwendungszweck konstruiert wurde. Es gibt verschiedene Ebenen der linguistischen Analyse, die im Abschnitt 2.2.2 näher erläutert werden. Im Gegensatz zu Menschen, die alle Ebenen verwenden, nutzen NLP-Systeme einzelne Ebenen bzw. Kombinationen der Ebenen. Daraus ergeben sich Unterschiede in der Anwendung, die mit Beispielen im Abschnitt 2.2.3 gezeigt werden sollen. NLP als Disziplin der Künstliche Intelligenz (KI) (Frochte, 2018) hat das Ziel eine menschenähnliche Sprachverarbeitung für unterschiedliche Aufgaben zu ermöglichen (Liddy, 2001).

2.2.2 Ebenen von NLP

Mit dem Ansatz der Ebenen der Sprache oder auch synchronem Modell genannt, kann ein NLP-System beschrieben werden. Durch die Interaktionen zwischen den Ebenen wird eine dynamische Sprachverarbeitung ermöglicht, wodurch bspw. die Mehrdeutigkeit von Wörtern durch den Einbezug des Kontextes gelöst werden kann. Jede Ebene vermittelt eine Bedeutung und durch die Nutzung aller Ebenen entsteht ein Verständnis. Daraus kann geschlussfolgert werden, dass je mehr Ebenen der Sprache ein NLP-System umsetzt, desto leistungsfähiger ist es. In den folgenden Absätzen werden die einzelnen Ebenen erläutert (Chopra et al., 2013; Liddy, 2001).

Die *phonologische Ebene* dient der Interpretation von Sprachlauten innerhalb und zwischen Wörtern. Dazu kann ein NLP-System eine gesprochene Eingabe erhalten, die Schallwellen analysieren und in ein digitales Signal kodieren. Anschließend ist die Interpretation mit verschiedenen Regeln oder der Nutzung eines Sprachmodells möglich. Ein Beispiel für die Ebene ist die Aussprache von Vokalen in Abhängigkeit von der Region. Es wird das „a“ in „ask“ u.a. als [æ], [a], [ɛ] oder [aː] ausgesprochen.

In der *morphologischen Ebene* wird die Struktur von Wörtern analysiert, identifiziert und beschrieben. Die Morpheme sind die kleinsten Bedeutungseinheiten, aus denen ein Wort besteht. Das Substantiv „preregistration“ enthält drei Morpheme mit dem Präfix „pre“, der Wurzel „registra“ und dem Suffix „tion“. Um die Bedeutung eines Wortes zu verstehen, kann ein NLP-System es in die einzelnen Morpheme zerlegen. Ein Beispiel ist, dass durch die Endung „ed“ an einem Verb festgestellt werden kann, dass die Handlung in der Vergangenheit stattgefunden hat.

Die Interpretation der Bedeutung der einzelnen Wörter wird auf der *lexikalischen Ebene* umgesetzt. Für das Verständnis auf Wortebene sind verschiedene Verarbeitungsschritte möglich. Einem Wort können POS-Tags hinzugefügt werden, sodass das Wort „eat“ als Verb markiert wird. Es ist auch die Nutzung eines Lexikons oder das Ersetzen von Wörtern, die nur eine mögliche Bedeutung haben, mit einer semantischen Repräsentation möglich.

Die *syntaktische Ebene* bietet die Analyse von Wörtern eines Satzes, um die grammatikalische Struktur eines Satzes zu beschreiben. Dafür sind ein Parser und die Grammatik der Sprache notwendig, damit der Satz mit seinen strukturellen Abhängigkeitsbeziehungen dargestellt werden kann. D.h. durch die Syntax wird den Wörtern eines Satzes eine Bedeutung aufgrund von Reihenfolge und Abhängigkeiten zugeordnet. Es wird durch die unterschiedliche Syntax der Sätze „The dog chased the cat“ und „The cat chased the dog“ eine unterschiedliche Bedeutung hervorgerufen.

Mithilfe der Analyse der Wörter im Satz und deren Bedeutung kann auf *semantischer Ebene* die Bedeutung eines Satzes festgestellt werden. Es kann bspw. die Mehrdeutigkeit eines Wortes durch die Nutzung des lokalen Kontextes oder Wissen über die Domäne aufgelöst werden. Ein Beispiel ist das Wort „file“, welches einen Ordner zur Aufbewahrung von Papieren oder ein Werkzeug zum Formen von Fingernägeln bezeichnen kann. Nach der Auswahl der Bedeutung kann sie in die semantische Repräsentation des Satzes aufgenommen werden.

In der *Diskurs-Ebene* werden Texte bestehend aus mehreren Sätzen interpretiert. Man versucht Verbindungen zwischen Sätzen zu finden, um Eigenschaften des Textes zu erhalten, die Bedeutung vermitteln. Der Satz „He wanted it“ ist abhängig vom Kontext, wodurch „it“ nur durch den vorherigen Satz ermittelt werden kann.

Die letzte Ebene ist die *pragmatische Ebene*. Sie bezieht sich auf den zielgerichteten Gebrauch von Sprache und die Verwendung des Kontextes über den Inhalt des Textes hinaus für das Verständnis. Das Ziel der Ebene ist eine zusätzliche Bedeutung eines Textes zu finden, ohne das sie kodiert ist. Dafür ist ein umfangreiches Weltwissen, sowie ein Verständnis von Absichten, Plänen und Zielen notwendig. Ein Beispiel ist der Satz „The city councilors refused the demonstrators a permit because they forced violence“, wo die Auflösung des Begriffes „they“ notwendig ist.

2.2.3 Anwendungen

Mithilfe von NLP können Computer die natürliche Sprache von Menschen verarbeiten und verstehen, um sie zu verwenden und nützliche Ausgaben zu erzeugen. In den folgenden Absätzen sollen mögliche Anwendungen von NLP vorgestellt werden (Chopra et al., 2013; Liddy, 2001; Sarkar, 2016).

Die *Text Summarization* ermöglicht die Erstellung einer Zusammenfassung eines Korpus von Textdokumenten mit den wichtigsten Punkten. Der Korpus, wessen Inhalt reduziert werden soll, kann eine Sammlung von Texten, Absätzen oder Sätzen sein. Für die Reduzierung des Inhaltes werden Schlüsselwörter, Phrasen und Sätze extrahiert, die eine wichtige Bedeutung haben.

In der *Machine Translation* wird zwischen zwei Sprachen unter Berücksichtigung der Syntax, Semantik und Grammatik übersetzt. Es werden nicht nur Wörter von einer in die andere Sprache ersetzt, sondern z. B. auch Textkorpora mit statistischen und linguistischen Techniken kombiniert. Ein bekanntes maschinelles Übersetzungssystem ist Google Translate.

Im *Question Answering System* werden Fragen durch die Verwendung von robusten und skalierbaren Systemen beantwortet. Die Eingabe der Frage erfolgt durch den Benutzer in natürlich-sprachlicher Form. Ein Beispiel sind die personalisierten Assistenten wie Siri oder Cortana sowie Chatbots für spezifische Domänen. Durch die genannten Beispiele zeigt sich, dass sie einen begrenzten Umfang abbilden, da das Verständnis nur für eine Teilmenge der wichtigsten Sätze und Phrasen vorhanden ist. Ein erfolgreiches System benötigt eine große Datenbank mit Wissen aus unterschiedlichen Domänen.

2.3 Maschinelles Lernen

In dem Abschnitt wird der zweite Teilbereich der Sentiment Analysis beschrieben. Dazu wird der Begriff Maschinelles Lernen definiert und weitere Anwendungsfälle erwähnt. Anschließend werden verschiedene Klassifikatoren und mögliche Werkzeuge zur Auswertung der Ergebnisse vorgestellt.

2.3.1 Definition

Maschinelles Lernen oder engl. Machine Learning (ML) ist ein Teilbereich der KI. Eine einheitliche Definition des Begriffes existiert nicht, weshalb im Folgenden einzelne Beschreibungen aufgezählt werden sollen.

> Jede Form von Leistungssteigerung, die durch gezielte Anstrengung erreicht wurde.
>
> Jede Verhaltensänderung, die sich auf Erfahrung, Übung oder Beobachtung zurückführen lässt.
>
> Durch Erfahrung entstandene, relativ überdauernde Verhaltensänderung bzw. -möglichkeiten. (Frochte, 2018)

Durch die Erläuterungen wird dargestellt, dass ein Modell bzw. eine Maschine aus Erfahrungen (Daten) lernen kann, um Verhaltensänderungen zu bewirken. Weitere Anpassungen des Verhaltens sind durch Adaption neuer Daten möglich.

Es ist keine statische Programmierung von Maschinen. Man unterscheidet drei verschiedene Kategorien von Lernalgorithmen: überwachtes, bestärkendes und unüberwachtes Lernen (Burkov, 2019; Frochte, 2018).

2.3.2 Anwendungen

Mithilfe des ML ergeben sich viele Möglichkeiten um Probleme zu lösen, die mit statischen Programmen schwer lösbar sind. In den folgenden Absätzen soll ein kleiner Abriss von möglichen Anwendungen des ML dargestellt werden (Kubat, 2017).

Bei der *Character Recognition* geht es um das Erkennen von (handgeschriebenen) Texten. Dazu wird ein Bild mit dem handgeschriebenen oder gescannten Text in ein maschinenlesbaren Text konvertiert. Es wird bspw. bei der automatisierten Vorverarbeitung verschiedener Formulare oder bei der Umwandlung eines geschriebenen Textes in ein Format, welches der Texteditor erkennen kann, genutzt.

In der *Text Classification* wird die Entscheidung getroffen, ob z. B. ein Text aus einer großen Sammlung von Dokumenten für ein bestimmtes Thema relevant ist oder in welcher Sprache eine Konversation geführt wurde. Die Umsetzung kann mit einer überschaubaren Menge von Daten anfangen, um einen Klassifikator zu trainieren. Anschließend können die restlichen Texte kategorisiert werden.

Die *Oil-Spill Recognition* ermöglicht das Aufspüren von Ölverschmutzungen, um die Verursacher zur Rechenschaft ziehen zu können. Es werden Radarbilder der Meeresoberfläche mithilfe von satellitengestützten Geräten gemacht, um sie anschließend auswerten zu können. Wenn auf dem gräulichen Hintergrund dunkle Regionen zu erkennen sind, ist das ein Indiz für einen Ölteppich, welcher illegal von einem Tanker entsorgt wurde.

2.3.3 Klassifikation

In der Arbeit sollen die gegebenen Texte nach deren negativen (0) und positiven (1) Stimmung kategorisiert werden. Die binäre Klassifikation ist eine Funktion die Vektoren aus dem k-dimensionalen Merkmalsraum auf eine Menge von Klassen C abbildet: $\mathbb{R}^k \rightarrow C$ mit $C \in \{0, 1\}$ (Frochte, 2018). Für das Beispiel Sentiment Analysis ist ein Merkmalsvektor ein Text mit Wörtern als Merkmale. Es existieren verschiedene Algorithmen wie Neuronale Netze, Fuzzy oder Evolutionsalgorithmen für die Umsetzung (Kruse et al., 2015). Die folgenden Absätze dienen zur Vermittlung von Grundwissen und können wahlweise gelesen werden. Es werden häufig verwendete Algorithmen (Joshi & Deshpande, 2018; Pang et al., 2002; Vaswani et al., 2017; Vishal & Sonawane, 2016; Xia et al., 2011) vorgestellt, die gute Ergebnisse in der Sentiment Analysis geliefert haben sowie den Algorithmus Decision Tree, welcher im Abschnitt 2.6 benötigt wird.

Logistic Regression

Die Logistische Regression oder auch Maximum Entropy genannt, basiert auf der Maximum-Likelihood-Schätzung. Man versucht eine Funktionskurve zu finden, die gut zu den Daten der binären Klassifikation passt. Die Funktion ist die logistische Funktion, wie in Abbildung 2.2 zu sehen. Sie ist eine sigmoide Funktion, „s-förmig“, symmetrisch und verläuft asymptotisch gegen $y = 0$ und $y = 1$ (Burkov, 2019; Géron, 2020).

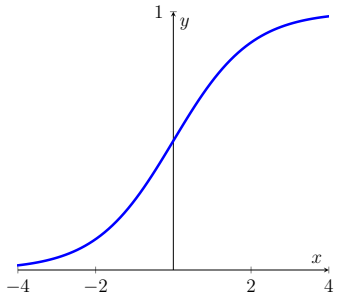


Abbildung 2.2: Logistische Funktion

Das Ergebnis der Funktion liegt zwischen 0 und 1. Es wird als Wahrscheinlichkeit interpretiert, d.h. wenn der Wert nahe bei 0 liegt, dann ist es unwahrscheinlich, dass der Text positiv ist. Wenn der Wert nahe bei 1 liegt, so ist es wahrscheinlich, dass der Text positiv ist. Für die Berechnung der Wahrscheinlichkeit p wird die folgende Formel genutzt. Anschließend erfolgt die Einteilung in die Klassen mithilfe eines Schwellenwertes (Burkov, 2019; Géron, 2020).

$$p = \frac{1}{1 + e^{-\theta \cdot x^T}} \tag{2.1}$$

Für die Berechnung der gewichteten Summe der Eingabemerkmale x ist der Parametervektor θ notwendig, welcher durch die Maximum-Likelihood-Schätzung bestimmt wird. Die Parameter sollen möglichst hohe Wahrscheinlichkeiten liefern, wenn der Text positiv ist. Ansonsten sollen es niedrige Wahrscheinlichkeiten werden, wenn der Text negativ ist. Die Schätzung maximiert die Likelihood-Funktion, die beschreibt, wie wahrscheinlich es ist, dass der Wert einer Funktion y durch die Merkmale x vorausgesagt werden kann. In der Praxis wird meist die Log-Likelihood genutzt, da die exponentielle Funktion schwieriger zu maximieren ist. Sie ist wie folgt definiert (Burkov, 2019; Géron, 2020).

$$L = -\frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot ln(p_i) + (1 - y_i) \cdot ln(1 - p_i) \tag{2.2}$$

Nach den Erläuterungen zur Berechnung der Likelihood L durch die Summe der Wahrscheinlichkeiten, kann ein Optimierungsverfahren wie z. B. das Gradientenabstiegsverfahren genutzt werden, um das Optimum zu finden. Dazu muss das Argument des natürlichen Logarithmus maximiert werden, weil der natürliche Logarithmus eine streng monotone Funktion ist (Burkov, 2019; Géron, 2020).

Naive Bayes

Der Naive Bayes basiert auf dem Theorem von Bayes, welches besagt, dass bei einer gegebenen Klassenvariable Y und einem Merkmalsvektor X folgende Beziehung besteht (Anandarajan et al., 2019; Jo, 2021).

$$P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)} \tag{2.3}$$

Mit der Formel können die Wahrscheinlichkeiten der einzelnen Merkmale geschätzt und die Likelihood eines neuen Merkmalvektors berechnet werden. Für die Berechnung der bedingten Wahrscheinlichkeit muss eine „naive“ Voraussetzung angenommen werden. Die Wahrscheinlichkeit, dass ein Wort zu einer bestimmten Klasse gehört, ist unabhängig von der Wahrscheinlichkeit, dass die anderen Wörter zu der Klasse gehören. Die Klassifizierungsregel beschreibt, dass die wahrscheinlichste Klassifizierung $\hat{y}$ des Merkmalvektors mit der bedingten Wahrscheinlichkeit eines Merkmalvektors bei gegebener Klasse $P(X|Y)$ und der Wahrscheinlichkeit der Klassifikation $P(Y)$ errechnet werden kann (Anandarajan et al., 2019; Jo, 2021).

$$\hat{y} = \text{argmax}\, P(Y) \cdot P(X|Y) \tag{2.4}$$

Das Ergebnis ist die Klasse bei der die Wahrscheinlichkeit am größten ist. Die Berechnung der Wahrscheinlichkeit und damit die Klassifizierung eines Merkmalvektors ist problematisch bei einer Häufigkeit eines Wortes von 0, da die Wahrscheinlichkeit ebenso 0 ergibt. Aus dem Grund wird ein kleiner Wert λ zu den Häufigkeiten addiert, wodurch die Wahrscheinlichkeit nicht 0 werden kann und die Klassifizierung stabilisiert wird (Anandarajan et al., 2019; Jo, 2021).

Die Entscheidung für die Klasse basiert auf einer einfachen Annahme, die sich in vielen Situationen bewährt hat. Der Algorithmus ist schnell im Vergleich zu komplexeren Methoden und er benötigt nur eine kleine Menge von Trainingsdaten, um die Parameter zu schätzen. Es ist ein guter Klassifikator, aber ein schlechter Schätzer, weshalb die Wahrscheinlichkeitsangaben für die Klassifizierung kritisch zu beachten sind. Es gibt verschiedene Umsetzungen des Naive-Bayes-Klassifikator. Sie unterscheiden sich durch die Annahme, die bzgl. der Verteilung von $P(X|Y)$ getroffen wird. (Anandarajan et al., 2019; Jo, 2021).

Support Vector Machine

Eine Support Vector Machine (SVM) zeichnet jeden Merkmalsvektor mit n Merkmalen in einen hochdimensionalen Raum (Géron, 2020; Lee, 2019). In Abbildung 2.3 ist eine einfache Darstellung im zweidimensionalen Raum zu sehen.

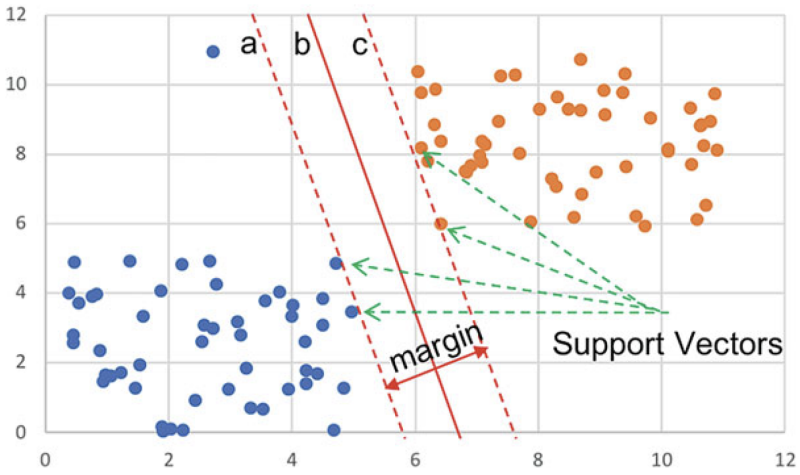


Abbildung 2.3: Support Vector Machine (Rebala et al., 2019)

Die zwei dargestellten Klassen können durch die Gerade b oder auch Entscheidungsgrenze genannt, voneinander getrennt werden. Sie sind linear separierbar. Außerdem wird ein größtmöglicher Abstand zu den Merkmalsvektoren der unterschiedlichen Klassen angestrebt, sodass der Margin $|\vec{ab}|$ bzw. $|\vec{bc}|$ möglichst groß wird. Die Art der Klassifikation nennt man Large-Margin-Klassifikation (Géron, 2020; Lee, 2019).

