Universität Regensburg

Implementierung und Evaluation verschiedener Machine Learning-Ansätze für die Sentiment-Analyse im Deutschen

Bachelorarbeit im Fach Medieninformatik

am Institut für Information und Medien, Sprache und Kultur (I:IMSK)

Vorgelegt von:	Niklas Donhauser
Adresse:	Stadtweg 6, 92334 Berching
E-Mail (Universität):	Niklas.Donhauser@stud.uni-regensburg.de
E-Mail (privat):	niklasdonhauser97@freenet.de
Matrikelnummer:	2111397
Erstgutachter:	Prof. Dr. Christian Wolff
Zweitgutachter:	Prof. Dr. Udo Kruschwitz
Betreuer:	Herr Jakob Fehle
Laufendes Semester:	7. Semester B.A. Medieninformatik / Medienwissenschaft
Abgegeben am:	13.02.2023

Inhaltsverzeichnis

1. Einleitung **8**

2. Verwandte Arbeiten **10**
2.1. Grundlagen der Sentiment Analyse 10
2.2. Verfahren 11
2.2.1. Lexikonbasierte Ansätze 11
2.2.2. Machine Learning-Ansätze 12
2.2.3. Kombination beider Ansätze 13
2.3. Anwendungsgebiete 13
2.4. Evaluationsmetriken 15

3. Ressourcen für die Sentiment Analyse **19**
3.1. Literarische Texte 19
3.2. Gemischte Korpora aus unterschiedlichen Domänen 20
3.3. Nachrichtenartikel 21
3.4. Produktbewertungen 22
3.5. Soziale Medien 24
3.6. Zusammenfassung 27

4. Zielsetzung der Arbeit **29**

5. Verarbeitung der Korpora **31**
5.1. Normalisierung der Datensätze 31
5.2. Bereinigung der Texteinheiten 32
5.3. Entfernung von Stoppwörtern 34
5.4. Umwandlung von Emoticons 35
5.5. Stemming 36
5.6. Einheitliche Groß- und Kleinschreibung 38

6. Definition und eigene Implementierung der Machine Learning-Ansätze **39**
6.1. Praktisches Vorgehen bei der Implementierung 39
6.2. Support Vector Machines 40
6.3. Naïve Bayes 41
6.4. Künstliche neuronale Netze 41
6.4.1. Convolutional Neural Networks 42
6.4.2. Recurrent Neural Networks 43
6.5. Transformer-Modelle 44
6.5.1. GBERT 45
6.5.2. GELECTRA 46
6.5.3. BERT Base Multilingual Uncased Sentiment 46

7. Ergebnisse **47**
7.1. Vergleich der vorverarbeitenden Schritte für Transformer-Modelle . . 48
7.2. Vergleich der Transformer-Modelle in der Ausführung Base und Large 50

7.3. Evaluation von Domänen und Ansätzen 52
7.3.1. Literarische Texte 52
7.3.2. Texte aus gemischten Domänen 53
7.3.3. Nachrichten Artikel 54
7.3.4. Produktbewertungen 55
7.3.5. Social Media 57
7.4. Betrachtung der einzelnen Polaritätsklassen 58

8. Diskussion **63**
8.1. Vorverarbeitungsschritte bei Transformern 63
8.2. Die Base und Large Varianten der Transformer-Modelle 64
8.3. Die Ergebnisse der Kreuzevaluation 65
8.4. Zusammenfassung der Beobachtungen 68

9. Zusammenfassung und Ausblick **69**

Literaturverzeichnis **71**

A. Laufzeiten der Methoden und Transformer-Varianten pro Domäne **79**

B. Durchschnittliche Accuracy- und F1-Werte der einzelnen Algorithmen auf Domänen **82**

C. Accuracy-Werte der einzelnen Klassen pro Korpus **85**

Erklärung zur Urheberschaft **87**

Erklärung zur Lizenzierung und Publikation dieser Arbeit **88**

Tabellenverzeichnis

1. Binäre Confusion Matrix 15

2. Übersicht aller verwendeten deutschen Korpora. Bei Angaben mit einem Stern*, sind die Korpora auf circa 70.000 Einträge gekürzt worden 28

3. Darstellung einer Texteinheit in einem Korpus 32
4. Wortstammreduktion des Wortes „ springen “ mittels des Snowball Stemmer 37

5. Durchschnittliche dichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich mit und ohne vorverarbeitenden Schritte 48
6. Durchschnittliche trichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich mit und ohne vorverarbeitenden Schritte 49
7. Durchschnittliche dichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich der beiden Ausführungen Base und Large 50
8. Durchschnittliche trichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich der beiden Ausführungen Base und Large 51
9. Accuracy-Werte für literarische Texte 52
10. F1-Werte für literarische Texte 53
11. Accuracy-Werte für Texte aus gemischten Domänen 53
12. F1-Werte für Texte aus gemischten Domänen 54
13. Accuracy-Werte für Nachrichtenartikel 54
14. F1-Werte für Nachrichtenartikeln 55
15. Accuracy-Werte für Produktbewertungen 56
16. F1-Werte für Produktbewertungen 56
17. Accuracy-Werte für Texte aus den sozialen Medien 57
18. F1-Werte für Texte aus den sozialen Medien 58
19. Angabe der durchschnittlichen Accuracy einer Klasse im dichotomen Fall für alle Ansätze 59
20. Angabe der durchschnittlichen Accuracy einer Klasse im trichotomen Fall für alle Ansätze 62

21. Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von jedem Modell bei dichotomen Domänen 79
22. Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von jedem Modell bei trichotomen Domänen 80
23. Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von den Varianten Base und Large im dichotomen Fall. Zusätzlich ist die zeitliche Differenz der beiden Varianten gegeben 80

24. Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von den Varianten Base und Large im trichotomen Fall. Zusätzlich ist die zeitliche Differenz der beiden Varianten gegeben 81

25. Durchschnittliche Accuracy der einzelnen Algorithmen im dichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen 82
26. Durchschnittliche Accuracy der einzelnen Algorithmen im trichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen 83
27. Durchschnittliche F1-Werte der einzelnen Algorithmen im dichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen 83
28. Durchschnittliche F1-Werte der einzelnen Algorithmen im trichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen 84

29. Angabe der besten und schlechtesten Accuracy-Werte innerhalb einer Klasse pro Korpus im dichotomen Fall. Außerdem Angabe des Ansatzes, der diese Werte erzeugte 85
30. Angabe der besten und schlechtesten Accuracy-Werte innerhalb einer Klasse pro Korpus im trichotomen Fall. Außerdem Angabe des Ansatzes, der diese Werte erzeugte 86

Zusammenfassung

Im englischsprachigen Raum werden für die Sentiment Analyse häufig maschinelle Lernverfahren eingesetzt. In anderen Sprachen wie dem Deutschen gibt es noch keine umfassende Analyse von verschiedenen *Machine Learning*-Ansätzen auf gleichsprachigen Korpora. Diese Arbeit versucht diese Lücke zu schließen, indem sieben verschiedene Ansätze auf 20 mit Sentiment annotierten Korpora evaluiert werden. Mithilfe der Kreuzevaluation können Empfehlungen für verschiedene Domänen in zukünftigen Arbeiten ausgesprochen werden.

Dazu wurden bei klassischen Ansätzen und neuronalen Netzen vorverarbeitende Schritt auf allen Korpora durchgeführt. Weiterhin untersuchten Unterstudien den Einfluss solcher vorverarbeitender Schritte auf Transformerbasierten Ansätzen. Außerdem wurden die *Base* und *Large* Varianten von zwei Transformer-Modellen miteinander verglichen.

Die Kreuzevaluation untersuchte alle Ansätze auf jedem Korpus in dichotomer und trichotomer Form, wobei Metriken wie *Accuracy* und der F1-Wert zur Einordnung der Ergebnisse verwendet wurden.

Im Durchschnitt zeigte das Modell *GBERT* die besten Ergebnisse und erreichte über alle Korpora verteilt einen F1-Wert von 78,23 %. Dabei wurde ein maximaler F1-Wert von 92,77 % in der Domäne der Produktbewertungen erreicht.

Aufgrund der Resultate in dieser Arbeit wird in den meisten Fällen bei transformerbasierten Ansätzen kein *Preprocessing* empfohlen. Die Verwendung der *Large*-Variante sollte ebenfalls in den meisten Fällen verwendet werden.

Abstract

In English-speaking countries, machine learning approaches are often used for sentiment analysis. In other languages such as German, there is not yet a comprehensive analysis of different *machine learning* approaches on same-language corpora. This paper attempts to fill this gap by evaluating seven different approaches on 20 with sentiment annotated corpora. With the help of the cross-evaluation, recommendations can be made for the different domains in the future work.

For this purpose, *preprocessing* steps were performed on all corpora for classical approaches and neural networks. Furthermore, sub-studies investigated the influence of such *preprocessing* steps on transformer-based approaches. In addition, the *Base* and *Large* variants of two transformer models were compared.

The cross-evaluation examined all approaches on each corpus in dichotomous and trichotomous polarity, using metrics such as *accuracy* and the F1-measure to rank the results.

On average, the *GBERT* model showed the best results, achieving an F1-measure of 78.23 % across all corpora. A maximum F1-measure of 92.77 % was achieved in the domain of product evaluations.

Based on the results in this work, no *preprocessing* is recommended in most cases for transformer-based approaches. The use of the *Large* variant should also be used in most cases.

1. Einleitung

Mit dem Aufkommen des Internets und der vermehrten Nutzung von sozialen Medien stieg auch die Möglichkeit die eigene Meinung zu verbreiten. Mit dieser Art der Meinungsäußerung wurde das Verlangen nach einer Echtzeit-Analyse dieser Aussagen immer dringender, da die Reichweite und die Ausbreitung dieser Stimmungen immer mehr zunahm (Pawar et al., 2016).

Bei der Sentiment Analyse handelt es sich um die Analyse von Meinungen, Emotionen und Bewertungen von Menschen auf Ereignisse, Personen und Produkte (Liu, 2012, S.7). Diese Auswertung kann mit verschiedenen Methoden erfolgen, es gibt Ansätze mit Lexika zur Ermittlung von Meinungen, aber auch Methoden des *Machine Learning* sind anwendbar. Besonders im englischen Sprachraum werden maschinelle Lernmethoden eingesetzt, da hier eine stetig steigende Anzahl an mit Sentiment annotierten Korpora vorliegt. Diese Datensätze werden benötigt um solche Methoden zu trainieren (Balazs & Velásquez, 2016) In anderen Sprachen gibt es nur wenige Datensätze, die sich in Qualität und Größe für die maschinelle Sentiment Analyse eignen (Fehle et al., 2021). Daher ist es umso wichtiger, die richtigen Ansätze und Vorbereitungen für Sprachen wie dem Deutschen auszuwählen, da so die Qualität der Ergebnisse gesteigert werden kann.

Mit dieser Arbeit soll eine Basis geschaffen werden, die bei der Auswahl des geeigneten Algorithmus als Entscheidungshilfe herangezogen werden kann. Anhand einer Kreuzevaluation der verschiedenen Ansätze auf den verfügbaren deuschen Korpora soll eine Entscheidungsgrundlage für die deutsche Sentiment Analyse geschaffen werden.

Verschiedene Arbeiten behandeln die Analyse von Ansätzen und Domänen in der englischen Sprache und geben Empfehlungen für Domänen wie Produktbewertungen(Tang et al., 2009) oder für die sozialen Medien (Yue et al., 2019) an. Andere

Arbeiten befassten sich mit der Sentiment Analyse in anderen Sprachen und zeigten eine umfassende Darstellung von Ansätzen und Hilfsprogrammen(Dashtipour et al., 2016; Lo et al., 2017) Die Wahl des richtigen Algorithmus wurde ebenfalls in anderen Arbeiten behandelt (L. Zhang et al., 2018; H. Zhang et al., 2014).

Der Aufbau der Arbeit ist wie folgt. In Kapitel zwei werden grundlegende Begriffe der Sentiment Analyse und verschiedene übergeordnete Ansätze erläutert. Außerdem werden Anwendungsgebiete der Sentiment Analyse aufgelistet. Weiterhin werden Evaluationsmetriken dargestellt, mithilfe dieser die verschiedenen Ansätze verglichen werden können. Im dritten Kapitel werden die verwendeten Korpora vorgestellt, die die Basis dieser Arbeit darstellen. Im vierten Kapitel wird auf Ziele und Vorgehen dieser Arbeit eingegangen. In Kapitel fünf wird die Verarbeitung der Korpora thematisiert. Das sechste Kapitel stellt alle verwendeten Ansätze in deren Grundzügen vor und erläutert das eigene Vorgehen. Die Ergebnisse werden im siebten Kapitel vorgestellt. Diese Resultate werden auf einzelnen Korpora und ganzen Domänen betrachtet. Weiterhin werden auch die beiden Unterstudien zu den Transformer-basierten Ansätzen im Bezug auf Ergebnisse behandelt. In Kapitel acht werden die Ergebnisse diskutiert und zusammengefasst. Das letzte Kapitel gibt eine Zusammenfassung und einen Ausblick auf das Thema.

2. Verwandte Arbeiten

Im folgenden Kapitel werden Grundbegriffe und das Konzept der Sentiment Analyse behandelt. Weiterhin werden Anwendungsgebiete und Verfahren der Sentimentbestimmung dargelegt. Ebenfalls werden alle Metriken definiert, die in dieser Arbeit zur Evaluation angewandt werden.

2.1. Grundlagen der Sentiment Analyse

Bei der Sentiment Analyse werden Stimmungen, Meinungen und Emotionen von Menschen im Bezug auf ein bestimmtes Thema, Produkt oder Ereignis untersucht. Die Polarität, die in einem Text vorkommt, wird in folgende Kategorien eingeordnet: Positiv, Negativ und Neutral. Auch die Einordnung in Gefühlskategorien und Emotionen ist möglich (Schmidt et al., 2018). Weiterhin gibt es noch metrische Zuordnungen, die die Stärke der Meinung ausdrücken kann. Texte, die mit einer neutralen Polarität gekennzeichnet sind, beinhalten meist keine Meinung. Dabei ist die Sentiment Analyse, auch *Opinion Mining* genannt, dem Feld der natürlichen Sprachverarbeitung zuzuordnen (Liu, 2012, S. 7-11 u. S.34).

Das analysieren von Meinungen in Texten, kann auf unterschiedlichen Ebenen durchgeführt werden. Bei der Dokumentebene wird das Sentiment für das gesamte Dokument bestimmt. Dabei wird angenommen, dass der gesamte Inhalt über die gleiche Entität handelt. Auf der Satzebene werden Meinungen innerhalb dessen Grenzen bestimmt. Einzelne Sätze können objektiv oder subjektiv eingeordnet werden. Wobei objektive Sätze Informationen vermitteln und subjektive Sätze, Meinungen und Standpunkte von Entitäten ausdrücken. Bei der Entitäts- oder Aspektebene besteht eine Meinung aus dem Sentiment und dem Ziel. Dabei können mehrere Meinungen innerhalb eines Satzes vorliegen. Dies kann zu Problemen führen, wenn ein Dokument mehrere Meinungen beinhaltet (Liu, 2012, S. 10-12). Die Analyse auf

Dokumentebene wird in dieser Arbeit zur Bestimmung des Sentiments verwendet. Dieses Vorgehen wurde bereits in anderen Arbeiten zur Sentimentbestimmung angewandt (Dey et al., 2016; Bütow et al., 2016)

2.2. Verfahren

Es gibt verschiedene Verfahren, die für die Sentiment Analyse verwendet werden. Nachfolgend werden einige dieser Vorgehen genauer erläutert.

2.2.1. Lexikonbasierte Ansätze

Um die Polarität eines Satzes oder eines Dokumentes zu bestimmen, benötigt man Wörter, die ein Sentiment beinhalten. Solche Wörter werden als *opinion words* bezeichnet. Eine Liste solcher Wörter wird in einem Sentiment Lexikon zusammengefasst. Diese Lexika können verwendet werden, um das Sentiment einer Texteinheit zu bestimmen. Bei dieser Art der Analyse können einige Probleme auftreten. Die *opinion words* können je nach Domäne auch das gegensätzliche Ausdrücken und sind nicht immer einer festen Polarität zugeordnet. Ebenso können diese Wörter in Fragesätzen verwendet werden. Somit kann kein Sentiment enthalten sein. Sarkasmus kann die Polarität umkehren und so eine Bestimmung des Sentiment erschweren. Ebenso gibt es Sätze, die ohne *opinion words* auskommen, aber dennoch eine Meinung beinhalten. Dies kann mit folgendem Beispiel verdeutlicht werden (Liu, 2012, S. 12-13).

„ Der Computer benötigt zu viel Strom. “

Dieses Beispiel zeigt das ein negatives Sentiment auch ohne *opinion word* ausgedrückt werden kann, da in diesem Fall das Gerät viele Ressourcen benötigt, dies kann negativ gewertet werden.

Die zuvor erwähnten *opinion words* besitzen einen numerischen Wert oder werden in Kategorien eingeordnet. Bei der numerischen Zuordnung wird dieser Wert entweder binär, wobei -1 für negativ und +1 für positiv steht oder auf einer Skala festgelegt. Bei einer Festlegung durch eine Skala können so einzelne Wörter mit der Intensität der Polarität ausgestattet werden. Um das Sentiment eines Dokumentes

beziehungsweise einer Texteinheit zu bestimmen, werden die einzelnen Werte der Wörter addiert, um so das Sentiment dieser Einheit zu bestimmen. Ein positives Ergebnis steht für eine Texteinheit mit positiver Polarität, hingegen ein negatives Ergebnis für eine negative Polarität steht (Fehle et al., 2021).

Um bessere Ergebnisse bei diesem Verfahren zu erzielen, werden verschiedene Vorverarbeitungsschritte angewandt. Dazu zählen Methoden wie das Entfernen von Stoppwörtern, Links, Nutzernamen und speziellen Charakteren. Weiterhin werden *Part of Speech* Informationen hinzugefügt. Auch Lemmatisierung und das Stemmen von Wörtern kann angewandt werden. Ebenso werden Texte *lowercased* und *Valence shifter* erkannt (Fehle et al., 2021).

2.2.2. Machine Learning-Ansätze

Es gibt verschiedene Ansätze im Bereich des maschinellen Lernens, um die Problemstellung der Sentiment Analyse zu beantworten. Darunter zählen klassische Methoden wie *Support Vector Machines* oder auch *Naïve Bayes*. Bei diesen Arten von Ansätzen werden Trainingsdaten verwendet, um das Modell zu trainieren. Diese schriftlichen Daten werden aus verschiedenen Domänen generiert beziehungsweise durch *Text Mining* gesammelt. Anschließend werden diese Daten mit weiteren Informationen annotiert, darunter fällt das Sentiment einer Texteinheit. Mithilfe dieser Informationen kann der Algorithmus eine Sentimentbestimmung vornehmen und so Texte klassifizieren (Tang et al., 2009). In anderen Sprachen wie dem Englischen konnten somit gute Ergebnisse in der Sentiment Analyse erzielt werden (Basarslan et al., 2020).

