Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Umesh Adhikari

Identifizierung der beratungsintensiven Kunden
mittels Kundendaten einer Kundenhotline

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Umesh Adhikari
[redacted]

Identifizierung der beratungsintensiven Kunden mittels Kundendaten einer Kundenhotline

Abschlussarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ing. Marina Tropfmann-Frick
Zweitgutachter: Prof. Dr. Ing. Martin Schultz

Eingereicht am: 06.06.2022

Umesh Adhikari

Thema der Arbeit
Identifizierung der beratungsintensiven Kunden mittels Kundendaten einer Kunden-hotline

Stitchwort
Sentimentanalyse, Natural Language Processing, NLP, Text Klassifikation, BERT

Kurzzusammenfassung
In dieser Arbeit wurde deutscher Text (als Memo) von einer technischen Kunden-hotline mit verschiedenen NLP Methoden analysiert. Dabei wurde experimentiert, ob firmendefinierte Stoppwörter definiert werden sollten. Außerdem wurde das Memo in Anfragetypen, wie zum Beispiel Störungsmeldung oder Technische Fragen, usw., klassifiziert. Anschließend wurde ein BERT-Modell mit einem eigenen Datensatz trainiert und evaluiert. Das Ergebnis dieser Arbeit zeigt, dass Kundendaten dazu verwendet werden können, beratungsintensive Kunden zu identifizieren.

Umesh Adhikari

Title of the paper
Identification of consulting-intensive clients with the help of customer data of a customer hotline

Keywords
Sentiment Analysis, Natural Language Processing, NLP, Text classification, BERT

Abstract
In this paper, German text (as a memo) from a technical customer hotline was analyzed using various NLP methods. Thereby it was experimented whether to define company defined stop words. Also, the memo was classified into request types, such as disturbance report or technical questions, etc. Subsequently, a BERT model was trained and evaluated with a custom data set. The result of this work shows that customer data can be used to identify customers who require intensive consulting.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

1. Einleitung

"Human behavior flows from three main sources: desire, emotion and knowledge." - Plato

Emotionen sind der stärkste Aspekt des menschlichen Lebens. Sie können auf viele verschiedene Arten und Weise ausgedrückt werden, unter anderem auch in Texten. Durch die zunehmende Popularität und Fortschritte des Internets und der Technologie wächst auch die Zahl der Webanwendungen, wie z. B. sozialen Netzwerken (Facebook[1], Twitter[2]), Nachrichtenportal, E-Commerce, u.a. Kundenportal eines Dienstleistungs-unternehmens, die reich an emotionalen Informationen der Benutzer oder Kunden enthalten. Die Analyse der Texte dieser Art von Webanwendungen kann Vorteile für verschiedene Anwendungsbereiche bringen, beispielsweise subjektive Suchmaschinen, die Vermarktung von Produkten und die Bestimmung der Kundenpräferenzen im webbasierten Dienst [1].

Besonders für die Dienstleistungsunternehmen ist es wichtig, die Emotionsdynamik der Kunden zu ermitteln, da sie die Qualität der Dienstleistungen beeinflusst. Das ist auch deshalb so wichtig, weil das Verhalten der Menschen auf ihren Emotionen basiert.

Wenn die Unternehmen die Kundenmeinung über ihre Produkte oder Dienstleistungen erfahren möchten, führen sie in der Regel traditionelle Umfragen durch. Alternativerweise kann dies auch durch ein automatisiertes Meinungsanalyse-System erledigt werden. Manuell wäre es kaum machbar, die textuellen Informationen aus Webportalen zu verarbeiten, um die wichtigsten Ansichten zu extrahieren und ihre Stimmung zu erkennen.

Computerlinguistik und Informationssuche bietet eine Lösung solcher Aufgabe. Diese Lösung gehört zu dem Bereich der Sentimentanalyse und Meinungsextraktion. Die Sentimentanalyse ist für die Klassifizierung der Texte auf Grund von Stimmungen oder Tonalität zuständig. Durch die Meinungsextraktion extrahiert man emotionale Lexikon und die Meinungen, die im Text enthalten sind.

1 https://www.facebook.com/
2 https://twitter.com/

1.1. Firmenprofil der wilhelm.tel GmbH

Diese vorliegende Abschlussarbeit wurde im Hause der wilhelm.tel GmbH[3] in Norderstedt erarbeitet und entstand im Zusammenhang mit dem operativen Einsatz eines maschinellen Lernmodells zur Erleichterung des Kundensupports.

Die wilhelm.tel GmbH ist ein Tochterunternehmen der Stadtwerke Norderstedt[4], die ein Energieversorgungsunternehmensgruppe ist und für rund 550 Mitarbeiter/-innen einen Arbeitsplatz bietet [20]. wilhelm.tel GmbH betreibt ein eigenes Glasfasernetze für Telefonie, Internet und Kabelfernsehen. Darüber hinaus bietet wilhelm.tel den WLAN-Dienst "MobyKlick" mit Gigabit-Geschwindigkeiten an fünftausend Standorten an, darunter alle S- und U-Bahnhöfe, Hamburger Bücherhallen sowie touristische Orte, nämlich die Elbphilharmonie, Speicherstadt, Reeperbahn. Im Jahre 1999 gegründete wilhelm.tel beschäftigt zurzeit ca. 125 Mitarbeiter/-innen [21].

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es herauszufinden, inwieweit der Einsatz von Künstlicher Intelligenz (KI) die beratungsintensiven Kunden eines Telekommunikations-unternehmens identifizieren kann und den Kundensupport anhand von Kundendaten im Textformat erleichtern kann. Wobei wird es für die Sentimentanalyse Bidirectional Encoder Representations from Transformers (BERT)[5] [2] als Methodik verwendet.

Das zweite Hauptziel der Arbeit besteht darin, die Kundendaten (Memos) zu klassifizieren, ob es sich um eine Beschwerde, ein Lob, eine Anregung oder ein Technische handelt.

3 https://www.wilhelm-tel.de/

4 https://www.stadtwerke-norderstedt.de/

5 BERT ist eine auf Transformer basierende maschinelle Lerntechnologie für NLP.

1.3 Aufbau der Arbeit

Kapitel 1: Im ersten Kapitel wird die Einleitung der Abschlussarbeit beschrieben. Außerdem wird die Firma „wilhelm.tel“ vorgestellt und das Ziel der Arbeit beschrieben.

Kapitel 2: In diesem Kapitel werden die wichtigsten grundlegenden Begriffe von Sentimentanalyse, Klassifikation der Texte und Klassifikation der Emotion eingeordnet.

Kapitel 3: Im fünften Kapitel geht es um die möglichen Ansätze der Klassifikation der Sentimentanalyse.

Kapitel 4: Das darauffolgende Kapitel erläutert die alternierende Lösungsmöglichkeit der Probleme.

Kapitel 5: Anschließend wird in diesem Kapitel das praktische Teil dieser Bachelorarbeit durchgeführt.

Kapitel 6: In diesem Kapitel werden das praktische Teil evaluiert.

Kapitel 7: Im letzten Kapitel wird die Abschlussarbeit zusammengefasst und ein Ausblick beschrieben.

2 Grundlagen

In diesem Kapitel werden wir uns mit den grundlegenden Begriffen der Technologien befassen, die in der Stimmungsanalyse zum Einsatz kommen. Darüber hinaus soll es dem Leser helfen, den praktischen Teil dieser Arbeit in diesem Anwendungskontext betrachten zu können.

2.2 Deep Learning

In den 1980er Jahren hatten Noel Entwistle und Paul Ramsden erstmals den Begriff "Deep Learning" vorgestellt, als sie den Unterschied zwischen Deep Learning und Surface Learning diskutierten [13]**.** Dann sind künstliche neuronale Netze ein Zweig des maschinellen Lernens geworden. Deep Learning kann in folgende Kategorien unterteilt werden [14]:

- Überwachtes Lernen (Supervised Learning)
- Halbüberwachtes Lernen (Semi-Supervised Learning)
- Unüberwachtes Lernen (Unsupervised Learning)

Das Hauptkonzept der Deep Learning-Algorithmen ist die automatische Extraktion von Repräsentationen aus Daten. Ein weiteres Kernkonzept, das eng mit Deep Learning verbunden ist, ist das Lernen der verteilten Darstellung von Daten. In diesem Fall kann jede Probe kompakt dargestellt werden, was zu einer umfassenderen Verallgemeinerung führt.

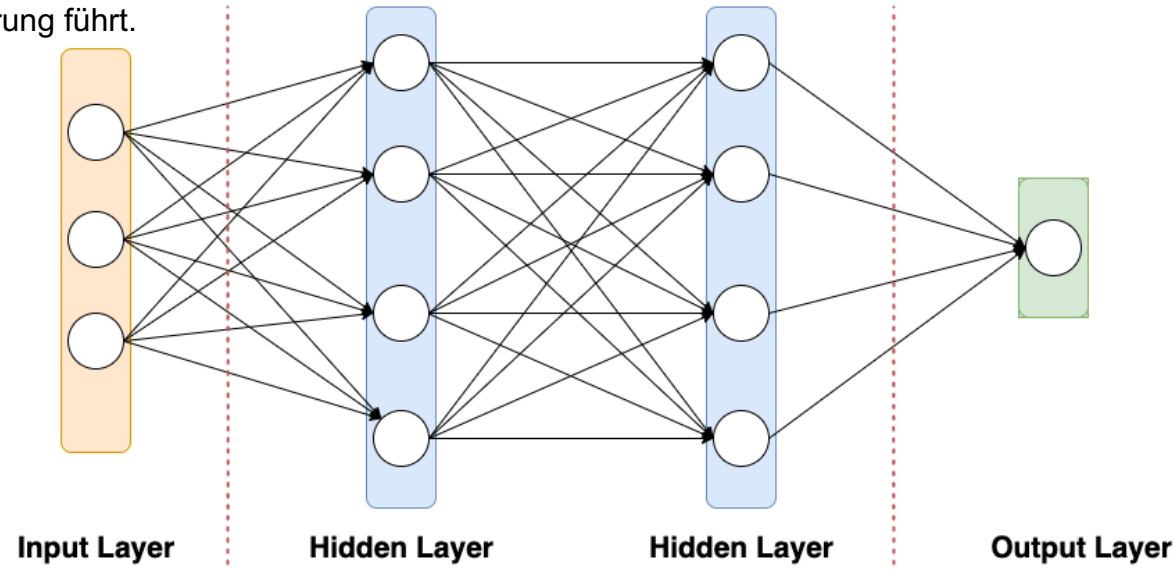


Abbildung 1: Deep Learning Architektur

Deep Learning-Algorithmen kann man sich einfach als tiefe Architekturen aus aufeinanderfolgenden Schichten vorstellen, nämlich Eingabeschicht (Input Layer), ein oder mehrere verborgen Schicht (Hidden Layer) und Ausgabeschicht (s. Abbildung 1). Jede Schicht wendet eine nichtlineare Transformation auf ihre Eingabe an und gibt die Darstellung aus. Die Eingabeebene enthält Ihre Eingabedaten. Die verborgene Schicht versucht, verschiedene Aspekte der Daten zu lernen, indem sie eine Fehler- oder Kostenfunktion minimiert. Die Ausgabeschicht besteht aus den Ausgabedaten. Das Ziel ist es, eine komplexe und abstrakte Datendarstellung hierarchisch zu erlernen, indem die Daten durch mehrere Transformationsschichten geleitet werden. Die Eingabedaten werden in die erste Schicht eingespeist, wie z.B. die Einbettung von Token. Die Ausgabe jeder Schicht ist die Eingabe für die nächste Schicht.