Der Algorithmus bekommt seinen Namen durch die Support Vectors (dt. Stützvektoren). Die Vektoren liegen auf den Geraden a und c und stützen die Entscheidungsgrenze. Solange sich die Stützvektoren nicht ändern, wird die Grenze durch neue Daten nicht beeinflusst. Die Entscheidungsgrenze ist eine (n-1)-dimensionale Hyperebene und die Menge aller Punkte mit $h = 0$. Die Entscheidungsfunktion h ist eine n-dimensionale Hyperebene und berechnet sich wie folgt (Géron, 2020; Lee, 2019).

$$h = w^T \cdot x + b \tag{2.5}$$

Der Parameter w sind die Gewichte der Merkmale und b ist der Bias-Term. Mithilfe der Formel kann ein Merkmalsvektor klassifiziert werden. Wenn das Ergebnis positiv ist, so ist der Text positiv, ansonsten ist er negativ. Die parallelen Geraden a und c mit dem gleichen Abstand zur Entscheidungsgrenze ergeben sich aus $h = -1$ und $h = 1$. Damit der Margin möglichst groß und das Modell verallgemeinert werden kann, muss der Gewichtsvektor w minimiert werden (Géron, 2020; Lee, 2019).

Es gibt zwei Ansätze wie die Minimierung des Gewichtsvektors umgesetzt werden kann. Die erste Möglichkeit nennt sich Hard-Margin, welche keine Verletzung des Margin zulässt. D.h. die Entscheidungsfunktion muss für positive Merkmalsvektoren einen Wert größer als 1 und für negative Merkmalsvektoren einen Wert kleiner als -1 zurückgeben. Zusammengefasst können die Bedingungen in dem Ausdruck $t \cdot h(x) \geq 1$ mit $t = -1$ für negative und $t = 1$ für positive Merkmalsvektoren. Die Zielfunktion für den SVM-Klassifikator wird in der folgenden Formel gezeigt (Géron, 2020; Lee, 2019).

$$\min \frac{1}{2} \cdot w \cdot w^T \text{ mit } t \cdot h(x) \geq 1 \tag{2.6}$$

Der Nachteil ist, dass der Hard-Margin sehr anfällig für Ausreißer ist. Eine Balance zwischen größtmöglichen Margin und einer begrenzten Anzahl von Margin-Verletzungen wird mit dem Soft-Margin erreicht. Dazu werden die Schlupfvariablen oder engl. Slack Variables ζ eingeführt, die eine Abweichung von der Forderung ermöglichen. Das Ziel ist es, die Slack Variable so klein wie möglich zu halten, um Verletzungen des Margins zu verringern sowie $\frac{1}{2} \cdot w^T \cdot w$ zu minimieren, sodass der Margin möglichst groß wird. Ein zusätzlicher Parameter C wird genutzt, um die Bedeutung des Strafterms zu variieren. Die Änderungen ergeben die folgende Formel (Géron, 2020; Lee, 2019).

$$\min \frac{1}{2} \cdot w^T \cdot w + C \cdot \sum_{i=1}^{n} \zeta_i \text{ mit } t \cdot h(x) \geq 1 - \zeta \text{ und } \zeta \geq 0 \tag{2.7}$$

Bisher musste der Datensatz immer linear separierbar sein. Das Problem kann mit dem Kernel-Trick (Géron, 2020) gelöst werden. Er basiert auf dem Mercer-Theorem, welches die Berechnung der Distanz zwischen Merkmalsvektoren für einen hochdimensionalen Merkmalsraum durch die Nutzung einer Kernel-Funktion ermöglicht. Es gibt verschiedene Arten von Kernel wie z. B. den gaußschen Radial Basis Function (RBF) oder den polynomiellen Kernel (Géron, 2020; Lee, 2019).

Decision Tree

Ein Entscheidungsbaum oder engl. Decision Tree ist ein azyklischer Graph zum Treffen von Entscheidungen, wie in Abbildung 2.4 zu sehen. Beginnend bei dem Wurzelknoten r wird für jeden Knoten n ein Merkmal des Merkmalsvektors untersucht. Wenn der Wert kleiner ist als der Schwellenwert, so wird der linke Zweig verfolgt, ansonsten wird der rechte Zweig genutzt. Zum Schluss wird ein Blattknoten l erreicht, wodurch die Entscheidung der Klassifizierung anhand der Häufigkeiten der enthaltenen Merkmale und ihrer Klassen ermittelt werden kann (Frochte, 2018; Géron, 2020). Es gibt verschiedene Algorithmen, um den Baum anzulernen, darunter zählen bspw. ID3 (Quinlan, 1986) und Classification and Regression Tree (CART) (Breiman et al., 1984).

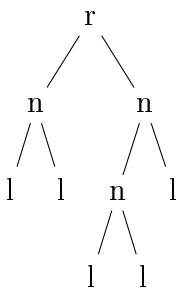


Abbildung 2.4: Binärer Entscheidungsbaum

In der Arbeit wird der CART-Algorithmus verwendet, sodass die Verarbeitung von quantitativen Merkmalen möglich ist. Der Algorithmus erzeugt Binärbäume und nutzt als Maß die Gini Impurity, welche die Unreinheit eines Knotens i beschreibt. Ein Knoten ist rein ($G_i = 0$), wenn sämtliche Merkmale zu einer Klasse k gehören. Das Ziel von CART ist es, dass die Unreinheit reduziert wird und damit die Trainingsmenge bei jeder Entscheidung homogener wird. Die Unreinheit kann mit folgender Formel und der Anzahl von Merkmalen einer Klasse in einem Knoten $p_{i,k}$ berechnet werden (Frochte, 2018; Géron, 2020).

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2 \tag{2.8}$$

Mithilfe der Formel kann ein Entscheidungsbaum trainiert werden. Dafür müssen die Daten anhand eines Merkmals m und eines Schwellenwertes t_m in zwei Untermengen aufgeteilt werden. Die Parameter werden ausgewählt, sodass die reinsten Untermengen erreicht werden. Der CART-Algorithmus minimiert die folgende Funktion mit n_l und n_r für die Anzahl der Merkmale in der linken bzw. rechten Untermenge (Frochte, 2018; Géron, 2020).

$$J_{m,t_m} = \frac{n_l}{n} \cdot G_l + \frac{n_r}{n} \cdot G_r \tag{2.9}$$

Nach der Berechnung der Untermengen wird das Verfahren rekursiv angewendet bis ein Abbruch-Kriterium erfüllt ist. Das Kriterium kann z. B. die maximale Tiefe des Baums sein oder es ist keine weitere Reduzierung der Unreinheit möglich. Wenn der Baum erstellt wurde, kann er zum Schluss rückwärts durchlaufen werden, um Zweige, die kaum zur Verkleinerung des Fehlers beitragen, zu entfernen und durch Blattknoten zu ersetzen. Das Bottom-Up-Verfahren nennt man Pruning (Frochte, 2018; Géron, 2020).

2.3.4 Bewertung

Für die Bewertung muss ein Datensatz in Trainings- und Testdaten aufgeteilt sein. Die Leistung eines Modells, welches anhand der Trainingsdaten erzeugt wurde, wird mithilfe der Testdaten bestimmt. In dem folgenden Abschnitt werden grundlegende Qualitätsmaße zur Beurteilung eines Modells erläutert (Burkov, 2019), welcher wahlweise gelesen werden kann.

Confusion Matrix

Die Confusion Matrix (dt. Konfusion Matrix) ist eine Tabelle zur Darstellung des Erfolgs eines Modells bei der Vorhersage der Klassenzugehörigkeit (Burkov, 2019). In der Abbildung 2.5 wird eine Confusion Matrix für eine binäre Klassifikation abgebildet.

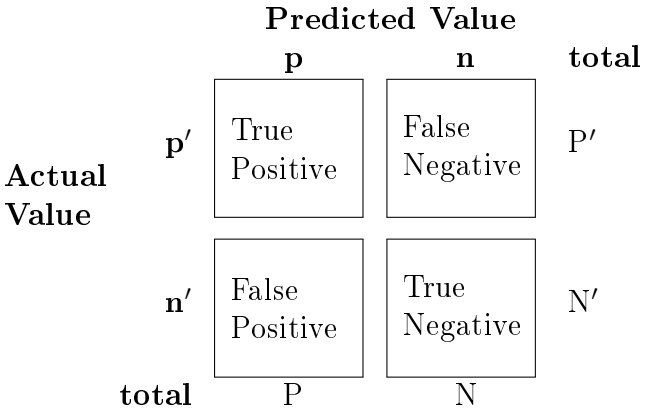


Abbildung 2.5: Confusion Matrix

In den Zeilen werden die tatsächlichen und in den Spalten die vorhergesagten Klassen dargestellt. Die genauen Bedeutungen der Zellen werden in der folgenden Aufzählung gelistet (Burkov, 2019; Géron, 2020).

True Positive (TP): Anzahl der *positiven* Texte, die *positiv* klassifiziert sind

True Negative (TN): Anzahl der *negativen* Texte, die *negativ* klassifiziert sind

False Positive (FP): Anzahl der *negativen* Texte, die *positiv* klassifiziert sind

False Negative (FN): Anzahl der *positiven* Texte, die *negativ* klassifiziert sind

Die Visualisierung ermöglicht einen schnellen Blick auf die Klassifikationsergebnisse. Ein perfekter Klassifikator hat nur die Werte der Hauptdiagonalen ungleich 0 (Burkov, 2019; Géron, 2020).

Accuracy

Die Accuracy (dt. Korrektklassifizierungsrate) ist ein einfaches Bewertungsmaß für den Klassifikator. Es wird berechnet durch die Division der Summe aller richtig klassifizierten Texte und der Gesamtzahl aller klassifizierten Texte, welches in der folgenden Formel mathematisch ausgedrückt wird (Burkov, 2019).

$$Accuracy = \frac{TP + TN}{P + N} \tag{2.10}$$

Man erhält ein Ergebnis zwischen 0 und 1. Ein hoher Wert nahe 1 zeigt, dass viele Texte richtig klassifiziert und ein niedriger Wert nahe 0, dass wenige Texte korrekt klassifiziert wurden (Burkov, 2019).

Das Maß ist ein nützliches Kriterium, wenn alle Klassen gleich wichtig sind, da es keine Unterscheidung zwischen den Fehlern FP und FN gibt (Japkowicz, 2006). Ansonsten sind andere Bewertungsmaße wie z. B. Precision (dt. Relevanz) oder Recall (dt. Sensitivität) zu nutzen (Géron, 2020).

Receiver Operating Characteristic

Die Receiver Operating Characteristic (ROC)-Kurve ist eine Kombination aus der False Positive Rate (FPR) und der True Positive Rate (TPR) bzw. Recall. TPR berechnet sich durch $\frac{TP}{P'}$ und beschreibt die Anzahl der richtig vorhergesagten positiven Texte. FPR ist die Anzahl der falsch klassifizierten negativen Texte und ergibt sich aus $\frac{FP}{N'}$. Durch die Kombination ist die Trennung der Leistung zwischen positiver und negativer Klasse möglich (Burkov, 2019; Géron, 2020). Die folgende Abbildung stellt eine beispielhafte ROC-Kurve dar.

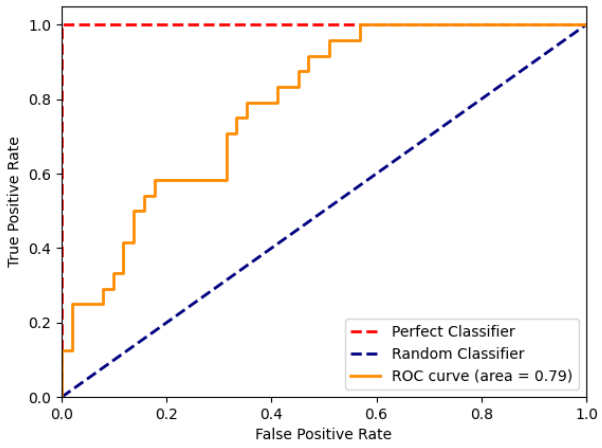


Abbildung 2.6: Receiver Operating Characteristic

In der Abbildung wird die ROC-Kurve eines zufälligen Klassifikators durch die gestrichelte blaue Linie und eines perfekten Klassifikators durch die gestrichelte rote Linie dargestellt. Ein guter Klassifikator liegt möglichst nahe an dem perfekten und weit weg vom zufälligen Klassifikator. Außerdem ist in der Grafik zu erkennen, dass je größer TPR ist, desto größer wird auch FPR (Burkov, 2019; Géron, 2020).

Die Klassifikatoren lassen sich mithilfe der Area Under Curve (AUC) vergleichen. Das Ergebnis ist ein Wert zwischen 0 und 1. Je größer der Wert ist, desto besser ist die Klassifikation. Für den zufälligen Klassifikator ergibt sich eine AUC von 0,5 und für den perfekten Klassifikator ist der Wert 1 (Burkov, 2019; Géron, 2020).

Um die ROC-Kurve nutzen zu können, müssen die Klassifikatoren in der Lage sein die Wahrscheinlichkeiten der Vorhersagen zurückzugeben. Außerdem sollte die Kurve nur genutzt werden, wenn die positive Klasse nicht selten vorkommt und wenn FP nicht wichtiger als FN ist, ansonsten kann die Precision-Recall-Kurve verwendet werden (Géron, 2020; Japkowicz, 2006).

2.4 Sentiment Analysis

In dem Abschnitt wird der Begriff Sentiment Analysis definiert. Es werden die unterschiedlichen Levels der Analyse und die Vektorisierung des Textes beschrieben.

2.4.1 Definition

Die *Sentiment Analysis* oder auch Opinion Mining genannt, ist ein Teilbereich des NLP und ein semantisches Problem. Es ist ein aktives Forschungsgebiet des NLP mit dem Ziel einen geschriebenen Text von einer Person über eine Entität wie z. B. ein Produkt, eine Person oder eine Dienstleistung zu analysieren und deren Stimmung zu erkennen. Für die Analyse des Textes ist kein vollständiges Verständnis des Textes notwendig, da der Fokus auf der Stimmung liegt. Sie kann negativ, neutral oder positiv sein. Neutral beschreibt die Abwesenheit einer Stimmung. Außerdem werden neben subjektiven Ausdrücken auch Ausdrücke ohne implizites Gefühl berücksichtigt, weil neutrale Beschreibungen bzw. objektive Sätze eine Stimmung vermitteln können (Liu, 2015, 2012).

Eine einfache Definition der Sentiment Analysis ist die Ermittlung aller Quadrupel (g, s, h, t) eines Dokumentes. Das Sentiment Target g ist die Entität, ein Teil oder ein Attribut der Entität, auf welches die Stimmung bezogen ist. Der Opinion Holder h ist die Person oder Organisation, die die Meinung vertritt und das Sentiment s ist das zugrundeliegende Gefühl. Die Zeit t beschreibt den Zeitpunkt an dem die Meinung veröffentlicht wurde, um einen möglichen Verlauf der Meinung darstellen zu können (Liu, 2015, 2012). Ein einfaches Beispiel ist der folgende Satz: „The film was great and I'll watch it again in a minute". Der Autor der Film-Rezension schreibt, dass der Film sehr schön war und er ihn noch einmal sehen möchte. Demnach ist zu schlussfolgern, dass das Sentiment positiv ist.

Das Problem der Sentiment Analysis kann bspw. mithilfe von Machine Learning (siehe Abschnitt 2.3) oder eines Lexikons mit positiven und negativen Einträgen gelöst werden (Liu, 2015, 2012).

Die Klassifikation des Sentiment über eine Domäne hinaus wird als *cross-domain bzw. domänenübergreifende Sentiment Analysis* beschrieben. Die Klassifikation reagiert empfindlich auf die Domäne. Ein Wort oder ein Sprachkonstrukt und dessen Stimmung kann sich von Domäne zu Domäne unterscheiden. Die Klassifikatoren, die mit den Trainingsdaten einer Domäne trainiert wurden, erzielen oft schlechte Ergebnisse für Testdaten einer anderen Domäne. Außerdem kommt erschwerend hinzu, dass ein Wort unterschiedliche Stimmungen in verschiedenen Domänen ausdrücken kann. Die Lösung bietet das Transfer Learning, um die Modelle für die neue Domäne verwenden zu können.

Es werden keine bzw. eine kleine Menge gelabelter Trainingsdaten für die neue Domäne genutzt. Die ursprüngliche Domäne wird auch als Quelldomäne und die neue Domäne als Zieldomäne bezeichnet (Liu, 2015, 2012).

2.4.2 Level der Analyse

Man unterscheidet drei verschiedene Ebenen an Granularität bei der Sentiment Analysis, die in dem folgenden Abschnitt erläutert werden.

Bei dem *Document Level* wird ein ganzes Dokument nach deren negativen und positiven Stimmung klassifiziert. Die neutrale Klasse wird aufgrund einer einfacheren binären Klassifikation ignoriert. Die Voraussetzung ist, dass das Dokument eine Meinung zu einer einzelnen Entität von einer einzelnen Person ausdrückt. Die Restriktion ermöglicht keine Untersuchung einzelner Entitäten, um die über sie ausgedrückte Stimmung zu ermitteln (Liu, 2015, 2012).

Das *Sentence Level* ermöglicht die Bestimmung des Sentiments eines einzelnen Satzes. Dadurch kann jeder Satz eine andere Stimmung haben und die Klassifikation ist aufgrund der geringen Informationen in einem Satz schwieriger. Zur Lösung der Aufgabe wird als Erstes überprüft, ob ein Satz eine Meinung ausdrückt. Anschließend kann die Klassifikation des Satzes in die negative und positive Klasse erfolgen (Liu, 2015, 2012).

Das letzte Level ist das *Aspect Level*. Es betrachtet die Meinungen und deren Ziele und entdeckt Stimmungen zu Entitäten (z. B. Restaurant) und/oder deren Aspekte (z. B. Service des Restaurants). Zur Lösung der Aufgabe müssen die Entitäten und deren Aspekte extrahiert werden. Anschließend können die Aspekte nach der Stimmung klassifiziert werden. Schlussendlich wird eine Zusammenfassung von Meinungen über Entitäten und/oder deren Aspekte erstellt (Liu, 2015, 2012).

Auf Grundlage der Datensätze, des Ziels den kompletten Text als positiv bzw. negativ zu klassifizieren und einer einfacheren domänenübergreifenden Sentiment Analysis wird in der Arbeit für das traditionelle maschinelle Lernen eine dokumentenbasierte Sentiment Analysis genutzt. Ein Review drückt eine Meinung eines Rezensenten zu einem bestimmten Film aus. Für einen Tweet kann angenommen werden, dass aufgrund der Länge des Textes eine Meinung durch den Autor zu einem bestimmten Thema geäußert wurde.

2.4.3 Text Vectorization

Um die Daten für die Sentiment Analysis nutzen zu können, müssen sie vorher bearbeitet werden wie in Abschnitt 2.1 und Abschnitt 3.1 beschrieben. Damit werden die Datensätze bereinigt und die Komplexität reduziert. Die Auswahl der Wörter hat einen großen Einfluss auf das Ergebnis der Analyse. Nach der Bereinigung der Texte muss die Umwandlung der Tokens in einen strukturierten Merkmalsraum erfolgen, da eine mathematische Modellierung als Teil des Klassifikators verwendet wird. Es gibt gängige Methoden zur Merkmalsextraktion wie TF, TF-IDF, Word2Vec (Goldberg & Levy, 2014) oder Global Vectors for Word Representation (GloVe) (Pennington et al., 2014), die zur Lösung des Verlustes der syntaktischen sowie der semantischen Beziehung genutzt werden können (Kowsari et al., 2019). In der Arbeit soll die Methode TF-IDF aufgrund guter Ergebnisse in der Sentiment Analysis genutzt werden (Madasu & Elango, 2019; Selamat & Zainuddin, 2014; Singh & Shashi, 2019). Sie basiert auf der n-Gramm-Technik.