Weitere Möglichkeiten für die Sentiment Analyse sind künstliche neuronale Netzwerke. Künstliche neuronale Netzwerke können in *feedforward neural networks* und *recurrent neural networks* aufgeteilt werden. Bei beiden Varianten werden unterschiedliche Schichten, auch *layers* genannt, angelegt. Diese sind mit Informationsverarbeitenden Einheiten beziehungsweise *Neuronen* ausgestattet. Dadurch das mehrere Schichten mit vielen Neuronen aneinandergereiht sind, spricht man auch vom *Deep Learning*. Spezialfälle, die später in dieser Arbeit verwendet werden, sind *Convo-*

lutional Neural Networks (*CNN*) und *Long short-term memory networks* (*LSTM*) eine Abwandlung eines *recurrent neural networks* (L. Zhang et al., 2018).

Mit dem Auftreten von Transformern im Jahr 2017 wurden neu Wege geschaffen, um verschiedene natürliche Sprachverarbeitungsaufgaben zu lösen. Dieses Modell besteht aus vielen komplexen *recurrent* und *convolutional Neural Networks*. Weiterhin gibt es im Aufbau einen *Decoder* und einen *Encoder*. Eine Weiterentwicklung ist das *BERT*-Modell. *BERT* steht für *Bidirectional Encoder Representations from Transformers*. Solche Modelle sind mit einer Vielzahl von textuellen Daten vortrainiert und werden an die finale Aufgabe angepasst. Dies wird durch das sogenannte *fine-tuning* vorgenommen (Devlin et al., 2019; Vaswani et al., 2017).

2.2.3. Kombination beider Ansätze

Bei der Kombination der beiden zuvor genannten Verfahren wird versucht bessere Ergebnisse bei der Sentiment Analyse zu erreichen. Ein Vorgehen hierbei ist es, verschiedene einzelne Verfahren hintereinander auszuführen. Dabei werden die zu bestimmenden Texteinheiten von einzelnen Methoden einer Polarität zugeordnet. Falls hierbei eine Methode keine eindeutige Bestimmung der Polarität vorlegen kann, wird die Texteinheit weitergegeben. Somit kann durch das Verbinden von regelbasierten, statistischen und klassischen Methoden die Genauigkeit der Sentiment Analyse erhöht werden (Prabowo & Thelwall, 2009).

Ein anderer Ansatz ist es, verschiedene Methoden zu verwenden, die die Polarität von Texteinheiten bestimmen und danach mittels Gewichtung eine einheitliche Bestimmung des Sentiment durchführt. Dabei ist zu beachten, dass diese Methode bei einer größeren Anzahl an Methoden sich wenig verbessert. Die Auswahl der Methoden sollte an den Datensatz und die Domäne angepasst werden (Gonçalves et al., 2013).

2.3. Anwendungsgebiete

Es gibt verschiedene Bereiche, in denen die Sentiment Analyse Verwendung findet. Ursprünglich eingesetzt in den Feldern der Produktbewertungen und Filmrezen-

sionen, wurden auch andere Bereiche wie soziale Medien, der Finanzsektor und der Gesundheitssektor eingebunden. So zeigten verschiedene Arbeiten, wie man Produktbewertungen mittels der Sentiment Analyse klassifizieren kann (Denecke, 2008). Auch die Einordnung von Rezensionen aus der Gastronomie in unterschiedliche Polaritätsklassen konnte mittels der Sentiment Analyse bewerkstelligt werden. So haben Kang et al. (2012) ein Sentimentlexikon mittels Restaurantbewertungen erstellt, das als *Feature* in den *Naïve Bayes* Algorithmus integriert werden kann. Dadurch konnte die Genauigkeit für die Sentiment Bestimmung einer Klasse verbessert werden (Kang et al., 2012). Ein weiteres Feld, das untersucht wurde, sind Texte aus den sozialen Medien. Hierbei wurden verschiedene Themen wie zum Beispiel Politik oder Sport auf Meinungen untersucht. So konnten Schmidt et al. (2022) aufzeigen, dass die *Tweets* über die deutsche Bundestagswahl eher negativ konnotiert waren und das es während der Wahlperiode starke Stimmungswechsel auf Twitter gab (Schmidt et al., 2022). Andere Autoren wie Wunderlich & Memmert (2020) analysierten *Tweets* durch einen lexikonbasierten Ansatz. Hierbei wurden Texte aus der Domäne Sport analysiert (Wunderlich & Memmert, 2020). Andere Arbeiten haben gezeigt, dass Emotionen ebenfalls extrahiert werden können. Eine besondere Herausforderung hierbei ist, dass die meisten Emotionen nicht explizit in Texten ausgedrückt werden. Die Autoren Balahur et al. (2012) haben in ihrer Arbeit bereits existierende Algorithmen verglichen und einen eigenen Ansatz präsentiert, der bessere Ergebnisse in der Klassifikation von Emotionen liefert (Balahur et al., 2012). Andere Arbeiten untersuchten die Sentiment Analyse im Bezug auf Infektionskrankheiten und der *COVID-19*-Pandemie. Hierbei wurden verschiedene Ansätze aufgeführt und dessen Zielsetzung dargelegt. Eine systematisch Übersicht über dieses Thema wird in der Arbeit von Alamoodi et al. (2021) aufgeführt. Dabei werden verschiedene Anwendungsfälle beschrieben, wie der Verlauf der öffentlichen Meinung oder die Ausbreitung von Fehlinformationen (Alamoodi et al., 2021). Ein anderer Bereich, in dem die Sentiment Analyse eingesetzt wird, ist die Wirtschaft. Hierbei wurden von Schumaker et al. (2012) verschiedene Finanznachrichtenartikel analysiert. Die Autoren konnten einen Zusammenhang von Aktienkursen und

den Stimmungen innerhalb der Artikel nachweisen, dies kann verwendet werden um Marktbewegungen vorherzusagen (Alessia et al., 2015; Schumaker et al., 2012).

2.4. Evaluationsmetriken

Um die verschiedenen Modelle und Methoden zu vergleichen, müssen bestimmte Metriken eingeführt werden. Im folgenden Abschnitt werden einige dieser Metriken erläutert und definiert.

Confusion Matrix

Bei der Sentiment Analyse gibt es verschiedene Arten von *labels* (Positiv, Negativ und Neutral) die einer Texteinheit zugeordnet werden können. Sind zwei *labels* für die Textklassifikation vorhergesehen, spricht man von einer binären oder dichotomen Polarität. Sind drei *labels* enthalten, wird eine trichotome Polarität angegeben. Bei der Einordnung in zwei verschiedene Klassen müssen zunächst die tatsächlichen Klassen bestimmt werden. Diese Daten werden meist von Menschen annotiert und werden auch als *gold labls* bezeichnet. Um nun die Metriken *Accuracy, Precision* und *Recall* zu ermitteln, kann man eine *confusion matrix* verwenden. Eine *confusion matrix* ist eine visuelle Darstellung in tabellarischer Form, die die Ergebnisse der Sentiment Analyse veranschaulicht. Die Dimensionen sind beim dichotomen Fall wie folgt.

Zum einen gibt es die vom Modell bestimmen *labels* und die *gold standard labels*. Dar-

	gold positive	gold negative	
modell positive	true positive (tp)	false negative (fp)	precision = $\frac{tp}{tp+fp}$
modell negative	false negative (fn)	true negative (tn)	
	recall = $\frac{tp}{tp+fn}$		accuracy= $\frac{tp+tn}{tp+fp+tn+fn}$

Tabelle 1.: Binäre Confusion Matrix

unter kann noch zwischen dem positiven und negativen Fall unterschieden werden (Jurafsky & Martin, 2022, S. 67; Fehle et al., 2021).

Accuracy, Precision und Recall

Aus dieser Tabelle lassen sich die drei Metriken *Accuracy, Recall* und *Precision* ab-

leiten. Bei der *Accuracy* handelt es sich um die Genauigkeit des Modells. Mithilfe der folgenden Formel kann die *Accuracy* in dichotomen, sowie allen anderen Fällen berechnet werden.

$$Accuracy\ (A) = \frac{\sum korrekt\ klassifizierte\ Einheiten}{\sum alle\ Einheiten}$$

Allerdings ist diese Metrik nicht sehr aussagekräftig, was sich vor allem bei stark unausgeglichenen Datensätzen zeigt. Dieser Umstand ist darauf zurückzuführen, dass hierbei nur die richtig klassifizierten Texteinheiten in Bezug auf die Gesamtmenge an Daten betrachtet werden (Jurafsky & Martin, 2022, S.67). Wenn ein Datensatz aus 1000 Texteinheiten besteht und 990 Texteinheiten davon ein positives und 10 Stück ein negatives Sentiment besitzen, gibt die *Accuracy* bei einer komplett positiven Klassifikation einen Wert von 99,9 % aus. In diesem Beispiel ist die negative Klasse in keinen Fall richtig klassifiziert worden. Um dieses Problem zu lösen, gibt es andere Metriken, die als Indikator für die Güte eines Modells verwendet werden können. Die *Precision* gibt an, wie viel Prozent einer Klasse (Positiv, Negativ und Neutral) tatsächlich als diese Klasse bestimmt wurde. Die *Precision* errechnet sich wie folgt:

$$Precision\ (P) = \frac{\sum korrekt\ als\ positivklassifizierte\ Einheiten}{\sum positiv\ klassifizierten\ Einheiten}$$

Der *Recall* misst den Anteil der tatsächlich vorhandenen Texteinheiten in dem Trainingsdatensatz, die vom System korrekt identifiziert wurden. Die Formel zur Berechnung des *Recalls* kann ebenfalls für dichotome und trichotome Polarität verwendet werden (Jurafsky & Martin, 2022, S.68; Fehle et al., 2021).

$$Recall\ (R) = \frac{\sum korrekt\ als\ positiv\ klassifizierte\ Einheiten}{\sum tatschlich\ positive\ Einheiten}$$

F1 Measure

Um beide Aspekte der Metriken *Recall* und *Precision* zu vereinen, wird die sogenannte *F-Measure* eingeführt. Diese Kennzahl kann wie folgt definiert werden.

$$F_\beta = \frac{(\beta^2\ +\ 1)\ x\ Precision\ x\ Recall}{\beta^2\ Precision\ +\ Recall}$$

Mithilfe des Parameters β kann die Gewichtung von *Precision* und *Recall* gesteuert werden. Wird $\beta < 1$ gewählt, wird die *Precision* stärker gewichtet. Wenn $\beta < 1$ ist, wird der *Recall* bevorzugt. Um beide Metriken gleich zu gewichten, wird der β Wert auf 1 gesetzt. Zur Vereinfachung wird nachfolgend der β-Wert als 1 betrachtet. Somit ist das F1-Maß das gewichtete harmonische Mittel aus *Precision* und *Recall* (Jurafsky & Martin, 2022, S.68f.).

Bei den trichotomen Klassifikationsaufgaben, also den Korpora die mit drei verschiedenen Klassen annotiert wurden, kann das F1-Maß mittels Makro- beziehungsweise Mikrodurchschnitt gebildet werden. Bei dem Makrodurchschnitt werden die einzelnen Klassen evaluiert und anschließend wird der Durchschnitt der Klassen errechnet. Der Makrodurchschnitt wird als gleichgewichteter Durchschnitt aller Klassen berechnet. Bei dem Mikrodurchschnitt werden die einzelnen Klassen nicht mehr betrachtet, sondern die Summe deren Werte wird berücksichtigt. Dadurch werden *Recall* und *Precision* so wie in Tabelle 1 berechnet, mit dem Unterschied, dass die *Confusion Matrix* aus drei mal drei Feldern besteht. Der Mikrodurchschnitt wird stärker von Klassen beeinflusst, die einen höheren Anteil besitzen. Der Makrodurchschnitt bildet auch kleine Klassen besser in seiner Gewichtung ab und ist somit eher relevant für Fälle, in denen jede Klasse von Bedeutung ist (Jurafsky & Martin, 2022, S.69).

Stratifizierte Kreuzvalidierung

Ein *Machine Learning*-Modell benötigt eine größtmögliche Menge an Daten, um Vorhersagen zu treffen. Dazu ist es erforderlich, die vorhandenen Daten in einen Trainingsdatensatz und einen Test- beziehungsweise Validierungsdatensatz aufzuteilen. Um die besten Ergebnisse zu erzielen, sollte der Trainingsdatensatz so groß wie möglich sein. Ebenfalls sollte der Testdatensatz nicht zu klein gewählt sein, da dieser so eventuell nicht mehr für den ganzen Korpus repräsentativ ist. Um möglichst viele Daten zum Trainieren des Modells zu erhalten, aber dennoch den gesamten Korpus zu testen, wird die sogenannte Kreuzvalidierung verwendet. Hierbei wird der Korpus in gleichmäßige k Teildatensätze aufgeteilt. Die Zahl k stellt hier die Anzahl der Teildatensätze dar, wobei einer dieser Teildatensätze für das Testen des

Modells verwendet wird, während die anderen *k-1* Teile für das Training verwendet werden. Dieser Vorgang wird *k*-fach wiederholt, bis jeder Teildatensatz mindestens einmal als Testmenge verwendet wurde. Die Ergebnisse aus den einzelnen Durchgängen werden am Ende addiert und durch die Anzahl an Durchgängen geteilt, um so die durchschnittlichen Ergebnisse des Modells zu bekommen (Jurafsky & Martin, 2022, S.69f.). Um eine andere Verteilung der Klassen in den Teildatensätzen zu vermeiden, sollte die stratifizierte Kreuzvalidierung verwendet werden, da diese Klassenproportionen auch in Teildatensätzen gleich behält (Berrar, 2019). Wird zum Beispiel ein Datensatz, der aus insgesamt 400 Texteinheiten mit 300 positiven und 100 negativen Fällen besteht, in vier gleich große Teilmengen ($k = 4$) partitioniert, so ergibt sich, dass jede Teilmenge aus 75 positiven und 25 negativen Fällen bestehen bleibt.

3. Ressourcen für die Sentiment Analyse

In den folgenden Kapiteln werden die in dieser Arbeit verwendeten Korpora vorgestellt. Insgesamt gibt es 20 verschiedene Korpora, die sich in vielerlei Hinsicht unterscheiden. So reicht die Größe der Datensätze von einigen hundert bis zu mehreren tausend Texteinheiten. Die Anzahl der Texteinheiten bezieht sich auf die Anzahl nach der Datenbereinigung, d.h. in einigen Fällen sind weniger Texteinheiten enthalten als im ursprünglichen Korpus. Auch die Qualität der annotierten Texteinheiten ist divers, da eine unterschiedliche Anzahl an Personen mit unterschiedlichen Vorkenntnissen diese Arbeit übernommen hat. Ebenso ist die Art der Polarität nicht in jedem Korpus identisch, so gibt es feine und auch gröbere Granularitäten. Die Datensätze wurden in fünf Oberkategorien eingeordnet: literarische und historische Texte (Abkürzung: LT), Korpora mit Texteinheiten aus verschiedenen Bereichen (MI), Nachrichtenartikel (NA), Produktbewertungen (RE) und Texte aus den sozialen Medien (SM). Die verwendeten Korpora wurden bereits in der Arbeit von Fehle et al. (2021) normalisiert und werden in dieser Arbeit weiterverarbeitet.

3.1. Literarische Texte

German Novel Dataset (GND) / LT01-Zehe

Dieser Datensatz besteht aus 270 Texteinheiten aus deutschen Märchen, die mittels *Crowdsourcing* im Hinblick auf Emotionen annotiert wurden. Zusätzlich hatten diese Sätze Annotationen von mindestens fünf verschiedenen Personen, wobei Sätze mit geringer Übereinstimmung aus dem Datensatz entfernt wurden. Der finale Korpus setzt sich aus 270 Texteinheiten mit 57 positiv, 89 negativ und 124 neutral annotierten Sätzen zusammen (Zehe et al., 2017).

Sentiment Annotation for Lessings´s Plays [1] / LT02-Schmidt

Dieser Datensatz besteht aus 704 Texteinheiten und wurde von neun Studierenden

aus dem Fachbereich der deutschen Literatur annotiert. Diese Texte stammten aus sechs verschiedenen Dramen von G.E. Lessing. Jeder Annotator beziehungsweise jede Annotatorin musste 200 beziehungsweise 183 Einheiten analysieren und das Sentiment angeben. Jede Einheit wurde mit zwei Annotationen versehen. Es gab sechs verschiedene Polaritäten: *negative, positive, neutral, mixed, uncertain* und *other*. Ebenso wurde eine binäre Polarität von den Personen festgelegt. Für diese Arbeit wurden alle Einträge, die nicht mit positiv, negativ oder neutral gekennzeichnet waren, entfernt. Der Datensatz besteht somit aus 202 positiven, 370 negativen und 132 neutralen Texteinheiten (Schmidt et al., 2019).

Sentiment Annotation for Lessings´s Plays [2] / LT03-Schmidt

Dieser Datensatz besteht aus 200 Texteinheiten aus verschiedenen Dramen von G.E. Lessing. Die Annotation wurde von fünf Novizen und einem Experten durchgeführt. Die Polarität wurde auf sechs verschiedene Klassen aufgeteilt, *very negativ, negative, neutral, mixed, positive* und *very positive*. Ebenso wurde eine binäre Polarität mit *negative* und *positive* festgehalten. Eine zusätzliche Skala hielt eine von acht Emotionen fest. Für diese Arbeit wurde die dichotome Polarität gewählt. Der Datensatz besteht aus 61 positiven und 139 negativen Texteinheiten (Schmidt et al., 2018).

3.2. Gemischte Korpora aus unterschiedlichen Domänen

Multi-layered Corpus for Sentiment Analysis (MLSA) / MI01-Clematide

MLSA besteht aus 270 per Hand annotierten Texteinheiten. Diese wurden aus dem DeWaC Korpus extrahiert, der aus verschiedenen deutschsprachigen Dokumenten aus dem Internet zusammengesetzt wurde. Das Korpus wurde auf drei unterschiedlichen Ebenen annotiert. Die erste Ebene annotiert die einzelnen Texteinheiten auf der Satzebene, die zweite Stufe beschäftigt sich mit einzelnen Wörtern und Phrasen, die letzte Ebene wurde auf der Ausdrucksebene annotiert. Ebene eins und zwei wurden von drei Personen bearbeitet, die dritte Ebene wurde von zwei Personen analysiert. Für diese Arbeit wurde die erste Ebene bestehend aus 69 positiven, 110 negativen und 91 neutralen Texteinheiten, verwendet (Clematide et al., 2012).

GermEval 2017 / MI02-Wojatzki

Der GermEval Datensatz besteht aus 26.680 Texteinheiten und wurde speziell für vier verschiedene Aufgaben erstellt. Für die Korpusbildung wurden Texte aus den sozialen Medien, Nachrichtenartikel und Mikroblocks erfasst, die von sechs geschulten Studierenden und einer kuratierenden Person annotiert wurden. Jede Texteinheit wurde durch zwei Studierende annotiert, bei Unstimmigkeiten entschied die kuratorische Leitung, welches Sentiment vergeben wird. Dies ergibt einen Korpus mit 1.537 positiven, 6.887 negativen und 18.256 neutralen Texteinheiten (Wojatzki et al., 2017).