Die Grundidee der Deep Learning-Algorithmen besteht darin, die nichtlinearen Transformationsschichten übereinander zu legen. Komplexere nichtlineare Transformationen können aus tieferen Schichten konstruiert werden [15]. Durch eine tiefe Architektur mit mehreren Repräsentationsebenen werden die Daten in abstrakte Repräsentationen übertragen. In diesem Fall können Deep Learning-Algorithmen als eine Art von Repräsentationslernalgorithmus betrachtet werden.

Das endgültige trainierte Modell kann als eine hochgradig nichtlineare Funktion der Eingabedaten betrachtet werden, die eine endgültige Darstellung konstruieren kann. Die zugrundeliegenden erklärenden Faktoren in den Daten können aus den nichtlinearen Transformationen durch die Schichten der tiefen Architektur extrahiert werden [15].

Die endgültige Darstellung (die Ausgabe der Ausgabe-Schicht) enthält die nützlichen Informationen in den Trainingsdaten, die durch den Deep-Learning-Algorithmus konstruiert wurden und als Merkmale bei der Erstellung von Klassifizierern mit hoher Effizienz im Vergleich zu den hochdimensionalen sensorischen Daten verwendet werden können.

In dieser vorliegenden Bachelorarbeit wird ein vortrainiertes Sprachrepräsentationsmodell verwendet, das auf Deep-Learning-Techniken basiert, um die Informationen aus dem Text zu extrahieren und den Text in Vektoren zu übertragen. Die Ausgabe des Sprachmodells heißt Arbeitseinbettungen und enthält die Informationen des Eingabetextes.

2.2 Klassifikation der Texte

Textklassifizierung (TK) ist der Prozess der Kategorisierung von Texten (z. B. Tweets, Nachrichtenartikel, Kundenrezensionen) in organisierte Gruppen. Typische TK-Aufgaben sind Stimmungsanalyse, Kategorisierung von Nachrichten und Themenklassifizierung von den Texten [3].

Die Trainingsmenge ist eine Menge $S = \{(x_i, y_i) \mid i = 1, \ldots, k\}$ von Paaren (x_i, y_i), wobei x einen Text, y ein Label und k die Anzahl der Kategorien repräsentieren. Hier y ein Label, was mit einem Text zugeordnet ist. Anhand der Klassen die Klassifizierungsfunktion bildet die Repräsentation der Text ab [4].

Die Anzahl der Klassen C kann 2 oder mehr als 2 sein. Sei $C=2$, so spricht man von einem binären Klassifikationsproblem, im Fall $C>2$ ist, dann von einem Multiklassenproblem.

Die Klassifizierung der Texte ist das grundlegende Problem im Natural Language Processing (NLP), die in viele Anwendungen verwendet werden kann, wie nämlich die Extraktion der Information aus dem Internetquellen, Klassifizieren von Kundenrezensionen und die Klassifizierung der Nachrichtenartikeln.

Im praktischen Teil dieser Abschlussarbeit es wird die Kundenrezensionen oder Kundenanfragen an den Kundensupport klassifiziert, ob es ein Rechnungsbezogen, Beschwerde, Kaufmännisch oder Vorschlag ist.

2.3 Sentimentanalyse (SA)

Bei einer Sentimentanalyse (SA) wird Emotionen, Meinungen und Subjektivität eines Textes betrachtet. Um verschiedene Sentiments zu extrahieren, benötigt man eine Texteingabe für die Durchführung einer SA [5].

Es gibt zwei folgende kern Begriffe für diesen Forschungsbereich:

1. Opinion Mining
2. Subjektivität Analysis

Opinion Mining oder auch als Sentiment Analysis bekannt, wird häufig in einem Kontext verwendet, in dem Meinungen zu verschiedenen Aspekten eines Objekts oder einer Entität ermittelt werden sollen. Subjektivität Analysis konzentriert sich vorherrschend auf die Erkennung von subjektiven Texten bzw. Textstellen im Kontrast zu objektiven Texten. Die SA hingegen wird überwiegend als Begriff für die Bestimmung der Polarität von Texten verwendet [6]. Bei der Stimmungsanalyse der Texte repräsentieren die Klassen positive, negative oder neutrale Polarität [7].

In der Stimmungsanalyse gibt es drei Hauptklassifizierungs-ebenen, nämlich Dokumentebene, Satzebene und Aspekt-ebene Stimmungsanalyse. Der Zweck der Emotionsanalyse (EA) auf Dokumentebene besteht darin, Meinungsdokumente dahingehend zu klassifizieren, dass sie positive oder negative Meinungen oder Emotionen ausdrücken. Es betrachtet das gesamte Dokument als eine grundlegende Informationseinheit. Die Sentimentanalyse auf Satzebene zielt darauf ab, dass in jedem Satz ausgedrückte Sentiment zu klassifizieren. Eigentlich kann man sich Sätze als ein kurzes Dokument vorstellen, so dass es keinen grundlegenden Unterschied zwischen der Dokumentenebene und der Satzebene gibt. Die aspektbasierte Stimmungsanalyse bezieht sich auf den Prozess der Ausgabe der Stimmungspolarität jedes Aspektworts in einem Satz mit einem Satz und einigen vordefinierten Aspektwörtern als Eingabedaten [16].

Sentimentanalyse hat divers Anwendungsgebiete, wie zum Beispiel die Bestimmung der Valenz von Filmkritik [8], oder Einsatz bei einem Unternehmen als Business Intelligente Systemen, um die Meinungen der Kunden zu analysieren [9].

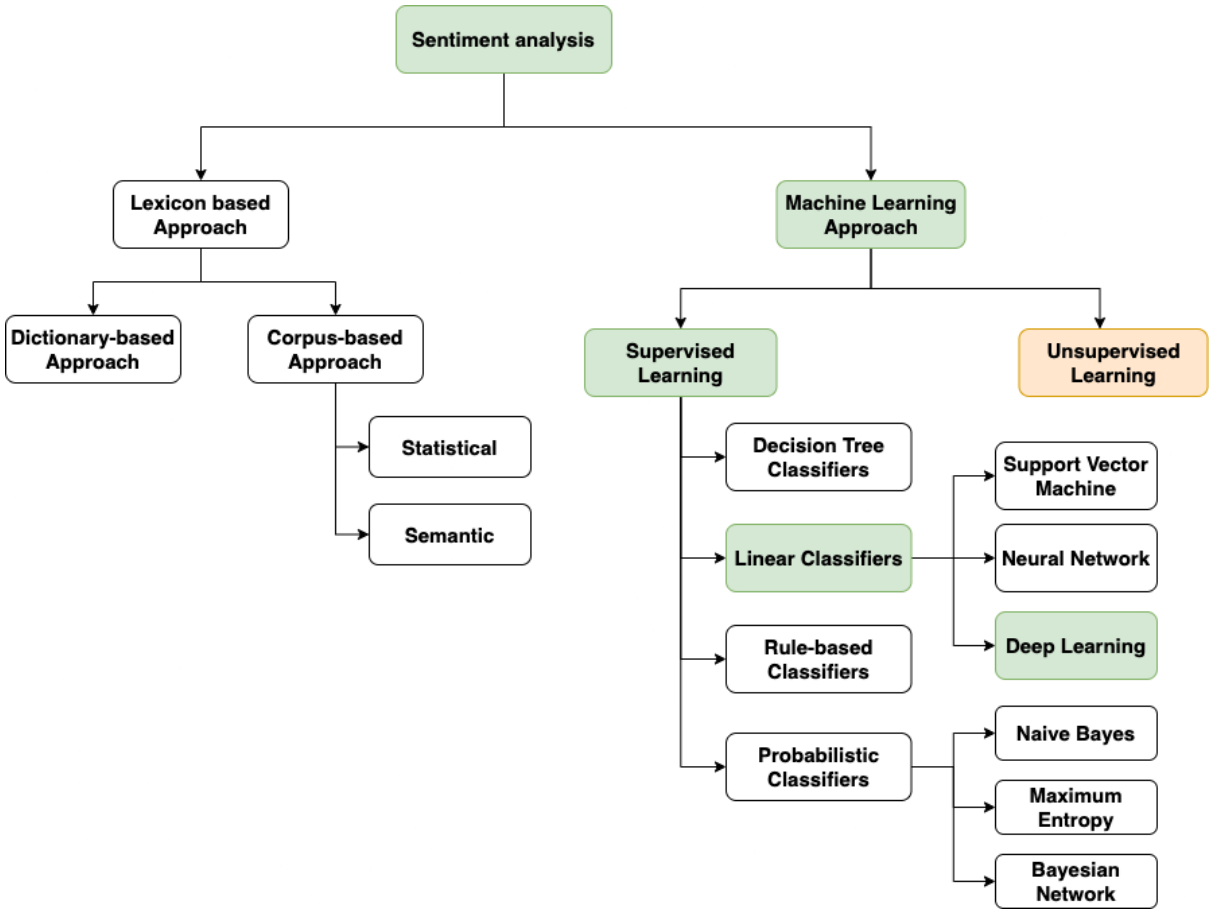


Abbildung 2: Techniken zur Klassifizierung des Sentiments [39]

2.4 Klassifizierung der Emotion

Die Klassifizierung von Emotionen kann als eine Unterart der Stimmungsanalyse betrachtet werden. Dabei berücksichtigt sie die Textklassifizierung in Kategorien, die Emotionen entsprechen. Die Basis-Emotionen sind die vordefinierten Klassen.

Es wird die automatisierte Klassifizierung der Emotion des Textes in verschiedenen Anwendungsbereich angewendet, beispielsweise Medien, wie Nachrichtenüberschriften [10], Soziale Netzwerken, wie Twitter [11] und automatisches audiobasiertes Emotionserkennungssystem [12].

2.5 Vortrainiertes Sprachmodel

Aufgrund der Effektivität des vortrainierten Sprachmodells in vielen nachgelagerten NLP-Aufgaben, hat es viel Aufmerksamkeit bekommen. Das Vortraining von Sprachmodellen hat sich als effizient erwiesen, um viele Aufgaben der natürlichen Sprachverarbeitung zu verbessern. Aufgaben der NLP, wie z.B. die Klassifizierung von Stimmungen. Die grundlegende Die Grundidee hinter dem vortrainierten Sprachmodell ist das Trainieren einer Worteinbettungsschicht aus einer großen Anzahl von Arten zu trainieren, so dass sie eine hervorragende Fähigkeit hat, die Informationen aus dem Kontext zu extrahieren. Denn es reicht nicht aus, verschiedene neuronale Architekturen der kodierenden Kontextdarstellung nur anhand der begrenzten Überwachungsdaten von Terminalaufgaben zu trainieren.

Bidirectional Encoder Representations from Transformers (BERT) ist ein vortrainiertes Sprachrepräsentationsmodell, das im Jahr 2018 vom Google AI-Team auf der Grundlage von Deep Learning-Techniken vorgeschlagen wurde [2]. Im Gegensatz zu anderen Sprachrepräsentationsmodellen kann BERT durch die gemeinsame Berücksichtigung von linkem und rechtem Kontext in allen Schichten tiefe bidirektionale Repräsentationen aus dem ungelabelten Eingabetext erzeugen. BERT wurde bereits in verschiedenen NLP-Aufgaben wie Textklassifizierung und Fragen-beantwortung eingesetzt und hat eine hervorragende Leistung erbracht [17].