Ein n-Gramm ist eine Menge von n Wörtern, die in der Reihenfolge im Text vorkommen. Es ist keine Repräsentation eines Textes, jedoch kann es als Merkmal zur Repräsentation eines Textes verwendet werden (Sarkar, 2016).

Die Methode Term Frequency-Inverse Document Frequency (TF-IDF) besteht aus zwei Komponenten. Die erste Komponente ist TF, welches die grundlegende Form der gewichteten Wortmerkmalsextraktion ist. Es wird jedes Wort auf eine Zahl abgebildet, die der Anzahl des Wortes im Text entspricht. Der Text wird in einen Vektor mit den Häufigkeiten der Wörter umgewandelt. Für $n = 1$ wird es auch als Bag of Words (BoW) bezeichnet. Die vereinfachte Darstellung des Textes führt zum Verlust der Reihenfolge der Wörter und die semantische Beziehung zwischen den Wörtern wird ignoriert. Für $n > 1$ können die extrahierten Merkmale mehr Informationen liefern (Salton & Buckley, 1988; Sarkar, 2016). Ein einfaches Beispiel für BoW sieht wie folgt aus.

I love this film. This film is the best.

BoW = { 'I': 1, 'love': 1, 'this': 2, 'film': 2, 'is': 2, 'the': 1, 'best': 1 }

Das Problem bei der Nutzung von TF ist, dass die häufig vorkommenden Wörter dominieren und die Methode nicht skalierbar ist, wenn das Vokabular zu groß wird, da die Anzahl aller genutzten Wörter die Größe des Vektors bestimmt (Salton & Buckley, 1988).

Aus dem Grund wird die zweite Komponente von TF-IDF genutzt, um das Ergebnis von TF zu erweitern, indem die Worthäufigkeit logarithmisch skaliert wird. Es wird der Effekt der häufig vorkommenden Wörter verringert und die Wörter mit niedriger TF erhalten ein höheres Gewicht. Die Gewichtung eines Terms in einem Dokument d durch TF-IDF wird mit der folgenden Gleichung berechnet (Jones, 1972).

$$W(d,t) = TF(d,t) \cdot log(\frac{N}{df(t)}) \tag{2.11}$$

N ist die Anzahl der untersuchten Dokumente und $df(t)$ ist die Anzahl der Dokumente, die den Begriff t im Korpus enthalten. Die Methode ist eingeschränkt, da die Ähnlichkeit von Wörtern aufgrund der unabhängigen Darstellung nicht berücksichtigt werden (Jones, 1972). Das Problem kann mit komplexeren Modellen wie Word2Vec oder GloVe gelöst werden (Kowsari et al., 2019).

2.5 Datensätze

Die Datengrundlage für die Betrachtungen in der Arbeit bilden der Datensatz Sentiment140[1] (Go et al., 2009) mit Twitter-Nachrichten und der Movie Review Datensatz in Version 2.0[2] (Pang & Lee, 2004) mit Rezensionen zu Filmen. Für die spätere Nutzung der Daten werden die geschriebenen Texte sowie die dazugehörige positive und negative Klasse verwendet. In den folgenden Abschnitten sollen die zu nutzenden Datensätze kurz beschrieben werden.

[1] Abrufbar durch: https://www.kaggle.com/kazanova/sentiment140
[2] Abrufbar durch: https://www.cs.cornell.edu/people/pabo/movie-review-data/

Film-Review

Der Datensatz von Pang und Lee enthält ausführliche und strukturierte Rezensionen zu Filmen von unterschiedlichen Nutzern. Es ist der De-Facto-Standarddatensatz (Aue & Gamon, 2005) mit 1.000 positiven und 1.000 negativen Rezensionen von vor 2002. Er wurde mithilfe eines Detektors erstellt, welcher subjektive Sätze einer Rezension selektieren und objektive Sätze entfernen kann. Anschließend erfolgt die Bewertung der Stimmung anhand des Ratings des Nutzers, welches im Anschluss aus der Rezension entfernt wurde.

Die Struktur des Datensatzes wird durch die zwei Verzeichnisse „pos“ und „neg“ definiert. Die Ordner umfassen die zu verarbeiteten positiven und negativen Textdateien, die pro Zeile einen subjektiven Satz enthalten. Eine Datei repräsentiert eine Rezension. Der Name setzt sich aus einem Tag für die verwendete Kreuzvalidierung (Géron, 2020) aus der Arbeit von Pang und Lee sowie dem Namen der Originaldatei zusammen.

Twitter

Der Social-Networking- und Microblogging-Dienst Twitter ermöglicht es den Benutzern eine Nachricht (Tweet) in Echtzeit zu posten. Der Datensatz Sentiment140 von Siddik besteht aus 0,8 Mio. positiven und 0,8 Mio. negativen Tweets, die über die Twitter-API im Jahr 2009 extrahiert wurden. Ein Tweet ist eine kurze Nachricht mit einer maximalen Länge von 140 Zeichen. Aufgrund der kurzen Nachrichten und der schnellen Möglichkeit des Versendens einer Nachricht kommt es zur Verwendung von Akronymen oder es werden Rechtschreibfehler gemacht. Es werden Emoticons genutzt, um den Gesichtsausdruck und damit die Stimmung zu vermitteln. Außerdem ist es möglich andere Nutzer in dem Tweet zu verlinken sowie Hashtags zur Markierung von Themen zu nutzen, um eine größere Reichweite zu erzielen.

Mithilfe der Twitter-API konnten die Tweets über Abfragen zu verschiedenen positiven und negativen Emoticons wie z. B. „:D“ oder „:(“ gefiltert werden. Die genutzten Emoticons wurden danach aus dem Tweet entfernt. Es wurden weitere Nachrichten gelöscht, die positive und negative Emoticons enthalten, die Tweets eines anderen Nutzers veröffentlichen (Retweet) oder doppelte Tweets aufgrund der wiederholten Abfragen aller zwei Minuten. Außerdem mussten Nachrichten herausgenommen werden, die das Emoticon „:P“ beinhalten, weil die API zum Zeitpunkt der Arbeit Tweets mit dem Emoticon „:P“ für die Abfrage „:(“ zurückgegeben hat. Das Resultat ist eine Datei, die pro Zeile einen Eintrag mit u.a. der Stimmung und der Nachricht enthält.

2.6 Ensemble Learning

Das Ensemble Learning ist eine Technik, um mehrere Klassifikatoren zu einem Klassifikator zusammenzuschließen. Die Gruppe von Klassifikatoren nennt man Ensemble und werden in Abschnitt 2.3.3 beschrieben. Das resultierende Ergebnis ist oft besser als die Vorhersage des besten Klassifikators. Es gibt verschiedene Algorithmen, um ein Ensemble zu trainieren (Géron, 2020). In dem folgenden Abschnitt sollen häufig verwendete Ensemble-Methoden vorgestellt werden (Dietterich, 2000; Opitz & Maclin, 1999; Wan & Yang, 2013; Zhang & Ma, 2012; Zhou, 2021), die gute Ergebnisse in der Sentiment Analysis geliefert haben.

Ein Ensemble kann ein starker Lerner werden und somit eine hohe Genauigkeit erreichen. Die Voraussetzung ist, dass jeder Klassifikator des Ensemble ein schwacher Lerner und dessen Genauigkeit geringfügig größer als zufälliges Raten (> 0.5) ist. Es müssen genügend schwache Lerner genutzt werden, die sich voneinander unterscheiden, um die Chance zu erhöhen, dass man differenzierte Arten von Fehlern erhält (Géron, 2020). Mithilfe von Ensemble-Methoden wird die Anzahl der Fehlklassifikationen für neue Daten gesenkt (Géron, 2020; Opitz & Maclin, 1999).

Es gibt drei Gründe, weshalb die Generalisierungsfähigkeit eines Ensemble besser ist als die eines einzelnen Klassifikators. Der erste Grund ist, dass die Trainingsdaten nicht genügend Informationen liefern, um einen einzelnen besten Klassifikator auszuwählen. Als zweiter Grund wird der Suchprozess des Lernalgorithmus erwähnt. Er ist unvollkommen, weshalb die Algorithmen nicht immer zum korrekten Ergebnis führen. Der letzte Grund ist, dass die Menge aller möglichen Klassifikatoren nicht den wahren Klassifikator enthalten muss (Thomas, 1997; Zhou, 2021).

2.6.1 Bagging

Bootstrap Aggregating oder kurz Bagging (Breiman, 1996) ermöglicht die Kombination von Klassifikatoren unter der Verwendung eines Algorithmus in dem Ensemble. Dazu werden die Daten in unterschiedliche Teilmengen aufgeteilt und nach dem Training wird die genutzte Teilmenge wieder zurückgelegt. Es ist möglich, dass die Daten für einen Klassifikator mehrmals oder Teile der Daten nicht verwendet werden (Out-of-Bag) (Géron, 2020).

Zum Schluss werden die erhaltenen Ergebnisse der Klassifikatoren mithilfe einer Aggregationsfunktion zusammengefasst. Es kann das Hard Voting genutzt werden, wodurch eine mehrheitsbasierte Entscheidung auf Basis des Modalwertes getroffen wird. Eine zweite Möglichkeit ist die Verwendung von Soft Voting. Es nutzt die Wahrscheinlichkeiten der Klasse und das Ergebnis ist die Klasse mit der höchsten über alle einzelnen Klassifikatoren gemittelten Wahrscheinlichkeiten. In Abbildung 2.7 wird die Methode schematisch dargestellt (Géron, 2020).

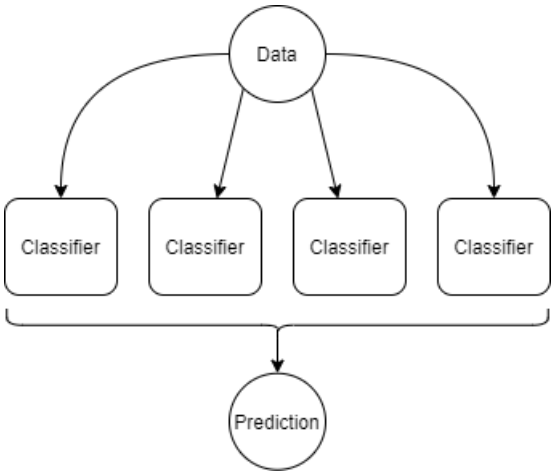


Abbildung 2.7: Bagging Methode

In der Abbildung ist gut zu erkennen, dass die Klassifikatoren parallel ausgeführt werden können, wodurch die Methode gut skalierbar ist. Eine spezielle Implementierung der Bagging Methode ist der Random Forest (Breiman, 2001). Das Ensemble besteht aus Entscheidungsbäumen. Der Algorithmus verwendet bei der Erstellung des Baumes ein Zufallselement, weshalb bei der Aufteilung der Knoten das beste Merkmal in einer zufälligen Teilmenge von Merkmalen ermittelt wird. Das führt zu einer höheren Diversität unter den Bäumen und das Modell liefert gute Ergebnisse (Géron, 2020).

2.6.2 Boosting

Boosting ist eine Kombination von schwachen Lernern, um einen starken Lerner zu erhalten. In dem Ensemble wird ein Algorithmus verwendet. Das Training der Klassifikatoren wird nacheinander ausgeführt, mit dem Ziel den Fehler des Vorgängers zu beheben. Die sequentielle Lernmethode kann nicht gut skaliert werden, weil auf das Training des vorherigen Klassifikators gewartet werden muss. Das Verfahren soll in dem folgenden Abschnitt anhand des AdaBoost Algorithmus (Schapire, 2003) beschrieben werden (Géron, 2020).

Adaptive Boosting oder kurz AdaBoost korrigiert den vorherigen Klassifikator durch die Beachtung der nicht abgedeckten bzw. falsch klassifizierten Trainingsdaten. In Abbildung 2.8 wird der Ablauf schemenhaft dargestellt. Als Erstes wird ein Klassifikator erstellt mit den Anfangsgewichten $\frac{1}{n}$ für die Trainingsdaten. Anschließend können die einzelnen Klassifikatoren trainiert und Vorhersagen getroffen werden. Mithilfe der falsch klassifizierten Trainingsdaten können die gewichtete Fehlerquote r_j und das Gewicht a_j für den j-ten Klassifikator berechnet werden. Je höher das Gewicht des Klassifikators ist, desto genauer ist die Vorhersage. Zum Schluss können die relativen Gewichte der falsch klassifizierten Trainingsdaten erhöht und die Gewichte aller Trainingsdaten normalisiert werden (Géron, 2020).

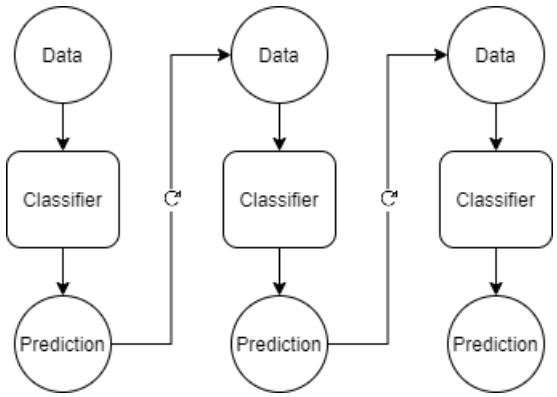


Abbildung 2.8: Boosting Methode

Um eine Vorhersage treffen zu können, berechnet der Algorithmus die Vorhersagen sämtlicher Klassifikatoren und deren Gewichte. Die vorhergesagte Klasse ist die Klasse, welche die Summe der Gewichte der Klassifikatoren $\sum_{j=1}^{N} a_j$ maximiert (Géron, 2020).

Es gibt weitere verschiedene Versionen des AdaBoost Algorithmus wie SAMME oder SAMME.R. SAMME steht für „Stagewise Additive Modeling using a Multiclass Exponential loss function“ und ermöglicht eine Klassifikation mit mehreren Klassen. Bei einer binären Klassifikation ist SAMME äquivalent zu AdaBoost. SAMME.R kann die Wahrscheinlichkeiten der Klassifikatoren nutzen, um eine Vorhersage zu treffen. Das „R“ steht für „Real“ (Hastie et al., 2009).

2.6.3 Stacking

In der Ensemble-Methode Stack Generalization oder kurz Stacking (Wolpert, 1992) werden die Vorhersagen aller Klassifikatoren in einem Ensemble mithilfe eines weiteren Klassifikators, statt einer trivialen Funktion wie beim Bagging, zusammengefasst. Der Klassifikator wird auch „Blender“ oder „Meta-Learner“ genannt. Die genutzten Algorithmen in einem Ensemble können verschieden sein (Géron, 2020).

Die Abbildung 2.9 stellt den Ablauf des Verfahrens schematisch dar. Es werden die Trainingsdaten in zwei Mengen aufgeteilt. Die erste Teilmenge wird genutzt, um die Klassifikatoren auf der ersten Stufe zu trainieren. Anschließend können mit der zweiten Teilmenge die trainierten Algorithmen getestet werden, um zu prüfen wie gut die Vorhersagen sind. Zum Schluss sind die Ergebnisse der Klassifikatoren aus der zweiten Teilmenge die Eingabewerte für das Training des Meta-Learner, welcher die endgültige Vorhersage berechnet (Géron, 2020).

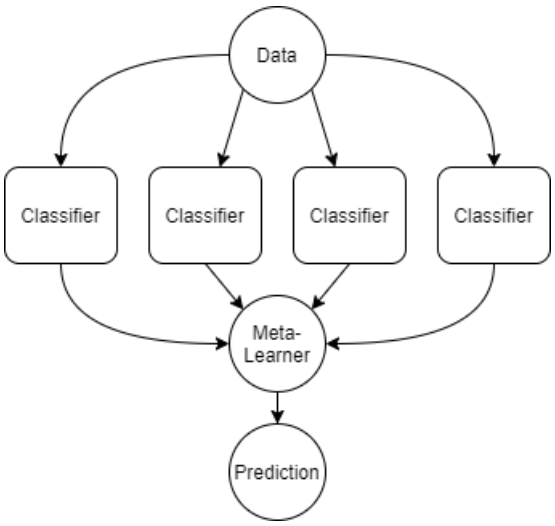


Abbildung 2.9: Stacking Methode

Durch die Abbildung kann man sehen, dass eine Skalierung möglich ist, indem die Klassifikatoren einzeln ausgeführt werden. Zusätzlich ist die Nutzung weiterer Meta-Learner möglich unter der Bedingung, dass genügend Daten vorhanden sind (Géron, 2020).

2.7 Transformer

Ein Transformer ist ein Deep-Learning-Verfahren und eine Architektur zur Umwandlung einer Sequenz in eine andere mithilfe von Encodern und Decodern. In früheren Arbeiten zur Sequenzmodellierung wurde das Framework Sequenz-zu-Sequenz (Sutskever et al., 2014) mit Techniken wie Recurrent Neural Network (RNN) (Graves, 2013) und Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) verwendet. Die Basis der Architektur eines Transformers ist der Attention-Mechanismus (Vaswani et al., 2017) zur Bestimmung der wichtigen Sequenzen für jeden Rechenschritt. In der folgenden Abbildung ist der Aufbau der Architektur dargestellt.

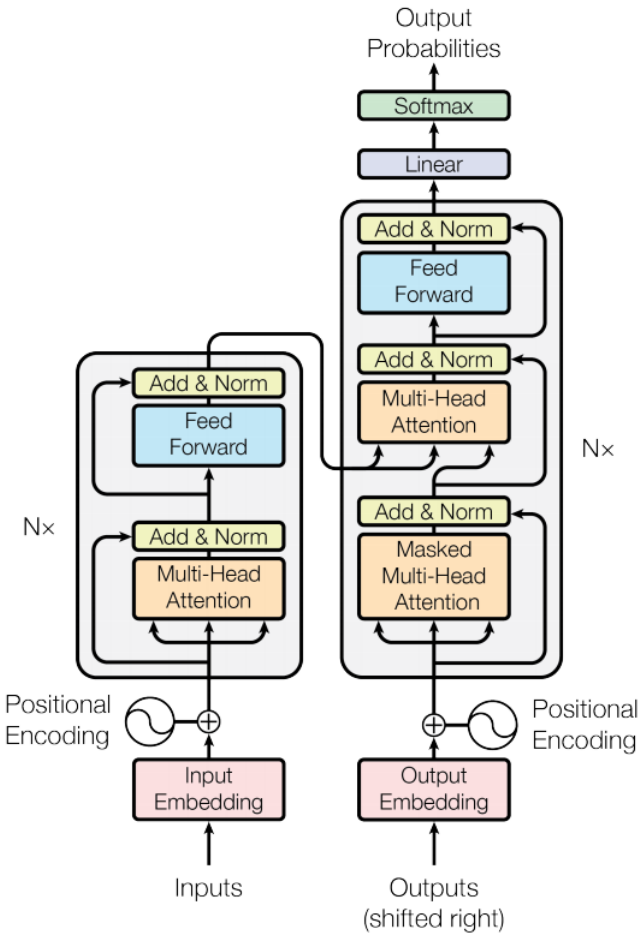


Abbildung 2.10: Architektur eines Transformers (Vaswani et al., 2017)

Der Encoder wird auf der linken und der Decoder auf der rechten Seite abgebildet. Sie bestehen aus unterschiedlichen Komponenten, die mehrfach übereinander gestapelt werden können. Ein Encoder besteht aus dem Self-Attention Mechanismus und einem neuronalen Feed-Forward-Netzwerk (Schmidhuber, 2014). Der Self-Attention Mechanismus dient zur Erzeugung von Ausgaben durch die Abwägung der Relevanz der Eingabe des vorherigen Encoders. Das neuronale Feed-Forward-Netzwerk ermöglicht die weitere Verarbeitung durch lineare Transformationen eines jeden Elementes der Sequenz. Die Ausgabe wird an den nächsten Encoder und Decoder weitergegeben. Der erste Encoder erhält als Eingabe die Positionsinformationen und die vektorisierte Eingabesequenz, die notwendig sind, damit der Transformer die Reihenfolge der Sequenz nutzen kann (Vaswani et al., 2017).