Rauh / MI03-Rauh

Das Korpus von Rauh (2018) besteht aus 1.425 Texteinheiten, welche zufällig aus allen Reden der Ministerpräsidenten im Deutschen Bundestag zwischen den Jahren 1991 und 2013 ausgewählt wurden. Die Annotation wurde von drei Studierenden der Politikwissenschaft durchgeführt. Dabei wurde jede Texteinheit mit einer der drei verschiedenen Polaritäten positiv, negativ oder neutral gekennzeichnet. Das endgültige Korpus besteht aus 333 positiven, 475 negativen und 617 neutralen Einheiten (Rauh, 2018).

3.3. Nachrichtenartikel

GerSEN / NA01-Bütow

Das Korpus besteht aus 2.334 Texteinheiten und setzt sich aus verschiedenen Artikeln deutscher Nachrichtenzeitungen mit Bezug auf Universitäten zusammen. Die Annotation wurde von drei Experten übernommen, die jeweils Teile des gesamten Korpus annotiert haben. Das gesamte Korpus wurde abschließend durch einen Experten überprüft, um Konsistenz und Qualität zu erhöhen. Der Datensatz besteht aus 372 positiven, 485 negativen und 1.477 neutralen Texteinheiten (Bütow et al., 2016).

GerOM / NA02-Ploch

Das GerOM-Korpus setzt sich aus insgesamt 851 Texteinheiten mit 71 positiven, 38 negativen und 742 neutralen Texteinheiten aus deutschen Nachrichtenartikeln zu-

sammen, die im Zeitraum vom 23.02.2012 bis 21.05.2012 gesammelt und anschließend auf Polarität annotiert wurden. Die Auswahl der Texteinheiten erfolgte anhand von drei Voraussetzungen: 250 Texteinheiten sollten ein direktes Zitat beinhalten, 250 Texteinheiten sollten bestimmte Verben enthalten und 500 Texteinheiten hatten keine Restriktionen (Ploch, 2015).

One Million Posts Corpus / NA03-Schabus

Dieser Datensatz besteht aus 3.401 annotierten deutschsprachigen Nachrichtenartikeln aus einer österreichischen Zeitschrift. Weiterhin wurden eine Million Texteinheiten ohne Polarität hinzugefügt. In der Kategorie *Sentiment* konnten die Werte positiv, negativ und neutral zugewiesen werden. Das Annotieren der Texteinheiten wurde durch vier professionelle Moderatoren übernommen. Das verwendete Korpus in dieser Arbeit besteht aus 43 positiven, 1.596 negativen und 1.762 neutralen Texteinheiten. Somit ist das gesamte Korpus 3.401 Einheiten groß (Schabus et al., 2017).

3.4. Produktbewertungen

University Sentiment Analysis Corpus for German and English (USAGE) / RE01-Klinger

Das 590 Texteinheiten große Korpus besteht aus Nutzerbewertungen der Webseite Amazon. Diese Bewertungen können in sieben Kategorien eingeordnet werden. Die Annotation der Bewertungen erfolgte durch zwei Personen, die die einzelnen Bewertungen auf Aspekt- und subjektiver Phrasenebene in die Polaritäten positiv, negativ und neutral einordneten. Um eine Einordnung auf Dokumentenebene zu erhalten, wurden die Sternebewertungen umgewandelt. Fünf und vier Sterne stehen für ein positives, drei Sterne für ein neutrales und zwei beziehungsweise ein Stern für ein negatives *Sentiment* stehen. Somit ergibt sich ein Datensatz bestehend aus 506 positiven, 50 negativen und 34 neutralen Texteinheiten. Die Summe der Texteinheiten ergibt nach Bereinigung 590 Stück (Klinger & Cimiano, 2014).

Sentiment Corpus of App Reviews (SCARE) / RE02-Sänger

Das ursprüngliche Korpus von Sänger et al. (2016) besteht aus 1.760 annotierten

Texteinheiten. Diese wurden über den *Google Play Store* gesammelt. Die Texteinheiten wurden aus elf verschiedenen Kategorien mit insgesamt 148 unterschiedlichen *Apps* ausgewählt. Der Zeitraum für die Akquise der Bewertungen war der Dezember 2014 bis zum Juni 2015. Die Annotation wurde auf Aspekt- und Subjektebene durchgeführt. Die vier Personen führten die Einordnung in zwei Durchgängen aus, wobei in der ersten Runde die Annotation erfolgte und in der zweiten Runde diese noch mal überarbeitet wurde. Um die Annotationen auf Dokumentenebene zu akquirieren, wurden die Sternebewertungen im *Play Store* verwendet. Dieser Schritt wurde in der Arbeit von Fehle et al. (2021) durchgeführt. Das Vorgehen war wie bei Korpus RE01 /USAGE. Bewertungen mit fünf oder vier Sternen wurden ein positives Sentiment, die Bewertungen mit drei Sternen ein neutrales Sentiment und ein bis zwei Sterne wurden ein negatives Sentiment zugeordnet. Somit ergibt sich eine Summe von 416.063 positiven, 185.204 negativen und 60.598 neutralen Texteinheiten. Da diese Anzahl von 661.865 Texteinheiten einen erheblichen Rechenaufwand bedeutete, wurde daraus ein zufälliger ausgeglichener Datensatz ausgewählt. Dieser besteht aus 70.000 Texteinheiten und wurde bei dichotomer Polarität in je 35.000 positiven und negativen Texteinheiten aufgeteilt. Bei trichotomer Polarität wurden 23.334 Einheiten jeweils positiv, negativ und neutral klassifiziert. Die Auswahl erfolgte nach dem Zufallsprinzip. (Sänger et al., 2016).

SentiLitKrit/ RE03-Du

Das Korpus von Du & Mellmann (2019) besteht aus 1.685 Texteinheiten. Diese wurden aus den literaturwissenschaftlichen Anthologien deutschsprachiger Literaturkritik im Zeitraum 1870-1889 entnommen. Aufgrund des Alters der Texte wurden diese digitalisiert und mittels *optical character recognition* auf Fehler untersucht. Weiterhin wurden diese Texte menschlich auf Fehler geprüft und anschließend überarbeitet. Die Annotation wurde mit den Polaritäten positiv, negativ und neutral durchgeführt. Eine Erweiterung auf insgesamt acht Kategorien sollte die Erkennung verbessern. Dadurch wurden einige neutrale Texteinheiten weiter unterteilt in positiv-negativ, neutral-positiv, neutral-negativ, positiv-Artefakte und negativ-Artefakte. Somit beläuft sich das Korpus, das verwendet wurde, auf 718 positiven,

290 negativen und 677 neutralen Texteinheiten (Du & Mellmann, 2019).

Filmstarts-Korpus/ RE04-Guhr

Dieser Korpus umfasst 70.433 Texteinheiten, die aus der deutschsprachigen Webseite *filmstarts.de*, im Jahr 2018 entnommen wurden. Da in den Rezensionen auch Sterne vergeben wurden, konnten die einzelnen Texte automatisiert einer Polarität zugeordnet werden. Texte mit einer Bewertung von vier oder mehr Sternen wurden als positiv, Bewertungen mit drei Sternen als neutral und Bewertungen mit weniger als drei Sternen als negativ eingestuft. Somit ergibt sich ein Korpus mit 39.615 positiven, 15.434 negativen und 15.384 neutralen Sentiment Angaben (Guhr et al., 2020).

Prettenhofer & Stein / RE05-Prettenhofer

Das Korpus von Prettenhofer & Stein (2010) besteht aus 296.032 Texteinheiten. Die Texteinheiten sind Nutzerrezensionen aus der deutschen Amazon-Webseite und können in folgende Kategorien eingeordnet werden: Bücher, DVDs und Musik. Der ursprüngliche Datensatz beinhaltet 159.810 positive und 136.222 negative Reviews. Die Einordnung dieser Klassen erfolgte nach der zu Korpus RE05-Prettenhofer erläuterten Methode. Fünf und vier Sterne wurden als positiv, eins und zwei als negativ gewertet. Drei Sterne Bewertungen wurden aus dem Korpus entfernt, somit enthält der Korpus keine neutralen Texteinheiten. Aufgrund der Größe der einzelnen Texteinheiten und der Größe des gesamten Korpus wurde ein ausbalancierter Korpus erstellt. Dieser besteht aus 35.000 positiven und 35.000 negativen Texteinheiten (Prettenhofer & Stein, 2010).

3.5. Soziale Medien

PotTS / SM01-Cieliebak

PotTS: The Potsdam Twitter Sentiment Corpus hat 7.428 Texteinheiten und besteht aus deutschen *Tweets* von der Webseite Twitter. Diese Texteinheiten wurden im Zeitraum von März bis September 2013 gesammelt. Die Dokumente lassen sich in eines der folgenden vier Bereiche einordnen: die Bundestagswahl in Deutschland, die päpstliche Konklave, allgemeine politische Diskurse und Alltagsgespräche. Die

Personen haben auf Aspekt-, Phrasen- sowie Dokumentebene annotiert. Das Vorgehen hierbei war, dass beide Personen jeweils die Hälfte des Datensatzes bearbeiteten und danach nochmals die Annotationen überprüft und verbessert wurden. Somit ergibt sich ein Korpus mit 1.703 positiven, 1.117 negativen und 4.608 neutralen Texteinheiten (Sidarenka, 2016).

SB10k / SM02-Sidarenka

Das Korpus SB10K besteht aus 7.772 mit Sentiment annotierten Texteinheiten. Die *Tweets* wurden im Zeitraum vom 01.08.2013 bis zum 31.10.2013 auf der Plattform Twitter akquiriert. Diese *Tweets* stammen aus verschiedenen Bereichen und repräsentieren somit eine Vielzahl an unterschiedlichen Themen. Die Annotation wurde von 34 Studierenden aus den Fachrichtungen Linguistik und Computerwissenschaft vorgenommen. Jede Texteinheit wurde von jeweils drei zufällig ausgewählten Personen bearbeitet und mit den Beschriftungen positiv, negativ, neutral und gemischt versehen. Wie bereits in der Arbeit von Fehle et al. (2021) sind nur *Tweets* mit einer Übereinstimmung von zwei Personen verwendet worden. Ebenso besteht dieser Korpus aus Texteinheiten von Sidarenka (2019), da der ursprüngliche Datensatz mittels *Tweet-IDs* nicht mehr vollständig reproduzierbar ist. Somit besteht das verwendete Korpus aus 3.349 positiven, 1.510 negativen und 2.435 neutralen Texteinheiten (Cieliebak et al., 2017).

Narr, Hülfenhaus & Albayrak / SM03-Narr

Der Datensatz von Narr et al. (2012) umfasst 1.658 Texteinheiten und besteht aus *Tweets* aus vier Sprachen. Die Daten wurden aus einem größeren Korpus extrahiert, wobei die Hälfte zufällig ausgewählt wurde und die andere Hälfte bestimmte Schlüsselwörter wie Audi, Sony, Nike etc. enthalten musste. Die Annotation wurde jeweils per Sprache mit drei Personen durchgeführt. Diese wurden über den Dienst *Amazon Mechanical Turk* rekrutiert und beschrifteten Texteinheiten mit den Werten positiv, negativ, neutral und irrelevant. Bei dieser Arbeit werden nur Dokumente verwendet, die mit einer Mehrheit der Personen gleich bewertet wurden. Somit setzt sich das finale Korpus aus 350 positiven, 237 negativen und 1.071 neutralen Texteinheiten zusammen (Narr et al., 2012).

Mozetič, Grčar & Smailović / SM04-Mozetič

Dieser Korpus von Mozetič et al. (2016) umfasst 64.501 Texteinheiten und besteht aus *Tweets* aus unterschiedlichen Sprachen. Insgesamt wurden Texteinheiten aus 13 Sprachen von 83 Personen annotiert. Der deutsche Anteil an Texteinheiten wurde im Zeitraum von Februar bis Juni 2014 gesammelt. Da nur die IDs der *Tweets* angegeben waren, mussten alle Einheiten erneut von Twitter extrahiert werden. Dadurch sind einige Texteinheiten verloren gegangen, da diese von der Seite Twitter bereits gelöscht wurden. Die finale Polarität wurde mittels Mehrheitsentscheidung bestimmt. Somit ergibt sich ein Datensatz aus 16.466 positiven, 36.364 negativen und 64.501 neutralen Texteinheiten (Mozetič et al., 2016).

German Irony Corpus / SM05-Stiegel

Das Korpus besteht aus 163 Texteinheiten aus der Domäne Fußball und enthält vermehrt ironische Texte. Die Annotation wurde manuell vorgenommen und ist in positiv, negativ und neutral eingeteilt. Dadurch ergibt sich ein Korpus mit 49 positiven, 107 negativen und 7 neutralen Texteinheiten (Siegel et al., 2017).

Short Texts about Celebrities / SM06-Momtazi

Das Korpus von Momtazi (2012) umfasst 490 Texteinheiten, diese stammen von sozialen Netzwerken und handelt von deutschen Prominenten. Dabei wurden Seiten wie Facebook, YouTube und Amazon verwendet, aber auch Blogs und Foren wurden durchsucht. Die Annotation wurde manuell durch drei Personen durchgeführt, diese haben die einzelnen Texte mit numerischen Werten annotiert. Für beide Polaritäten wurden Werte zwischen null und eins zugewiesen. Die Bestimmung der Polarität für eine Texteinheit erfolgte mittels der Subtraktion des Wertes für Negativität von dem Wert der Positivität. Ebenso mussten zwei der drei Personen die gleiche Polarität vergeben, damit jene Texteinheit in dieser Arbeit weiterverwendet wird. Somit ergibt sich ein Datensatz von 278 positiven, 190 negativen und 22 neutralen Texteinheiten (Momtazi, 2012).

3.6. Zusammenfassung

Die nachfolgende Tabelle 2 führt Informationen zu den einzelnen Korpora auf. Angegeben sind die Gesamtmengen an Texteinheiten pro Korpus. Ebenfalls wird die Menge an Texteinheiten pro Polarität aufgelistet. Die Anzahl bezieht sich auf die bereits vorverarbeiteten Korpora. Es wurden doppelte sowie fehlerhafte Einträge entfernt. Die Abkürzungen wurden aus dem Werk von Fehle et al. (2021) übernommen und werden weiter in dieser Arbeit verwendet.

Korpus	Kürzel	Pos	Neu	Neg	Total
German Novel Dataset	LT01-Zehe	57	124	89	270
Lessing ´s Plays [1]	LT02-Schmidt	202	132	370	704
Lessing ´s Plays [2]	LT03-Schmidt	61	0	139	200
Multi-layered Corpus for Sentiment Analysis	MI01-Clematide	69	91	110	270
GermEval 2017	MI02-Wojatzki	1.537	18.257	6.887	26.680
Rauh	MI03-Rauh	333	617	475	1.425
GerSEN	NA01-Bütow	372	1.477	485	2.334
GerOM	NA02-Ploch	71	742	38	851
One Million Posts Corpus	NA03-Schabus	43	1.762	1.596	3.401
USAGE	RE01-Klinger	506	34	50	590
SCARE*	RE02-Sänger	416.063	60.598	185.204	661.865
SentiLitKrit	RE03-Du	718	677	290	1.685
Filmstarts-Korpus	RE04-Guhr	39.615	15.384	15.434	70.433
Prettenhofer & Stein*	RE05-Prettenhofer	159.810	0	136.222	296.032
SB10k	SM01-Cieliebak	1.703	4.608	1.117	7.428
PotTS	SM02-Sidarenka	3.349	2.435	1.510	7.772
Narr, Hülfenhaus & Albayrak	SM03-Narr	350	1.071	237	1.658
Motetič, Grčar & Smailović	SM04-Mozetič	16.466	36.364	11.671	64.501
German Irony Corpus	SM05-Siegel	49	7	107	163
Short Texts about Celebrities	SM06-Momtazi	278	22	190	490
Total		641.652	144.401	362.221	1.148.274

Tabelle 2.: Übersicht aller verwendeten deutschen Korpora. Bei Angaben mit einem Stern*, sind die Korpora auf circa 70.000 Einträge gekürzt worden

4. Zielsetzung der Arbeit

In dieser Arbeit sollen verschiedene Methoden zur Sentiment Analyse im Deutschen auf unterschiedlichen Domänen angewandt werden, um so Aussagen über deren Genauigkeit zu treffen.

Die Akquise der verwendeten Literatur erfolgte durch *Google Scholar*, *IEEE*, *Springer Verlag*, *ACM Digital Library* und *ACL Anthology*. Weiterhin wurden die Literaturverzeichnisse in anderen Arbeiten verwendet, um nach weiterer relevanter Literatur zu suchen.

Anhand einer Kreuzevaluation zwischen den verschiedenen *Machine Learning*-Ansätzen und den 20 Korpora soll ein Vergleich erstellt werden. Dieser kann als Referenz verwendet werden, um den Algorithmus passend zur Domäne zu wählen, was eine Steigerung der Genauigkeit zur Folge hat.

Dazu wurden für die Sentiment Analyse übliche Vorverarbeitungsschritte auf die Korpora angewandt, um bessere Ergebnisse bei der Klassifikation zu erzielen. Ferner wurden Stoppwörter entfernt, die Wörter wurden mittels *Stemmer* auf ihren Wortstamm reduziert, Emoticons in Text umgewandelt und Elemente ohne Inhalt beziehungsweise ohne Sentiment wie Nutzernamen oder Hashtags entfernt. Diese vorverarbeitenden Schritte wurden bei traditionellen Methoden wie *Support Vector Machines* bereits eingesetzt und führten zu besseren Ergebnissen (Ahmad et al., 2018). Bei den sogenannten Transformer-Modellen wie *GBERT* soll weiterhin untersucht werden, ob geringfügiges vorverarbeiten der Texte zu einer Leistungssteigerung führen kann. Dazu sollen ebenfalls Stoppwörter entfernt, Emoticons umgewandelt und störende Elemente entfernt werden.

Um eine Aussage über die Leistung der einzelnen Algorithmen zu treffen, werden die Metriken wie *Accurcay* und *F1-Measure* verwendet. Darüber hinaus wird die Verteilung der Polaritäten in den einzelnen Korpora untersucht, so dass Aussa-

gen über unterschiedliche Klassen hinsichtlich der korrekten Polaritätsbestimmung getroffen werden können.

Die technische Umsetzung dieser Aufgabe wird mit der Programmiersprache *Python* realisiert. Dazu werden *Libraries* wie das *NLP-Toolkit NLTK* (Bird et al., 2009), *Scikit-learn* (Pedregosa et al., 2011), keras (Chollet, 2018) und die *Hugging-Face* Implementierung für Transformer-Modelle verwendet.

5. Verarbeitung der Korpora

Im folgenden Kapitel wird auf die Vorverarbeitung der Daten eingegangen. Hierbei wurden die Korpora normalisiert, bereinigt und durch weitere Vorverarbeitungsschritte wie dem *Stemming* aufbereitet, um die Textqualität zu erhöhen. Mithilfe der *Library Pandas* werden die Korpora als *Dataframe* (McKinney et al., 2010) importiert. Anschließend werden die Texteinheiten durch Funktionen von *Pandas* manipuliert und erneut gespeichert. Die verwendete Entwicklungsumgebung ist ein *JupyterLab-System*, das auf einem Server der Universität Regensburg installiert ist. Dieser Server ist mit einem *Intel Xeon W-2255 Prozessor*, *64GB DDR4 Ram* und einer *Quadro RTX 6000 mit 24GB DDR6* Speicher ausgestattet. Im Zeitraum der Benutzung wurden keine weiteren Prozesse ausgeführt, um eine konstante Leistung zu gewährleisten.