Aufgrund des gewählten Feinabstimmungsansatzes (fine-tuning) gibt es bei der Verwendung von BERT keine spezifische Architektur für nachgelagerte NLP-Aufgaben. Als intelligenter Agent sollte er die Verwendung von menschlichem Vorwissen bei der Modell-entwicklung minimieren und dieses Wissen aus Daten lernen stattdessen. In BERT werden zwei verschiedene Ziele für das Training des Sprachmodells verwendet und nicht das häufig verwendete Ziel der Vorhersage des nächsten Wortes. Das erste ist das maskierte Sprachmodell, bei dem das Modell den maskierten Token aus ihrem Kontext vorhersagen muss. Das andere Ziel ist die Vorhersage der nächsten Sequenz, bei der das Modell lernen muss, ob Sequenz B auf Sequenz A folgt. Diese beiden Ziele ermöglichen es dem Modell langfristige Abhängigkeiten besser zu lernen.

- **Masked Language Model (MLM):** Das Modell lernt, der zufällig maskierte Token in Sequenz A und Sequenz B vorherzusagen.
- **Next-Sentence Prediction:** Damit BERT langfristige Abhängigkeiten besser lernen kann, muss das Modell lernen, ob eine Sequenz B auf natürliche Weise auf die vorherige Sequenz A folgt. Die Sequenz A und die Sequenz B stammen also aus demselben Dokument, so dass die Sequenz A auf die Sequenz B folgt.

Bei BERT verwenden die Autoren den Transformator als Basiskomponente und nicht Recurrent Neural Network (RNN) oder Convolutional Neural Network (CNN) [2]. Der Transformer basiert ausschließlich auf dem Mechanismus der Selbstbeobachtung. Im Vergleich zu einem RNN oder einem CNN hat der Transformator folgende Vorteile: Erstens kann es die Rechenressourcen und die Berechnungsgeschwindigkeit reduzieren. Zweitens kann die Berechnung parallelisiert werden, was bei RNN unmöglich ist. Ansonsten hat der Transformator eine gute Leistung beim Erlernen weitreichender Abhängigkeiten.

In der Praxis ist es einfach, durch eine Feinabstimmung des BERT mit einer zusätzlichen Ausgabeschicht für verschiedene NLP-Aufgaben wie Textklassifizierung und Fragenbeantwortung ein hervorragendes Leistungsmodell zu erstellen, ohne dass die aufgabenspezifische Architektur zu stark verändert werden muss.

3 Ansätze der Sentimentanalyse

Grundsätzlich kann man im Bereich der Stimmungsanalyse zwischen den folgenden zwei Arten unterscheiden:

I. Lexikalisch basierten Ansätzen
II. Lernbasierten Ansätzen

Die lexikalisch basierten Ansätze basiert auf Lexika und hat Daten ohne Label. Im Gegensatz lernbasierte Ansätze verwendet gelabelte Daten. Außerdem gibt es noch einen Ansatz, der aus einer Mischung beider Ansätze besteht, der als Hybridansatz bezeichnet wird. In der vorliegenden Abschlussarbeit ist der Schwerpunkt hauptsächlich auf lexikalische Ansätze beschränkt.

Vor der eigentlichen Ausführung der Sentimentanalyse ist bei all diesen Ansätzen im ersten Schritt eine Vorverarbeitung erforderlich, um irrelevante Teile der Daten zu entfernen und den Text in eine für die Analyse geeignete Form zu bringen [33]. Irrelevante Teile der Daten können z. B. HTML-Tags, Skripte oder Werbung sein, die die Leistung und Genauigkeit der nachfolgenden Analyse beeinträchtigen. Der Hauptteil der Vorverarbeitung der Daten besteht jedoch aus Techniken der natürlichen Sprachverarbeitung wie Tokenisierung, Splitting, Streichung der Stoppwörtern oder Stemming. Die Aufteilung der Text in Wörter und Symbole heißt Tokenization. Das Splitting ist für die Erkennung der Datensatzgrenzen verantwortlich. Die Stopwörter enthalten keine bedeutenden Informationen über die Stimmung [34]. Daher sie haben keinen Mehrwert bei der Sentimentanalyse. Unter Stemming versteht man die Bildung von Wortstämmen zur Vermeidung unnötiger Redundanz.

3.1 Lexikalisch basierten Ansätzen

Der lexikalische Ansatz folgt der Grundidee, alle meinungsbildenden Wörter im Text mit einem semantischen Lexikon zu vergleichen, in dem ihre Stimmungspolarität festgehalten ist. Dieser Ansatz verwendet das Bag of Words-Modell, das heißt, dass alle Wörter im Text als ungeordnete Menge betrachtet werden, wobei die Reihenfolge vernachlässigt wird.

Es gibt mehrere Möglichkeiten, ein solches semantisches Lexikon oder Meinungslexikon zu erstellen. Der Verwendungskontext hat einen entscheidenden Einfluss auf die verwendeten Wörter, daher muss der Verwendungskontext berücksichtigt werden. Außerdem können Wörter in verschiedenen Kontexten unterschiedliche Bedeutungen haben und somit eine unterschiedliche Polarität aufweisen. Tatsächlich kann ein Wort unterschiedliche Polaritäten anzeigen, je nachdem, welchen Aspekt es modifiziert, insbesondere bei mehrdeutigen Wörtern wie "hoch" , das im Schnipsel "hohe Qualität" eine positive Ausrichtung hat, im Schnipsel "hoher Preis" jedoch eine negative Ausrichtung [18]. Es ist zwar möglich, solche entsprechenden Wörter manuell zu suchen und einzutragen, aber dieses Verfahren kostet enorm viel Zeit und Arbeitsaufwand. Durch die Verwendung von Wörterbüchern können solche Probleme behandelt werden.

Die semi-automatisch Erstellung eines semantischen Lexikons ist die eine andere Möglichkeit. Zunächst wird manuelle eine Gruppe eindeutig positiver oder negativer Wörter, die sogenannten Seeds, die selbständig anwachsen soll, bestimmt. Die Qualität des späteren Lexikons hängt von ihnen ab, deswegen müssen diese Wörter jedoch gut überlegt sein. Es sollten also Begriffe gewählt werden, die möglichst eindeutig und kontextunabhängig sind. Es wird in Wörterbüchern automatisiert nach Synonymen der Wörter gesucht und diese dann dem Wörterbuch hinzugefügt. Diese Methode hat jedoch den Nachteil, dass die Ähnlichkeit eines Wortes mit der Entfernung vom ursprünglichen Seed abnimmt. Dies garantiert jedoch nicht, dass alle Wörter mit einem solchen Pfad die gleiche Polarität haben. Die alle Wörter auf einem Pfad, ähnlich einer Normalverteilung, lassen sich nicht eindeutig einer Polarität zuordnen und sollten daher nicht für ein Meinungslexikon verwendet werden. [31]. Nach Ansicht der Autoren sind die Wörter besser, die am Rande der Normalverteilung liegen und daher eindeutig einer Polarität zugeordnet werden können. Außerdem kann die Stärke der Pfade auch bei der Erstellung eines semantischen Wörterbuchs berücksichtigt werden, was mit ähnlichen Gewichtungen wie beim Page Rank-Algorithmus von Google geschehen kann.

3.1.1 German Polarity Clues (GPC)

Es wurde eine semi-automatische Übersetzung vorhandener englischsprachiger Sentiment-Quellen verwendet, um die deutschen Polarity Clues zu erstellen. Da sich die vorhandenen Quellen hinsichtlich der Anzahl der enthaltenen Begriffe und Merkmale stark unterscheiden, wurde in drei Schritten eine neue Ressource entwickelt [29].

Zuerst die am häufigsten verwendeten englischsprachigen Quellen, wie SentiWordNet, SentiSpin, wurden in einem Experiment durchgeführt, d.h. ihre Leistung wurde in Abhängigkeit von der Menge ihrer Merkmale getestet. Danach wurden die vorhandenen Wörterbücher mit Hilfe einer Sprachsoftware ins Deutsche übersetzt. Für Wörter, für die es mehrere Übersetzungsmöglichkeiten gibt, wurde eine maximale Anzahl von drei gewählt. Außerdem alle Übersetzungen eines Wortes wurden mit der Polarität des Originals versehen (positiv, negativ oder neutral).

Infolgedessen weicht der Gesamtumfang des übersetzten Wörterbuchs vom Original ab, aber auch bei einzelnen Wörtern kann es zu Mehrdeutigkeiten kommen. Daher wurde in einem dritten Schritt erneut eine manuelle Bewertung aller Wörter des resultierenden GPC durchgeführt. Dazu wurden 290 verneinende Ausdrücke (z.B. nicht-schlecht) und positive oder negative Synonyme, die noch nicht vorhanden waren, hinzugefügt, so dass sich insgesamt 10141 Begriffe ergaben.

Overall Features:	10,141
No. Positive Features:	3,220
No. Negative Features:	5,848
No. Neutral Features:	1,073
No. Negation Features:	290
No. Noun Features:	4,408
No. Verb Features:	2,728
No. Adj/Adv Features:	2,604

Tabelle 1: German Polarity Clues Feature [29]

3.1.2 SentiWS

Sentiment Wortschatz, abgekürzt SentiWS, wurde im Jahr 2010 von Remus et al. [30] für die deutschsprachige Sentimentanalyse veröffentlicht. Der Wortschatz enthält insgesamt 32 734 Wörter, davon 16 406 positive und 16 328 negative Wortformen, die in einem Intervall von 1 bis -1 gewichtet werden können. Nomen, Verben, Adjektive und Adverbien sind die enthaltenen Wortformen des Wortschatzes.

		Positive	**Negative**
Adjectives	Baseforms	784	698
	Inflections	11 782	10 604
Adverbs	Baseforms	6	4
	Inflections	0[3]	0[3]
Nouns	Baseforms	584	686
	Inflections	521	806
Verbs	Baseforms	312	430
	Inflections	2 453	3 100
All	Baseforms	1 650	1 818
	Inflections	14 756	14 510
	Total	**16 406**	**16 328**

Tabelle 2: Überblick über den Inhalt des Wörterbuchs [30]

Ein Eintrag in SentiWS enthält für das jeweilige Wort dessen Part of Speech (POS), der nach dem Stuttgart-Tübingen Tag Set (STTS) abgebildet wurde und sich immer auf die entsprechende Basisform bezieht. Außerdem bezieht ein Eintrag noch die gewichtete Polarität und die flektierten Formen des Wortes ein, falls vorhanden.

Word	**POS Tag**	**Weight**	**Inflections**
harmonisch	ADJX[2]	0,5243	harmonisehe, ...' harmonischst
Krise	NN	0,3631	Krisen

Tabelle 3: Das Schema der SentiWS-Einträge [30]

Es wurde drei Hauptquellen verwendet, um die semantische Ausrichtung der Wörter in das Lexikon zu bestimmen. Zunächst wurden die positiven und negativen Kategorien des General Inquirer Lexikons (GI) erfasst und halbautomatisch mit Google Übersetzer übersetzt, gefolgt von einer manuellen Überprüfung auf Korrektheit. Außerdem wurden einige spezielle Wörter mit Bezug zur Finanzbranche manuell hinzugefügt, da dies der ursprüngliche Zielbereich des Lexikons ist.

Die zweite Quelle basiert auf der Textanalyse von Produktrezensionen, die als negativ und positiv gekennzeichnet waren. Um die Wörter zu sammeln, wurden Text-analytische Methoden eingesetzt, die in einer besonders großen Anzahl positiver oder negativer Bewertungen auftauchten, und diese dann manuell dem Wörterbuch hinzugefügt.

Als letzte Quelle wurde das Deutsche Kollokationswörterbuch verwendet, das unter anderem Wörter zusammenfasst, die häufig mit bestimmten Substantiven vorkommen. Die semantische Nähe wurde verwendet, um die individuellen Polaritätsgewichte für die Ausgangswörter zu berechnen.