Der Decoder besteht aus dem Self-Attention Mechanismus, dem Attention-Mechanismus für die Encodings und einem neuronalen Feed-Forward-Netzwerk. Die Funktionsweise des Decoders ist vergleichbar mit der des Encoder mit dem Unterschied, dass der Decoder ein zusätzliches Element nutzt, um die relevanten Informationen aus den erhaltenen Encodings zu extrahieren. Der erste Decoder erhält auch die Positionsinformationen und die vektorisierten Ausgangssequenzen, die maskiert und eins nach rechts verschoben werden müssen, sodass die Vorhersage mit der Ausgabe nicht möglich ist. Nach dem letzten Decoder wird eine letzte lineare Transformation durchgeführt und eine Softmax-Schicht genutzt, um die Ausgabewahrscheinlichkeit über das Vokabular zu erstellen (Vaswani et al., 2017).

Die Transformer ermöglichen das Training auf großen Datensätzen durch starke Parallelisierung, wodurch vortrainierte Systeme wie BERT (Devlin et al., 2018) oder Generative Pre-trained Transformer (GPT) (Radford et al., 2019) unüberwacht angelernt werden können. Zur Evaluierung der Systeme gibt es verschiedene Benchmarks wie General Language Understanding Evaluation (GLUE) (A. Wang et al., 2018) oder Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018). Am Ende können die erstellten Modelle auf eine bestimmte Aufgabe wie z. B. Sentiment Analysis oder maschinelle Übersetzung durch überwachtes Lernen fein abgestimmt werden.

2.8 Few-Shot Learning

Das Few-Shot Learning (Y. Wang et al., 2019) ist das Training eines vortrainierten Modells mit einer sehr kleinen Menge an Trainingsdaten für das Transfer Learning. Normalerweise werden Modelle für maschinelles Lernen auf großen Datenmengen trainiert, wie z. B. bei der Feinabstimmung eines Transformers aus Abschnitt 2.7.

Durch das überwachte Lernen des Modells mit nur wenigen Daten ergeben sich unterschiedliche Vorteile. Es sind weniger Daten notwendig, um eine zuverlässige Verallgemeinerung und ein robustes Modell zu erhalten. Außerdem können Kosten reduziert werden, weil die großen Datensätze nicht beschriftet werden müssen und die Notwendigkeit entfällt, dass spezifische Features für verschiedene Aufgaben hinzugefügt werden müssen, wenn ein gemeinsamer Datensatz verwendet wird (Y. Wang et al., 2019).

Man unterscheidet beim Few-Shot Learning zwei besondere Fälle bei der sich die Anzahl der Trainingsdaten mit Labels auf einen (One-Shot Learning) bzw. gar keinen Datensatz (Zero-Shot Learning) beschränkt (Y. Wang et al., 2019).

Kapitel 3

Methodik

Die Methodik beschreibt die Durchführung der Methoden. Es sollen mithilfe eines Experiments quantitative Ergebnisse ermittelt werden, um neue Erkenntnisse zu gewinnen und die Forschungsfrage zu beantworten. In der folgenden Abbildung wird der Ablauf des Experiments dargestellt.

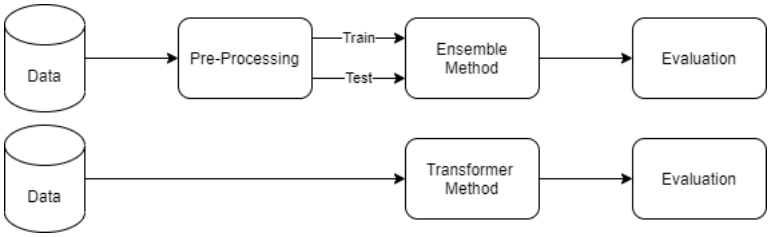


Abbildung 3.1: Big Picture

Zu Beginn des Experiments werden die originalen Datensätze heruntergeladen. Die Daten für das Ensemble Learning werden vorverarbeitet und in einen Vektor von Zahlen transformiert. Anschließend werden die Daten in Trainings- und Testdaten aufgeteilt. Für das Transfer Learning werden die Reviews als Trainingsdaten und ein prozentualer Anteil der Tweets als zusätzliche Trainingsdaten genutzt. Das Ensemble von (unterschiedlichen) Klassifikatoren wird mit einer Ensemble-Methode trainiert, um anschließend mithilfe der Testdaten die Leistung zu visualisieren und auszuwerten.

Für die Transformer werden die heruntergeladenen Datensätze verwendet. Anschließend kann bspw. das fein abgestimmte Modell auf die Review-Daten genutzt werden, um die Klassifikation der Tweets durchzuführen. Am Ende werden die Ergebnisse visualisiert und ausgewertet. In den folgenden Abschnitten werden die dargestellten Schritte beschrieben.

3.1 Vorverarbeitung

Die Vorverarbeitung der Daten für das Ensemble Learning soll auf Basis der beschriebenen Schritte aus Abschnitt 2.1 erfolgen mit dem Unterschied, dass statt dem Stemming die Lemmatisierung (Anandarajan et al., 2019) genutzt werden soll. Die Lemmatisierung löst das Problem, dass ein einzelnes Wort in Abhängigkeit vom Kontext oder der Wortart (POS) mehrere Bedeutungen haben kann, indem die Wortart in die Regeln zur Gruppierung der Stammformen einbezogen wird. Dadurch werden separate Regeln für mehrdeutige Wörter in Abhängigkeit von der Wortart ermöglicht. Jedoch basiert die Verbesserung auf Kosten der zusätzlichen Komplexität.

Für die Auswahl der einzelnen Schritte wurde ein Experiment umgesetzt, um die Auswirkung der Vorverarbeitungsschritte auf den jeweils anderen Datensatz zu ermitteln. Im Anhang A sind die Ergebnisse des Experimentes zusammengefasst. Aufgrund der gleichbleibenden Resultate unter der Nutzung von Unigrammen und der besseren Ergebnisse gegenüber Bi- und Trigrammen sollen die Vorverarbeitungsschritte aus den Experimenten 5 und 6 verwendet werden, sodass irrelevante Daten entfernt sind. Es werden folgende Schritte genutzt: Tokenization, Lemmatisierung, Erweiterung von Akronymen, Ersetzung von Negationen, Entfernen von Stoppwörtern, Entfernen von Nummern, Entfernen von bedeutungslosen Wörter und Entfernen von Wörtern mit weniger als drei Zeichen. Zusätzlich wird weiteres Rauschen entfernt durch das Löschen von Satzzeichen, Links und Nutzernamen (Jianqiang & Xiaolin, 2017).

Für einen kleinen Einblick in die vorverarbeiteten Film-Rezensionen und Twitter-Nachrichten stellen die Abbildungen 3.2 und 3.3 jeweils die zwanzig häufigst vorkommenden Wörter dar. Daraus ist zu erkennen, dass das Wort „film“ mit 11.168 und das Wort „movie“ mit 6.978 am häufigsten in den Rezensionen vorkommt. Für die Tweets sind es die Wörter „going“ mit 125, gefolgt von „day“ mit 105 und „good“ mit 103 Vorkommnissen.

Aufgrund der begrenzten Anzahl der Review-Daten aus dem Standarddatensatz soll die Anzahl der Tweets ebenfalls beschränkt werden. Es sollen 1.000 positive und 1.000 negative Tweets genutzt werden, die zufällig aus dem ganzen Twitter-Datensatz genommen werden.

Am Ende der Vorverarbeitung für die Ensemble-Methoden werden die Daten in Trainings- und Testdaten aufgeteilt. Die Trainingsdaten bestehen aus den Review-Daten und einem prozentualen Anteil von 0 % bis 50 % der Twitter-Daten. Dadurch kann das Transfer Learning umgesetzt werden und es bleiben genügend Testdaten zur Evaluierung übrig. Anschließend können die Daten mit TF-IDF vektorisiert werden.

Für die Transformer wird keine Vorverarbeitung der Daten ausgeführt. Der Grund ist, dass die vortrainierten Modelle keine Vorverarbeitung durchgeführt haben. Jedoch wird die Anzahl der Testdaten auf 1.000 mit 500 positiven und 500 negativen Tweets durch die begrenzte Anzahl an Testdaten für die Ensemble-Methoden beschränkt.

3.2 Methoden

Die Umsetzung des Ensemble Learning soll mithilfe der Ensemble-Methoden aus Abschnitt 2.6 und der Klassifikatoren aus Abschnitt 2.3.3 ermöglicht werden. Für die Methode Bagging und AdaBoost soll der Klassifikator SVM genutzt werden, weil er in unterschiedlichen Arbeiten (Khalid et al., 2020; Pang et al., 2002) die besten Ergebnisse für die Sentiment Analysis geliefert hat. Dabei ist zu beachten, dass ein Ensemble aus schwachen Lernen bestehen muss. Um die Bedingung zu erfüllen, wird der RBF-Kernel verwendet (Li et al., 2008). Außerdem soll die Aggregationsfunktion mit Soft Voting umgesetzt werden, wodurch eine bessere Vorhersage als mit Hard Voting getroffen werden kann (Géron, 2020). Für die Methode Random Forest wird der Algorithmus Decision Tree verwendet. Die Ensemble-Methode Stacking nutzt die Klassifikatoren SVM, Naive Bayes und Logistic Regression.

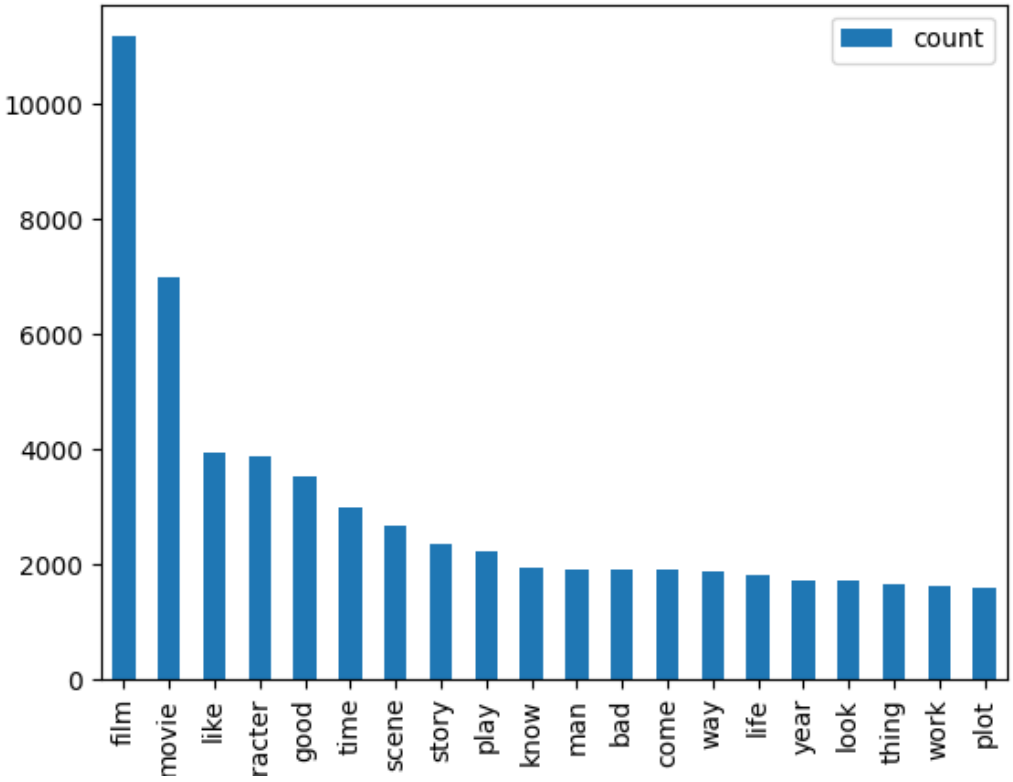


Abbildung 3.2: Häufigsten Wörter in Film-Rezensionen

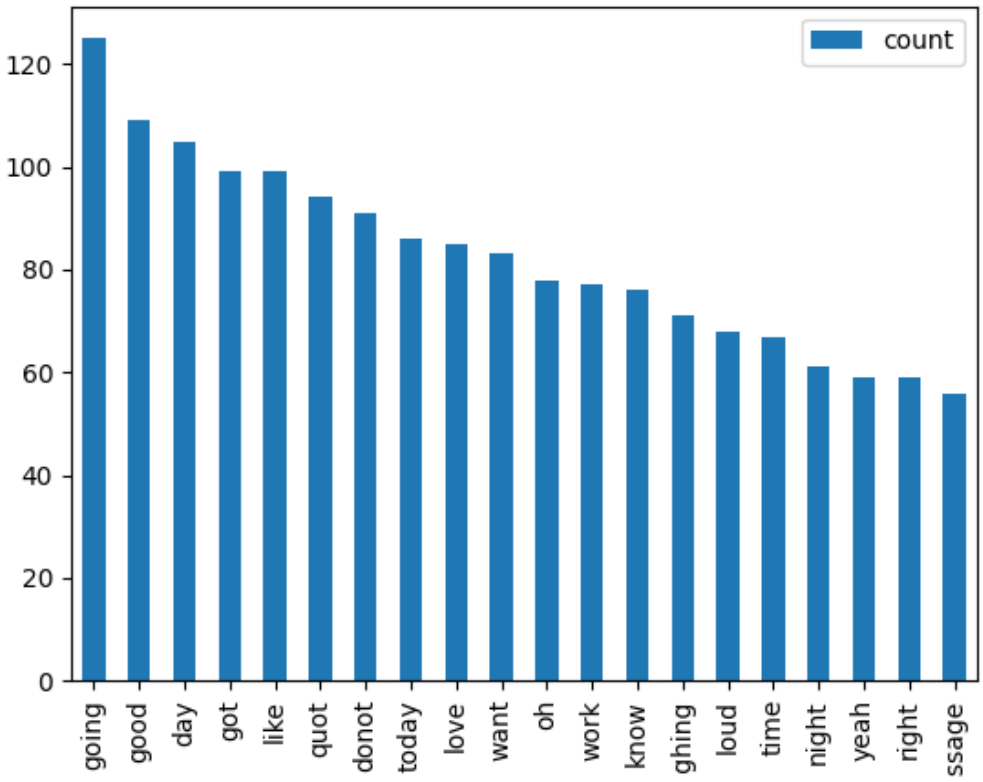


Abbildung 3.3: Häufigsten Wörter in Tweets

Für das Transfer Learning unter der Nutzung von Transformern soll ein fein abgestimmter Transformer genutzt werden, der eine gute Accuracy für die Sentiment Analysis von über 90 % erreicht hat und das BERT Modell (Devlin et al., 2018) nutzt, sodass ein Vergleich mit dem Modell von Siebert et al. (2019) möglich ist. Der gewählte Transformer wird häufig verwendet, basiert auf DistilBERT (Sanh et al., 2019) und wurde nur auf Review-Daten angepasst. Bei den Experimenten zu den Transformern wird auf existierende Modelle zurückgegriffen, um den Umfang der Arbeit zu beschränken.

Zum Schluss soll das Few-Shot Learning genutzt werden. Der Vorteil bei der Weiterentwicklung von Modellen ist, dass die Abhängigkeit von großen Mengen an annotierten Daten für nachgelagerte Aufgaben langsam abnimmt. Dadurch können extrem große Sprachmodelle bei nachgelagerten Aufgaben mit weit weniger aufgabenspezifischen Daten konkurrenzfähig sein (Brown et al., 2020). Für die Umsetzung wurde sich für einen Transformer entschieden, der auf Bidirectional and Auto-Regressive Transformer (BART) (Lewis et al., 2019) basiert, Accuracies von 96 % erreicht und auf dem Multi-Genre Natural Language Inference (MLNI)-Datensatz trainiert wurde, wodurch eine leistungsfähigere Feinabstimmung erreicht werden kann (Yin et al., 2019). Es soll das Zero-Shot Learning umgesetzt werden, um die allgemeine Machbarkeit zu zeigen.

3.3 Auswertung

Die Bewertung der Klassifikation soll mit den verschiedenen Qualitätsmaßen aus Abschnitt 2.3.4 ermöglicht werden. Die Confusion Matrix wird genutzt, um einen Überblick über die Klassifikationsergebnisse zu erhalten. Das Maß Accuracy gibt die korrekt klassifizierten Texte an und kann genutzt werden, weil alle Klassen gleich wichtig sind (Japkowicz, 2006) und um mit anderen Arbeiten vergleichbar zu sein. Außerdem soll die ROC-Kurve verwendet werden, um die Unterschiede der Klassifikation in den verschiedenen Klassen zu ermitteln. Es kann genutzt werden, weil die Klassifikatoren die Wahrscheinlichkeiten der Klassen berechnen können und die Datensätze ausgeglichen sind (Japkowicz, 2006).

Für den Vergleich der Ergebnisse des Transfer Learning sollen die Ensemble-Methoden mit jeweils 70 % der Tweets bzw. Film-Reviews trainiert und jeweils 30 % aufgrund der geringen Anzahl an Daten evaluiert werden. Außerdem sollen die knappen Entscheidungen während der Klassifikation mit den Ensemble- und Transformer-Methoden aufsummiert werden, um zu erkennen, ob die Ergebnisse zufällig sind. Der Ausdruck „knapp“ bedeutet, dass die Differenz zwischen den Wahrscheinlichkeiten der Klassen kleiner gleich 5 ist. Der Wert wurde vom Autor festgelegt, aufgrund dessen, dass eine Vorhersage der Klasse mit bspw. den Wahrscheinlichkeiten $(0.475, 0.525)$ als zufällig bezeichnet werden soll.

3.4 Verwendete Software

Die Entwicklung des Projektes soll mit der Programmiersprache Python umgesetzt werden. Es gibt andere Möglichkeiten das Projekt zu entwickeln, wie die Nutzung von RapidMiner (Mierswa et al., 2006) oder Weka (Frank et al., 2016). Aufgrund der Erfahrungen und des einfacheren Einstiegs wurde sich für Python entschieden.

Für die Implementierung des Projektes wurde die Software-Bibliothek scikit-learn (Pedregosa et al., 2011) verwendet. Die Bibliothek wurde für das maschinelle Lernen entwickelt und basiert auf NumPy und SciPy für die mathematischen Berechnungen und auf matplotlib für die visuellen Darstellungen. Eine bessere und komfortablere Möglichkeit zur Datenmanipulation bietet Pandas (The pandas development team, 2021) als Erweiterung von NumPy. Die Analyse von Texten soll mithilfe der NLP-Bibliothek spaCy (Honnibal et al., 2021) ermöglicht werden. Es gibt andere Bibliotheken zur Textanalyse wie z. B. NLTK, die jedoch nicht so gute Ergebnisse für die Accuracy geliefert haben (Omran & Treude, 2017). Für die Umsetzung der Transformer-Methoden wird huggingface (Wolf et al., 2020) verwendet, welches viele vortrainierte Modelle zur Verfügung stellt. Andere Bibliotheken wie bspw. spaCy basieren auf den Modellen von hugginggface. Das Projekt mit der Implementierung des Experimentes und den erhaltenen Ergebnissen wurde auf GitHub[1] hochgeladen .