5.1. Normalisierung der Datensätze

Die Korpora wurden bereits in der Arbeit von Fehle et al. (2021) normalisiert und ermöglichen so ein einheitliches Vorgehen bei der Weiterverarbeitung. In der nachfolgenden Tabelle 3 wird ein beispielhafter Aufbau einer einzelnen Texteinheit dargestellt.

Der optionale Wert *summary* ist bei verschiedenen Korpora enthalten (NA03-Schabus, RE01-Klinger, RE02-Sänger, RE05-Prettenhofer) und kann als Zusammenfassung beziehungsweise als Überschrift der entsprechenden Texteinheit aufgefasst werden. Diese Textspalte kann ebenfalls für die Sentiment Analyse verwendet werden und gegebenenfalls unterschiedlich gewichtet werden. In dieser Arbeit wurde die *summary* nicht verwendet.

Ein weiterer optionaler Prioritäts-Wert ist der *additional sentiment*-Eintrag. Dieser ist in verschiedenen Korpora (LT02-Schmidt, LT03-Schmidt, MI01-Clematide, MI03-Rauh, NA03-Schabus, SM03-Narr und in SM06-Momtazi) enthalten. Jedoch wur-

id	Eindeutiger numerischer Wert für eine Texteinheit
text	Inhalt der Texteinheit, muss mindestens ein Wort enthalten
sentiment	Annotierte Polarität der Texteinheit, angegeben in positive, negative, neutral
summary (optional)	Zusammenfassung oder Titel einer Texteinheit, nicht immer angegeben
additional sentiment (optional)	Angabe einer zusätzlichen Polarität, wenn mehrere Personen den gleichen Wert vergeben haben.

Tabelle 3.: Darstellung einer Texteinheit in einem Korpus

den hierfür mehrere gleiche Annotationen gewählt. Je nach Datensatz haben hier zwei bis vier Personen das gleiche Sentiment vergeben. Für genauere Informationen können die verwendeten Arbeiten in Kapitel 3 analysiert werden. Dieser Wert kann ebenfalls verwendet werden, um die Sentiment Analyse zu verfeinern beziehungsweise auf Schwierigkeiten in der Klassifikationen einzugehen. Zum Beispiel kann dieser Wert aufzeigen, dass nicht alle Personen den gleichen Wert für das Sentiment vergeben haben. Dieser optionale Wert wurde ebenfalls nicht in dieser Arbeit verwendet.

5.2. Bereinigung der Texteinheiten

Es ist notwendig, die Texteinheiten zu bereinigen, um bessere Ergebnisse mit klassischen Methoden und künstlichen neuronalen Netzen zu erzielen. Im Rahmen dieses Schrittes werden nicht relevante Elemente aus Texteinheiten entfernt. Je nach Gebiet werden die Schritte, die benötigt werden, angepasst, um so das beste Ergebnis zu erzielen (Alam & Yao, 2019; Arras et al., 2017).

Allgemeines Vorgehen

Texte aus dem digitalen Umfeld wie *Tweets* oder Blogeinträge enthalten üblicherweise viel *Noise* und nicht relevante Elemente oder Informationen (Liu, 2012, S.89). Diese Vielzahl von Textbausteinen erhöht die Dimensionalität der Textklassifikation, das bedeutet jeder dieser Bausteine stellt eine eigene Dimension in einem Klassifikationsalgorithmus dar. Dem kann durch eine Vorverarbeitung der Daten ent-

gegengewirkt werden. Dazu gehören verschiedene Schritte, wie das Entfernen von speziellen Zeichen, die nur im digitalen Kontext auftreten (zum Beispiel HTML-Tags), die Entfernung von sogenannten *white space*, die Ausschreibung von Abkürzungen, das Entfernen von Stoppwörtern, das Entfernen von *Hashtags* und das Umwandeln von *Emoticons* (Haddi et al., 2013). Für bestimmte Korpora müssen außerdem noch weitere Schritte unternommen werden. So haben die Autoren Hemalatha et al. (2012) in ihrer Arbeit einige Methoden aufgeführt, die bei der Verwendung eines Korpus von Twitter beachtet werden sollen. Darunter zählen das Entfernen von Verlinkungen beziehungsweise URLs, das Entfernen von Nutzernamen und Twitter spezifischen Zeichen wie „ RT “ oder „ Re“. Auch wird empfohlen, spezielle Zeichen zu entfernen wie „ []() “, da dies ebenfalls zur Rauschentfernung beitragen kann (Hemalatha et al., 2012). Dieses Vorgehen wird auch in anderen Arbeiten angewandt (Parveen & Pandey, 2016).

Eigenes Vorgehen

Um die Genauigkeit der Klassifikationsalgorithmen zu verbessern, wurden in dieser Arbeit verschiedene Schritte durchgeführt. Dabei wurden bestimmte Vorgänge für alle Korpora angewandt und andere nur auf spezifische Korpora. Die nun folgenden Schritte wurden auf allen Texteinheiten durchgeführt. Zuerst wurden alle Texteinträge entfernt, die keinen Inhalt beziehungsweise keine Zeichen enthalten. Anschließend werden bestimmte Zeichen, die mehrmals hintereinander auftreten, entfernt. Danach werden alle Sonderzeichen, die nicht besonders definiert wurden, ebenfalls aus den Texten gelöscht. Die Korpora, die aus der Domäne „ soziale Medien “ stammen, werden nun gesondert weiterverarbeitet. Diese Datensätze werden von *Hastags*, Nutzernamen, Namensnennungen und den twitterspezifischen Zeichen für *Retweets* und *Replies* bereinigt. Ebenso werden die Texte von Verlinkungen und URLs befreit. In der Domäne der „ literarischen Texte “ werden ebenfalls Besonderheiten betrachtet. Hierbei werden die Sprecherattribute aus den Texten entfernt. Diese beinhalten den Namen der Person, die den darauf folgenden Texte spricht. Da dieser immer am Anfang einer Texteinheit steht, werden solche Attribute wie zum Beispiel „ ANGELO: “ oder „ ODOARDO: “ entfernt. Folgende Beispiele sollen ver-

deutlichen, wie die Texteinheiten verarbeitet wurden. Diese Textbeispiele wurden bereits weiteren Schritten unterzogen, wie Stoppwortentfernung und *Stemming*.

Beispiel 1 (aus SM04-Mozetič):

„ RT @onlinevitalshop: Jüngste Kritiken an #Vitaminen sind manipulierte Angstmache http://t.co/BEY1uzeyqi “

Ergebnis für Beispiel 1:

„Jung Kritik manipuliert Angstmach “

Beispiel 2 (aus RE03-Du):

„ ”Ganghofer: Auch im Humor ist Ganghofer dem ältem Erzähler zweifellos überlegen; die „Fuhrmännin“ wird ihm Schmidt nicht nachschreiben. “

Ergebnis für Beispiel 2:

„Auch Humor Ganghof alt Erzahl zweifellos uberlegen; fuhrmannin Schmidt nachschreiben. “

5.3. Entfernung von Stoppwörtern

Bei der Stoppwortentfernung werden Wörter, die in einer Sprache oft vorkommen und keine Semantik besitzen, aus den Texteinheiten entfernt. Dieser Ansatz wird häufig verwendet und hilft bei der Dimensionalitätsreduktion.

Allgemeines Vorgehen

Die zugrunde liegende Idee von Stoppwörtern lieferte Luhn (1958) in seiner Arbeit, in der er automatisch die signifikantesten Wörter und Sätze aus Artikeln extrahierte (Luhn, 1958). Weiterhin kann man Stoppwörter als häufige und gewöhnliche Wörter bezeichnen. Dazu zählen Wörter wie : „ der, die, das, hatte, jeder“. Durch das Entfernen dieser Wörter werden Dimensionen aus dem Text entnommen, wodurch der jeweilige Algorithmus effizienter arbeiten kann (Asghar et al., 2014). Es gibt verschiedene Listen mit solchen Wörtern. Die Implementation wird von *Libraries* wie *NLTK-Toolkit* oder von *Solr* gewährleistet. Um noch bessere Ergebnisse zu erzielen, können diese Listen erweitert werden. Mit Hilfe des *Zipf's Law* können weitere Wörter, die sehr häufig oder sehr selten im Text vorkommen, hinzugefügt werden. Mit der *Mutual Information* Methode werden Wörter, die in keinen oder sehr geringen

Zusammenhang mit einer Polarität stehen, auch als Stoppwörter klassifiziert (Vijayarani et al., 2015). Bei der Lexikon-basierten Sentiment Analyse kann ein solches Vorgehen helfen, maschinell erzeugte Lexika zu bereinigen, da so bei der Analyse diese Wörter nicht abgeglichen werden müssen (Fehle et al., 2021).

Eigenes Vorgehen

In dieser Arbeit wurde die Stoppwortentfernung mittels des *NLTK-Toolkits* durchgeführt (Bird et al., 2009). Die Liste besteht aus 232 häufigen Wörtern der deutschen Sprache. Zu Beginn der Vorverarbeitung wurde jede Texteinheit auf diese Wörter geprüft und zutreffende Wörter wurden entfernt.

5.4. Umwandlung von Emoticons

Bei Texten, die aus der Domäne *social media* stammen, treten vermehrt Emoticons auf. Diese beinhalten Semantik und sollten aus diesen Grund umgewandelt werden, um diese Informationen bei der Klassifikation zu nutzen.

Allgemeines Vorgehen

Im Bereich der sozialen Medien verwendeten Nutzende eine Vielzahl an Emoticons, die Teilweise auch häufig in Sätzen platziert werden, um Gefühle und Emotionen auszudrücken (Wang & Castanon, 2015). Damit eine effiziente Nutzung dieser Informationen gewährleistet werden kann, sollten die Zeichenketten und in das entsprechende Wort umgewandelt werden. Eine Liste von Emoticons und deren Übersetzung findet sich in der Arbeit von Parveen & Pandey (2016), die die Ergebnisse der Sentiment Analyse eines Twitter-Datensatzes verbessert hat. Andere Arbeiten, wie zum Beispiel Hogenboom et al. (2013) deuten an, dass Texte mit Emoticons sogar von dessen Sentiment dominiert werden. Emoticons sollten eher immer mit in die Analyse von Stimmungen hineinfließen (Hogenboom et al., 2013). Besonders bei der Lexikon-basierten Sentiment Analyse können solche Sonderzeichen zu Problemen führen, da Zeichen, die nicht im Lexikon vertreten sind, nicht klassifiziert werden können. Auch können mittels Emoticons sarkastische Sätze besser bestimmt werden, da diese meist mit einer Stimmung versehen sind (Yadav & Pandya, 2017).

Eigenes Vorgehen

Um die Klassifikation der Texteinheiten zu verbessern, sind auch in dieser Arbeit Emoticons umgewandelt worden. Dazu wurden zwei verschiedene Ansätze gewählt, die unterschiedliche Arten von Emoticons übersetzen sollen. Mittels einfacher regulären Ausdrücke sollten Zeichenketten, die als Emoticons interpretiert werden, gefunden werden und in Text übersetzt werden. Eine Liste über Emoticons und dessen Bedeutung wurde aus der Arbeit von Parveen & Pandey (2016) entnommen und verwendet, um Emoticons umzuwandeln (Parveen & Pandey, 2016). Um diesen Vorgang zu verdeutlichen, folgt ein Beispiel:

> Aus folgenden Satzzeichen „ :D , :-D, =D, =-D “ wird das Wort „ lachen “ gebildet.

Der zweite Ansatz funktioniert über die *Library Emoji* (Çevikel, 2018). Mithilfe dieser *Library* können einzelne Zeichen, die als Emoticon zählen, in Text umgewandelt werden. Folgendes Beispiel soll dies verdeutlichen.

> Dieses Emoticon „ 👍“ wird mit „ daumen hoch “ übersetzt.

5.5. Stemming

Es gibt verschiedene Herangehensweisen, um deutsche Texte zu vereinfachen und die Dimensionalität zu verringern. Ein Ansatz hierbei ist das *Stemming*, das nun nachfolgend beschrieben wird.

Allgemeines Vorgehen

Die Hauptaufgabe des *Stemmings* ist es, verschiedene grammatikalische Formen von Nomen, Verben, Adjektiven und Adverbien zu reduzieren. Somit sollen Flexionen eines Wortes auf einen gemeinsamen Wortstamm reduziert werden. Um dies umzusetzen, werden üblicherweise die Suffixe und Präfixe eines Wortes entfernt. Bei der Verwendung eines *Stemmers* können Fehler auftreten, so können Wörter, die unterschiedliche Wortstämme besitzen, auf den gleichen Wortstamm zurückgeführt, oder es werden Wörter, die den gleichen Wortstamm haben, nicht *gestemmt*. Es gibt verschiedene Arten von *Stemmern*, diese können eingeordnet werden in folgende Klassen: Trunkierungsmethoden, statistische Methoden und gemischte Methoden.

Im Folgenden werden ausgewählte Trunkierungsmethoden genauer betrachtet. Um einige dieser *Stemmer* zu nennen: Lovins Stemmer, Porters Stemmer und Husk Stemmer (Jivani et al., 2011).

Der sogenannte *Snowball stemmer* ist eine Weiterentwicklung des *Porter stemmers* und stellt ein *Framework* bereit, um eigene *Stemmer* für andere Sprachen zu entwickeln. Für bestimmte Sprachen, wie dem Deutschen wurden bereits solche *Stemmer* implementiert und sind im *Snowball stemmer* verfügbar. Durch das Anwenden von bestimmten Regeln kann dieser Algorithmus Wörter verändern und diese auf die Grundform reduzieren. Dadurch kann durch geringfügiges *stemming* bereits die Leistung verbessert werden (Singh & Gupta, 2016).

Ausgangswort	Sprung	springen	springend	sprang
Stemming	sprung	spring	springend	sprang

Tabelle 4.: Wortstammreduktion des Wortes „ springen “ mittels des Snowball Stemmer

In Tabelle 4 wird mit einem Beispiel verdeutlicht, wie ein *Stemmer* funktioniert. Dabei werden Wörter wie „ springen “ auf „ spring “ reduziert, während andere Wörter wie „ sprang “ unverändert bleiben. Neben dem *Stemming*, gibt es auch noch das Lemmatisieren. Dabei wird ein Wort auf dessen Grundform reduziert. Besonders in der deutschen Sprache, in der viele verschiedene Wortformen vorkommen, kann dies vorteilhaft sein (Fehle et al., 2021).

Eigenes Vorgehen

Um die Klassifikation der verschiedenen Polaritäten zu verbessern, wurde in dieser Arbeit ein *Stemmer* verwendet. Da der *Porter Stemmer* bereits in vielen anderen Arbeiten verwendet wurde und dabei gute Ergebnisse erzielte, wird hier die deutsche Version verwendet (Ghosh & Sanyal, 2017; Krouska et al., 2016). Der *Snowball Stemmer* ist bereits in der *Library* des *NLTK-Toolkits* integriert und wird deshalb auch so eingebunden.

5.6. Einheitliche Groß- und Kleinschreibung

In der englischen Sprache werden viele Wörter bereits kleingeschrieben. Zwar existieren Ausnahmen wie Satzanfänge, Namen, Orte und Geschehnisse, jedoch überwiegt die Anzahl an kleingeschriebenen Wörtern (Bindra, 2016). Daher macht es im Englischen Sinn, Text bei der Vorverarbeitung zu bereinigen, indem man alle Wörter in die Kleinschreibweise umwandelt (Kim, 2018; Basarslan et al., 2020). In anderen Sprachen wie dem Deutschen werden auch andere Wörter wie Nomen in der Großschreibweise verwendet. Dies kann bei fast gleichen Wörtern zu unterschiedlichen Polaritäten führen. Als Beispiel kann hier das Wort „ Würde“ aufgeführt werden. Als Nomen besitzt dieses Wort meist eine positive Polarität. Hingegen wird das Wort „ würde“ als Verb mit keiner Polarität in Verbindung gesetzt. Dennoch kann eine ausbleibende Umwandlung in die Kleinschreibweise auch zu Problemen führen. Vor allem in subjektiv formulierten und wenig strukturierten Texten, wie zum Beispiel Diskussionsbeiträgen und Texten aus sozialen Medien, treten vermehrt Rechtschreibfehler auf. Im Fall von fehlerhafter Groß- und Kleinschreibung können unter anderem bei der Lexikon-basierten Sentiment Analyse Wörter aus dem Text nicht mit den Wörtern aus dem Lexikon übereinstimmen. Dies kann dazu führen, dass die Genauigkeit der Klassifikation abnimmt (Fehle et al., 2021). Da in dieser Arbeit die beiden Transformer-Modelle *GBERT* und *GELECTRA* in der *cased* Version, also mit Berücksichtigung der Groß- und Kleinschreibung, verwendet werden, werden die Korpora diesbezüglich nicht verändert.

6. Definition und eigene Implementierung der Machine Learning-Ansätze

Im folgenden Kapitel werden die sieben *Machine Learning*-Ansätze definiert, Besonderheiten erklärt und die eigene Implementierung aufgezeigt.

6.1. Praktisches Vorgehen bei der Implementierung

Die Verarbeitung der Korpora war bei allen sieben Ansätzen im dichotomen und trichotomen Fall recht ähnlich. Die Besonderheiten der jeweiligen Ansätze werden in den folgenden Kapiteln noch ausführlicher erklärt, das praktische Vorgehen wird folgend kurz beschrieben. Zuerst werden allen benötigten Bibliotheken in das *Jupyter Notbook* importiert. Dazu zählen folgende Programmbibliotheken: *Pandas* (McKinney et al., 2010), *NumPy* (Oliphant et al., 2006), *re*, *logging*, *time*, *sklearn* (Pedregosa et al., 2011). Anschließend werden alle Korpora mit der gleichen Polarität mittels *Pandas* importiert, da die Vorgehensweise für dichotome und trichotome Korpora etwas unterschiedlich ist. Die Texteinheiten aus der *Tab Seperated Values* (*tsv*) Datei werden nun durch Funktion der Bibliothek *sklearn* in einen Trainings- und einen Testdatensatz aufgeteilt. Dazu wird die stratifizierte Kreuzevaluierung verwendet. Nach der Aufteilung der Texteinheiten erfolgt bei einigen Ansätzen eine Transformation der Texte durch Tokenisierung oder Vektorisierung. Anschließend wird das Modell initialisiert und die erforderlichen Parameter werden festgelegt. Dieser Schritt wird in den folgenden Kapiteln der einzelnen Ansätze ausführlicher beschrieben. Nach der Erstellung des Modells werden die Texteinheiten des Trainingsdatensatzes zum Training des Modells verwendet, während die anderen Texte des Datensatzes zur Validierung des Modells herangezogen werden. Die gewonnenen Daten werden nun ausgewertet, indem die Vorhersagen des Modells mit den tat-

sächlichen Werten der Texteinheiten verglichen werden. Funktionen aus der Bibliothek *sklearn* berechnen verschiedene Metriken, wie *Recall, Precision, Accuracy* und den F1-Wert. Weiterhin werden die *Confussion Matrix* und der *Classification Report* ausgelesen und deren Inhalte werden transformiert. Dieser Schritt ist notwendig, um diese Informationen schlussendlich mit den anderen Werten in einer Tsv-Datei zu speichern. Die Schritte des Trainings des Modells und der Speicherung der Dateien werden viermal durchgeführt.