3.2 Lernbasierten Ansätzen

Lernbasiert Ansätzen werden im Bereich Data Mining zur Klassifizierung der Informationen eingesetzt und können dem Bereich des maschinellen Lernens zugeordnet werden. Machine Learning basiert im Wesentlichen auf dem Prinzip, dass grundlegende Entscheidungen von gesammelten Erfahrungen getroffen werden, aus denen Muster oder Regeln abgeleitet werden, mit dem Ziel Daten automatisch in Zielgruppen einordnen zu können. Die gesammelte Erfahrung bezieht sich auf die Eigenschaften der Daten oder Objekte, die qualitativer oder quantitativer Natur sein können. In diesem Kontext der Begriff Quantitativ beschreibt, dass ein Merkmal in der Regel jeden beliebigen Wert innerhalb eines Intervalls annehmen kann, wie z.B. Körpergröße oder Temperatur. Im Gegensatz zu quantitativen Merkmalen die qualitativen Merkmale beschreibt nur bestimmte Werte oder Ausprägungen, z.B. Geschlechtsangaben.

Es wird eine Datenmatrix erstellt, um eine Klassifizierung mit einem lernbasierten Ansatz durchzuführen, bei dem alle Objekte mit ihren Merkmalen eingegeben werden. Die Ausprägungen werden oft mit Zahlen kodiert, um die Objekte leichter vergleichen zu können. Anhand dieser Matrix wird versucht, die Ähnlichkeit oder Unähnlichkeit (Distanz) der einzelnen Objekte zu berechnen, um sie einer Klasse zuzuordnen [32]. Es gibt verschiedene Methoden zur Berechnung der Ähnlichkeit, die als Ähnlichkeits- oder Distanzmaße bezeichnet werden. Es wird untersucht, wie nahe den Eigenschaften von Objekten beieinander liegen und geht davon aus, dass ähnliche Objekte zu denselben Gruppen und unähnliche Objekte zu verschiedenen Gruppen gehören. Oft wird das Abstandsmaß noch auf das Intervall [0;1] skaliert, wobei 1 die maximale Ähnlichkeit und 0 die maximale Unähnlichkeit darstellt. Die Stimmungsanalyse basiert im Wesentlichen auf binären Ähnlichkeitsmodellen, d.h. ein Merkmal ist entweder vorhanden oder nicht vorhanden. Somit untersuchen die verschiedenen Maße Eigenschaften nur, um zu sehen, ob sie gleich oder ungleich sind, und unterscheiden sich nur darin, wie sie gemessen werden. Ähnlichkeitsmaße vergleichen Kombinationen von Werten mit gleichen Werten wie 1/1 oder 0/0. Distanzmaße hingegen bewerten Wertepaare aus 0/1 und 1/0.

Das am häufigsten verwendete Distanzmaß ist der euklidische Abstand. Der Abstand zwischen zwei Punkten entspricht der direkten Linie zwischen ihnen in einem Vektorraum (s. u. *Abbildung 3*). Der Abstand kann mit der folgenden Formel berechnet werden:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Formel 1: Euklidischen Distanz [35]

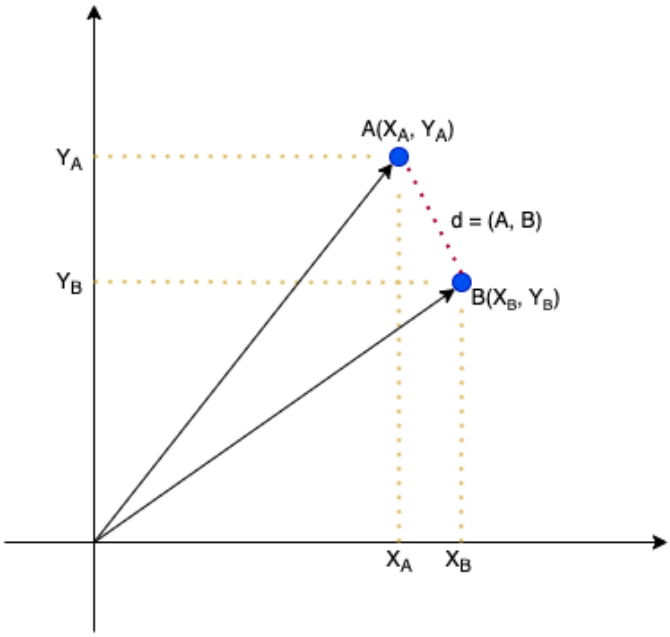


Abbildung 3: Modell der Euklidischen Distanz

Bei der Textklassifizierung werden auch Maschinelles Lernverfahren verwendet. Dabei wird davon ausgegangen, dass ein Dokument zu einem höherdimensionalen Raum von Dokumenten (Sätzen oder Wörtern) gehört, die verschiedenen Klassen enthält. Die Aufgabe des maschinellen Lernens besteht nun darin, die Dokumente anhand von Beispieldaten den Klassen zuzuordnen, was dann später an neuen Dokumenten getestet wird.

3.2.1 Naive Bayes

Die naive Bayes-Methode basiert auf den Bayes-Theoremen und ordnet dem Datensatz eine bedingte Wahrscheinlichkeit zu.

$$P(c \mid x) = \frac{P(x \mid c)\, P(c)}{P(x)}$$

Formel 2: Naive Bayes [36]

wobei,

- $P(c \mid x)$ bezeichnet die bedingte Wahrscheinlichkeit, dass das Dokument x zu der Klasse c gehört.
- $P(x \mid c)$ beschreibt die bedingte Wahrscheinlichkeit, die ein Dokument x unter allen Dokumenten der Klasse c zu finden.
- $P(c)$ bezeichnet die Wahrscheinlichkeit, die ein Dokument der Klasse c unter allen Dokumenten vorkommt.
- $P(x)$ ist die Wahrscheinlichkeit des Dokuments x im Korpus.

Es ist einen einfachen und effizienten Algorithmus, der auf verschiedene reale Probleme angewendet werden kann. Naive Bayes gehört zu Supervised Learning und arbeitet mit einem Wahrscheinlichkeitsmodell (*s.o. Formel 2*).

In der Stimmungsanalyse ist dies die Wahrscheinlichkeit, dass ein Wort und in der Summe ein ganzes Dokument als positiv oder negativ eingestuft werden kann. Der große Vorteil dieser Methode ist, dass man ein Modell mit einem relativ kleinen Trainingssatz trainieren kann [23]. Der größte Nachteil des Naive Bayes-Algorithmus ist jedoch seine grundlegende Annahme, dass die Attribute unabhängig voneinander sind. Diese Annahme ist in der Realität nicht haltbar, da die Attribute einer Datenmatrix regelmäßig korrelieren [24].

3.2.2 Support Vektor Maschine

Support Vektor Maschine, kurz bezeichnet SVM, sind die Kombination aus linearer Modellierung und instanzbasiertem Lernen und sind natürlich Algorithmen und keine Maschinen. Aufgrund ihrer Eigenschaften gehört die SVM zur Gruppe der überwachten Lernmethoden [37].

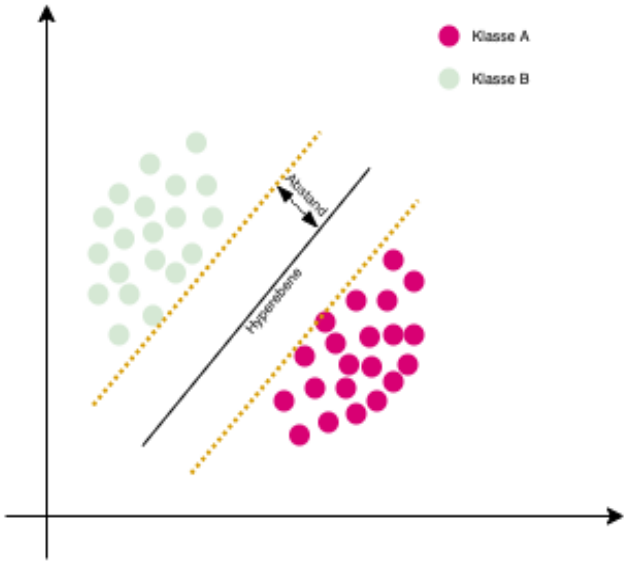


Abbildung 4: Support Vektor Maschine [27 s. 97-98]

Dieses Verfahren bestimmt die Klasse von Objekten basierend ihrer Position in einem n-dimensionalen Vektorraum, wobei n die Anzahl der Objekte bezeichnet. Wie bei den meisten Lernalgorithmen wird eine Menge von Trainingsdaten als Ausgangspunkt verwendet, deren Klassenzugehörigkeit bereits bekannt ist. Das Ziel ist es, eine Linie oder Trennfläche, die auch Hyperebene genannt (s. *Abbildung 4*), im Vektorraum zu finden, die die einzelnen Klassen möglichst sauber trennt und deren Abstand zueinander maximal ist. Oft sind die einzelnen Vektoren jedoch so relativ zueinander im Raum, dass sie nicht so einfach voneinander zu trennen sind. Dies wäre zum Beispiel nur durch eine Schlangenlinie möglich, deren Funktion ist aber schwer zu bestimmen (s. u. *Abbildung 5*).

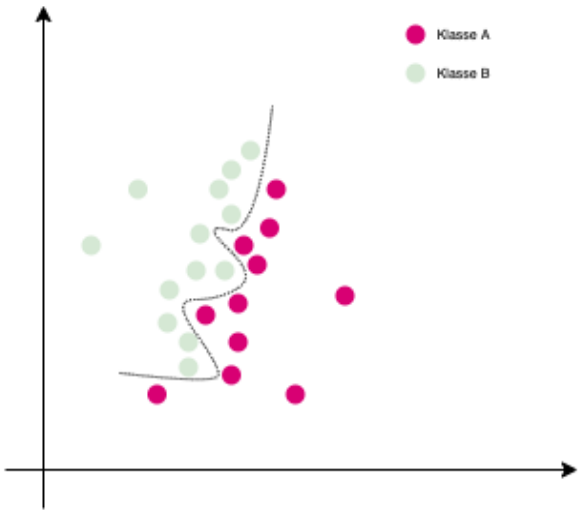


Abbildung 5: SVM Klassifizierung Probleme

Wenn das Problem jedoch in einen höherdimensionalen Raum übertragen wird, reicht es aus, eine einfache gerade Ebene zur Trennung zu verwenden, die mit Hilfe des Stützvektors berechnet werden kann (s. *Abbildung 6*). Dies ist der senkrechte Vektor in der Hyperebene und zeigt auf das nächste Dokument.