[1] Projekt: https://github.com/flaxel/sentiment_analysis

Kapitel 4

Ergebnisse

In dem Abschnitt werden die vorgestellten Methoden mit der Programmiersprache Python umgesetzt und die Ergebnisse beschrieben. Es wird geprüft, ob ein Transfer Learning eines mit Review-Daten trainierten Modells auf die Zieldomäne Twitter möglich ist. Die Resultate der Experimente sind die Vorhersagen der negativen und positiven Texte, die mit 0 und 4 kodiert werden. Die folgenden Abschnitte beschreiben die Ergebnisse der einzelnen Methoden.

4.1 Ensemble-Methoden

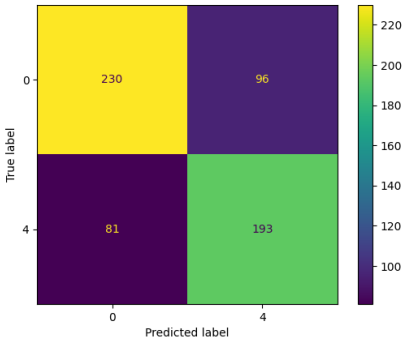
Der Abschnitt stellt die Ergebnisse des Transfer Learning unter Nutzung der Ensemble-Methoden aus Abschnitt 2.6 vor. Es wird zur Auswertung der Methoden ein prozentualer Anteil der Texte verwendet, sodass 1.000 bis 2.000 Texte als Testdaten verwendet werden. In den Abbildungen und Tabellen wird der prozentuale Anteil an Trainingsdaten angegeben. Zusätzlich werden die Resultate der Ensemble-Methoden beschrieben, die nur auf die Tweets bzw. Review-Daten trainiert und evaluiert wurden. Für die Evaluierung werden 600 der 2.000 Texte als Testdaten genutzt.

Bagging

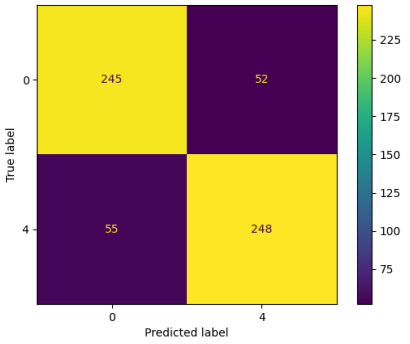
Die erste Ensemble-Methode ist Bagging, die in scikit-learn mit dem BaggingClassifier umgesetzt wurde. Mithilfe der vorhergesagten Klassen kann ausgewertet werden wie gut die Anpassung an die neue Domäne ist. In der Abbildung 4.1 wird ein Überblick über die Klassifikationsergebnisse mit den Confusion Matrixes dargestellt.

Es zeigt, dass die Ensemble-Methoden, die mit Reviews oder Tweets trainiert wurden, weniger Fehlklassifikationen haben als die Ensemble-Methoden für das Transfer Learning. Der prozentuale Anteil der Fehlklassifikationen sinkt stetig nach dem zusätzlichen Training mit den Tweets, jedoch bleibt es mit mehr als 30 % weiterhin hoch. Die resultierenden Accuracies und die Anzahl der knappen Entscheidungen werden in der Tabelle 4.1 zusammengefasst.

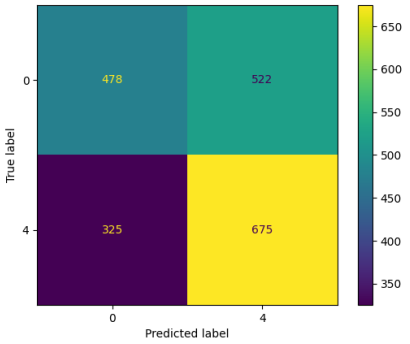
Für die Methode Bagging ergeben sich unter der Nutzung von Tweets oder Reviews gute Accuracies und wenige knappe Entscheidungen während der Klassifikation, die einen prozentualen Anteil von 6,33 % und 3,5 % ausmachen. Die Ergebnisse des Transfer Learning zeigen, dass die Accuracy steigt, je mehr Tweets als Trainingsdaten verwendet werden und die Accuracy der Ensemble-Methode mit Tweets wird erreicht. Jedoch sinkt die Anzahl der knappen Entscheidungen im Verhältnis zur Anzahl der Testdaten nicht kontinuierlich.



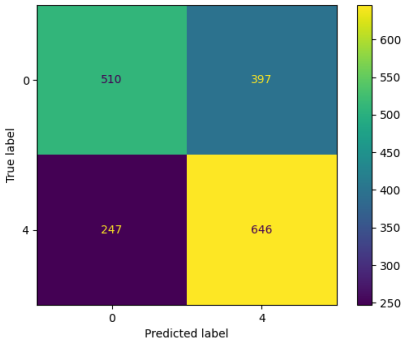
(a) Tweets



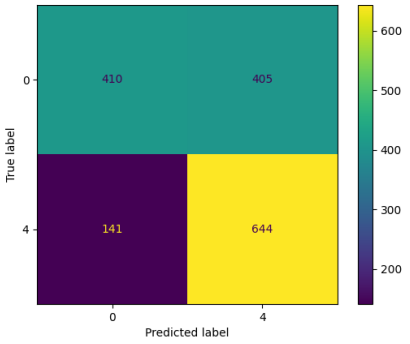
(b) Reviews



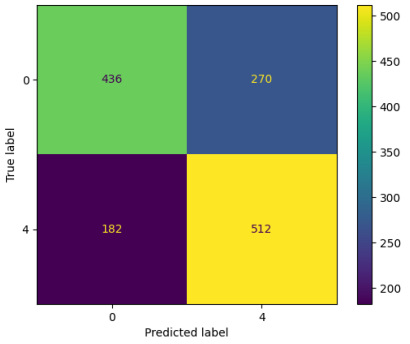
(c) Reviews & Tweets (0 %)



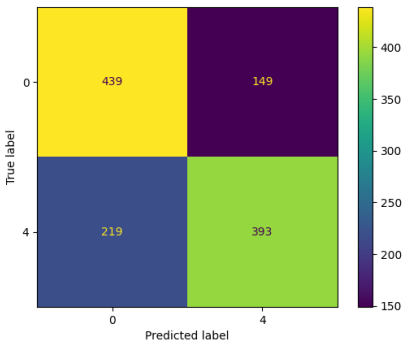
(d) Reviews & Tweets (10 %)



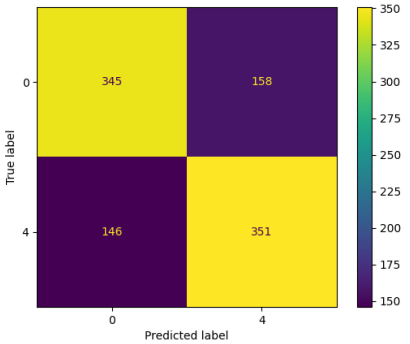
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)

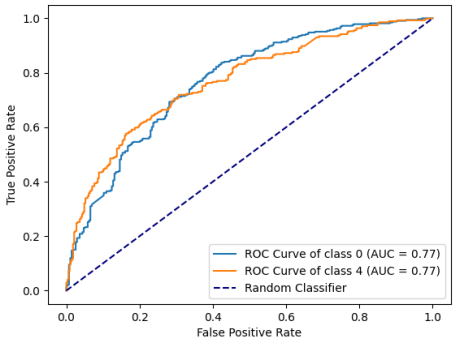


(g) Reviews & Tweets (40 %)

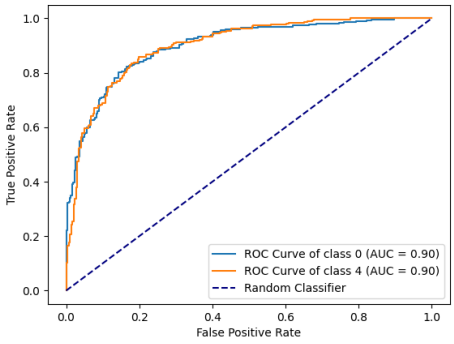


(h) Reviews & Tweets (50 %)

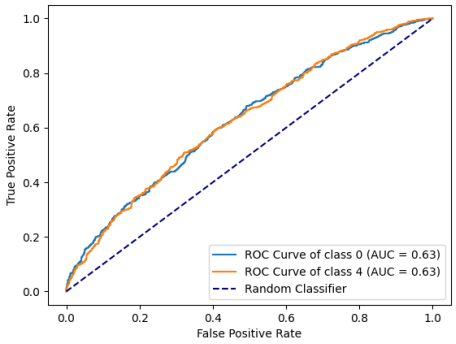
Abbildung 4.1: Confusion Matrixes für Bagging



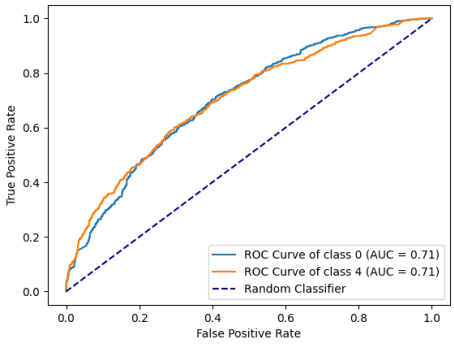
(a) Tweets



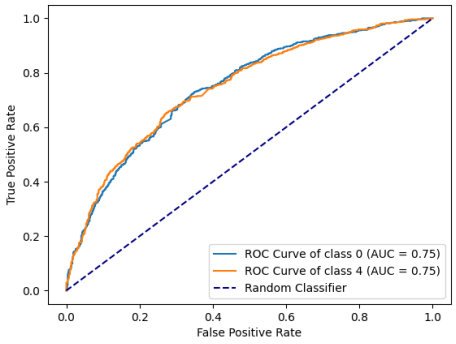
(b) Reviews



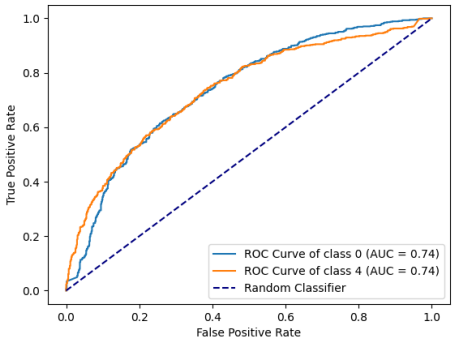
(c) Reviews & Tweets (0 %)



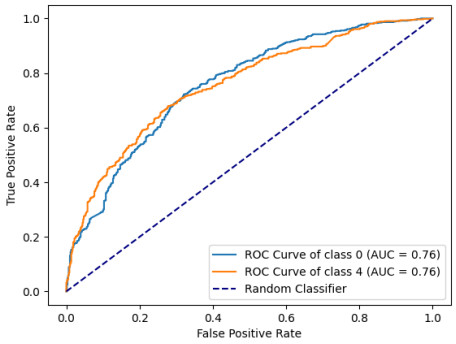
(d) Reviews & Tweets (10 %)



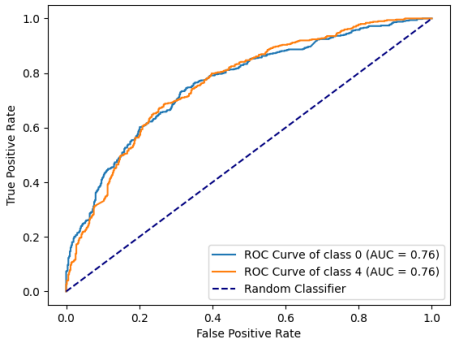
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)



(g) Reviews & Tweets (40 %)



(h) Reviews & Tweets (50 %)

Abbildung 4.2: ROC-Kurven für Bagging

	Accuracy	Close Predictions
Tweets	0,70	38 von 600
Reviews	0,82	21 von 600
Reviews & Tweets (0 %)	0,58	227 von 2.000
Reviews & Tweets (10 %)	0,64	132 von 1.800
Reviews & Tweets (20 %)	0,66	98 von 1.600
Reviews & Tweets (30 %)	0,68	77 von 1.400
Reviews & Tweets (40 %)	0,69	70 von 1.200
Reviews & Tweets (50 %)	0,70	65 von 1.000

Tabelle 4.1: Ergebnisse für Bagging

Der beste Wert wird für die Nutzung von 30 % der Tweets als Trainingsdaten mit 5,5 % erreicht und ist damit geringer als für die Ensemble-Methode mit Tweets.

Die Abbildung 4.2 repräsentiert die erhaltenen ROC-Kurven. Sie zeigen, dass sich die Leistung der Methode nicht in den Klassen unterscheidet. Die besten Werte für AUC werden für die mit Tweets bzw. Review-Daten trainierten Ensemble-Methoden erreicht. Durch das Transfer Learning und der wachsenden Anzahl an Tweets für das Training steigt der Wert für AUC bis die Differenz zur Ensemble-Methode mit Tweets 0,01 beträgt.

Random Forest

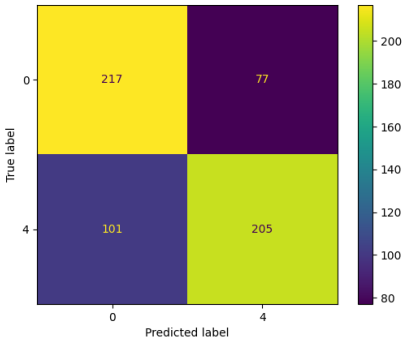
Die nächste Ensemble-Methode ist Random Forest, die in scikit-learn mit dem RandomForestClassifier genutzt werden kann. Die Ergebnisse der Klassifikation werden ausgewertet und in Abbildung 4.3 wird ein Überblick mithilfe der Confusion Matrixes dargestellt.

Die Abbildungen zeigen, dass durch das Training der Ensemble-Methode mit Tweets der prozentuale Anteil der Fehlklassifikation stetig gesenkt werden kann, jedoch mehr als 31 % der Daten nicht korrekt klassifiziert werden. Die Ensemble-Methoden, die nur auf Tweets oder Reviews trainiert wurden, liefern vergleichsweise einen niedrigeren prozentualen Anteil an Fehlklassifikationen. Weitere Werte zur Beurteilung der Leistung des Modells sind in Tabelle 4.2 zusammengefasst.

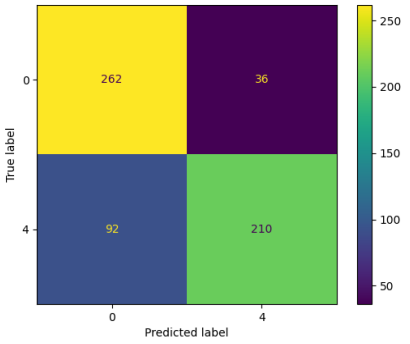
	Accuracy	Close Predictions
Tweets	0,70	43 von 600
Reviews	0,79	75 von 600
Reviews & Tweets (0 %)	0,55	681 von 2.000
Reviews & Tweets (10 %)	0,59	107 von 1.800
Reviews & Tweets (20 %)	0,62	110 von 1.600
Reviews & Tweets (30 %)	0,66	103 von 1.400
Reviews & Tweets (40 %)	0,68	118 von 1.200
Reviews & Tweets (50 %)	0,69	72 von 1.000

Tabelle 4.2: Ergebnisse für Random Forest

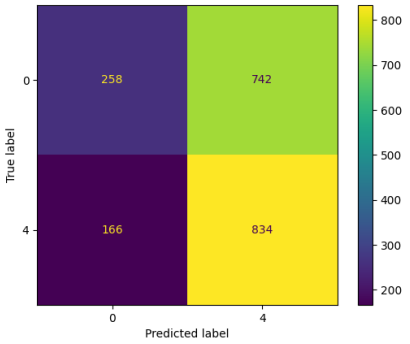
Durch die Tabelle wird gezeigt, dass die Ensemble-Methoden, die auf Tweets oder Reviews trainiert wurden, gute Accuracies erreichen. Die Ensemble-Methode mit dem Transfer Learning zeigt unter der zunehmenden Verwendung von Tweets als Trainingsdaten steigende Accuracies und nähert sich weiter der Accuracy der Ensemble-Methode mit Tweets an.