6.2. Support Vector Machines

Bei den *Support Vector Machines* handelt es sich um „sparse kernel decision machines“, welche statistische Methoden verwenden, um eine Klassifizierung vorzunehmen (Awad et al., 2015, S.39). Dieses diskriminative Lernverfahren versucht laut Awad et al. (2015) mithilfe einer unterscheidenden Funktion Daten in verschiedene Klassen einzuordnen. Dieses Vorgehen wird auch als *Maximal Margin Classifier* beschrieben. Dabei kann zwischen einer *Hard-* und *Soft-Margin* unterschieden werden, diese geben an wie die Hyperebene aufgebaut ist. Folgende Punkte können aufgeführt werden, wieso dieses Verfahren oft eingesetzt wird. *Support Vector Machines* gehören zu den sparsamen Verfahren, da zwar zu Beginn alle Daten im Speicher vorliegen müssen, aber nach der Identifikation aller Parameter nur noch die sogenannten *Support Vectors* benötigt werden. Diese *Support Vectors* bestimmen die Hyperebene, welche die Datenklassen voneinander abgrenzt. Weiterhin werden die Eingabedaten durch das Kernel-Verfahren in einen höherdimensionalen Raum abgebildet, was zur Folge hat, dass immer eine lineare Trennung der Daten vorgenommen werden kann. Dies trägt zur Reduzierung des Rechenaufwandes in der Optimierungsphase bei. Ein weiterer Punkt ist, dass dieses Verfahren bei der Berechnung der Hyperebene versucht, den maximalen Abstand zwischen den Klassen zu finden, da bei neuen Datenpunkten eine leichte Abweichung auftreten kann (Awad et al., 2015, S.39 ff.) Die Verwendung von *Support Vector Machines* innerhalb der Sentiment Analyse wurde bereits in vielen anderen Arbeiten umgesetzt. So zeigten Ahmad et al. (2018) F1-Werte von 74,70 % bis zu 84,10 %, die Arbeit von Ahmad

et al. (2017) erreichte 57,20 % bis zu 69,90 % und die Arbeit von Zainuddin & Selamat (2014) zeigte F1-Werte von 87,30 % bis 89,20 %.

Eigene Implementierung

Dieses Modell wird mithilfe der Bibliothek *sklearn* implementiert (Pedregosa et al., 2011). Als Modellparameter wurden eine *batch-size* von 1800 MB und der *kernel* mit dem Wert *linear* verwendet. Die Texteinheiten wurden durch den *CountVectorizer* transformiert.

6.3. Naïve Bayes

Der *Naïve Bayes*-Algorithmus gehört zu den linearen Klassifizierungsverfahren und basiert auf dem *Bayes*-Theorem. Dieser probabilistische Ansatz geht davon aus, dass die *Features* innerhalb eines Datensatzes voneinander unabhängig sind, was in der Praxis eher selten der Fall ist. Weiterhin wird mit der *Bayes*-Wahrscheinlichkeit versucht, die A-posteriori-Wahrscheinlichkeit mit den Trainingsdaten zu maximieren (Raschka, 2014). Auch dieser Ansatz wurde bereits in vielen anderen Arbeiten behandelt, so zeigten Zuo (2018) *Accuracy*-Werte von 60,00 % bis 74,95%, andere Arbeiten, wie die von Dey et al. (2016) erreichten *Accuracy*-Werte bis zu 82,43 %.

Eigene Implementierung

Der *Multinomial Naïve Bayes*-Algorithmus wurde mithilfe der Bibliothek *sklearn* implementiert und darüber hinaus wurden keine Parameter hinzugefügt oder angepasst (Pedregosa et al., 2011). Die verwendeten Datensätze wurden mithilfe des *CountVectorizer* transformiert.

6.4. Künstliche neuronale Netze

Künstliche neuronale Netze haben ihren Ursprung in der Architektur menschlicher Neuronen. Heutzutage bestehen diese Netze aus einer Vielzahl kleiner Einheiten, die Vektoren als Eingabe verarbeiten und einen einzelnen Wert als Ausgabe generieren. Aufgrund ihrer Architektur werden diese Netze auch als *Feed-Forward Networks* bezeichnet, da jede Schicht Informationen an die nächste Schicht weitergibt (Jurafsky & Martin, 2022, S.133).

6.4.1. Convolutional Neural Networks

Die sogenannten *Convolutional Neural Networks* (*CNN*) gehören zu den *Feed-Forward*-Netzwerken und bestehen aus Eingabeschicht, Merkmalsextraktionsschicht und Klassifikationsschicht. Die erste Schicht verarbeitet Rohdaten, indem diese zu *embeddings* verarbeitet werden. Bei der Merkmalsextraktion werden *convolution* und *pooling* Schichten verwendet, um die Merkmale in den Texten zu extrahieren. In den *convolution layers* werden Merkmalsdetektoren zur Erzeugung der *feature map* verwendet, die *pooling layers* hingegen sortieren irrelevante Merkmale aus. Die letzte Schicht wird zur Bestimmung der Klassen verwendet und stellt somit die Ausgabe des künstlichen neuronalen Netzwerkes dar. Vorteile von *CNNs* sind, dass diese im Vergleich zu anderen neuronalen Netzen weniger Parameter besitzen und daher weniger Zeit zum Trainieren benötigen. In der Sentiment Analyse können *CNNs* kontextabhängige, begrenzte Merkmale aus den Texteinheiten herausfiltern wie Habimana et al. (2020) gezeigt haben. Auch in der Literatur wurden *convolutional neural networks* bereits für die Sentiment Analyse erfolgreich eingesetzt, so zeigte die Arbeit von Dos Santos & Gatti (2014) *Accuracy*-Werte bis zu 85,70 %. Die Autoren Liao et al. (2017) zeigten mit ihrem Ansatz *Accuracy*-Werte von 75,39 %.

Eigene Implementierung

Das *CNN* wurde mittels der *keras* Bibliothek implementiert und mit folgenden Parametern und Schichten ausgestattet (Chollet, 2018). Das Netzwerk besteht aus einem *Embedding layer*, einem *Conv1D layer*, einem *MaxPooling1D layer* und zwei *Dense layer*. Der *Embedding layer* nimmt als Eingabedimension die Anzahl an verschiedenen *Tokens*, die Ausgabedimension beträgt 100 und die Eingabelänge wird mit der längsten Texteingabe bestimmt. Der *Conv1D layer* und der erste *Dense layer* bekommen als Aktivierungsfunktion den Wert *relu* übergeben. Der zweite *Dense layer* wird mit der Aktivierungsfunktion *sigmoid* bestückt. Das Modell wird beim kompilieren durch eine Verlustfunktion (*binary crossentropy*) und einem *optimizer* (*adam*) ausgestattet, die verwendete Metrik ist *Accuracy*. Die Texteinheiten werden mittels eines *Tokenizer* transformiert und anschließend zu Sequenzen weiterverarbeitet.

6.4.2. Recurrent Neural Networks

Wie auch die *CNNs* gehören die *RNNs* zu den *Feed-Forward neural networks*. Diese Art von künstlichen neuronalen Netzen durchlaufen Zyklen die notwendig sind, um Aktivierungsfunktionen auf eingehende Daten anzuwenden. Dies ist auch der Grund weshalb man einem solchen Netz ein Erinnerungsvermögen nachsagt. Die Vorteile dieser Architektur sind, dass Eingangsdaten in beliebiger Länge und Abhängigkeit modelliert werden können. Außerdem ist es möglich, während der Datenverarbeitung an jeder Stelle auf Kontextinformationen zuzugreifen. Der größte Nachteil eines solchen Netzes ist, dass Abhängigkeiten eine gewisse Distanz zueinander benötigen, obwohl ein *RNN* für weitreichende Abhängigkeiten gut geeignet ist. Dieses Problem wird auch als *explodierender Gradient* bezeichnet und kann durch die Verwendung eines *Long Short Time Memory*-Ansatzes gelöst werden. Dieser Ansatz wird um einen *gating mechanism* erweitert, der es dem Netzwerk ermöglicht, Informationen über einen längeren Zeitraum zu speichern oder Informationen aus früheren Zyklen zu vergessen (L. Zhang et al., 2018). In anderen Arbeiten wurden solche *recurrent neural networks* und deren Abwandlung *LSTM* bereits evaluiert, so zeigte die Arbeit von Baktha & Tripathy (2017) *Accuracy*-Werte bis zu 82,20 %. In der Arbeit von Monika et al. (2019) wurden *Accuracy*-Werte über 75,00 % erreicht.

Eigene Implementierung

Das hierbei verwendete *RNN* ist ein *Long short Time Memory*-Ansatz, der durch die *keras* Bibliothek implementiert wurde. Dieses neuronale Netz setzt sich aus einem *Embedding layer*, einem *LSTM layer* und einem *Dense layer* zusammen. Der *Embedding layer*, hat eine Eingabedimension von 2000, eine Ausgabedimension von 128 und eine Eingabelänge die der Größe des *Arrays* der Trainingsdaten entspricht. Der *LSTM layer* besitzt eine Ausgabedimension von 196, einen *Dropout* von 0,2. Der *Dense layer* hat eine *softmax* Aktivierungsfunktion. Das Modell wird ebenso wie bei dem *CNN*-Ansatz mit dem *optimizer adam* und der Metrik *Accuracy* bestückt. Außerdem ist die Verlustfunktion *categorical cross entropy*. Die Texteinheiten werden durch einen *Tokenizer* transformiert und anschließend zu Sequenzen weiterverarbeitet.

6.5. Transformer-Modelle

Mit dem Auftreten der Transformer-Architektur wurden bisherige Modelle, die auf komplexe Verbindungen zwischen *recurrent* und *convolutional neural networks* basierten abgelöst. Diese neuen Modelle basieren auf dem sogenannten *attention mechanism* und haben einen *Encoder*, der die *Features* extrahiert und einen *Decoder*, der die Ausgabe umwandelt. Der *attention mechanism* bildet eine Abfrage auf *key-value* Paare ab, wobei diese drei Elemente und die Ausgabe als Vektoren interpretiert werden. Weiterhin wird dieses Ergebnis aus der gewichteten Summe der einzelnen Werte berechnet, wobei die Gewichtung aus der Kompatibilitätsfunktion zwischen Abfrage und zugehörigen Schlüssel berechnet wird. Der *Encoder* wandelt eine Eingabesequenz von Symbolen auf eine Sequenz mit fortlaufender Darstellung um und verwendet dazu sechs Schichten mit jeweils zwei Unterschichten, die aus einem „multi-head self-attention mechanism“ und einem „positionwise fully connected feed-forward network“ bestehen (Vaswani et al., 2017). Der *Decoder* übernimmt die Sequenz mit kontinuierlicher Darstellung und wandelt sie in eine Ausgabesequenz von Symbolen um, wobei eine ähnliche Struktur wie beim *Encoder* verwendet wird, die um einen „multi-head attention mechanism“ erweitert wird, der auf die Ausgabe des *Encoder* fokussiert ist (Vaswani et al., 2017).

Basierend auf der ursprünglichen Architektur der Transformer wurde mit *BERT* (*Bidirectional Encoder Representations from Transformers*) ein neues Modell geschaffen. Dieses Modell ist vortrainiert und kann durch das *fine-tuning* auf verschiedene Aufgaben angepasst werden. Bei dem Vortrainieren des Modells werden zwei verschiedene Aufgaben angewandt. So wird ein maskiertes Sprachmodell als erste Aufgabe verwendet, indem einige *Tokens* als Eingabe maskiert werden und diese danach vorhergesagt werden. Die zweite Aufgabe befasst sich mit der Vorhersage eines Folgesatzes, dies dient dazu die Beziehung zwischen Sätzen zu erlernen, was für ein Sprachmodell wichtig ist (Devlin et al., 2019).

Ebenso wie das *BERT*-Modell basiert *ELECTRA* (*Efficiently Learning an Encoder that Classifies Token Replacements Accurately*) auf der Architektur der Transformer. Bei diesem Ansatz wird jedoch nicht das maskierte Sprachmodell verwendet son-

dern das *replaced token detection*-Verfahren wird zum Vortrainieren verwendet. Dabei werden einige *Tokens* durch andere Wörter ausgetauscht, die durch ein kleines Generator Netzwerk erzeugt werden. Anstatt die originalen Wörter vor dem Austausch vorherzusagen, wird bestimmt, welche *Tokens* ausgetauscht wurden. Durch dieses Vorgehen werden alle *Tokens*, die verwendet wurden, miteinbezogen, anstatt nur der maskierten *Tokens* (Clark et al., 2020).

Eigene Implementierung

Die drei folgenden Transformer-Modelle *GBERT, GELECTRA* und *Bert Base Multilingual Uncased Sentiment* wurden mithilfe der Bibliothek *SimpleTransformers* implementiert (Rajapakse et al., 2019-2022). Der Aufbau für alle drei Modelle ist identisch und wird daher nur einmal beschrieben. Die Modelle werden durch die Bibliothek *SimpleTransformers* initialisiert und mit folgenden Parametern ausgestattet. Die *batch size* für das Trainieren und für das Evaluieren beträgt 32. Außerdem wurde für das Training und die Evaluation das *multiprocessing* ausgeschaltet, da dies zur Stabilität des Systems beiträgt. Die Anzahl der Trainingsepochen beträgt für alle Korpora vier Durchgänge. Weiterhin wurden in den Modellen die Anzahl an *labels* definiert und die Verwendung der Grafikkarte wurde aktiviert. Das jeweilige Modell wird über die Seite *Hugging Face* [1] bezogen und durch Angabe des übergeordneten Modells (*ELECTRA* oder *BERT*) sowie des spezifischen Modellnamens in den Programmcode integriert.

6.5.1. GBERT

Die Weiterentwicklung von *BERT* auf eine andere Sprache, wie dem Deutschen, wird mithilfe von anderen Textdaten für das Vortrainieren vorgenommen. Hierbei wurden vier große Korpora verwendet, die insgesamt aus 160 GB Text bestehen. Diese Texte setzten sich aus Wikipedia-Artikeln, Internetseiten und Texten aus deutschen Gerichtssälen zusammen. Das Modell *GBERT Base* wurde auf allen Daten trainiert und besitzt kein *Whole Word Masking*, die Variante *Large* wird zusätzlich mit der *Whole Word Masking* Methode trainiert (Chan et al., 2020). Dieses Modell

[1] https://huggingface.co/

wurde bereits in der Arbeit von Schmidt et al. (2022) verwendet und erzielt dort *Accuracy*-Werte von 93,30 %. Auch die Arbeit von Idrissi-Yaghir et al. (2023) zeigte gute F1-Werte bis zu 96,10 %.

6.5.2. GELECTRA

Die verwendeten Datensätze zum Trainieren der deutschen *ELECTRA*-Variante entsprechen den Datensätzen, die bereits im vorherigen Abschnitt besprochen wurden. Die *Base*-Variante wurde auf den Datensätzen *OPUS* und *The Wikipedia dump for German* trainiert, die *Large*-Variante wurde auf allen Datensätzen trainiert (Chan et al., 2020). Diese Version von *ELECTRA* wurde ebenfalls in anderen Arbeiten bereits verwendet (Schmidt et al., 2021; Idrissi-Yaghir et al., 2023).

6.5.3. BERT Base Multilingual Uncased Sentiment

Dieses Modell wurde speziell für Produktbewertungen in sechs verschiedenen Sprachen trainiert. Dabei kann es die Texteinheiten auf einer Skala von eins bis fünf Sternen einordnen. Die Anzahl der Bewertungen, die für das Training in deutscher Sprache verwendet wurden, beträgt 137.000.[2]

[2]Modell von der Seite Hugging Face: https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

7. Ergebnisse

In diesem Kapitel werden die Ergebnisse der Kreuzevaluation untersucht. Zu diesem Zweck wurden die verschiedenen Domänen mit allen Ansätzen betrachtet. Die folgenden Metriken werden verglichen: *Accuracy, Precision, Recall* und der gewichtete durchschnittliche F1-Wert. Diese Werte sind aus dem Durchschnitt der vier Durchgänge gebildet worden. Alle F1-Werte sind mittels des gewichteten Durchschnitts kalkuliert worden, daher wird dieser Wert vereinfacht als F1-Wert angegeben. Bei der nachfolgenden Betrachtung wurden bei den Transformer-Modellen *GBERT* und *GELECTRA*, die Variante *Large* gewählt. Diese erzielte bessere Ergebnisse und wird daher in dieser Auswertung verwendet. Die Unterschiede zwischen dem *Base* und *Large* Modell können in Kapitel 7.2 betrachtet werden. Genauere Angaben zu den Laufzeiten der Algorithmen sind in Anhang A zu finden [3]. Die folgenden Zeiten sind als Mittelwerte zu interpretieren. Diese beziehen sich auf einen Ansatz innerhalb einer Domäne. Durchschnittliche *Accuracy*- und F1-Werte von allen Ansätzen über die verschiedenen Domänen sind in Anhang B zu finden. Die verwendeten Abkürzungen in den Tabellen sind wie folgt: SVM - *Support Vector Machines*; NB - *Naïve Bayes*; CNN - *Convolutional Neural Network*; RNN - *Recurrent Neural Network*; GB - *GBERT*; GE - *GELECTRA*; BBMUS - *Bert Base Multilingual Uncased Sentiment*; Di - dichotome Polarität; Tri - trichotome Polarität; ACC - *Accuracy*; F1 - F1-Wert. Der beste Wert eines Korpuses wird fett gedruckt und in grüner Farbe in der Tabelle hinterlegt. Außerdem sind alle Werte in den folgenden Tabellen als Prozent-Angaben zu verstehen.

[3] Angaben zu den Zeiten aller Korpora sind im digitalen Anhang (/4_Tabellen) zu finden

7.1. Vergleich der vorverarbeitenden Schritte für Transformer-Modelle

In diesem Abschnitt werden die Resultate der Vorverarbeitungsschritte der Transformer-Modelle besprochen. Zu diesem Zweck wurde die Variante *Base* der Modelle *GBERT* und *GELECTRA* verwendet, da der Zeitaufwand für das Training der Klassifizierung der Texteinheiten geringer ist. Es wurden alle Korpora zur Evaluierung verwendet und folgende Schritte wurden ausgeführt: Emoticons wurden übersetzt, *Hashtags*, Links und Nutzernamen wurden entfernt, leere Einträge wurden gelöscht, Twitter spezifische und spezielle Zeichen wurden entfernt. Die Abkürzung *PP* steht in Tabelle 5 und 6 für *Preprocessing*.