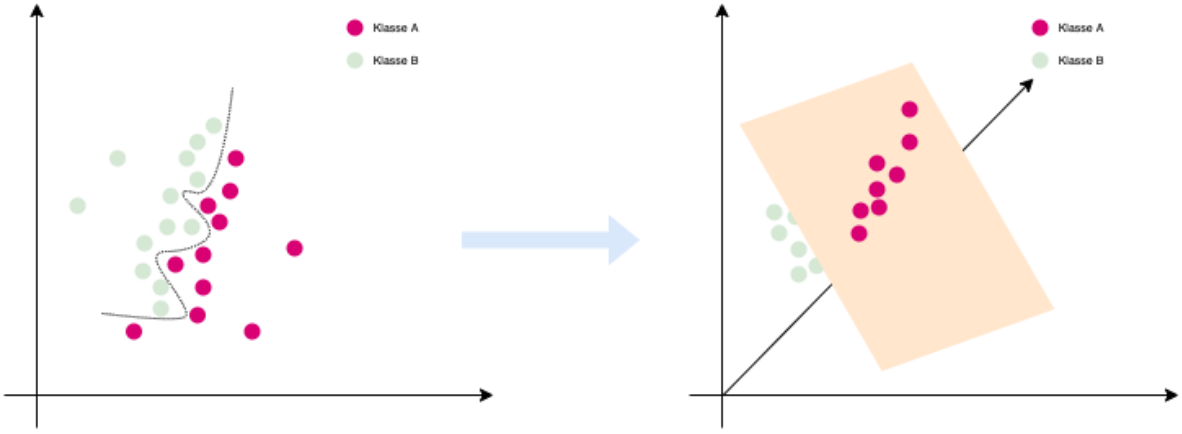


Abbildung 6: SVM in einem höherdimensionalen Raum

Eine einzelne Dimension ist eine Eigenschaft eines Objekts. Bei Texten wären dies zum Beispiel Wörter, wobei jedes Wort eine eigene Dimension darstellt. Da das Vorhandensein eines Wortes einzeln geprüft werden muss, gibt es nur binäre Dimensionen.

Später werden die Hyperebenen wieder zurücktransformiert und es können nicht-lineare Trennlinien entstehen, die die Klassen dennoch vollständig trennen. Um ein neues Dokument zu klassifizieren, muss man nur seinen Vektor berechnen und seinen Abstand zum Support-Vektor bestimmen.

Ein großer Vorteil von SVM ist, dass es bei ihrer Verwendung schwieriger ist, zu einer Überanpassung (overfitting) zu kommen, da die Komplexität der Klassen bereits im Algorithmus berücksichtigt wird [28]. Die SVM hat jedoch ein Nachteil. Bei fehlenden oder unregelmäßigen Daten ist eine Vorverarbeitung notwendig, und das resultierende Modell ist oft schwer zu interpretieren.

3.2.3 Maximale Entropie

Ein weiterer Klassifikator, der häufig für die Stimmungsanalyse verwendet wird, ist die maximale Entropie. Entropie bezieht sich in diesem Fall auf den Grad der Unsicherheit, so dass eine niedrige Entropie einen hohen Grad an Sicherheit darstellt und andersherum. Im Bereich der Textklassifizierung wird die Entropie auf Wahrscheinlichkeitsverteilung angewandt, die, ähnlich wie bei Naive Bayes, aus verschiedenen Variablen bestehen, z.B. aus Wörtern, Sätzen und deren Auftrittswahrscheinlichkeit [26]. Das Ziel ist es, eine Verteilung von Objekten in Klassen zu finden, die die maximale Entropie aufweist, da dann davon ausgegangen werden kann, dass alle Fakten auch berücksichtigt werden und die Wahrscheinlichkeiten möglichst gleichmäßig verteilt sind.

3.2.4 Cluster

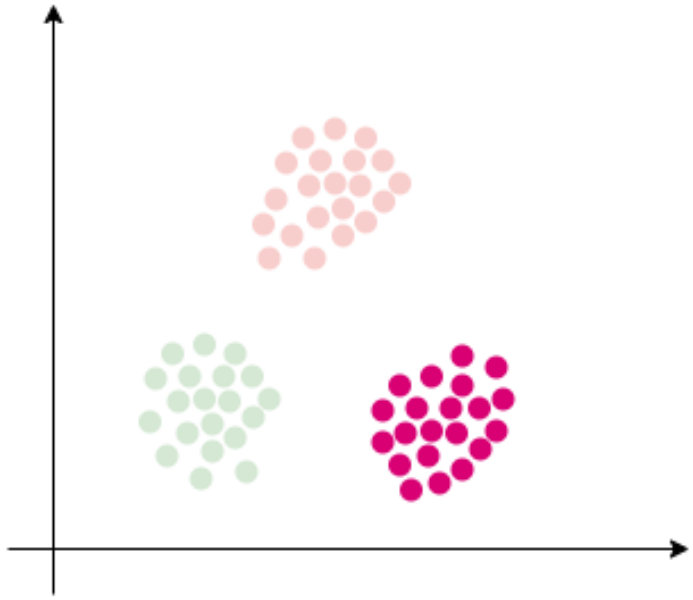


Abbildung 7: K-Nearest Neighbor Klassifikation [38]

Cluster-Klassifikatoren, wie der K-Nearest Neighbor (KNN) Algorithmus, sind instanzbasierte Lernmethoden, die die Zuordnung von Objekten zu Klassen auf der Grundlage ihrer Nähe zu anderen Objekten bestimmen [38 s. 153-156]. Ähnlich wie bei den Support-Vektor-Maschinen werden die Objekte als Vektoren in einem höherdimensionalen Vektorraum betrachtet, deren Abstand zueinander bestimmt wird. Vorzugsweise wird auf das euklidische Abstandsmaß zurückgegriffen.

Der Hauptvorteil der Verwendung von Cluster-Algorithmus besteht darin, dass er in der Lage ist, Klassen oder Gruppen nahezu optimal voneinander zu trennen. Der Nachteil ist jedoch, dass dieses Verfahren nicht so gut lernfähig ist und die Anzahl der zu ermittelnden Klassen in der Regel unbekannt ist [25].

4 Forschungsfragen und Lösung des Problems

Stoppwörter sind für das Verständnis der öffentlichen Stimmung bedeutungslos. Sie kommen häufig in den Texten vor, aber enthalten keine wichtigen Informationen. Das Ergebnis der statistischen Analyse der Dokumente zeigt, dass einige Wörter mit einer geringen Häufigkeit normalerweise genau das Gegenteil bewirken [19] . Zum Beispiel kommen deutsche Artikel ("der", "die", "das") in deutschen Texten häufig vor. Diese Wörter werden nur aus grammatikalischen Gründen verwendet, ohne wichtige Informationen zum Verständnis des Textes.

In diesem Kapitel werden die grundlegende und Forschungsfragen dieser Arbeit beschrieben. Diese Bachelorarbeit wird sich mit den folgenden grundlegenden Fragen befassen:

4.1 Wann ist ein Kunde beratungsintensiv? Kann man einen beratungsintesive Kunde lokalisieren?

Bei Störungen, technischer Beratung, vertragsbezogenen Fragen sowie rechnungsbezogenen Fragen sind die Kunden oft beratungsintensiv. Das gilt auch dann, wenn sie einen Termin vereinbaren oder verschieben möchten und ein Produkt wechseln wollen. Die untenstehende Tabelle beschreibt die Anzahl der Fragen der Kunden:

Thema	Anzahl der Fragen
Störung	9328
Technische Beratung	3252
Vertragsfragen	2860
Rechnungsfragen	1848
Techniker-Termin	1843
Produktänderung	961

Tabelle 4: Wann ist ein Kunde beratungsintensiv?

4.2 Sind die automatisch generierten Einträge (Memos) mit einer Stimmung verbunden?

Nein, die automatisch genierten Einträge enthalten kein Sentiment. Daher dürfen generierte Einträge nicht in den Datensatz aufgenommen werden. Nach Angaben von Herrn Mohr, es wurde bei der Auswahl von Einträgen alle automatisch generierten Einträge ausgeworfen [22].

Außerdem werden es in dieser Arbeit folgenden Themen untersucht:

4.3 Ist es notwendig, zusätzlich firmendefinierte Stoppwörter zu ermitteln, um ein besseres Ergebnis zu erzielen?

Ja, aus Sicht des Unternehmens ist es wichtig, eigene Stoppwörter zu definieren. Wörter, die häufig verwendet werden und keine wichtigen Informationen über das Sentiment enthalten, sollten als separate Stoppwörter definiert werden.

Experten der Abteilung Service Center und Abteilungsleiter IT-IE haben vorgeschlagen, die folgenden Wörter als unternehmensdefinierte Stoppwörter zu definieren:

netcheck, otrs, ggf, an, ab, axiros, noc, kann, beim, wird, mal, frau, sich, mit, von.

4.4 Ändert sich die Lösung nach der Streichung der Standard- und unternehmensbezogenen Stoppwörter?

Ja, die Lösung ändert sich nach der Entfernung der vom Unternehmen definierten Stoppwörter. Es gibt nicht so viele Änderungen, aber es gibt einige Änderungen bei Sentiment.

Sentiment	**Count**
Positive	5426
Negative	6574
Neutral	15390

Tabelle 5: Standard Stoppwörter

Sentiment	**Count**
Positive	5422
Negative	6575
Neutral	15393

Tabelle 6: Firmendefinierten Stoppwörter

Memo Nummer	Sentiment vorher	Sentiment nachher
3751	Positive	Neutral
5332	Positive	Neutral
16855	Positive	Negative
17296	Positive	Neutral

Tabelle 7: Änderungen in Sentiment

4.1 Gibt es ein Unterschied in Sentiment zwischen Telefonat und Face-2-Face?

Ja, bei Face2Face gibt es deutlich mehr neutrale Stimmungen als bei Telefonanrufen. Für die positive und negative Stimmung spielen Telefonanrufe und Face2Face ebenfalls eine Rolle, s. u. *Abbildung 8 und 9*.

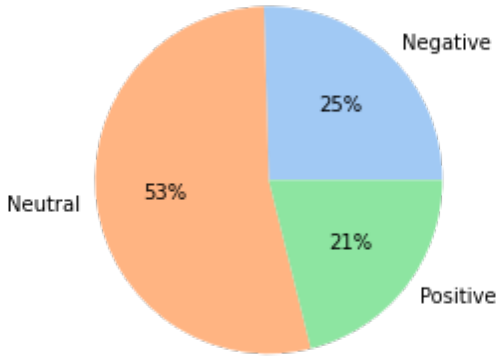


Abbildung 8: Sentiment in Anruf

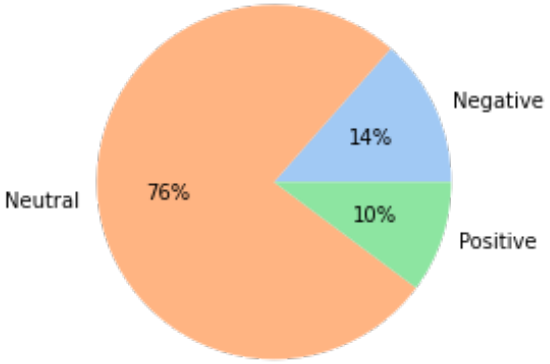


Abbildung 9: Sentiment in Face2Face

5 Softwareimplementierung

Dieses Kapitel umfasst die Implementierung eines Algorithmus für maschinelles Lernen für die Stimmungsanalyse in deutscher Sprache sowie die Evaluierung des Modells und das Bereitstellen des Models für einen operativen Einsatz bei wilhelm.tel.

5.1 Anforderungen

In diesem Abschnitt werden die Anforderungen für die Implementierung einer operationalen maschinellen Lernen Algorithmus für die Sentimentanalyse beschrieben.

5.1.1 Interview

Die Informationen sind mit Hilfe von Interviews und E-Mails gesammelt, an denen der zuständige Fachbereichleiter der Abteilung IT-Integration bei wilhelm.tel, Herr Mohr, aktiv beteiligt ist. Das Ziel der Interviews ist es, herauszufinden, wie man ein besseres maschinelles Lernmodell für die deutschsprachige Sentimentanalyse entwickelt und es bei einer technischen Kundenhotline zum Einsatz bringt.