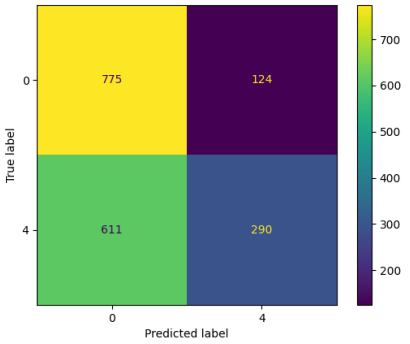
(a) Tweets



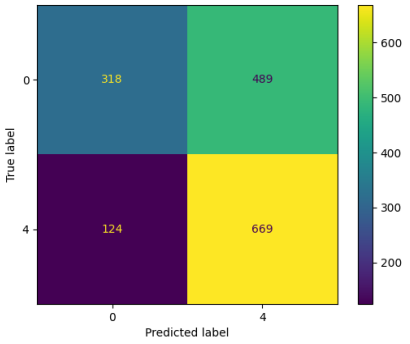
(b) Reviews



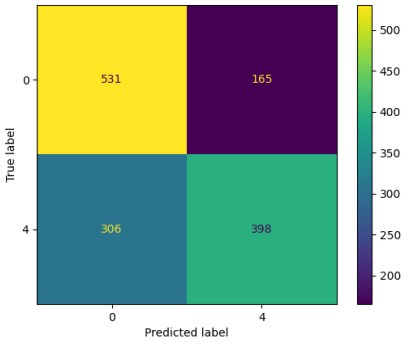
(c) Reviews & Tweets (0 %)



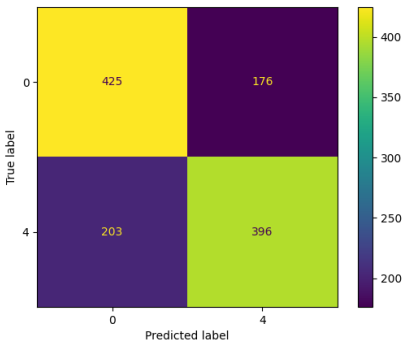
(d) Reviews & Tweets (10 %)



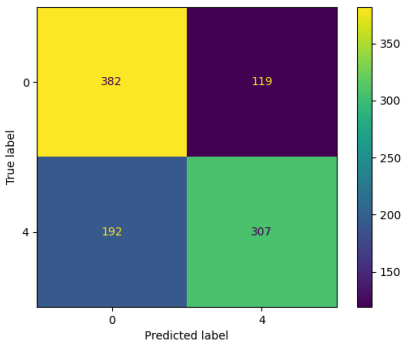
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)

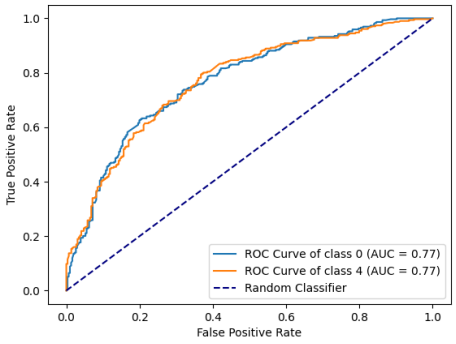


(g) Reviews & Tweets (40 %)

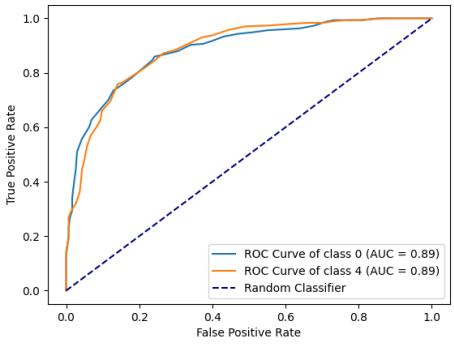


(h) Reviews & Tweets (50 %)

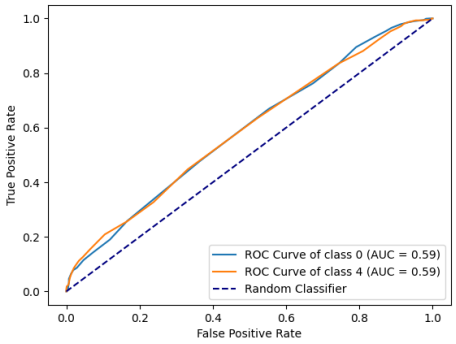
Abbildung 4.3: Confusion Matrixes für Random Forest



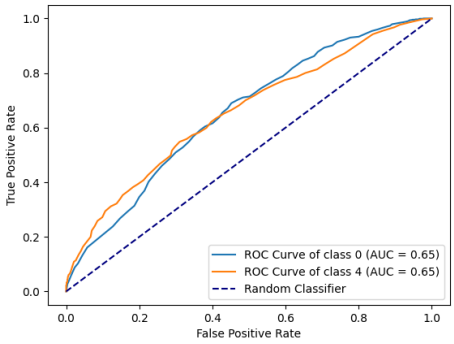
(a) Tweets



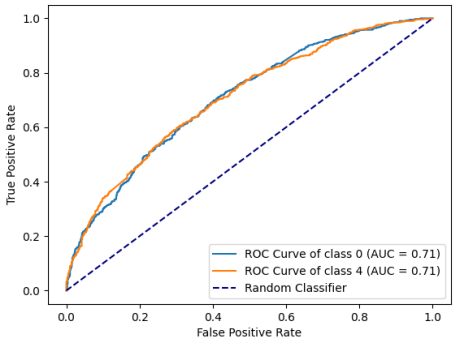
(b) Reviews



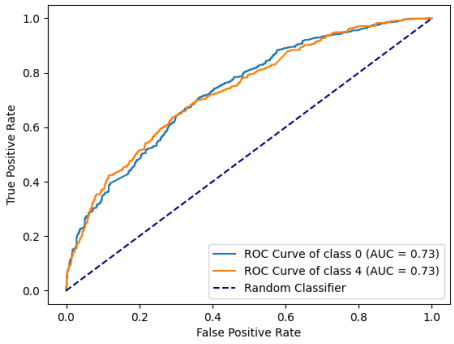
(c) Reviews & Tweets (0 %)



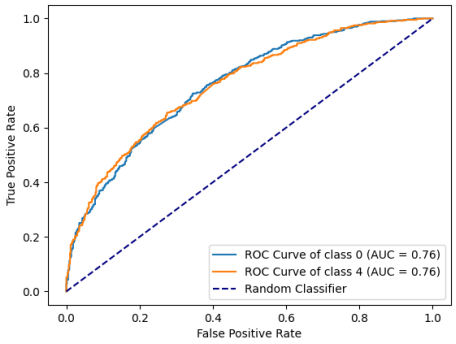
(d) Reviews & Tweets (10 %)



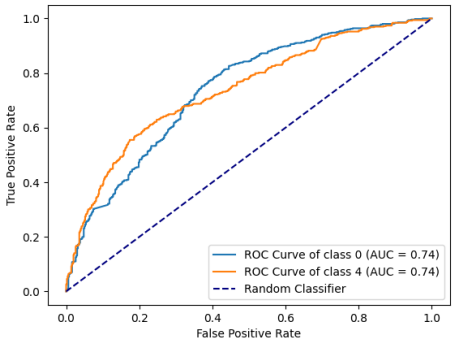
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)



(g) Reviews & Tweets (40 %)



(h) Reviews & Tweets (50 %)

Abbildung 4.4: ROC-Kurven für Random Forest

Die Anzahl der knappen Entscheidungen wird durch die Nutzung der Tweets für das Training stark gesenkt. Jedoch bleibt die Anzahl weiterhin hoch. Im Vergleich zu der Ensemble-Methode mit Tweets (7,16 %) und Review-Daten (12,5 %) ist der prozentuale Anteil der knappen Entscheidungen unter der Verwendung von bspw. 10 % der Tweets als Trainingsdaten mit 5,94 % geringer.

In der Abbildung 4.4 werden die ROC-Kurven dargestellt. Sie zeigen, dass sich die Leistung der Methode zwischen den Klassen nicht unterscheidet. Die besten Werte für AUC werden für die Ensemble-Methoden erreicht, die auf Tweets oder Reviews trainiert wurden. Durch das zusätzliche Training der Ensemble-Methode mit Tweets wird der Wert von AUC erhöht und eine minimale Differenz zur Ensemble-Methode mit Tweets von 0,01 erreicht.

AdaBoost

AdaBoost ist die nächste Ensemble-Methode, die in scikit-learn mit dem AdaBoostClassifier implementiert werden kann. Für das Training und das anschließende Testen benötigte die Methode die meiste Zeit mit 30 min im Vergleich zu den anderen Methoden mit 2 min bis 7 min. Die Resultate der Klassifikation werden zur Auswertung verwendet und einen Überblick über die Klassifikationsergebnisse erhält man mit den Confusion Matrixes aus Abbildung 4.5.

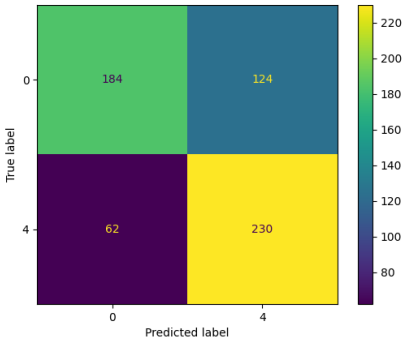
In den Abbildungen ist zu sehen, dass der prozentuale Anteil der Fehlklassifikationen bei den Ensemble-Methoden mit Tweets oder Reviews geringer als bei den Ensemble-Methoden zum Transfer Learning ist. Es zeigt sich, dass durch die Verwendung von Tweets als Trainingsdaten die Fehlklassifikationen gesenkt werden können, aber der Wert bleibt mit mehr als 32 % weiterhin hoch. In der Tabelle 4.3 werden weitere Qualitätsmaße zur Auswertung der Ensemble-Methode zusammengefasst.

	Accuracy	Close Predictions
Tweets	0,69	466 von 600
Reviews	0,83	97 von 600
Reviews & Tweets (0 %)	0,56	1.994 von 2.000
Reviews & Tweets (10 %)	0,63	1.238 von 1.800
Reviews & Tweets (20 %)	0,64	1.470 von 1.600
Reviews & Tweets (30 %)	0,66	1.344 von 1.400
Reviews & Tweets (40 %)	0,68	1.014 von 1.200
Reviews & Tweets (50 %)	0,66	938 von 1.000

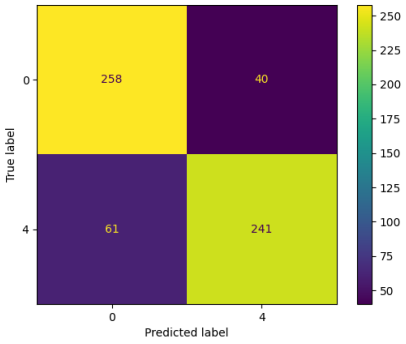
Tabelle 4.3: Ergebnisse für AdaBoost

Durch die Tabelle wird ersichtlich, dass die auf Reviews trainierte Ensemble-Methode eine gute Accuracy erreicht. Die Ergebnisse des Transfer Learning zeigen, dass die Accuracy durch die Verwendung von Tweets als Trainingsdaten verbessert werden kann. Jedoch sind die erzielten Accuracies unter der Verwendung von Tweets zufällig auf Grundlage der Anzahl der knappen Entscheidungen. Fast alle Ergebnisse basieren auf über 90 % knappen Entscheidungen.

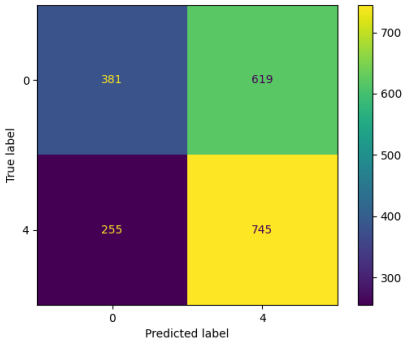
Die ROC-Kurven für die Ensemble-Methode AdaBoost werden in Abbildung 4.6 dargestellt. Sie zeigen, dass sich die Leistung der Methode zwischen den zwei Klassen nicht unterscheidet. Der größte Wert für AUC wird für die Ensemble-Methode erreicht, die mit den Review-Daten trainiert wurde.



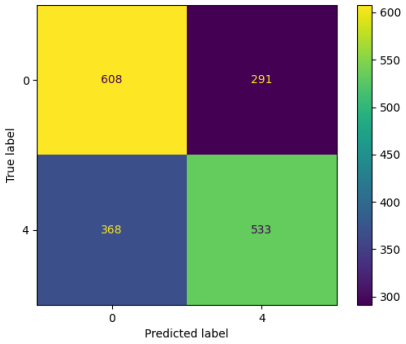
(a) Tweets



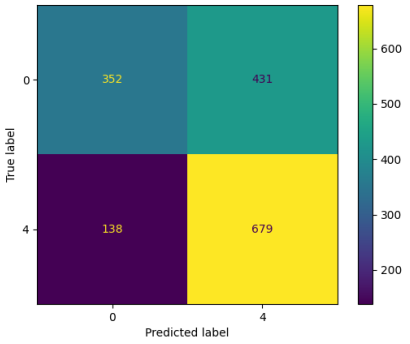
(b) Reviews



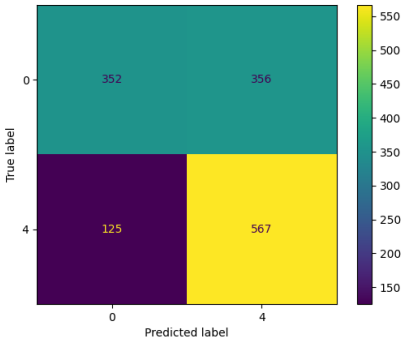
(c) Reviews & Tweets (0 %)



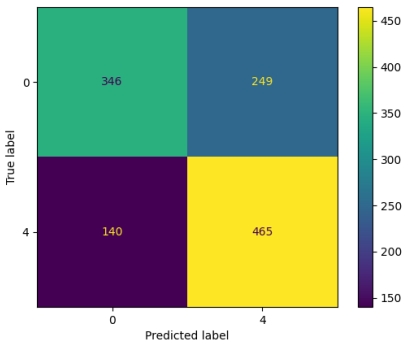
(d) Reviews & Tweets (10 %)



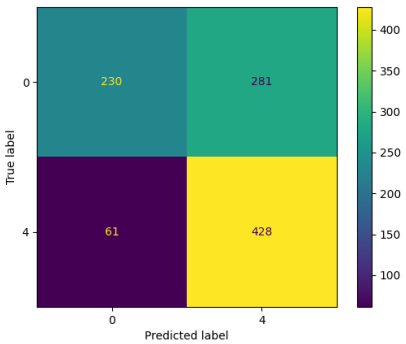
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)

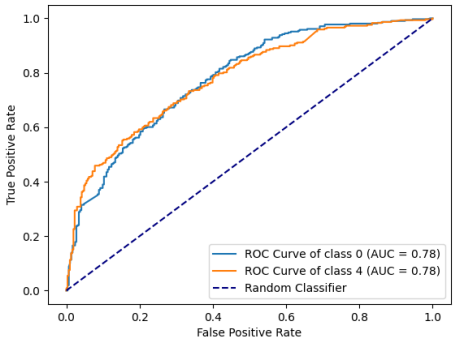


(g) Reviews & Tweets (40 %)

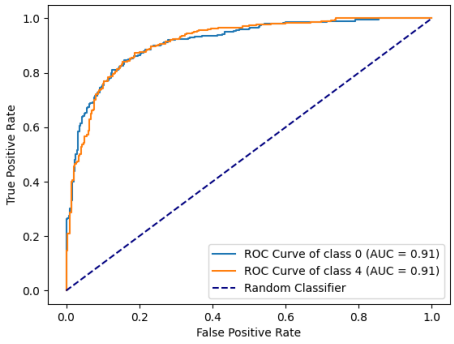


(h) Reviews & Tweets (50 %)

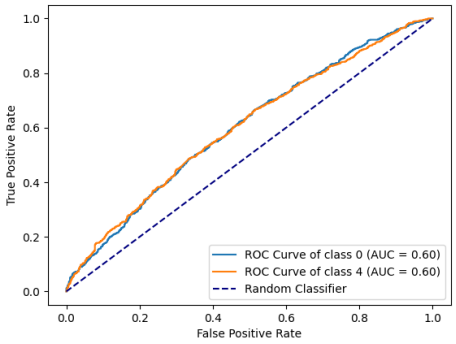
Abbildung 4.5: Confusion Matrixes für AdaBoost



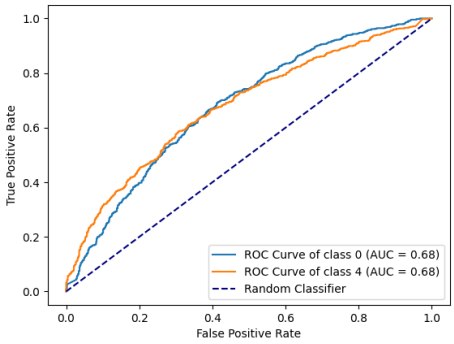
(a) Tweets



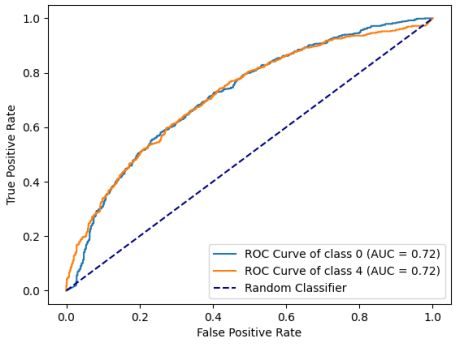
(b) Reviews



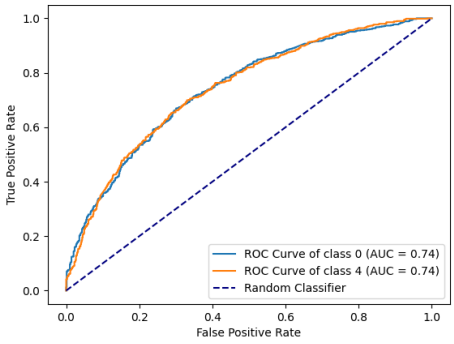
(c) Reviews & Tweets (0 %)



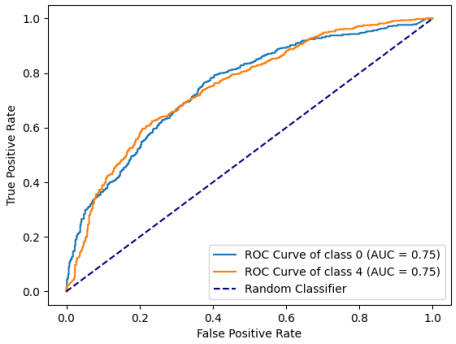
(d) Reviews & Tweets (10 %)



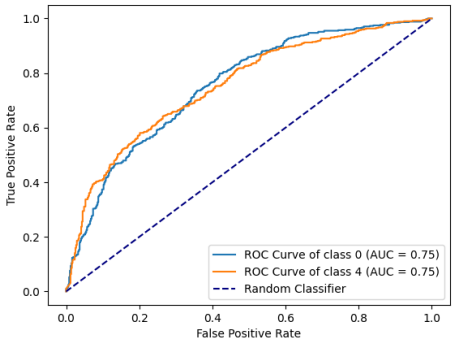
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)



(g) Reviews & Tweets (40 %)



(h) Reviews & Tweets (50 %)

Abbildung 4.6: ROC-Kurven für AdaBoost

Für die mit Tweets trainierten Ensemble-Methoden kann kein zuverlässiger Wert von AUC aufgrund der vielen knappen Entscheidungen getroffen werden.

Stacking

Die letzte Ensemble-Methode ist Stacking, die in scikit-learn mit dem StackingClassifier verwendet werden kann. Mithilfe der erhaltenen Klassen kann die Methode ausgewertet werden und in der Abbildung 4.7 wird ein Überblick der Klassifikationsergebnisse dargestellt.

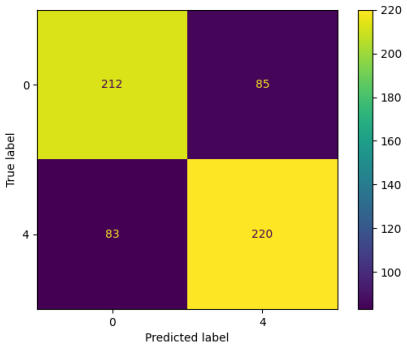
Es ist zu erkennen, dass die auf Reviews oder Tweets trainierten Ensemble-Methoden weniger Fehlklassifikationen haben als die Ensemble-Methoden für das Transfer Learning. Durch das zusätzliche Training mit Tweets kann der prozentuale Anteil an Fehlklassifikationen gesenkt werden. Dennoch ist der Anteil der Fehlklassifikationen mit mehr als 29 % weiterhin hoch. Die sich daraus ergebenden Accuracies und die Anzahl der knappen Entscheidungen werden in Tabelle 4.4 zusammengefasst.

	Accuracy	Close Predictions
Tweets	0,72	32 von 600
Reviews	0,81	11 von 600
Reviews & Tweets (0 %)	0,59	131 von 2.000
Reviews & Tweets (10 %)	0,64	92 von 1.800
Reviews & Tweets (20 %)	0,67	77 von 1.600
Reviews & Tweets (30 %)	0,67	59 von 1.400
Reviews & Tweets (40 %)	0,69	46 von 1.200
Reviews & Tweets (50 %)	0,70	35 von 1.000

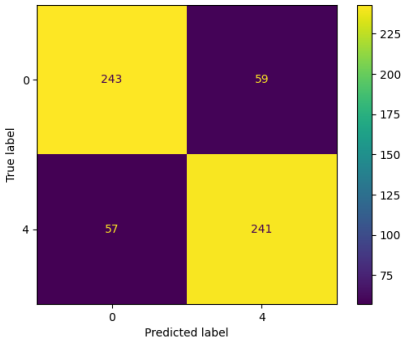
Tabelle 4.4: Ergebnisse für Stacking

Für die Methode Stacking erhält man gute Accuracies für die Ensemble-Methoden mit Tweets oder Reviews. Es ergeben sich wenige knappe Entscheidungen während der Klassifikation, die einen prozentualen Anteil von 5,33 % und 1,83 % bilden. Die Ensemble-Methoden für das Transfer Learning erzielen steigende Accuracies, die sich an den Wert der Ensemble-Methode mit Tweets annähern, und eine sinkende Anzahl an knappen Entscheidungen, je mehr Tweets als Trainingsdaten genutzt werden. Im Vergleich zu der Ensemble-Methode mit Tweets ist der prozentuale Anteil der knappen Entscheidungen ab der Verwendung von 20 % der Tweets als Trainingsdaten geringer.