Domäne	**GBERT**				**GELECTRA**			
	Ø Accuracy		**Ø F1**		**Ø Accuracy**		**Ø F1**	
	Original	**PP**	**Original**	**PP**	**Original**	**PP**	**Original**	**PP**
Literarische Texte	67,50	**68,87**	60,58	**61,53**	65,05	**67,50**	**51,63**	51,37
Gemischte Domäne	**84,59**	83,47	**83,57**	81,61	**79,83**	79,27	**74,89**	74,36
Nachrichten-artikel	**88,64**	88,30	**87,24**	86,44	**85,52**	85,30	**80,51**	80,29
Produkt-bewertungen	**90,79**	90,69	**90,53**	90,39	89,42	**89,55**	87,79	**88,06**
soziale Medien	**83,85**	83,06	**83,38**	82,41	**77,94**	76,33	**74,44**	72,47

Tabelle 5.: Durchschnittliche dichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich mit und ohne vorverarbeitenden Schritte

In Tabelle 5 werden die Werte im dichotomen Fall dargestellt. Alle *Accuracy*- und F1-Werte sind im Durchschnitt pro Domäne angegeben. Durch das *Preprocessing* wurde in beiden Fällen keine bessere Leistung erzielt. Die Werte für *Accuracy* zeigen, dass in fast allen Fällen unverarbeitete Texte zu bevorzugen sind. Auch bei den F1-Werten sind unverarbeitete Texte zu bevorzugen, um höhere Werte zu erreichen. Eine Ausnahme bildet hier die Domäne der literarischen Texte, wobei hier bei beiden Modellen leicht bessere Werte erzielt werden konnten. Außerdem führt eine Vorver-

arbeitung bei Produktbewertungen zu leicht besseren F1- und *Accuracy*-Werten. Im Gesamtdurchschnitt werden durch das Weglassen von vorverarbeitenden Schritten *Accuracy*-Werte erreicht, die um 0,20 %P (*GBERT*) beziehungsweise 0,48 %P (*GELECTRA*) besser sind. Ebenso sind die F1-Werte um 0,58 %P (*GBERT*) beziehungsweise 0,54 %P (*GELECTRA*) besser im unverarbeiteten Fall. Eine zeitliche Änderung bei der Gesamtdauer des Evaluationsvorganges konnte nicht beobachtet werden.

Domäne	**GBERT**				**GELECTRA**			
	Ø Accuracy		**Ø F1**		**Ø Accuracy**		**Ø F1**	
	Original	**PP**	**Original**	**PP**	**Original**	**PP**	**Original**	**PP**
Literarische Texte	**53,36**	52,88	**49,05**	47,09	**52,72**	50,66	**38,97**	35,55
Gemischte Domäne	**72,96**	71,98	**72,68**	71,50	**62,45**	62,06	**56,15**	55,88
Nachrichten-artikel	**79,74**	79,14	**79,07**	78,33	77,66	**78,08**	74,89	**75,75**
Produkt-bewertungen	**79,20**	78,70	**78,43**	77,90	**77,79**	77,32	**75,42**	74,35
soziale Medien	**75,19**	74,49	**74,21**	72,53	**69,24**	68,14	**64,35**	62,82

Tabelle 6.: Durchschnittliche trichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich mit und ohne vorverarbeitenden Schritte

Die Werte in Tabelle 6 stellen *Accuracy*- und F1-Werte bei trichotomer Polarität dar. Durch das Auslassen von *Preprocessing* kann in beiden Modellen ein Zuwachs an *Accuracy* und F1-Wert erzielt werden. Im Durchschnitt beträgt dieser Wert 0,65 %P (*GBERT*) beziehungsweise 0,72 %P (*GELECTRA*)bei der *Accuracy*. Die F1-Werte steigen im Schnitt um 1,22 %P (*GBERT*) beziehungsweise 1,09 %P (*GELECTRA*). Eine Ausnahme bilden hier Nachrichtenartikel, diese können durch das *GELECTRA*-Modell mithilfe von *Preprocessing* besser klassifiziert werden. Eine zeitliche Änderung konnte bei der Gesamtdauer des Evaluationsvorganges ebenso wie bei dem dichotomen Fall nicht beobachtet werden.

7.2. Vergleich der Transformer-Modelle in der Ausführung Base und Large

Um die besten Ergebnisse mit den Transformer-Modellen *GBERT* und *GELECTRA* zu erzielen, wurde ein Vergleich der Ausführungen *Base* und *Large* durchgeführt. Diese beiden Varianten unterscheiden sich in ihren Hyperparametern. Die beiden *Base* Variationen besitzen 12 *layer*, die beiden *Large* Varianten sind mit 24 *layern* ausgestattet. Ebenso verfügen beide über eine unterschiedliche Anzahl von *attention heads*, nämlich 12 beziehungsweise 16 Stück. Die *batch size* unterscheidet sich auch von einer Variante zur anderen. So hat *GBERT Base* eine *batch size* von 128, *GBERT Large* 2048, *GELECTRA Base* 256 und *GELECTRA Large* 1024 (Chan et al., 2020).

Domäne	**GBERT**				**GELECTRA**			
	Ø Accuracy		**Ø F1**		**Ø Accuracy**		**Ø F1**	
	Base	**Large**	**Base**	**Large**	**Base**	**Large**	**Base**	**Large**
Literarische Texte	67,50	**68,52**	**60,58**	60,55	65,05	**69,79**	51,63	**61,56**
Gemischte Domäne	84,59	**91,03**	83,57	**91,03**	79,83	**85,04**	74,89	**82,90**
Nachrichten-artikel	88.64	**93,32**	87,24	**92,45**	**85,52**	83,70	**80,51**	78,86
Produkt-bewertungen	90,79	**92,99**	90,53	**92,77**	89,42	**93,31**	87,79	**93,06**
soziale Medien	83,85	**84,90**	83,38	**83,82**	77,94	**83,14**	74,44	**90,95**

Tabelle 7.: Durchschnittliche dichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich der beiden Ausführungen Base und Large

Die beiden Varianten wurden mit allen Datensätzen evaluiert. In Tabelle 7 sind die Ergebnisse für die dichotome Polarität aufgeführt. Dazu wurden die durchschnittlichen *Accuracy* und F1-Werte für jede Domäne gebildet. Bei dem *GBERT* Modell ist die *Large* Variante in jeder Domäne besser geeignet mit einem durchschnittli-

chen Zuwachs von 3,08 %P. Bei den F1-Werten hat die *Large* Variante ebenso besser abgeschnitten. Hier ist ein Zuwachs von 3,06 %P im Durchschnitt erzielt worden. Das *GELECTRA* Modell konnte bei den *Accuracy* Werten sogar eine Steigerung von 3,45 %P im Durchschnitt erreichen. Auch die F1-Werte verbesserten sich in der *Large* Variante im Schnitt um 5,62 %P.

Da die *Large* Varianten mehr Schichten und mehr *attention heads* aufweisen, steigt auch die für die Verarbeitung eines Datensatzes benötigte Zeit. Alle Zeiten sind im Anhang A zu finden. Die durchschnittliche Zunahme beträgt 34 Minuten, die maximale Zunahme 2 Stunden und 17 Minuten für das *GBERT* Modell. Bei dem *GELECTRA* Modell erhöhte sich die benötigte Zeit im Durchschnitt ebenfalls um 34 Minuten, der maximale Zuwachs beträgt 2 Stunden und 17 Minuten.

Domäne	**GBERT**				**GELECTRA**			
	Ø Accuracy		**Ø F1**		**Ø Accuracy**		**Ø F1**	
	Base	**Large**	**Base**	**Large**	**Base**	**Large**	**Base**	**Large**
Literarische Texte	53,36	**55,80**	49,05	**51,37**	52,72	**55,98**	38,97	**47,83**
Gemischte Domäne	72,96	**75,69**	72,68	**75,42**	62,45	**73,40**	56,15	**71,93**
Nachrichten-artikel	79,74	**80,66**	79,07	**79,86**	77,66	**81,53**	74,89	**80,35**
Produkt-bewertungen	79,20	**80,95**	78,43	**79,59**	77,79	**82,12**	75,42	**81,46**
soziale Medien	75,19	**76,89**	74,21	**75,41**	69,24	**76,24**	63,35	**73,61**

Tabelle 8.: Durchschnittliche trichotome Accuracy und F1-Werte der Transformer-Modelle GBERT und GELECTRA im Vergleich der beiden Ausführungen Base und Large

Die Werte in Tabelle 8 repräsentieren die Auswertungen bei trichotomer Polarität. Das *GBERT* Modell erreicht bei der *Accuracy* im Schnitt einen Zuwachs von 1,91 %P bei der Variante *Large*. Die F1-Werte konnten um 1,64 %P gesteigert werden. Den

größten Gesamtzuwachs zeigte *GELECTRA*. Hierbei wurden mit der *Large* Variante im Durchschnitt 5,88 %P bei der *Accuracy* und 9,08 %P bei den F1-Werten gewonnen. Ebenso wie im dichotomen Fall stieg die benötigte Zeit, um einen Datensatz zu klassifizieren. Im Durchschnitt wurden 42 Minuten mehr benötigt, wobei ein maximaler Anstieg um 2 Stunden und 4 Minuten zu verzeichnen ist.

7.3. Evaluation von Domänen und Ansätzen

7.3.1. Literarische Texte

In diesem Unterkapitel werden die Ergebnisse für literarische und historische Texte zusammengefasst. In Tabelle 9 werden für dichotome und trichotome Polaritäten die *Accuracy*-Werte angegeben. Der beste Ansatz in der dichotomen Klassifikation ist *GELECTRA* mit durchschnittlich 69,79 % *Accuracy*. Der schlechteste Ansatz ist das *RNN* mit durchschnittlich 61,16 % *Accuracy*. Bei der trichotomen Klassifikation

ACC	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	LT01-Zehe	63,68	56,17	60,96	56,10	**67,12**	64,34	60,23
	LT02-Schmidt	55,94	63,64	64,69	57,87	69,93	**75,52**	65,56
	LT03-Schmidt	66,50	66,50	**69,50**	**69,50**	68,50	**69,50**	69,00
Tri	LT01-Zehe	42,59	40,37	45,19	43,71	**46,27**	45,91	41,11
	LT02-Schmidt	44,18	51,85	56,11	54,69	65,34	**66,05**	57,95

Tabelle 9.: Accuracy-Werte für literarische Texte

schneidet *GBERT* mit 53,36 % Genauigkeit am besten ab. Das *SVM* hingegen erreicht im Durchschnitt nur 43,38 % *Accuracy* und ist somit das schlechteste Modell. Weiterhin sind in Tabelle 10 F1-Werte angegeben. Es zeigt sich, dass im dichotomen Fall *Support Vector Machines* gute Ergebnisse liefern Die beiden Transformer-Modelle *GBERT* und *GELECTRA* erzielen ähnlich gute Ergebnisse. Das schlechteste Modell ist herbei das *CNN*. Bei der trichotomen Auswertung fällt auf, dass die beiden Transformer-Modelle *GBERT* und *GELECTRA* die besten Werte liefern. Das *CNN* ist das schlechteste Modell in diesem Fall. Die minimale Zeit für die dichotome Klassifikation beträgt weniger als eine Sekunde, die maximale Zeit 1 Minute

und 47 Sekunden. Die schnellsten Verfahren sind NB und SVM. Die Transformer-basierten Ansätze *GBERT* und *GELECTRA* benötigen am längsten. Die trichotome Klassifikation wird minimal unter einer Sekunde und maximal in 2 Minuten und 43 Sekunden durchgeführt. Die Aufteilung der schnellsten und langsamsten Algorithmen ist ebenso wie im dichotomen Fall.

F1	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	LT01-Zehe	**63,34**	53,98	46,18	45,09	60,11	52,72	53,20
	LT02-Schmidt	55,26	59,09	50,81	54,71	64,18	**74,97**	64,74
	LT03-Schmidt	**63,20**	61,28	57,00	57,00	57,35	57,00	59,19
Tri	LT01-Zehe	40,86	38,56	28,66	35,19	**41,48**	34,63	32,29
	LT02-Schmidt	43,96	43,70	46,46	46,08	**61,27**	61,04	56,25

Tabelle 10.: F1-Werte für literarische Texte

7.3.2. Texte aus gemischten Domänen

In diesem Unterkapitel werden die Ergebnisse der Texte aus unterschiedlichen Domänen besprochen. In Tabelle 11 sind die *Accuracy*-Werte für die dichotome und trichotome Klassifikation dargestellt. Bei der dichotomen Polarität schneidet *GBERT*

ACC	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	MI01-Clematide	68,16	68,69	61,45	62,01	**86,57**	69,81	73,18
	MI02-Wojatzki	86,23	87,01	86,56	85,98	**94,21**	93,98	87,77
	MI03-Rauh	62,87	63,86	59,03	60,64	**92,33**	91,34	78,84
Tri	MI01-Clematide	50,35	49,60	40,74	38,52	**68,10**	60,37	47,76
	MI02-Wojatzki	73,44	74,41	75,13	74,20	84,24	**84,54**	73,61
	MI03-Rauh	48,21	49,76	45,97	45,61	74,74	**75,30**	63,09

Tabelle 11.: Accuracy-Werte für Texte aus gemischten Domänen

mit durchschnittlich 91,03 % *Accuracy* am besten ab. Das *Recurrent Neural Network* kann hingegen nur 69,02 % Genauigkeit erzielen und ist somit der schlechteste Ansatz. Bei der trichotomen Klassifikation kann *GBERT* die besten Ergebnisse liefern und einen Wert von 72,96 % erreichen. Das schlechteste Ergebnis wird mittels ei-

nes *RNNs* produziert, wobei hier der Wert von 52,78 % vorliegt. Die F1-Werte sind in Tabelle 12 hinterlegt. Der beste Ansatz für dichotome Datensätze ist *GBERT*, der schlechteste Ansatz ist das *Convolutional Neural Network*. Dies verhält sich ebenso bei der trichotomen Klassifikation. Die benötigte Zeit für die Klassifikation im dicho-

F1	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	MI01-Clematide	66,03	66,78	46,78	48,00	**86,61**	63,47	72,97
	MI02-Wojatzki	85,96	84,82	85,74	84,93	**94,15**	93,93	87,49
	MI03-Rauh	62,68	62,98	44,09	60,40	**92,35**	91,33	78,76
Tri	MI01-Clematide	48,80	48,03	23,59	28,00	**67,42**	56,06	44,86
	MI02-Wojatzki	72,92	72,51	74,32	72,49	84,21	**84,44**	73,33
	MI03-Rauh	47,08	48,69	44,17	45,07	74,62	**75,28**	63,01

Tabelle 12.: F1-Werte für Texte aus gemischten Domänen

tomen Fall beträgt unter einer Sekunde (NB) bis hin zu 16 Minuten 02 Sekunden (GE). Bei der Klassifikation mit drei Polaritäten beträgt die kürzeste Rechenzeit 1 Sekunde (NB), die längste Zeit ist 47 Minuten 54 Sekunden (GE).

7.3.3. Nachrichten Artikel

In diesem Unterkapitel werden die Resultate der Klassifikation von Nachrichten Artikel beschrieben. In Tabelle 13 sind die *Accuracy*-Werte abgebildet. In der dicho-

ACC	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	NA01-Bütow	70,01	77,60	67,10	71,88	**96,85**	87,97	80,51
	NA02-Ploch	73,38	71,56	65,15	65,15	**85,19**	65,15	72,52
	NA03-Schabus	97,25	97,38	97,38	97,38	97,93	**97,99**	97,13
Tri	NA01-Bütow	63,28	66,62	64,31	62,85	82,35	**82,99**	70,22
	NA02-Ploch	87,90	88,25	87,19	87,54	89,42	**90,13**	89,19
	NA03-Schabus	56,66	61,42	59,75	55,92	70,21	**71,48**	60,10

Tabelle 13.: Accuracy-Werte für Nachrichtenartikel

tomen *Sentiment* Bestimmung schneidet der *GBERT* Ansatz mit 93,32 % *Accuracy* am besten ab. Der Ansatz mit der schlechtesten *Accuracy* ist das *CNN* mit 76,54 %.

Bei der trichotomen Klassifikation hat das *GBERT* Modell einen Durchschnittswert von 79.74 % und ist somit das beste Ergebnis. Mit 68,77 % ist das *RNN* der schlechteste Ansatz. Die F1-Werte sind in Tabelle 14 dargestellt. Dabei ist zu sehen, dass die beste Leistung durch *GBERT* im dichotomen Fall erzielt wird. Das schlechteste Modell ist das *Convolutional Neural Network*. Bei der trichotomen Klassifikation kann das Modell *GELECTRA* empfohlen werden. Hingegen liefert das *Recurrent Neural Network* die schlechtesten Ergebnisse und ist daher nicht empfehlenswert. Die be-

F1	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	NA01-Bütow	69,50	77,45	62,88	71,86	**96,85**	87,58	80,54
	NA02-Ploch	70,62	68,13	51,41	51,41	**83,14**	51,41	68,74
	NA03-Schabus	96,28	96,08	96,08	96,08	97,36	**97,59**	96,59
Tri	NA01-Bütow	60,53	61,07	61,99	60,87	82,38	**82,95**	69,39
	NA02-Ploch	85,38	83,59	81,23	83,09	**87,25**	87,11	87,22
	NA03-Schabus	56,04	60,74	58,81	55,36	69,93	**70,98**	59,80

Tabelle 14.: F1-Werte für Nachrichtenartikeln

nötigte Zeit bei dieser Domäne beträgt unter einer Sekunde (NB) im schnellsten Ansatz. Bei dem langsamsten Ansatz wird eine Zeit von 4 Minuten und 37 Sekunden (GE) benötigt. Bei der trichotomen Polarität werden unter einer Sekunde (NB) bis hin zu 11 Minuten 14 Sekunden (GE) benötigt.

7.3.4. Produktbewertungen

In diesem Unterkapitel werden die Ergebnisse der *Sentiment* Bestimmung für Produktbewertungen beschrieben. Die Werte in Tabelle 15 repräsentieren die *Accuracy*. Bei den dichotomen Polaritäten kann der Ansatz *GELECTRA* mit einer durchschnittlichen *Accuracy* von 93,31 % überzeugen. Der *CNN*-Ansatz schneidet hierbei am schlechtesten ab und besitzt einen *Accuracy*-Wert von 84,38 %. Bei der trichotomen Klassifikation kann *GBERT* das beste Ergebnis liefern, mit 79,20 %. Das *CNN* liefert mit 69,43 % das schlechteste Ergebnis. Die F1-Werte sind in Tabelle 16 zu betrachten. *GELECTRA* liefert hier die besten Ergebnisse, das *CNN* hingegen die

ACC	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	RE01-Klinger	90,83	91,01	91,01	90,83	94,60	**94,78**	94,78
	RE02-Sänger	85,83	85,74	85,01	86,38	92,18	**92,43**	86,59
	RE03-Du	74,31	73,61	71,73	71,92	90,18	**90,77**	76,49
	RE04-Guhr	87,81	87,98	88,68	88,82	94,49	**94,66**	88,69
	RE05-Prettenhofer	84,63	84,05	85,48	85,68	93,51	**93,92**	86,66
Tri	RE01-Klinger	85,43	85,76	85,76	85,93	86,78	**90,51**	88,14
	RE02-Sänger	63,95	65,37	62,34	65,99	**73,42**	72,97	66,34
	RE03-Du	65,22	63,38	57,68	62,37	83,68	**84,87**	66,17
	RE04-Guhr	70,50	71,92	71,93	74,36	79,92	**80,14**	72,94

Tabelle 15.: Accuracy-Werte für Produktbewertungen

schlechtesten Ergebnisse. Bei der trichotomen Klassifikation wird *GELECTRA* empfohlen. Das *Convolutional Neural Network* erzielt erneut den letzten Platz und ist somit der schlechteste Ansatz. Die benötigte Zeit liegt zwischen 7 Sekunden (NB)

F1	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	RE01-Klinger	87,79	86,72	86,72	86,63	93,46	93,66	**94,69**
	RE02-Sänger	85,83	85,72	85,01	86,38	92,18	**92,43**	86,59
	RE03-Du	73,00	69,16	63,26	70,17	90,21	**90,68**	75,46
	RE04-Guhr	87,75	87,30	88,64	88,66	94,47	**94,63**	88,62
	RE05-Prettenhofer	84,63	84,02	85,46	85,68	93,51	**93,92**	86,66
Tri	RE01-Klinger	80,69	79,19	79,19	79,57	81,47	**87,87**	86,18
	RE02-Sänger	63,58	65,38	62,39	65,93	**73,48**	73,05	66,34
	RE03-Du	64,37	60,76	60,47	61,06	83,52	**84,81**	65,90
	RE04-Guhr	70,25	69,47	71,61	73,16	79,89	**80,10**	72,97

Tabelle 16.: F1-Werte für Produktbewertungen

und 3 Stunden, 34 Minuten und 12 Sekunden (GE) für Texte mit zwei Polaritäten. Texte mit drei Polaritäten benötigen 6 Sekunden (NB) bis hin zu 3 Stunden, 3 Minuten und 42 Sekunden (GE).