5.1.2 Nicht-Funktionale Anforderungen

1. Identifizierung der beratungsintensiven Kunden, s.o. *4.1*.
2. Definieren der Unternehmens Stoppwörter, s.o. *4.3*.
3. Bestimmung der Stimmung zwischen Anruf und Face2Face, s.o. *4.5*

5.2 Entwurf

Im Kapitel 5.1 wurde die Anforderungen analysiert. In diesem Abschnitt wird eine Lösung für die oben genannten Anforderungen entworfen.

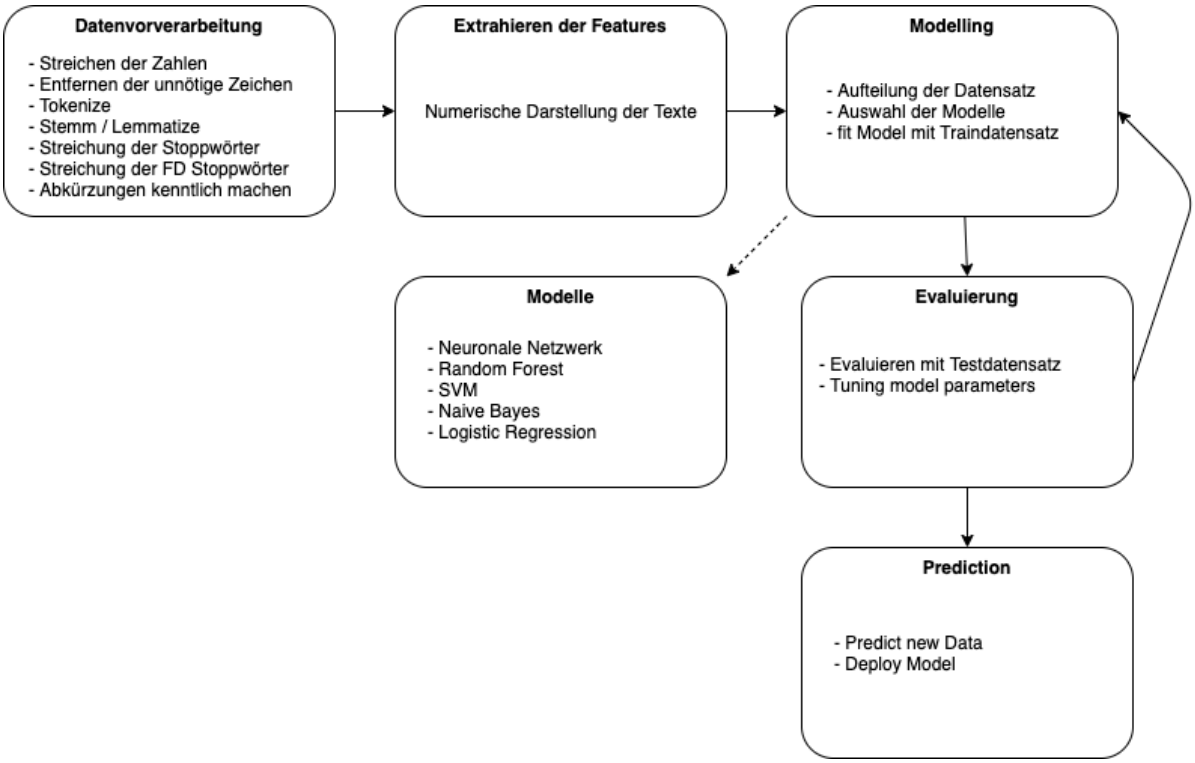


Abbildung 10: Sentimentanalyse Workflow

5.3 Implementierung

Dieser Abschnitt beschreibt die Implementierung anhand der Anforderungen und des Entwurfs. Es wird in diesem Kapitel über die Datenvorverarbeitung, Extrahieren der Features, das Erstellen des Modells und Evaluierung des Modells beschrieben.

5.3.1 Datensatz

Bevor man eine Sentimentanalyse durchführt, muss man zunächst Rohdaten für die Analyse beschaffen und vorverarbeiten, wenn man nicht bereits annotierte Texte verwendet. Der Datensatz für die Analyse wird von wilhelm.tel GmbH zur Verfügung gestellt. Der Datensatz enthält die Daten eines Zeitraums vom 01.01.2022 bis 31.03.2022 [22] und hat die folgenden Attribute:

MEMO_DATE	Zeitstempel des Eintrags
MEMO	Texteintrag (Memo)
CONTACT_REASON_ID	ID des Kontaktgrundes
DESCRIPTION	Kontaktgrund
Contact_Type_ID	ID des Kontakttyps
CONTACT_TYPE_GRUND	Kontakttyp

Tabelle 8: Attribute des Datensatzes

Der Datensatz enthält ungelabelten Texte aus Technische Kundenhotline, die in einer korrekten Form für die Analyse eingebracht werden müssen. Das nächste Kapitel beschreibt die Vorverarbeitung der Daten.

5.3.2 Datensatzvorverarbeitung

In diesem Kapitel werden geklärt, wie man die Rohdaten vorverarbeiten kann und in einer Form zur Sentimentanalyse bringen kann.

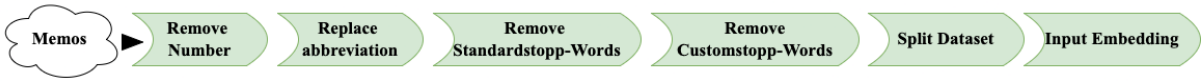


Abbildung 11: Datenvorverarbeitung

1. **Streichung der Zahlen:** Der Datensatz enthält auch Zahlen, wie z.B. die Ticketnummer mit oder ohne Rautenzeichen (#), die keine interessanten Informationen zur Stimmungsanalyse beitragen. Sie wurden aus dem Memo entfernt, um den Inhalt des Memos zu verfeinern.

2. **Ersetzen von Abbreviationen:** Die zu analysierende Datensatz enthält Firmendefinierte Abbreviationen, die wohl uns helfen können, um ein besseres Ergebnis zu erhalten. Aus diesem Grund werden die Abbreviationen mit vollständiger Form, wie z.B. **KD** mit **Kunde**, ersetzt.

   Nach Rücksprache mit den „Experten der Abteilung Service Center“ und „Fachbereichsleiter IT-IE“ wurden Spezifische Abkürzungen kenntlich gemacht (z.B. fb: FritzBox, wr: Werksreset, sr: Stromreset).

3. **Streichung der NaN-Werte:** Fehlende Werte oder deren Ersatzwerte können zu großen Fehlern in den Analyseergebnissen führen. Einige Zeilen der Datensatz enthalten NaN-Werte. Alle NaN-Werte wurden zeilenweise aus dem Datensatz entfernt.

4. **Streichung der Standardstoppwörter:** Stopwörter enthalten keine bedeutenden Informationen für die Stimmungsanalyse. Die meisten Forscher sind der Meinung, dass Stoppwörter bei der Klassifizierung von Stimmungen eine negative Rolle spielen und entfernen sie daher vor der Auswahl der Merkmale [19].

5. **Streichung der firmendefinierten Stoppwörter:** In dieser Arbeit wird ein Experiment für die Stimmungsanalyse mit und ohne firmendefinierte Stopp-wörter durchgeführt und verglichen, ob es eine Verbesserung der Stimmung nach der Streichung der firmendefinierten Stoppwörter gibt.

Durch ein „Experteninterview mit der Abteilung Service Center“ und „Fachbereichsleiter IT-IE“ wurden einige Stoppwörter abgestimmt (z.B. gegebenenfalls, ab, kann). Darüber hinaus wurden interne System-bezeichnungen als Stoppwörter definiert (z.B. netcheck, axiros).

6. **Aufteilung der Datensatz:** Um das Vertrauen in die Vorhersagefähigkeit des Modells zu erhöhen, wird eine Validierung durchgeführt. Die zu analysierende Datensatz wird auf Train und Testdatensatz aufgeteilt. In dieser Arbeit werden die verfügbaren Daten gleichmäßig nach dem Zufallsprinzip in 70% Trainingsdaten und 30% Testdaten aufgeteilt.

5.3.3 Tokenization

Bei der Tokenisierung wird der Text in Token umgewandelt, bevor er in Vektoren umgewandelt wird. Es ist auch einfacher, unnötige Token herauszufiltern. Zum Beispiel Aufteilung eines Dokuments in Absätze oder Sätze in Wörter. In diesem Fall zerlegen wir die MEMO in Wörter.

5.3.4 Beschriftung der unbeschrifteten Memos

Mit Hilfe von bereits veröffentlichter Bibliothek wie TextBlobDE und Lexikon SentiWS wurden ungelabelten Memos separat beschriftet und verglichen, ob es einen Unterschied zwischen den Stimmungen beider Methoden gibt. Da SentiWS mehr positive und negative Wortschatz enthält, gibt relative besseres Ergebnis.

Sentiment	Count
Positive	7314
Negative	2602
Neutral	17549

Tabelle 10: Labelling mit TextBlobDE

Sentiment	Count
Positive	5426
Negative	6574
Neutral	15390

Tabelle 9: Labelling mit SentiWS

5.3.5 Model

In der vorliegenden Abschlussarbeit wurde das BERT-Modell von Transformer für die Stimmungsanalyse verwendet. Dabei wurde das SentiWS-Lexikon für die Beschriftung der Memos verwendet.

BERT verfügt über einen eigenen Tokenizer. Es wird in dieser Arbeit bei dem Modell eine bereits trainierte Version des Tokenizers verwendet. BERT erwartet mehrere "spezielle" Token als Eingabe. [CLS] steht für Klassifizierung und markiert den Beginn einer neuen zu klassifizierender Eingabe. [SEP] markiert die Trennung zwischen Sätzen. Schließlich wird [PAD] als Platzhalter verwendet, um alle Vektoren auf dieselbe feste Länge aufzufüllen.

Nach mehreren Experimenten und Durchführung wurde es folgende Parameter für das Modell festgelegt:

```
MAXLEN = 192
BATCH_SIZE = 16
EPOCHS = 8
LEARNING_RATE = 1e-5
DATA_LENGTH = len(df)
```
, wobei df bezeichnet Dataframe