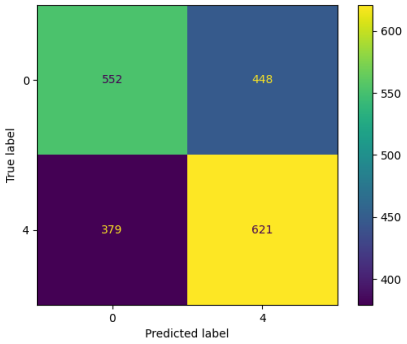
Zum Schluss werden die ROC-Kurven für die Ensemble-Methode Stacking in Abbildung 4.8 dargestellt. Es zeigt, dass sich die Leistung der Methode nicht zwischen den zwei Klassen unterscheidet. Die größten Werte für AUC wurden mit den Ensemble-Methoden erreicht, die mit den Tweets oder Reviews trainiert wurden. Mithilfe des zusätzlichen Trainings der Ensemble-Methode mit den Tweets kann der Wert von AUC erhöht werden und es wird eine minimale Differenz zur Ensemble-Methode mit Tweets von 0,02 erzielt.



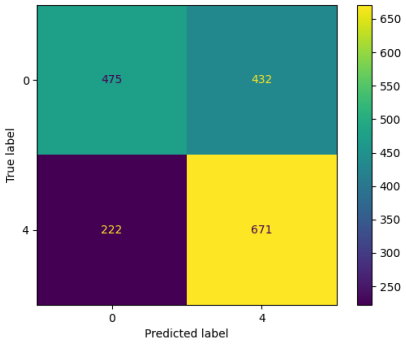
(a) Tweets



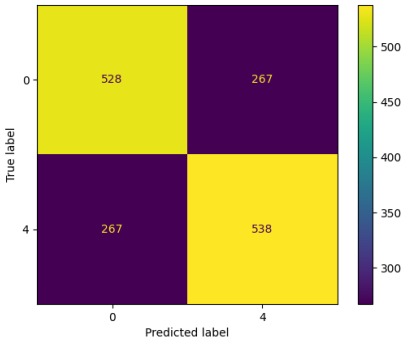
(b) Reviews



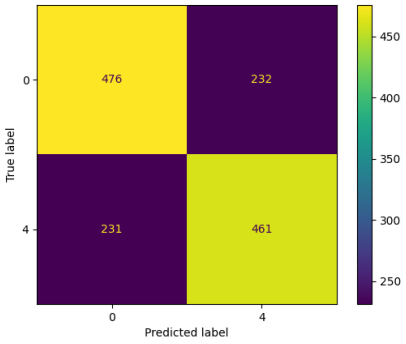
(c) Reviews & Tweets (0 %)



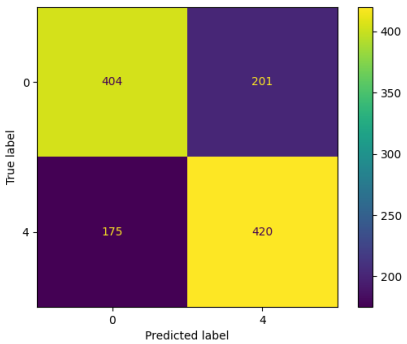
(d) Reviews & Tweets (10 %)



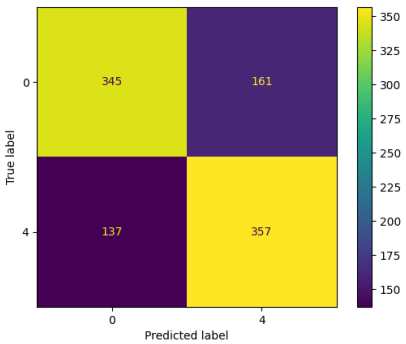
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)

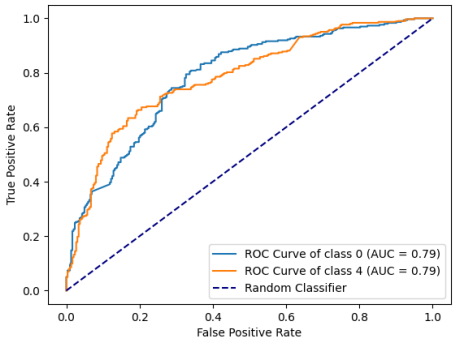


(g) Reviews & Tweets (40 %)

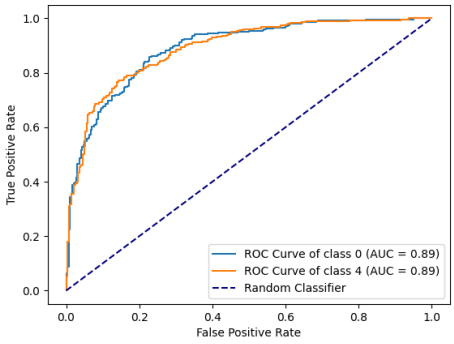


(h) Reviews & Tweets (50 %)

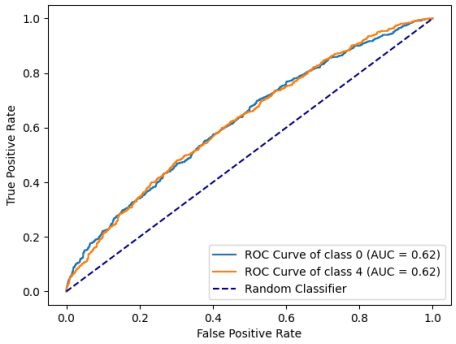
Abbildung 4.7: Confusion Matrixes für Stacking



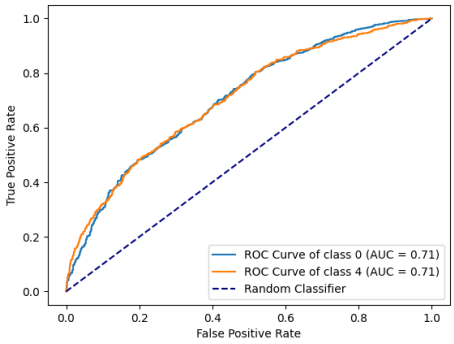
(a) Tweets



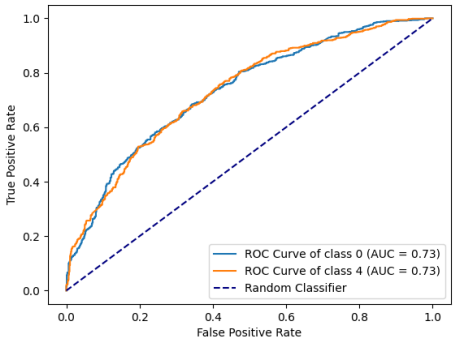
(b) Reviews



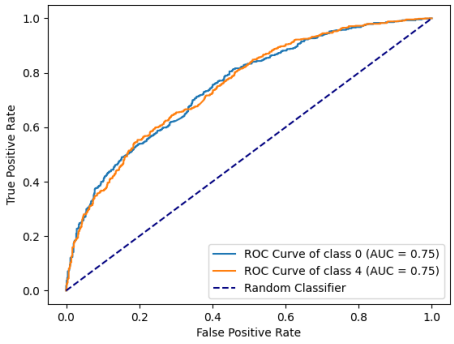
(c) Reviews & Tweets (0 %)



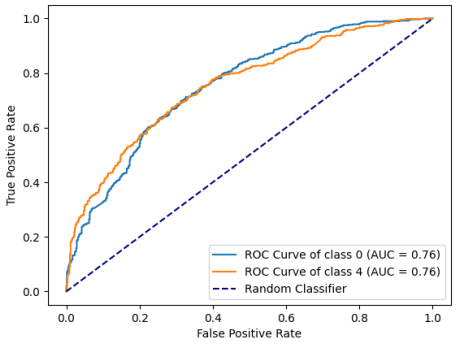
(d) Reviews & Tweets (10 %)



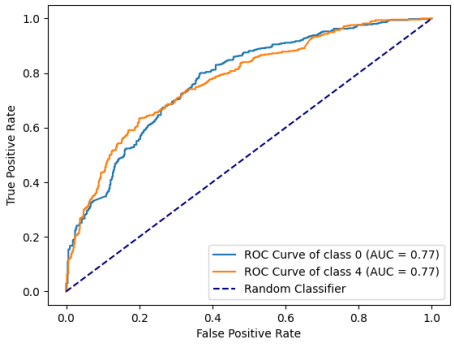
(e) Reviews & Tweets (20 %)



(f) Reviews & Tweets (30 %)



(g) Reviews & Tweets (40 %)



(h) Reviews & Tweets (50 %)

Abbildung 4.8: ROC-Kurven für Stacking

4.2 Transformer-Methoden

Der Abschnitt stellt die Ergebnisse des Transfer Learning unter Nutzung von Transformern aus Abschnitt 2.7 vor. Für die Evaluierung der Transformer werden 1.000 Texte als Testdaten verwendet. Die Umsetzung mit huggingface ist mithilfe der Pipelines „sentiment-analysis“ und „zero-shot-classification“ möglich. Mit den erhaltenen gelabelten Daten kann eine Auswertung erfolgen und einen Überblick über die Ergebnisse der Klassifikation erhält man mit den Confusion Matrixes aus Abbildung 4.9.

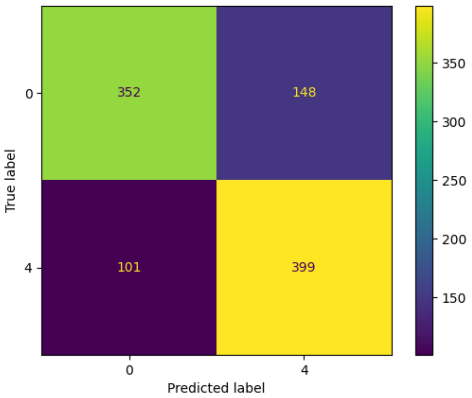
Mithilfe der Confusion Matrixes ist zu sehen, dass die Transformer mit RoBERTa und Zero Shot die wenigsten Fehlklassifikationen im Experiment aufzeigen. Der DistilBERT-basierte Transformer hat auf Grund des Trainings mit Reviews mehr Fehlklassifikationen. Die daraus resultierenden Accuracies und die Anzahl der knappen Entscheidungen während der Klassifikation werden in Tabelle 4.5 gelistet.

	Accuracy	Close Predictions
RoBERTa	0,75	0
DistilBERT	0,70	6
Zero Shot	0,75	39

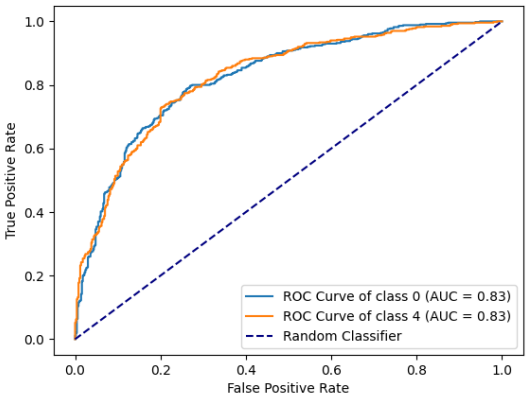
Tabelle 4.5: Ergebnisse für Transformer

Im Vergleich zu den Ensemble-Methoden erhält man für die Transformer mit RoBERTa sowie Zero Shot bessere und mit DistilBERT vergleichbare Accuracies. Die Anzahl der knappen Entscheidungen kann durch die Feinabstimmung der Transformers auf 0 % und 0,6 % gesenkt werden. Für den Transformer mit Zero Shot ist die Menge an knappen Entscheidungen mit 3,9 % vergleichsweise hoch.

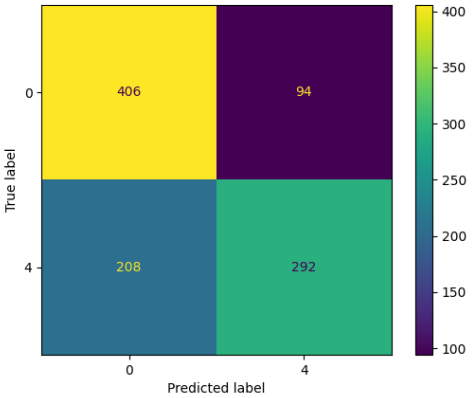
Die Abbildung 4.9 stellt die erhaltenen ROC-Kurven dar. Es ist zu erkennen, dass sich die Leistung der Methoden nicht zwischen den Klassen unterscheidet. Die besten Werte für AUC werden für die Transformer mit RoBERTa und Zero Shot erreicht. Der Wert für den Review-trainierten Transformer mit DistilBERT ist vergleichsweise niedrig.



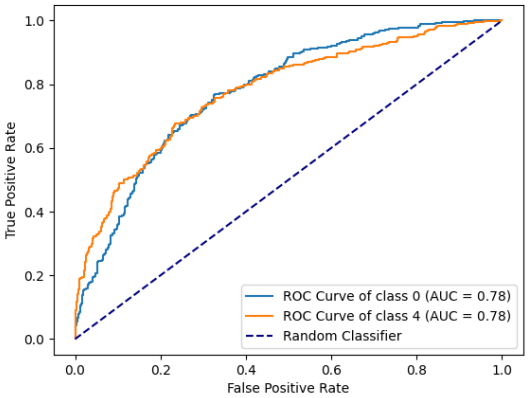
(a) Confusion Matrix für RoBERTa



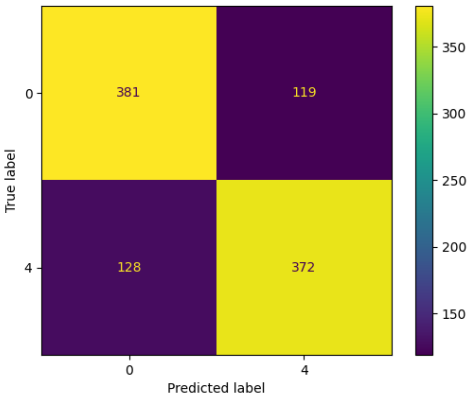
(b) ROC-Kurve für RoBERTa



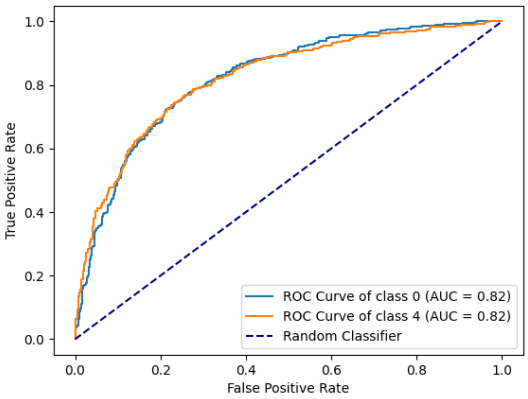
(c) Confusion Matrix für DistilBERT



(d) ROC-Kurve für DistilBERT



(e) Confusion Matrix für Zero Shot



(f) ROC-Kurve für Zero Shot

Abbildung 4.9: Confusion Matrixes und ROC-Kurven für Transformer

Kapitel 5

Diskussion

Im Rahmen der Arbeit wurde das Transfer Learning eines auf Review-Daten trainierten Modells zur Klassifikation von Tweets umgesetzt. Es wurde ein Experiment mit Ensemble- und Transformer-Methoden durchgeführt, die mit Trainingsdaten angelernt bzw. mit Testdaten evaluiert werden konnten. Mit den erhaltenen Klassifikationsergebnissen wird eine Auswertung der Methoden ermöglicht. Der folgende Abschnitt beschreibt die wichtigsten Ergebnisse und deren Interpretation sowie Beschränkungen und weitere Möglichkeiten für Forschungsansätze.

Die Ergebnisse der Klassifikation zeigen, dass die Leistungen der Methoden sich nicht zwischen den Klassen unterscheiden. Die Ensemble-Methoden für das Transfer Learning erzielen eine Accuracy von über 70 % und nähern bzw. erreichen die Werte der Ensemble-Methode mit Tweets. Der prozentuale Anteil von knappen Entscheidungen ist mit überwiegend 3 % bis 10 % gering. Es fällt auf, dass die Anzahl der knappen Entscheidungen der Ensemble-Methoden für das Transfer Learning kleiner als die der Ensemble-Methoden für Tweets ist. Die Ensemble-Methode AdaBoost konnte die Ergebnisse aufgrund der vielen knappen Entscheidungen nicht erzielen, weshalb kein zuverlässiges Ergebnis erhalten werden kann.

Für die Transformer-Methoden können bessere Werte für die Qualitätsmaße als für die Ensemble-Methoden erreicht werden. Es werden Accuracies von bis zu 75 % erzielt und die Anzahl der knappen Entscheidungen tendiert gegen 0. Jedoch bleiben die Ergebnisse hinter den suggerierten Werten der Publikationen der Transformer (Lewis et al., 2019; Siebert et al., 2019).

Es konnten die Erwartungen für die Ensemble-Methoden erfüllt werden, da die Werte der Qualitätsmaße der Ensemble-Methoden mit Transfer Learning annähernd gleich der Werte der Ensemble-Methode mit Tweets sind. Jedoch konnten die Werte für die Ensemble-Methode mit Reviews nicht erreicht werden. Ein Grund könnte die fehlende Struktur, fehlende Satzzeichen z. B. für „totallysweet.biz“ oder Rechtschreibfehler wie „cannott“ bzw. „soooooomebody“ in den Nachrichten von Twitter sein.

Außerdem enthält ein Merkmalsvektor von Twitter aufgrund der Länge des Textes und der Vorverarbeitungsschritte weniger Merkmale als ein Merkmalsvektor eines Reviews, weshalb weniger Wörter für die Klassifikation zur Verfügung stehen. Es ist jedoch aufgefallen, dass die Ensemble-Methoden für das Transfer Learning weniger knappe Entscheidungen liefern, wodurch die Ergebnisse der Methoden robuster sind.

Dennoch konnten die Ergebnisse der Ensemble-Methoden eine Accuracy von über 80 % wie in den Arbeiten von Aue und Gamon (2005) und Peddinti und Chintalapoodi (2011) nicht erreichen. Daraus kann geschlussfolgert werden, dass die Ergebnisse für eine domänenübergreifende Sentiment Analysis abhängig von der verwendeten Methode und der Quell- bzw. Zieldomäne sind. Die Resultate könnten verbessert werden, indem bspw. die Merkmale für die Klassifikation selektiert und die Kompatibilität der Domänen überprüft werden.

Die Erwartungen an die Transformer-Methoden durch die Ergebnisse der Publikationen (Lewis et al., 2019; Siebert et al., 2019) von über 90 % Accuracy konnten nicht erreicht werden. Der Grund sind zwei aktuelle Probleme in der Forschung: „(a) high unexplained variability in accuracies scattered across publications, and (b) state-of-the-art reports on popular benchmarks that can suggest levels of performance that may not be attainable for all real-world applications“ (Siebert et al., 2019). Aus den Gründen ist ein Einsatz in der Praxis zur aktuellen Zeit nicht vorstellbar ohne das Auftreten der unterschiedlichen Ergebnisse zu begreifen.