7.3.5. Social Media

Der folgende Abschnitt befasst sich mit der Auswertung der Domäne *Social Media*. Die *Accuracy*-Werte können in Tabelle 17 betrachtet werden. Bei der dichotomen

ACC	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	SM01-Cieliebak	71,24	71,56	70,60	71,63	**89,65**	89,50	76,99
	SM02-Sidarenka	75,57	74,71	74,42	74,81	84,17	**85,18**	76,23
	SM03-Narr	70,87	72,75	59,96	71,90	**89,28**	80,94	77,67
	SM04-Mozetič	74,72	76,37	73,48	73,63	**86,27**	86,15	77,10
	SM05-Siegel	77,56	78,21	68,59	68,59	**78,85**	73,08	74,36
	SM06-Momtazi	64,74	65,60	58,76	60,26	81,20	**83,97**	73,29
Tri	SM01-Cieliebak	69,56	70,30	67,26	68,81	70,33	**80,92**	73,22
	SM02-Sidarenka	65,15	59,38	64,81	63,37	77,10	**77,13**	67,37
	SM03-Narr	69,48	69,30	65,80	67,43	**81,00**	79,25	72,20
	SM04-Mozetič	62,23	63,70	59,58	63,90	**68,86**	68,39	64,55
	SM05-Siegel	74,22	74,25	65,64	65,64	**74,88**	66,25	68,67
	SM06-Momtazi	60,20	63,26	57,34	60,00	79,15	**85,51**	69,38

Tabelle 17.: Accuracy-Werte für Texte aus den sozialen Medien

Polarität liefert das *GBERT* Modell (84,90 %) die besten Ergebnisse. Der schlechteste Algorithmus ist das *Convolutional Neural Network*, das durchschnittlich nur 67,63 % erreicht. Bei der trichotomen Polarität werden Werte bis zu 71,31 % erreicht. Dies wird mittels *GBERT* ermöglicht. Die geringste *Accuracy* wird durch das *SVM* (60,32 %) erreicht. In der Tabelle 18 werden die F1-Werte dargestellt. Hierbei ist zu erkennen, dass das Modell *GBERT* in beiden Fällen am besten abschneidet. Das *CNN* liefert im dichotomen und trichotomen Fall die schlechtesten Ergebnisse. Die benötigte Zeit für negative und positive Klassifikation beträgt weniger als eine Sekunde (NB) bis hin zu 48 Minuten 16 Sekunden (GE). Für die Klassifikation von drei Fällen wird eine Sekunde (NB) bis 1 Stunde, 48 Minuten 42 Sekunden (GE) benötigt.

F1	Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Di	SM01-Cieliebak	70,73	69,68	69,58	71,52	**89,61**	89,47	76,94
	SM02-Sidarenka	75,13	71,94	73,51	74,09	84,18	**85,31**	76,01
	SM03-Narr	70,06	71,40	51,72	71,51	**89,17**	77,90	77,58
	SM04-Mozetič	74,52	76,07	73,24	73,59	**86,26**	86,16	77,03
	SM05-Siegel	76,32	**77,37**	55,82	55,82	76,28	64,30	68,16
	SM06-Momtazi	63,92	63,05	45,88	59,20	77,42	**82,52**	72,92
Tri	SM01-Cieliebak	68,61	67,23	66,97	67,65	80,31	**80,86**	72,61
	SM02-Sidarenka	64,91	56,52	63,62	62,78	**77,16**	77,13	67,17
	SM03-Narr	66,90	65,02	63,60	65,59	**81,09**	78,51	71,64
	SM04-Mozetič	61,37	61,69	58,88	61,85	**68,78**	68,27	64,05
	SM05-Siegel	71,32	**72,11**	52,02	52,02	71,43	53,35	60,47
	SM06-Momtazi	57,82	59,99	43,31	54,19	73,74	**83,57**	67,69

Tabelle 18.: F1-Werte für Texte aus den sozialen Medien

7.4. Betrachtung der einzelnen Polaritätsklassen

Das folgende Kapitel beschäftigt sich mit der Klassifikation einzelner Polaritäts-Klassen. Hierzu werden für jede Domäne die durchschnittlichen *Accuracy*-Werte pro Polarität berechnet. Bei den angegebenen Zahlen handelt es sich um Prozentsätze, wobei die einzelnen Werte wie unten beschrieben berechnet werden: Für jeden Datensatz in einer Domäne wird die *Accuracy* für jeden Fall berechnet. Anschließend wird der Mittelwert für die gesamte Domäne und der entsprechenden Polarität gebildet. Dazu werden die richtig klassifizierten Einheiten einer Polarität durch die Summe der richtig und falsch klassifizierten Einheiten dieser Klasse geteilt. Der Durchschnitt wird berechnet, indem die einzelnen *Accuracy*-Werte jedes Korpus einer Domäne addiert und dann durch die Anzahl dieser Korpora geteilt werden.

In Tabelle 19 werden die dichotomen Korpora betrachtet. In der Domäne der literarischen Texte gibt es zwei Werte die besonders Auffallen. So erzielte das *CNN* bei positiven Texten keine richtige Klassifikation (0,00 %) und bei negativen Texten wurde eine Klassifikationsgenauigkeit von 100,00 % generiert. Der beste Ansatz für

Domäne	SVM		NB		CNN		RNN		GB		GE		BBMUS	
	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg
Literarische Texte	35,92	75,11	22,50	82,76	0,00	100,00	13,66	86,02	22,28	92,60	22,40	94,95	23,69	86,59
Gemischte Domäne	50,08	82,86	41,32	86,76	16,90	98,23	32,22	87,90	86,20	92,34	66,76	94,83	65,60	85,43
Nachrichten-artikel	51,66	73,19	53,81	72,57	44,18	64,54	58,78	56,15	73,43	88,63	74,35	64,23	61,65	75,21
Produkt-reviews	90,17	59,05	90,47	53,23	92,32	51,35	90,42	56,86	94,63	83,00	95,44	80,98	90,51	73,07
Soziale Medien	76,64	60,97	82,53	53,29	72,56	47,38	64,21	65,92	83,65	80,82	78,22	80,12	73,13	71,38
Ansatz Gesamt	60,89	70,24	58,13	69,72	45,19	72,30	51,86	70,57	72,04	87,48	67,43	83,02	62,91	78,34

Tabelle 19.: Angabe der durchschnittlichen Accuracy einer Klasse im dichotomen Fall für alle Ansätze

positive Texte ist somit das *SVM* (35,92 %), der schlechteste Ansatz für negative Texte ist ebenfalls das *SVM* (75,11 %). In Texteinheiten aus gemischten Domänen zeigte *GBERT* mit 86,20 % die beste Klassifikation von positiven Texteinheiten, das *CNN* erzielt nur 16,90 % bei diesen Texten und ist somit das schlechteste. Die negativen Texte konnten mittels *CNN* (98,23 %) am besten klassifiziert werden. Die schlechteste Leistung zeigte das *SVM* mit 82,86 % Genauigkeit. Bei den Nachrichtenartikeln zeigte das *CNN* (44,18 %) das schlechteste Ergebnis und *GELECTRA* (74,35 %) das beste Ergebnis bei positiver Polarität. Bei negativen Fällen zeigte das *RNN* (58,78 %) eine geringe Genauigkeit, *GELECTRA* (74,35 %) schnitt am besten ab. In der Domäne der Produktbewertungen zeigte *GELECTRA* mit 95,44 % die besten Ergebnisse, das *SVM* (90,17 %) hatte das schlechteste Ergebnis. Bei der Klassifikation von negativen Texten wurde *GBERT* (83,00 %) am besten eingestuft, das *CNN* (51,35 %) am schlechtesten. In den Texten der sozialen Medien zeigte *GBERT* (83,65 %) die beste Leistung, das *RNN* (64,21) die schlechteste Leistung. Bei der Klassifikation der negativen Texte ist *GBERT* (80,82 %) am stärksten, das *RNN* (47,38 %) am schwächsten. Dadurch ergibt sich über alle Domänen folgendes Bild. Der beste Ansatz für positive Texte ist *GBERT* (72,04 %), der schlechteste Ansatz ist das *CNN* (45,19 %). Bei negativen Texteinheiten schneidet *GBERT* (87,48 %) ebenfalls am besten ab. Das schlechteste Ergebnis hier liefert der *NB*-Ansatz. Diese Werte sind die Durch-

schnittswerte aus allen Domänen.

Die Tabelle 20 zeigt die durchschnittliche *Accuracy* im trichotomen Fall. Im Fall der literarischen Texte ist der *GBERT*-Ansatz (25,52 %) der beste im Bezug auf positive Texteinheiten der *CNN*-Ansatz (0,00 %) der schlechteste. Bei negativen Texten kann der *NB*-Algorithmus (66,84 %) die besten Ergebnisse liefern. Das *CNN* (42,30 %) liefert die schlechtesten Werte. Texteinheiten mit neutraler Polarität werden am besten mit einem *CNN* (80,25 %) klassifiziert, hingegen werden diese mittels *NB* (26,52 %) am schlechtesten erkannt. Im Fall von Texten aus gemischten Domänen werden positive Texteinheiten mittels *GBERT* (65,89 %) am besten und mittels *RNN* (19,01 %) am schlechtesten eingeordnet. Negative Texte werden durch *GELECTRA* (81,31 %) sehr gut erkannt. Durch das Verwenden von einem *SVM*-Ansatz (55,88 %) werden diese Texte am schlechtesten erkannt. Mithilfe von *GELECTRA* (73,42 %) können neutral Texte gut klassifiziert werden, durch ein *CNN* (50,45 %) werden neutrale Texte nicht gut erkannt Nachrichtenartikel können durch *GBERT* (49,04 %) am besten und mittels *CNN* (8,06 %) am schlechtesten im positiven Fall klassifiziert werden. Texteinheiten mit negativer Polarität werden durch *GBERT* (52,08 %) am besten und durch *NB* (26,36 %) am schlechtesten bestimmt. Die neutralen Texte können durch *GELECTRA* (87,55 %) sehr gut klassifiziert werden. Das *RNN* (80,25 %) zeigt in diesem Fall die schlechtesten Ergebnisse. Bei Produktbewertungen kann *GELECTRA* (89,32 %) überzeugen. Das *CNN* erreicht nur 77,76 % Genauigkeit bei positiven Texteinheiten. Bei negativen Texten schneidet *GELECTRA* am besten mit 74,22 % ab, der *NB*-Ansatz (35,41 %) belegt den letzten Platz. Mittels *GBERT* (52,82 %) können neutrale Texte am besten klassifiziert werden. Durch das Verwenden des *NB*-Ansatzes (37,28 %) werden die schlechtesten Ergebnisse erzielt werden. Die Klassifikation von positiven Beiträgen aus der Domäne soziale Medien, gelingt durch *GBERT* (72,73 %) am besten. Von der Verwendung eines *RNN* (51,81 %) wird abgeraten. Bei Texteinheiten mit negativer Polarität sollte *GELECTRA* (72,64 %) verwendet werden. Der *NB*-Ansatz (35,68 %) erzielt die niedrigsten Werte. Mithilfe von *Support Vector Machines* können 53,84 % neutrale Texte klassifiziert werden. Dies entspricht dem besten Wert. Der *NB*-Ansatz kann 51,12 % erzielen und stellt daher

der schlechteste Ansatz dar. Zusammenfassend können bei positiven Texten Ansätze wie *GBERT* (60,50 %) am besten klassifizieren, *CNNs* (32,11 %) am schlechtesten. Bei negativen Texten kann *GELECTRA* mit 65,60 % überzeugen. Der *CNN*-Ansatz schneidet erneut mit 43,66 % am schlechtesten ab. Bei den neutralen Texten kann ebenfalls *GELECTRA* (66,70 %) die besten Ergebnisse erzielen. Der *NB*-Ansatz (53,12 %) überzeugt nicht. In Anhang C werden die besten und schlechtesten Ansätze angegeben und deren Werte werden dargestellt. Dazu werden die dichotomen und trichotomen Korpora getrennt betrachtet.

Domäne	SVM			NB			CNN			RNN			GB			GE			BBMUS		
	Pos	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**	**Pos**	**Neg**	**NEU**
Literarische Texte	20,14	45,88	53,27	14,53	66,84	26,52	0,00	42,30	80,25	1,49	57,89	61,67	25,52	61,60	58,16	19,19	52,30	66,32	17,24	43,23	68,63
Gemischte Domäne	32,06	55,88	61,76	24,43	56,74	63,02	21,31	62,48	50,45	19,01	56,70	53,51	65,89	78,97	71,92	50,20	81,31	73,42	43,12	57,46	67,18
Nachrichten-artikel	18,90	29,60	82,17	9,60	26,36	87,67	8,06	30,60	83,68	13,37	28,96	80,25	49,04	52,08	85,96	43,88	47,52	87,55	41,72	34,26	81,10
Produkt-reviews	80,76	43,86	44,31	87,38	35,41	37,28	77,63	45,32	40,93	84,42	39,41	41,93	89,32	59,43	52,82	89,63	74,22	52,30	82,19	59,50	45,95
Soziale Medien	58,47	44,16	53,84	60,93	35,68	51,12	53,53	37,61	52,02	51,81	40,75	52,57	72,73	69,99	53,62	63,68	72,64	53,91	58,06	55,55	52,39
Ansatz Gesamt	42,07	43,87	59,07	39,37	44,21	53,12	32,11	43,66	61,47	34,02	44,74	57,98	60,50	64,41	64,50	53,32	65,60	66,70	48,47	50,00	63,05

Tabelle 20.: Angabe der durchschnittlichen Accuracy einer Klasse im trichotomen Fall für alle Ansätze

8. Diskussion

Im folgenden Kapitel werden die Ergebnisse der Kreuzevaluation diskutiert. Außerdem wird auf die Varianten *Base* und *Large* der Transformer-Modelle eingegangen und die Notwendigkeit einer Vorverarbeitung bei Transformer-basierten Modellen wird erörtert.

8.1. Vorverarbeitungsschritte bei Transformern

Die Auswertungen aus Kapitel 7.1 zeigt, bei der Betrachtung dichotomer Polarität, das in den meisten Domänen keine vorverarbeitenden Schritte notwendig sind. Die Domäne der literarischen Texte bildet hier eine Ausnahme. Hierbei zeigte sich, dass *GBERT* höhere F1-Werte erreicht, wenn vorverarbeitende Schritte unternommen werden. Ebenso erreichten *GBERT* und *GELECTRA* bessere *Accuracy*-Werte bei der Verwendung dieser Schritte. Ein möglicher Grund dafür, dass sich die Ergebnisse hier verbesserten, ist das Entfernen der Sprecherrollen am Anfang jeder Texteinheit. Da hier oft der gleiche Name angegeben ist, könnte dies zu einer verzerrten Klassifikation geführt haben. In den anderen Domänen gab es eine solche explizite Namensnennung nicht. Außerdem zeigte sich, dass das *Preprocessig* bei Produktbewertungen im Fall des Modells *GELECTRA* einen positiven Einfluss hat. Ein Grund, weshalb die anderen Domänen von den vorverarbeitenden Schritten nicht profitierten, kann an der Architektur der Transformer-Modelle liegen. Da bei solchen Modellen alle im Text verfügbaren Informationen zur Klassifizierung herangezogen werden, kann das Herausfiltern bestimmter Elemente zu einer Verschlechterung der Genauigkeit führen, da nun Elemente mit Informationen fehlen. Bei den Korpora, die aus trichotomen Texteinheiten bestehen, wird in den meisten Fällen keine Vorverarbeitung empfohlen, da so die F1- und *Accuracy*-Werte sinken können. Eine Ausnahme bildet hier die Domäne der Nachrichtenartikel. Bei dem *GELECTRA*

Modell werden, die Werte durch das *Preprocessing* gesteigert. Ein eindeutiger Grund kann in diesem Fall allerdings nicht bestimmt werden.

Obwohl bei dem Vorverarbeiten der Korpora bestimmte Elemente entfernt werden, gibt es keine zeitliche Änderung bei der Laufzeit der Klassifikationsaufgabe. Es ist auch zu beachten, dass die Texteinheiten weniger stark vorverarbeitet werden, wie zum Beispiel die Texte, die für das *SVM* oder die neuronalen Netze verwendet wurden. Daher wird der Text weniger stark in seiner Dimensionalität verändert, was die ähnlichen Zeiten bei den Durchläufen mit und ohne vorverarbeitender Schritte erklärt.

Zusammenfassend lässt sich sagen, das in den meisten Fällen auf vorverarbeitende Schritte bei Transformern verzicht werden kann. Eine Ausnahme bilden hier besondere Texte wie historische und literarische Arbeiten, da hier der Satzbau sich stärker unterscheidet. Andere Arbeiten zeigten in unterschiedlichen Domänen, dass Transformer-basierte Ansatz am besten mit unverarbeiteten Texten arbeiten. So verwendeten Schmidt et al. (2022) keine vorverarbeitenden Schritte und erzielten *Accuracy*-Werte von 93,30 % In der Arbeit von Mathew & Bindu (2020) konnten *Accuracy*-Werte von 94,08 % erreicht werden.

8.2. Die Base und Large Varianten der Transformer-Modelle

Die Ergebnisse aus Kapitel 7.2 zeigen, dass die *Large* Versionen von *GBERT* und *GELECTRA* in vielen Domänen bessere Resultate erzielen können. Diese Ergebnisse sind im Einklang mit der Arbeit von Devlin et al. (2019). Diese Arbeit zeigte, dass größere Versionen des *GBERT*-Modells auch bessere Ergebnisse lieferten (Devlin et al.; 2019). Auch im Falle des Modells *GELECTRA*, konnte das Ergebnis für trichotome Polaritäten bestätigt werden (Clark et al., 2020). Hierbei wurde in allen Domänen einen Anstieg in *Accuracy* und F1-Werten verzeichnet. Bei der dichotomen Polarität zeigte sich ein Anstieg dieser Werte in den Domänen Produktbewertungen, soziale Medien und bei Texten aus gemischten Domänen. Auffälligkeiten zeigten sich in den Bereichen der literarischen Texte und der Nachrichtenartikel. So erzielte *GBERT* in der Version *Large* schlechtere Ergebnisse bei den F1-Werten. Ein mögli-

cher Grund dafür könnte der andersartige Text sein. Da die im *Pretraining* verwendeten Daten kaum ältere literarische Texte verwendeten. (Chan et al.; 2020). Ebenso zeigte sich, dass *GELECTRA* in der Domäne der Nachrichtenartikel bei beiden Metriken schlechtere Ergebnisse aufwies. Hierbei kann der Korpus NA01-Clematide als möglicher Grund genannt werden. Die Variante *Large* erzielte hierbei schlechtere Ergebnisse. Eine Aussage, weshalb gerade dieser Korpus niedrigere Werte bei der *GELECTRA Large* Version lieferte, kann nicht mit Sicherheit getroffen werden.