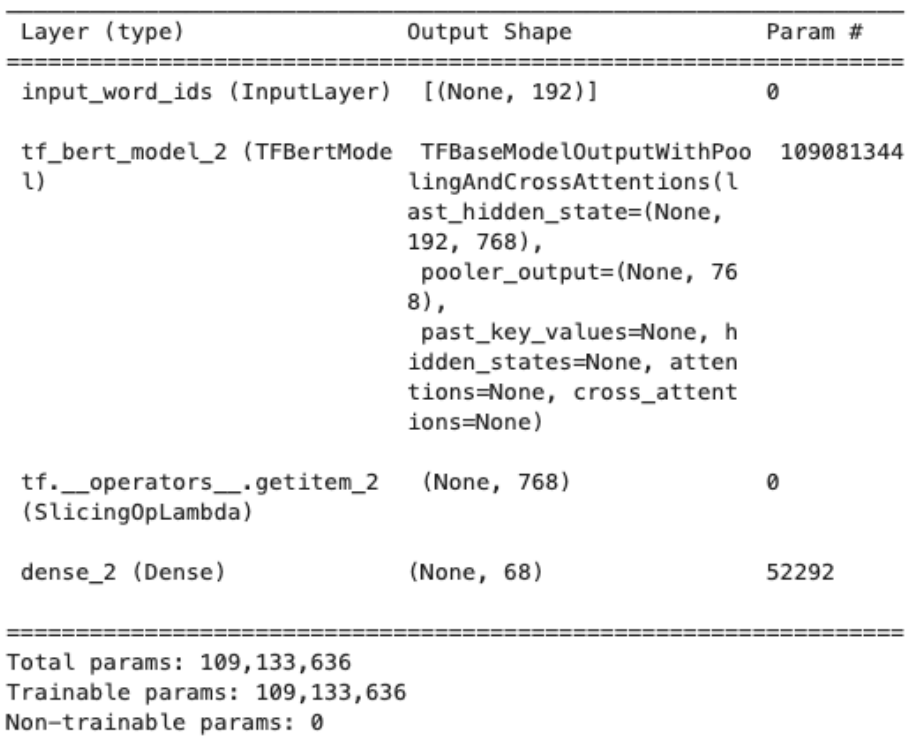
Da wir drei Sentimentklassen (Positive, Negative und Neutral) haben, müssen wir Softmax als Aktivierungsfunktion und Categorical-Crossentropy als Verlustfunktion entscheiden.

```
def build_model(transformer, max_len=MAXLEN):
    input_word_ids = tf.keras.layers.Input(
        shape=(max_len,), dtype=tf.int32, name="input_word_ids"
    )
    sequence_output = transformer(input_word_ids)[0]
    cls_token = sequence_output[:, 0, :]
    out = tf.keras.layers.Dense(3, activation="softmax")(cls_token)
    model = tf.keras.models.Model(inputs=input_word_ids,
outputs=out)
    model.compile(
        tf.keras.optimizers.Adam(lr=LEARNING_RATE),
        loss=" sparse_categorical_crossentropy",
        metrics=["accuracy"],
    )
    return model

transformer_layers = TFBertModel.from_pretrained("bert-base-german-cased")
model = build_model(transformer_layers, max_len=MAXLEN)
model.summary()
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_word_ids (InputLayer)  [(None, 192)]            0

 tf_bert_model (TFBertModel)  TFBaseModelOutputWithPoo  109081344
                             lingAndCrossAttentions(l
                             ast_hidden_state=(None,
                             192, 768),
                              pooler_output=(None, 76
                             8),
                              past_key_values=None, h
                             idden_states=None, atten
                             tions=None, cross_attent
                             ions=None)

 tf.__operators__.getitem (S  (None, 768)            0
 licingOpLambda)

 dense (Dense)               (None, 3)                 2307

=================================================================
Total params: 109,083,651
Trainable params: 109,083,651
Non-trainable params: 0
```

Abbildung 12: Überblick des Modelles für Sentiment Klassifizierung

Für die Klassifizierung der Memos wurde genauso wie Klassifizierung des Sentiments ein BERT-Modell erstellt. Wir haben insgesamt 68 Klassen, dazu eine Kundenanfrage gehören kann.



Layer (type)	Output Shape	Param #
input_word_ids (InputLayer)	[(None, 192)]	0
tf_bert_model_2 (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 192, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	109081344
tf.__operators__.getitem_2 (SlicingOpLambda)	(None, 768)	0
dense_2 (Dense)	(None, 68)	52292

Total params: 109,133,636
Trainable params: 109,133,636
Non-trainable params: 0

Abbildung 13: Überblick des Kategorie-Klassifizierungsmodelles

5.3.6 Trainieren des Models

Insgesamt haben wir ca. 110 Millionen trainierbare Parameter. Praktischerweise wurden fast alle davon bereits trainiert. Da es die bereits trainierten Gewichte für sie verwendet wird, muss sie nur noch etwas angepasst werden.

Als nächstes definieren wir Callbacks, die während des Trainings verwendet werden. Der EartlyStopping Callback stoppt das Training, wenn der Validierungsverlust zwischen den Epochen nicht mehr abnimmt. Dadurch wird eine Überanpassung vermieden. ModelCheckpoint speichert Checkpoints des Modells nach jeder Epoche.

```
callbacks = [
    tf.keras.callbacks.EarlyStopping(
        monitor="val_loss", verbose=1, patience=2,
restore_best_weights=True
    ),

     tf.keras.callbacks.ModelCheckpoint(
        "Model_{epoch:02d}_{val_loss:.4f}.h5",
        monitor="val_loss",
        save_best_only=True,
        verbose=1,
     )
]


steps_per_epoch = int(np.floor((len(train_ids) / BATCH_SIZE)))
print(
    f"Model Params:\nbatch_size: {BATCH_SIZE}\nEpochs: {EPOCHS}\n"
    f"Step p. Epoch: {steps_per_epoch}\n"
    f"Learning rate: {LEARNING_RATE}")

history = model.fit(
    train,
    batch_size=BATCH_SIZE,
    epochs=EPOCHS,
    steps_per_epoch=steps_per_epoch,
    validation_data=test,
    verbose=1,
    callbacks=callbacks
)
```

6 Evaluierung

In diesem Kapitel wird die Evaluierung des Modells beschrieben.

Dieser Bachelorarbeit konzentriert sich für die zwei Hauptaufgaben der NLP. Zum einen die Klassifizierung der Memos in verschiedene Kategorien von Kundenanfragen und zum anderen die Analyse der Stimmung.

Das Modell für die Klassifizierung der Stimmung hat bereits in der ersten Epoche 86% Trainingsgenauigkeit und 93% Validierungsgenauigkeit. Auf der anderen Seite hat es 34% Trainingsverlust und 18% Validierungsverlust. In der zweiten Epoche hat es aber sich stark verbessert. Der Trainingsverlust wurde von 34% auf 15% gesungen und die Trainingsgenauigkeit um 9% gestiegen. Das Training wurde bis zur vierten Epoche durchgeführt. Da es keine weiteren Verbesserungen gab, wurde das Training dann beendet.

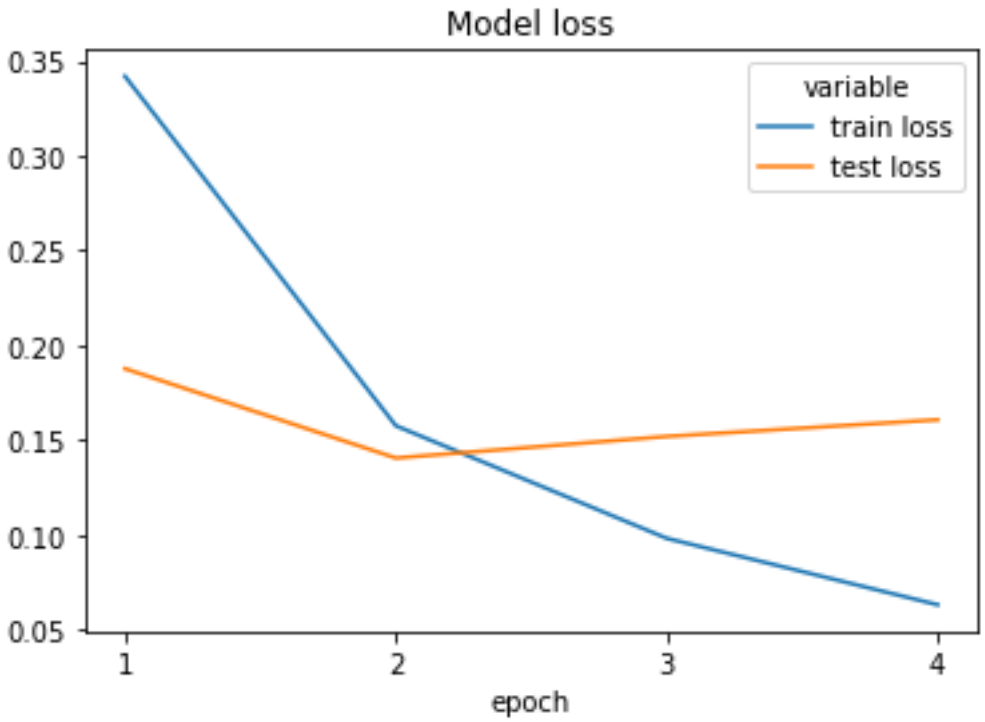


Abbildung 14: Sentiment Klassifizierung Loss

Das Modell zur Klassifizierung von Kundenanfragen macht langsam Fortschritte. Im ersten Durchlauf hat das Modell eine Trainingsgenauigkeit von über 50%. In weiteren Durchläufen hat sich das Modell merklich verbessert und in der fünften Epoche hat das Modell eine Trainingsgenauigkeit von 73%.