Durch die Ergebnisse hat sich gezeigt, dass ein Review-trainiertes Modell gute Ergebnisse für die Klassifikation von Tweets im Vergleich zu den Ensemble-Methoden liefert. Die Werte für die Qualitätsmaße können durch die Feinabstimmung auf die Twitter-Daten verbessert werden. Es fällt auf, dass bei der Verwendung von RoBERTa und DistilBERT fast keine knappen Entscheidungen getroffen werden, wodurch das Modell robuster wird. Die Transformer-Methode mit Zero-Shot Learning erzielte gute Accuracies, aber mehr knappe Entscheidungen, weshalb die Methode ohne Feinabstimmung weniger robuster ist.

Im Vergleich zu den Ensemble-Methoden konnten die Transformer-Methoden bessere Ergebnisse für die Qualitätsmaße erzielen. Die Ursache ist u.a. die Größe des Datensatzes, mit welchem das Modell trainiert wurde oder in Bezug auf die Ebenen von NLP die Nutzung von Positionsinformationen der Wörter als Eingabe für den Transformer. Außerdem konnte gezeigt werden, dass die Feinabstimmung eines Modells mit Tweets bzw. das Zero-Shot Learning die besten Ergebnisse in der Arbeit geliefert haben. Es kann geschlussfolgert werden, dass durch ein zusätzliches bzw. umfängliches Training ein konkurrenzfähiges Modell erstellt werden kann, welches zuverlässige und fast keine knappen Entscheidungen liefert.

Die Lesenden sollten beachten, dass die Arbeit auf Ensemble-Methoden, die in früheren Arbeiten gute Ergebnisse für die Sentiment Analysis geliefert haben und auf Transformer-Methoden, die vortrainiert und bereits existierend sind, basieren. Weitere Methoden wurden in der Arbeit nicht berücksichtigt, um den Umfang der Arbeit zu beschränken.

Außerdem wurden durch den Standarddatensatz der Film-Reviews nur subjektive Sätze verwendet, die entweder eine positive oder negative Stimmung vermitteln. Objektive bzw. neutrale Sätze wurden nicht betrachtet, die vor der Sentiment Analysis mithilfe von Subjectivity Detection gefiltert werden können. Die Anzahl der Daten für das Training und das Testen des Modells wurden auf 2.000 Datensätze aufgrund der Größe des Datensatzes von Film-Reviews beschränkt. Für die Sentiment Analysis mit dem traditionellen maschinellen Lernen wurde die Annahme getroffen, dass der Autor sich zu einer Entität äußert, um die dokumentenbasierte Analyse durchführen zu können und es wurde TF-IDF genutzt, wodurch die Ähnlichkeit von Wörtern durch die unabhängige Darstellung nicht berücksichtigt wird.

Zukünftige Forschungsvorhaben könnten an die Forschungsarbeit anknüpfen, indem größere Datensätze verwendet werden und für die Ensemble-Methoden eine erweiterte Vorverarbeitung genutzt wird, um z. B. Rechtschreibfehler zu korrigieren und Satzstrukturen zu überprüfen. Weiterhin könnte die Text Vectorization mithilfe von Word Embeddings (Rudkowsky et al., 2018) durchgeführt werden, um Ähnlichkeiten zwischen den Wörtern zu erkennen. Für die erhaltenen Merkmalsvektoren kann eine Feature Selection bspw. mit Chi-square oder Count Difference (Madasu & Elango, 2019) verwendet werden. Die Analyse kann durch die Anpassung der Parameter der Klassifikatoren an die gegebenen Daten optimiert und durch die Nutzung der Aspekt-basierten Sentiment Analysis kann die Annahme, dass sich ein Autor zu einer Entität äußert, verworfen werden. Zusätzlich können weitere Ensemble-Methoden wie z. B. Voting mit unterschiedlichen Ensemblen implementiert werden, um die Ergebnisse der Arbeit zu stützen bzw. zu verbessern.

Bei den Transformer-Methoden gilt es zu ergründen, weshalb unterschiedliche Ergebnisse in den Publikationen erreicht werden. Außerdem können weitere Transformer implementiert werden, um die Ergebnisse der Arbeit zu stützen bzw. zu verbessern, wie z. B. der Transformer ERNIE 2.0 (Sun et al., 2020), welcher gute Ergebnisse auf dem GLUE Dashboard erreicht hat. Nach der Umsetzung des Zero-Shot Learnings in der Arbeit kann das Few-Shot Learning realisiert werden, um den Transformer an die Zieldomäne anzupassen und bessere Ergebnisse zu erzielen.

Kapitel 6

Schlussfolgerung

Die vorliegende Arbeit setzte sich mit der Forschungsfrage „Ist die Nutzung eines trainierten Modells auf Review-Daten mit Twitter-Daten (Tweets) möglich?“ auseinander. Für die Lösung der Frage wurde eine quantitative Forschung in Form eines Experimentes mit Ensemble- und Transformer-Methoden durchgeführt. In dem folgenden Abschnitt werden die wichtigsten Ergebnisse der Arbeit zusammengefasst und ein Ausblick für weitere Forschungsansätze dargestellt.

Es konnte gezeigt werden, dass die Leistung der Methoden sich nicht zwischen den Klassen unterscheiden. Mithilfe der Ensemble-Methoden konnten Accuracies von über 70 % und überwiegenden 3 % bis 10 % knappen Entscheidungen erreicht werden. Eine Analyse der Ensemble-Methode AdaBoost war nicht möglich aufgrund der vielen knappen Entscheidungen während der Klassifikation.

Es wurde festgestellt, dass die Anzahl der knappen Entscheidungen beim Transfer Learning geringer als unter der Verwendung von Tweets oder Reviews ist, wodurch robustere Ergebnisse erzielt werden. Die Ergebnisse der Forschung knüpfen an die Arbeit von Aue und Gamon (2005) an, aber sie erzielen keine Accuracy von über 80 %. Daraus konnte geschlussfolgert werden, dass die domänenübergreifende Sentiment Analysis von der verwendeten Methode und der genutzten Domänen abhängt.

Durch die Transformer-Methoden wurden Accuracies von bis zu 75 % und 0 % bis 4 % knappen Entscheidungen erzielt. Der Ansatz basiert auf der Arbeit von Siebert et al. (2019), die Accuracies von über 90 % erreicht haben. Die Variabilität der Ergebnisse in den Publikationen sowie Versprechungen von Performance, die im realen Anwendungsfall nicht erreicht werden können, sind ein aktuelles Problem in der Forschung.

Mithilfe des Experimentes konnte festgestellt werden, dass durch die Feinabstimmung der Transformer die Anzahl der knappen Entscheidungen gesenkt werden kann, wodurch die Methode robuster wird. Ein konkurrenzfähiges Modell kann mithilfe der Feinabstimmung auf Tweets oder mit einem umfänglichen Training erstellt werden.

Die Ergebnisse des Experimentes zeigen, dass die Nutzung eines trainierten Modells auf Review-Daten zur Klassifikation von Tweets möglich ist. Die besten Ergebnisse können durch die Anpassung des Modells an die Twitter-Daten bzw. durch Verwendung des Zero-Shot Learning erreicht werden.

Für zukünftige Arbeiten sollte der Fokus auf den Transformer-Methoden aufgrund der besseren Ergebnisse in den Publikationen liegen. Dennoch könnten die Ensemble-Methoden u.a. durch die Nutzung von Word Embeddings für die Text Vectorization oder der Aspekt-basierten Sentiment Analysis verbessert werden. Im Hinblick auf die Transformer-Methoden sollten die Differenzen zwischen den Ergebnissen von Publikationen erforscht werden. Weiterhin können weitere Methoden implementiert und das Few-Shot Learning umgesetzt werden. Durch die Bereitstellung des Codes für das Experiment kann es wiederholt durchgeführt und erweitert werden.

Anhang A

Analyse der Vorverarbeitungsschritte

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,69	0,65	0,67	0,66
Bigramm	0,57	0,53	0,55	0,53
Trigramm	0,50	0,52	0,48	0,51

Tabelle A.1: Accuracies unter Nutzung von Tweets mit Tier 1 (Zin et al., 2017)

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,69	0,68	0,66	0,67
Bigramm	0,54	0,55	0,51	0,53
Trigramm	0,50	0,51	0,49	0,51

Tabelle A.2: Accuracies unter Nutzung von Tweets mit Tier 2 (Zin et al., 2017)

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,68	0,67	0,66	0,66
Bigramm	0,53	0,54	0,52	0,54
Trigramm	0,52	0,51	0,50	0,51

Tabelle A.3: Accuracies unter Nutzung von Tweets mit Tier 3 (Zin et al., 2017)

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,82	0,78	0,84	0,83
Bigramm	0,81	0,71	0,70	0,81
Trigramm	0,80	0,68	0,74	0,79

Tabelle A.4: Accuracies unter Nutzung von Reviews mit Erweiterung der Akronymen und Ersetzung der Negationen

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,69	0,67	0,67	0,68
Bigramm	0,55	0,57	0,51	0,55
Trigramm	0,51	0,52	0,50	0,50

Tabelle A.5: Accuracies unter Nutzung von Tweets mit Tier 3 (Zin et al., 2017), Erweiterung der Akronymen und Ersetzung der Negationen

	Bagging	Random Forest	AdaBoost	Stacking
Unigramm	0,82	0,80	0,85	0,84
Bigramm	0,78	0,70	0,71	0,76
Trigramm	0,65	0,50	0,64	0,66

Tabelle A.6: Accuracies unter Nutzung von Reviews mit Tier 3 (Zin et al., 2017), Erweiterungen der Akronymen und Ersetzung der Negationen

Literaturverzeichnis

Anandarajan, M., Hill, C. & Nolan, T. (2019). *Practical Text Analytics*. Springer International Publishing. https://doi.org/10.1007/978-3-319-95663-3

Aue, A. & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. *Proceedings of recent advances in natural language processing (RANLP)*, *1*(3.1), 2–1.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). CART. *Classification and Regression Trees, Wadsworth and Brooks/Cole, Monterey, CA*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/bf00058655

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL].

Burkov, A. (2019). *Machine Learning kompakt*. MITP Verlags GmbH.

Chopra, A., Prashar, A. & Sain, C. (2013). Natural language processing. *International journal of technology enhancements and emerging engineering research*, *1*(4), 131–134.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems* (S. 1–15). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

Frank, E., Hall, M. A. & Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for „Data Mining: Practical Machine Learning Tools and Techniques“* (4. Aufl.). Morgan Kaufmann.

Frochte, J. (2018). *Maschinelles Lernen Grundlagen und Algorithmen in Python*. Carl Hanser Verlag GmbH & Co. KG.

Géron, A. (2020). *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow*. Dpunkt.Verlag GmbH.

Glorot, X., Bordes, A. & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. *ICML*.

Go, A., Bhayani, R. & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12).

Goldberg, Y. & Levy, O. (2014). *Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. arXiv: 1402.3722 [cs.CL].

Graves, A. (2013). *Generating Sequences With Recurrent Neural Networks*. arXiv: 1308.0850 [cs.NE].

Hastie, T., Rosset, S., Zhu, J. & Zou, H. (2009). Multi-class AdaBoost. *Statistics and Its Interface*, *2*(3), 349–360. https://doi.org/10.4310/sii.2009.v2.n3.a8

Hoang, M., Bihorac, O. A. & Rouces, J. (2019). Aspect-Based Sentiment Analysis using BERT. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 187–196.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Honnibal, M., Montani, I., Van Landeghem, S. & Boyd, A. (2021). explosion/spaCy: v2.3.7: Bug fix for download CLI. https://doi.org/10.5281/ZENODO.1212303

Japkowicz, N. (2006). Why question machine learning evaluation methods. *AAAI workshop on evaluation methods for machine learning*, 6–11.

Jianqiang, Z. & Xiaolin, G. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. *IEEE Access*, *5*, 2870–2879. https://doi.org/10.1109/access.2017.2672677

Jo, T. (2021). *Machine Learning Foundations*. Springer International Publishing. https://doi.org/10.1007/978-3-030-65900-4

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Joshi, S. & Deshpande, D. (2018). Twitter Sentiment Analysis System. *International Journal of Computer Applications*, *180*(47), 35–39. https://doi.org/10.5120/ijca2018917319

Khalid, M., Ashraf, I., Mehmood, A., Ullah, S., Ahmad, M. & Choi, G. S. (2020). GBSVM: Sentiment Classification from Unstructured Reviews Using Ensemble Classifier. *Applied Sciences*, *10*(8), 2788. https://doi.org/10.3390/app10082788

Korovkinas, K., Danėnas, P. & Garšva, G. (2019). SVM and k-Means Hybrid Method for Textual Data Sentiment Analysis. *Baltic Journal of Modern Computing*, *7*(1). https://doi.org/10.22364/bjmc.2019.7.1.04

Kowsari, Meimandi, J., Heidarysafa, Mendu, Barnes & Brown. (2019). Text Classification Algorithms: A Survey. *Information*, *10*(4), 150. https://doi.org/10.3390/info10040150

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C. & Steinbrecher, M. (2015). *Computational Intelligence*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-10904-2

Kubat, M. (2017). *An Introduction to Machine Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-319-63913-0

Lee, W.-M. (2019). *Python Machine Learning*. WILEY.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. & Zettlemoyer, L. (2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. arXiv: 1910.13461 [cs.CL].

Li, X., Wang, L. & Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, *21*(5), 785–795.

Liddy, E. D. (2001). Natural language processing. *In Encyclopedia of Library and Information Science*.

Liu, B. (2015). *Sentiment analysis : mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167. https://doi.org/10.2200/s00416ed1v01y201204hlt016

Madasu, A. & Elango, S. (2019). Efficient feature selection techniques for sentiment analysis. *Multimedia Tools and Applications*, *79*(9-10), 6313–6335. https://doi.org/10.1007/s11042-019-08409-z

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. & Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. https://doi.org/10.1145/1150402.1150531

Müller, M., Salathé, M. & Kummervold, P. E. (2020). *COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter*. arXiv: 2005.07503 [cs.CL].

Omran, F. N. A. A. & Treude, C. (2017). Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments. *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. https://doi.org/10.1109/msr.2017.42

Opitz, D. & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, *11*, 169–198. https://doi.org/10.1613/jair.614

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q. & Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. *Proceedings of the 19th international conference on World wide web - WWW '10*. https://doi.org/10.1145/1772690.1772767

Pang, B. & Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *Proceedings of the ACL*.

Pang, B., Lee, L. & Vaithyanathan, S. (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*. arXiv: cs/0205070 [cs.CL].

Peddinti, V. M. K. & Chintalapoodi, P. (2011). Domain Adaptation in Sentiment Analysis of Twitter. *Proceedings of the 5th AAAI Conference on Analyzing Microtext*, (6), 44–49.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Peng, M., Zhang, Q., Jiang, Y.-g. & Huang, X.-J. (2018). Cross-domain sentiment classification with target domain specific information. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2505–2513.

Pennington, J., Socher, R. & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Rajpurkar, P., Jia, R. & Liang, P. (2018). *Know What You Don't Know: Unanswerable Questions for SQuAD*. arXiv: 1806.03822 [cs.CL].

Rebala, G., Ravi, A. & Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-030-15729-6

Rudkowsky, E., Haselmayer, M., Wastian, M., Jenny, M., Emrich, Š. & Sedlmair, M. (2018). More than Bags of Words: Sentiment Analysis with Word Embeddings. *Communication Methods and Measures*, *12*(2-3), 140–157. https://doi.org/10.1080/19312458.2018.1455817

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Sanh, V., Debut, L., Chaumond, J. & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv: 1910.01108 [cs.CL].

Sarkar, D. (2016). *Text Analytics with Python*. Apress. https://doi.org/10.1007/978-1-4842-2388-8

Schapire, R. E. (2003). The Boosting Approach to Machine Learning: An Overview. *Nonlinear Estimation and Classification* (S. 149–171). Springer New York. https://doi.org/10.1007/978-0-387-21579-2_9

Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Selamat, A. & Zainuddin, N. (2014). Sentiment Analysis Using Support Vector Machine. *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*. https://doi.org/10.1109/I4CT.2014.6914200

Siebert, C., Hartmann, J., Heitmann, M. & Schamp, C. (2019). Accuracy of Automated Sentiment Analysis. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3489963

Singh, A. K. & Shashi, M. (2019). Vectorization of text documents for identifying unifiable news articles. *Int. J. Adv. Comput. Sci. Appl*, *10*.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H. & Wang, H. (2020). ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8968–8975. https://doi.org/10.1609/aaai.v34i05.6428

Sutskever, I., Vinyals, O. & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 3104–3112.

The pandas development team. (2021). pandas-dev/pandas: Pandas 1.3.0rc1. https://doi.org/10.5281/ZENODO.3509134

Thomas, G. D. (1997). Machine learning research: Four current directions. *Artificial Intelligence, Magazine*, *18*(4), 97–136.

Tsakalidis, A., Papadopoulos, S. & Kompatsiaris, I. (2014). An Ensemble Model for Cross-Domain Polarity Classification on Twitter. *Web Information Systems Engineering – WISE 2014* (S. 168–177). Springer International Publishing. https://doi.org/10.1007/978-3-319-11746-1_12

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Vishal, A. & Sonawane, S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, *139*(11), 5–15. https://doi.org/10.5120/ijca2016908625

Wan, S. & Yang, H. (2013). Comparison among Methods of Ensemble Learning. *2013 International Symposium on Biometrics and Security Technologies*. https://doi.org/10.1109/isbast.2013.50

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. & Bowman, S. R. (2018). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. arXiv: 1804.07461 [cs.CL].

Wang, Y., Yao, Q., Kwok, J. & Ni, L. M. (2019). *Generalizing from a Few Examples: A Survey on Few-Shot Learning*. arXiv: 1904.05046 [cs.LG].

Whitehead, M. & Yaeger, L. (2010). Sentiment mining using ensemble classification models. *Innovations and advances in computer sciences and engineering* (S. 509–514). Springer.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., ... Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241–259.

Xia, R. & Zong, C. (2011). A POS-based ensemble model for cross-domain sentiment classification. *Proceedings of 5th international joint conference on natural language processing*, 614–622.

Xia, R., Zong, C. & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, *181*(6), 1138–1152. https://doi.org/10.1016/j.ins.2010.11.023

Yin, W., Hay, J. & Roth, D. (2019). *Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach*. arXiv: 1909.00161 [cs.CL].

Zhang, C. & Ma, Y. (Hrsg.). (2012). *Ensemble Machine Learning*. Springer-Verlag GmbH.

Zhou, Z.-H. (2021). Ensemble Learning. *Machine Learning* (S. 181–210). Springer Singapore. https://doi.org/10.1007/978-981-15-1967-3_8

Zin, H. M., Mustapha, N., Murad, M. A. A. & Sharef, N. M. (2017). The effects of pre-processing strategies in sentiment analysis of online movie reviews. *AIP Conference Proceedings*, *1891*(1). https://doi.org/10.1063/1.5005422

Selbstständigkeitserklärung

Hiermit versichere ich, Falk Puschner, dass ich die vorliegende Masterarbeit mit dem Titel „Domänenübergreifende Sentiment Analysis auf Twitter-Tweets und Film-Reviews“ selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Werdau, den 30. September 2021

Falk Puschner