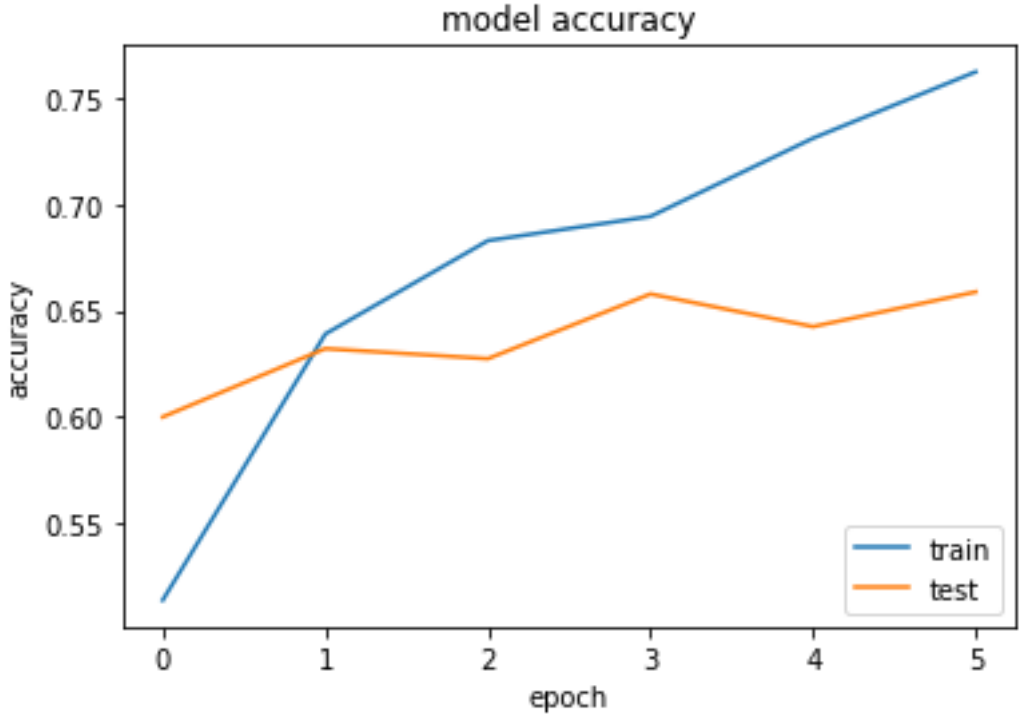


Abbildung 15: Memo Klassifizierung Accuracy

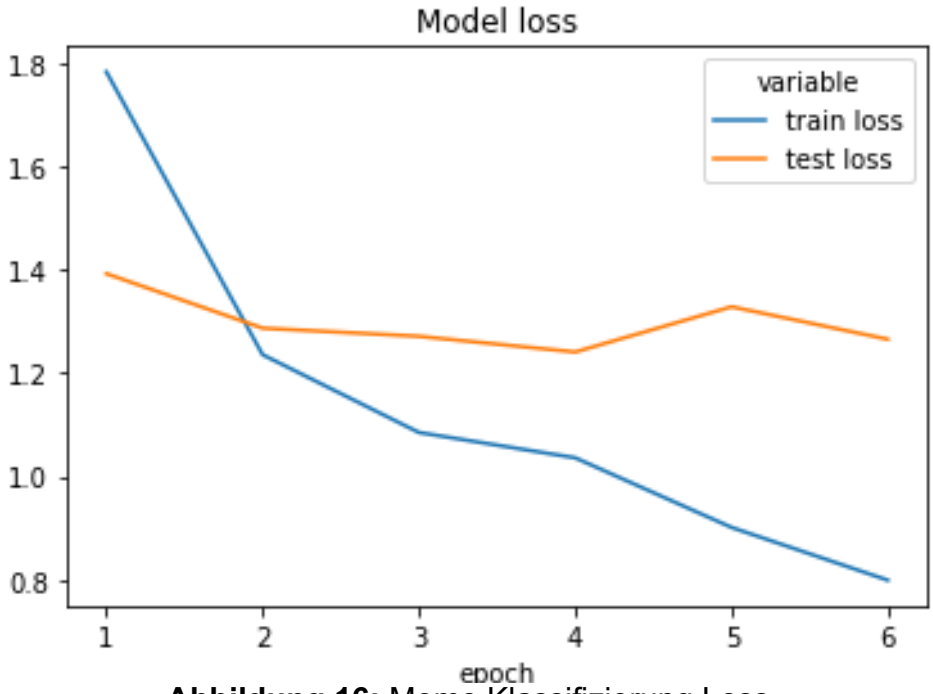


Abbildung 16: Memo Klassifizierung Loss

```
def preprocess_text(memos):
    input_ids = []
    for text in memos:
        encoded_sent = tokenizer.encode_plus(
            text=text,
            add_special_tokens=True
            max_length=MAXLEN,
            pad_to_max_length=True,
            return_attention_mask=False)
        input_ids.append(encoded_sent.get("input_ids"))
    return input_ids

memo = ["Internetverbindung seit heute Morgen abgebrochen, Grund
unbekannt."]
memo_ids = preprocess_text(memo)

predictions = model.predict(memo_ids)
predictions = np.argmax(predictions, axis=1)
predictions = [categories[pred] for pred in predictions]

predictions
['Störung']
```

7 Zusammenfassung

Dieses Kapitel fasst diese Abschlussarbeit zusammen und beschreibt ein Ausblick.

7.1 Zusammenfassung

Diese Arbeit beschäftigt sich mit Methoden des maschinellen Lernens im Bereich der natürlichen Sprachverarbeitung (NLP). Es wurde herausgefunden, inwieweit die Stimmung von deutschen Texten analysiert und klassifiziert werden kann, indem die Textdaten aus der technischen Kundenhotline verwendet wurden. Dabei wurden eine Datenvorverarbeitung und explorativer Datenanalyse auch durchgeführt.

Es wurde mit verschiedenen Lexika wie SentiWS und der frei verfügbaren Bibliothek TextBlobDE verwendet, um das Sentiment des Textes zu bestimmen. Durch das Experiment kann man beschlossen, dass SentiWS mehr effektiver, s.o. *Tabelle 9* und *Tabelle 10*.

Es wurde noch separat für die Kategorienklassifizierung der Memos ein Bert-Model entwickelt und evaluiert. Das Modell kann entscheiden, ob eine Kundenanfrage eine Störung, Terminvereinbarung oder Beratung usw. ist. Genauso für die Sentimentanalyse wurde auch ein BERT-Modell trainiert und evaluierte. Dieses Modell ist zuständig für die Klassifizierung der Stimmung der Kundenanfrage.

7.2 Ausblick

Insgesamt kann die vorliegende Arbeit noch erweitert werden und in der Zukunft für die Textklassifizierung und Stimmungsanalyse eingesetzt werden.

Abkürzungsverzeichnis

B		
	BERT	Bidirectional Encoder Representations from Transformers
D		
	DL	Deep Learning
E		
	EM	Emotionsanalyse
K		
	KI	Künstliche Intelligenz
	KNN	K-Nearest Neighbor
M		
	ML	Machine Learning
N		
	NLP	Natural Language Processing
P		
	POS	Part of Speech
T		
	TK	Textklassifizierung
S		
	SA	Sentimentanalyse, Stimmungsanalyse
	SVM	Support Vektor Machine

Literaturverzeichnis

1. Galit B. Yom-Tov, Shelly Ashtar, Daniel Altman, Michael Natapov, Neta Barkay, Monika Westphal, and Anat Rafaeli. 2018. Customer Sentiment in Web-Based Service Interactions: Automated Analyses and New Insights. In Companion Proceedings of the The Web Conference 2018 (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1689–1697. DOI:https://doi.org/10.1145/3184558.3191628
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805
3. Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep Learning--based Text Classification: A Comprehensive Review. ACM Comput. Surv. 54, 3, Article 62 (April 2022), 40 pages. DOI:https://doi.org/10.1145/3439726
4. B. S. Harish, S. Manjunath, and D. S. Guru. 2012. Text classification using symbolic similarity measure. In Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology (CCSEIT '12). Association for Computing Machinery, New York, NY, USA, 311–314. DOI:https://doi.org/10.1145/2393216.2393269
5. Detlev Frick, Andreas Gadatsch, Jens Kaufmann Birgit Lankes, Christoph Quix, Andreas Schmidt, Uwe Schmitz. 2021. Data Science Konzepte, Erfahrungen, Fallstudien und Praxis, DOI:https://doi.org/10.1007/978-3-658-33403-1 - [S. 259-260]
6. Jyoti and Seema Rao. 2016. A Survey on Sentiment Analysis and Opinion Mining. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing (AICTC '16). Association for Computing Machinery, New York, NY, USA, Article 53, 1–5. DOI:https://doi.org/10.1145/2979779.2979832
7. Melpomeni Alexa, Melanie Siegel. 2021. Tools für Social Listening und Sentiment-Analyse. DOI:https://doi.org/10.1007/978-3-658-33468-0 - [S. 88-89]
8. Kamil Topal and Gultekin Ozsoyoglu. 2016. Movie review analysis: emotion analysis of IMDb movie reviews. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '16). IEEE Press, 1170–1176.
9. Cássio Castaldi Araujo Blaz and Karin Becker. 2016. Sentiment analysis in tickets for IT support. In Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16). Association for Computing Machinery, New York, NY, USA, 235–246. DOI:https://doi.org/10.1145/2901739.2901781
10. Zornitsa Kozareva, Borja Navarro, Sonia Vázquez, and Andrés Montoyo. 2007. UA-ZBSA: a headline emotion classification through web information. In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval '07). Association for Computational Linguistics, USA, 334–337.
11. Olivier Janssens, Maarten Slembrouck, Steven Verstockt, Sofie Van Hoecke, and Rik Van de Walle. 2013. Real-time emotion classification of Tweets. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13). Association for Computing Machinery, New York, NY, USA, 1430–1431. DOI:https://doi.org/10.1145/2492517.2492577
12. Singla, Chaitanya and Singh, Sukhdev and Pathak, Monika, Automatic Audio Based Emotion Recognition System: Scope and Challenges (April 1, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, Available at SSRN: DOI: https://ssrn.com/abstract=3565861
13. Entwistle, N., & Ramsden, P. (1983). Understanding Student Learning (Routledge Revivals) (1st ed.). Routledge. DOI:https://doi.org/10.4324/9781315718637
14. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). DOI:https://doi.org/10.1038/nature14539

15. Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M. et al. Deep learning applications and challenges in big data analytics. Journal of Big Data 2, 1 (2015). DIO: https://doi.org/10.1186/s40537-014-0007-7
16. Melanie Siegel, Melpomeni Alexa (2020). Sentiment-Analyse deutschsprachiger Meinungsäußerungen, Springer Vieweg. DOI: https://doi.org/10.1007/978-3-658-29699-5 [S. 17-33, 49-69]
17. Yuwen Zhang, Zhaozhuo Xu, BERT for Question Answering on SQuAD 2.0 (2018).
18. CAO, Yanfang; ZHANG, Pu; XIONG, Anping. Sentiment analysis based on expanded aspect and polarity-ambiguous word lexicon. International Journal of Advanced Computer Science and Applications, 2015, 6. Jg., Nr. 2.
19. ZOU, Feng, et al. Automatic construction of Chinese stop word list. In: Proceedings of the 5th WSEAS international conference on Applied computer science. Stevens Point, WI, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006. S. 1010-1015.
20. Weiß, Oliver, Stadtwerke Norderstedt – die Unternehmensgruppe – Pressinformation – 01. Januar 2021
21. Weiß, Oliver, wilhelm.tel GmbH. – Pressinformation – 01. Januar 2021
22. Mohr, Stefan, Fachbereichsleiter – wilhelm.tel GmbH
23. Patel, Vishakha & Prabhu, Gayatri & Bhowmick, Kiran. (2015). A Survey of Opinion Mining and Sentiment Analysis. International Journal of Computer Applications. 131. 24-27. 10.5120/ijca2015907218.
24. SEITER, Mischa. Business analytics: effektive Nutzung fortschrittlicher Algorithmen in der Unternehmenssteuerung. Vahlen, 2017.
25. MAGIDSON, Jay; VERMUNT, Jeroen. Latent class models for clustering: A comparison with K-means. Canadian journal of marketing research, 2002, 20. Jg., Nr. 1, S. 36-43.
26. MCCALLUM, Andrew, et al. A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. 1998. S. 41-48.
27. ABERHAM, Jana; KURUC, Fabrizio. Support Vector Machine. In: Wie Maschinen lernen. Springer, Wiesbaden, 2019. S. 95-103.
28. HEINERT, Michael. Support Vector Machines–Teil 1: Ein theoretischer Überblick. Zeitschrift für Geodäsie, Geoinformation und Landmanagement, 2010, 3. Jg., S. 179-189.
29. Waltinger, Ulli. (2010). GermanPolarityClues: A Lexical Resource for German Sentiment Analysis.
30. REMUS, Robert; QUASTHOFF, Uwe; HEYER, Gerhard. Sentiws-a publicly available german-language resource for sentiment analysis. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). 2010.
31. GODBOLE, Namrata; SRINIVASAIAH, Manjunath; SKIENA, Steven. Large-Scale Sentiment Analysis for News and Blogs.
32. WUNDER, Johannes. Analyse des Verhaltens verschiedener Clusterverfahren nach Imputation fehlender Daten. 2014. Doktorarbeit.
33. PRADHA, Saurav; HALGAMUGE, Malka N.; VINH, Nguyen Tran Quoc. Effective text data preprocessing technique for sentiment analysis in social media data. In: 2019 11th international conference on knowledge and systems engineering (KSE). IEEE, 2019. S. 1-8.
34. SCHMIDT, Thomas; BURGHARDT, Manuel. An evaluation of lexicon-based sentiment analysis techniques for the plays of gotthold ephraim lessing.
35. de la Fraga, L.G., Silva, I.V., Cruz-Cortés, N. (2007). Euclidean Distance Fit of Ellipses with a Genetic Algorithm. In: Giacobini, M. (eds) Applications of Evolutionary Computing. EvoWorkshops 2007. Lecture Notes in Computer Science, vol 4448. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-71805-5_39
36. D. Isa, L. H. Lee, V. P. Kallimani and R. RajKumar, "Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine," in IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 9, pp. 1264-1272, Sept. 2008, doi: 10.1109/TKDE.2008.76.

37. Goodfellow, Ian, et al. Deep Learning. Das umfassende Handbuch : Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze, mitp, 2018. ProQuest Ebook Central,http://ebookcentral.proquest.com/lib/hawhamburg-ebooks/detail.action?docID=5598176.
38. STEINBACH, Michael; TAN, Pang-Ning. kNN: k-nearest neighbors. The top ten algorithms in data mining, 2009, S. 151-162.
39. MEDHAT, Walaa; HASSAN, Ahmed; KORASHY, Hoda. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 2014, 5. Jg., Nr. 4, S. 1093-1113.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den 06.06.2022