Durch das Verwenden der *Large* Versionen der beiden oben genannten Transformer-Modellen, stieg auch die benötigte Zeit, um die Klassifikationsaufgabe abzuschließen. Bereits beim *pretraining* zeigte sich, dass *Large* Versionen mehr Zeit benötigten als die Basismodelle (Chan et al.; 2020). Es ist wenig überraschend, dass die größeren Modelle bei der *finetuning* Aufgabe ebenfalls mehr Zeit benötigten. Besonders bei der Domäne der Produktbewertungen stieg diese Trainingszeit stark an, mit einem maximalen Zuwachs von 2 Stunden und 17 Minuten. Der Grund hierfür ist die große Menge an Texteinheiten.

Zusammenfassend lässt sich festhalten, dass die größeren Modelle für trichotome Polaritäten aufgrund ihrer Leistungssteigerung zu bevorzugen sind, wenn die benötigte Zeit von begrenztem Belang ist. Bei positiven und negativen Texteinheiten kann die Domäne eine Rolle spielen, weshalb eine Auswahl der entsprechenden Variante sinnvoll erscheint.

8.3. Die Ergebnisse der Kreuzevaluation

Im folgenden Abschnitt werden jeweils Besonderheiten und allgemeine Rückschlüsse auf den einzelnen Domänen aufgezählt.

Literarische Texte

Die Domäne der literarischen Texte stellt eine besondere Herausforderung für Klassifikationsalgorithmen dar. Hierbei zeigte sich, dass vor allem die künstlichen neuronalen Netze *CNN* und *RNN* Probleme bei der Sentimentbestimmung hatten. So ordnete das *CNN* in jedem Korpus alle Texteinheiten der negativen Polarität zu. Auch das *RNN* zeigte dieses Verhalten in dem Korpus LT03-Schmidt. Eine mögli-

che Erklärung diesbezüglich ist, dass *LSTMs* gut bei verschwindenden Gradienten funktionieren. Dieses Problem wird besser mit einer Abfolge von Datenpunkten, wie bei Videos bewältigt, was bei einer Textklassifikation nicht vorliegt (Alaparthi & Mishra, 2020). Dieses Phänomen ist auch in anderen Domänen zu beobachten. Die Transformer-basierten Ansätze zeigten eine starke Priorisierung einzelner Klassen im dichotomen Fall. Dies wird besonders anhand des Korpus LT03-Schmidt deutlich. Die dichotome Klassifikation wurde am besten durch das *SVM* gelöst, wobei auch die Ansätze *GBERT* und *GELECTRA* gute F1-Werte zeigten.

Gemischte Domäne

In dieser Domäne zeigten die beiden Ansätze *GBERT* und *GELECTRA* die besten Resultate im Bezug auf *Accuracy* und F1-Werte. Auffällig ist hier, dass das *CNN* und das *RNN* erneut eine starke Priorisierung der negativen Klasse aufzeigten. Eine mögliche Erklärung diesbezüglich ist die stärker ausgeprägte Anzahl an negativen Texten in der dichotomen Version. Weiterhin sinkt die Klassifikationsgenauigkeit der einzelnen Polaritäten, wenn die Anzahl der vorhandenen Klassenkategorien steigt. Somit sollten bei der Sentimentbestimmung bei Texten aus unterschiedlichen Domänen Transformer-basierte Ansätze gewählt werden, da diese die besten Ergebnisse liefern.

Nachrichtenartikel

Anhand der Domäne der Nachrichtenartikel wird erneut gezeigt, dass die Transformer-basierten Ansätze *GBERT* und *GELECTRA* die besten *Accuracy*- und F1-Werte generieren. Andere Arbeiten zeigten ebenfalls, dass solche Ansätze bei Nachrichtenartikeln die besten Ergebnisse erzielten (Mishev et al., 2020). Bei der dichotomen Klassifikation zeigte sich, dass alle Ansätze, mit Ausnahme der Transformer, Schwierigkeiten damit hatten, die Texteinheiten im Korpus NA03-Schabus zu bestimmen. Dies liegt vermutlich an der sehr starken unausgeglichenen Verteilung der positiven und negativen Klassen. Hier zeigt sich erneut, dass die negative Klasse besser klassifiziert wird. Im trichotomen Fall haben dieselben Algorithmen außerdem deutliche Schwierigkeiten bei der Bestimmung der positiven Texteinheiten. Im Fall von NA03-Schabus kann dies wieder mit der unausgeglichenen Verteilung der Klassen

begründet werden.

Produktbewertungen

Auch in der Domäne der Produktbewertungen zeigen die Transformer-basierten Ansätze erneut die besten Ergebnisse in *Accuracy* und im F1-Wert. Auch in anderen Arbeiten sind die verwendeten Transformer-basierten Ansätze den übrigen Algorithmen überlegen (Trueman et al., 2022). Im Korpus RE01-Klinger wurden Texteinheiten mit positiver Polarität durch künstliche neuronale Netze und den *NB*-Ansatz priorisiert. Dies kann auch auf eine ungleiche Verteilung der Klassen zurückgeführt werden. Aber auch in den anderen Korpora dieser Domäne wird deutlich, dass positive Texteinheiten besser erkannt werden. Dies ist unabhängig von der Anzahl der Klassen. Besonders hervorzuheben sind hier die Korpora RE02-Sänger und RE04-Guhr, da in diesen Fällen die Anzahl der verschiedenen Polaritäten ausgeglichen wurde. Dennoch klassifizieren alle Ansätze die positiven Texteinheiten im trichotomen Fall besser als in den beiden anderen Fällen.

Soziale Medien

Die Domäne der sozialen Medien zeigt, dass die Transformer-basierten Ansätze erneut die besten *Accuracy*- und F1-Werte liefern. Andere Arbeiten zeigten ebenfalls, dass Transformer-basierte Ansätze in dieser Domäne *State of the Art* Ergebnisse liefern (Schmidt et al., 2022; Nair et al., 2021). Das Korpus SM05-Siegel stellt hierbei eine Besonderheit dar, da dieser Korpus vermehrt aus ironischen Texten besteht. Überraschenderweise ist hier der *Naïve Bayes* Ansatz eine gute Alternative zu *GBERT* und *GELECTRA*. Bei der Klassifikation von neutralen Texten im Korpus SM06-Momtazi zeigte sich, dass alle Ansätze in diesen Fall keine guten Ergebnisse lieferten. Die Ursache dafür dürfte erneut an dem Ungleichgewicht der Klassen liegen. Im dichotomen Fall sind positive Texte durch die Ansätze *SVM, NB* und *CNN* besser zu identifizieren als negative Texte, wobei der beste Ansatz immer noch *GBERT* ist.

8.4. Zusammenfassung der Beobachtungen

Die Verwendung von vorverarbeitenden Schritten wird in der Regel bei dem Modell *GBERT* nicht empfohlen. Eine Ausnahme bilden hier dichotome literarische Texte, diese können so leicht bessere Werte generieren. Das Modell *GELECTRA* profitiert bei dichotomen Produktbewertungen von vorverarbeitenden Schritten. Im Fall von trichotomen literarischen Texten oder Nachrichtenartikeln kann ebenso eine Steigerung erzielt werden. Die Verwendung der *Large* Variante wird in trichotomen Texten empfohlen, bei der dichotomen Klassifikation kann die *Base* Variante nur in der Domäne Nachrichtenartikeln bessere Resultate liefern. In den meisten Fällen zeigen die Transformer-basierten Modelle die besten Ergebnisse in den unterschiedlichen Domänen. Hierbei setzt sich vor allem das Modell *GBERT* durch, wobei der *GELECTRA*-Ansatz ebenfalls sehr gute Ergebnisse lieferte. Nur in besonderen Fällen, wie der Klassifikation von positiven Elementen bei literarischen Texten gab es andere Algorithmen, die leicht bessere Ergebnisse erzielten. Schlussendlich bieten Transformer-basierte Ansätze gute *Accuracy*- und F1-Werte, obwohl die Trainingszeit auch dementsprechend größer ist.

9. Zusammenfassung und Ausblick

In dieser Arbeit wurden sieben verschiedene *Machine Learning*-Ansätze anhand von bis zu 20 verschiedenen Korpora evaluiert. Für die klassischen Ansätze wurden Vorverarbeitungsschritte durchgeführt. Unter anderem wurde vorab untersucht, ob diese Vorverarbeitungsschritte auch bei Transformer-basierten Modellen zu einer Verbesserung der Klassifikationsgenauigkeit führen. Ebenso wurde untersucht, ob die Klassifikationsgenauigkeit durch verschiedenen Varianten der Transformer-Modelle gesteigert werden kann. Die verwendeten Ansätze und deren Implementierung wurden ausführlich dargelegt und anschließend wurde eine Kreuzevaluation all dieser Ansätze auf allen Korpora in dichotomer und trichotomer Polarität ausgeführt. Die Auswertung identifizierte die besten Ansätze pro Korpus und zeigte Besonderheiten in den Domänen, wie zum Beispiel die schlechte Klassifikationsgenauigkeit bei literarischen Texten oder die durchweg gute Leistung der Transformer-basierten Modelle.

Mithilfe dieser Arbeit können kommende Studien gezielt einen passenden Ansatz für die Sentiment Analyse in der entsprechenden Domäne auswählen. Die Erkenntnisse aus den Unterstudien können verwendet werden, um Transformer-basierte Ansätze bereits zu Beginn zu optimieren. Dennoch gibt es einige Aspekte, die in dieser Arbeit nicht näher untersucht werden konnten.

Es gibt eine Vielzahl von Studien, die durchgeführt werden können, um die Ergebnisse zu verfeinern und weiterzuführen. So könnten die verwendeten Korpora hinsichtlich der Anzahl der Klassenelemente ausgeglichen und kleine Korpora desselben Fachgebiets zusammengeführt werden, um einen größeren Datensatz für das Training der Modelle zur Verfügung zu haben. Darüber hinaus können Transformer-basierte Ansätze mit Hyperparametern weiter optimiert werden, um noch bessere Ergebnisse zu erzielen. Dies könnte dazu beitragen, dass Transformer-

Modelle auch in speziellen Domänen eine höhere Klassifikationsgenauigkeit erreichen.

Die Erstellung deutscher Korpora, die anschließende Sentiment Analyse mit verschiedenen *Machine Learning*-Ansätzen und deren Auswertung sind wichtige Beiträge im Bereich der natürlichen Sprachverarbeitung. Besonders in nicht-Englischen Sprachen herrscht ein Mangel an solchen Ressourcen und derer systematischen Bewertung. Diese Arbeit soll einen wichtigen Beitrag zur Sentiment Analyse im Deutschen bringen und gibt Anregungen für zukünftige Fragestellungen und Forschungsarbeiten.

Literaturverzeichnis

Ahmad, M., Aftab, S. & Ali, I. (2017). Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, *177* (5), 25–29.

Ahmad, M., Aftab, S., Bashir, M. S., Hameed, N., Ali, I. & Nawaz, Z. (2018). Svm optimization for sentiment analysis. *Int. J. Adv. Comput. Sci. Appl*, *9* (4), 393–398.

Alam, S. & Yao, N. (2019). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Computational and Mathematical Organization Theory*, *25* (3), 319–335.

Alamoodi, A. H., Zaidan, B. B., Zaidan, A. A., Albahri, O. S., Mohammed, K., Malik, R. Q., ... others (2021). Sentiment analysis and its applications in fighting covid-19 and infectious diseases: A systematic review. *Expert systems with applications*, *167*, 114155.

Alaparthi, S. & Mishra, M. (2020). Bidirectional encoder representations from transformers (bert): A sentiment analysis odyssey. *arXiv preprint arXiv:2007.01127*.

Alessia, D., Ferri, F., Grifoni, P. & Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, *125* (3).

Arras, L., Montavon, G., Müller, K.-R. & Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. *arXiv preprint arXiv:1706.07206*.

Asghar, M. Z., Khan, A., Ahmad, S. & Kundi, F. M. (2014). A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research*, *4* (3), 181–186.

Awad, M., Khanna, R., Awad, M. & Khanna, R. (2015). Support vector machines for classification. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 39–66.

Baktha, K. & Tripathy, B. (2017). Investigation of recurrent neural networks in the field of sentiment analysis. In *2017 international conference on communication and signal processing (iccsp)* (S. 2047–2050).

Balahur, A., Hermida, J. M. & Montoyo, A. (2012). Detecting implicit expressions of emotion in text: A comparative analysis. *Decision support systems*, *53* (4), 742–753.

Balazs, J. A. & Velásquez, J. D. (2016). Opinion mining and information fusion: a survey. *Information Fusion*, *27*, 95–110.

Basarslan, M. S., Kayaalp, F. et al. (2020). *Sentiment analysis with machine learning methods on social media.* Ediciones Universidad de Salamanca (España).

Berrar, D. (2019). *Cross-validation.*

Bindra, A. (2016). *English grammar: Rules and usage*. Notion Press.

Bird, S., Klein, E. & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit*. OŔeilly Media Inc.

Bütow, F., Lommatzsch, A. & Ploch, D. (2016). Creation of a german corpus for internet news sentiment analysis. *Project report, Berlin Institute of Technology, AOT*.

Chan, B., Schweter, S. & Möller, T. (2020). German's next language model. arxiv 2020. *arXiv preprint arXiv:2010.10906*.

Chollet, F. (2018). *Deep learning mit python und keras: das praxis-handbuch vom entwickler der keras-bibliothek*. MITP-Verlags GmbH & Co. KG.

Cieliebak, M., Deriu, J. M., Egger, D. & Uzdilli, F. (2017). A twitter corpus and benchmark resources for german sentiment analysis. In *5th international workshop on natural language processing for social media, boston ma, usa, 11 december 2017* (S. 45–51).

Clark, K., Luong, M.-T., Le, Q. V. & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Clematide, S., Gindl, S., Klenner, M., Petrakis, S., Remus, R., Ruppenhofer, J., ... Wiegand, M. (2012). *Mlsa—a multi-layered reference corpus for german sentiment analysis.* University of Zurich.

Dashtipour, K., Poria, S., Hussain, A., Cambria, E., Hawalah, A. Y., Gelbukh, A. & Zhou, Q. (2016). Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cognitive computation*, *8*, 757–771.

Denecke, K. (2008). Using sentiwordnet for multilingual sentiment analysis. In *2008 ieee 24th international conference on data engineering workshop* (S. 507-512). doi: 10.1109/ICDEW.2008.4498370

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dey, L., Chakraborty, S., Biswas, A., Bose, B. & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

Dos Santos, C. & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: technical papers* (S. 69–78).

Du, K. & Mellmann, K. (2019). *Sentimentanalyse als instrument literaturgeschichtlicher rezeptionsforschung: Ein pilotprojek*. Niedersächsische Staats-und Universitätsbibliothek Göttingen.

Fehle, J., Schmidt, T. & Wolff, C. (2021). *Lexicon-based sentiment analysis in german: Systematic evaluation of resources and preprocessing techniques.* KONVENS 2021 Organizers.

Ghosh, M. & Sanyal, G. (2017). Preprocessing and feature selection approach for efficient sentiment analysis on product reviews. In *Proceedings of the 5th international conference on frontiers in intelligent computing: Theory and applications* (S. 721–730).

Gonçalves, P., Araújo, M., Benevenuto, F. & Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the first acm conference on online social networks* (S. 27–38).

Guhr, O., Schumann, A.-K., Bahrmann, F. & Böhme, H. J. (2020). Training a broad-coverage german sentiment classification model for dialog systems. In *Proceedings of the 12th language resources and evaluation conference* (S. 1627–1632).

Habimana, O., Li, Y., Li, R., Gu, X. & Yu, G. (2020). Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, *63*, 1–36.

Haddi, E., Liu, X. & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia computer science*, *17*, 26–32.

Hemalatha, I., Varma, G. S. & Govardhan, A. (2012). Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *1* (2), 58–61.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F. & Kaymak, U. (2013). Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th annual acm symposium on applied computing* (S. 703–710).

Idrissi-Yaghir, A., Schäfer, H., Bauer, N. & Friedrich, C. M. (2023). Domain adaptation of transformer-based models using unlabeled data for relevance and polarity classification of german customer feedback. *SN Computer Science*, *4* (2), 142.

Jivani, A. G. et al. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2* (6), 1930–1938.

Jurafsky, D. & Martin, J. H. (2022). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.*

Kang, H., Yoo, S. J. & Han, D. (2012). Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, *39* (5), 6000-6010. Zugriff auf `https://www.sciencedirect.com/science/article/pii/S0957417411016538` doi: https://doi.org/10.1016/j.eswa.2011.11.107

Kim, K. (2018). An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis. *Expert Systems with Applications*, *109*, 49–65.

Klinger, R. & Cimiano, P. (2014). The usage review corpus for fine grained multi lingual opinion analysis. In *Proceedings of the ninth international conference on language resources and evaluation* (*lrec'14*) (S. 2211–2218).

Krouska, A., Troussas, C. & Virvou, M. (2016). The effect of preprocessing techniques on twitter sentiment analysis. In *2016 7th international conference on information, intelligence, systems & applications* (*iisa*) (S. 1–5).

Liao, S., Wang, J., Yu, R., Sato, K. & Cheng, Z. (2017). Cnn for situations understanding based on sentiment analysis of twitter data. *Procedia computer science*, *111*, 376–381.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5* (1), 1–167.

Lo, S. L., Cambria, E., Chiong, R. & Cornforth, D. (2017). Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review*, *48*, 499–527.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, *2* (2), 159–165.

Mathew, L. & Bindu, V. (2020). A review of natural language processing techniques for sentiment analysis using pre-trained models. In *2020 fourth international conference on computing methodologies and communication* (*iccmc*) (S. 340–345).

McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Bd. 445, S. 51–56).

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T. & Trajanov, D. (2020). Evaluation of sentiment analysis in finance: From lexicons to transformers. *IEEE Access*, *8*, 131662-131682. doi: 10.1109/ACCESS.2020.3009626

Momtazi, S. (2012). Fine-grained german sentiment analysis on social media. In *Proceedings of the eighth international conference on language resources and evaluation (lrec'12)* (S. 1215–1220).

Monika, R., Deivalakshmi, S. & Janet, B. (2019). Sentiment analysis of us airlines tweets using lstm/rnn. In *2019 ieee 9th international conference on advanced computing (iacc)* (S. 92–95).

Mozetič, I., Grčar, M. & Smailović, J. (2016). Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, *11* (5), e0155036.

Nair, A. J., Veena, G. & Vinayak, A. (2021). Comparative study of twitter sentiment on covid-19 tweets. In *2021 5th international conference on computing methodologies and communication (iccmc)* (S. 1773–1778).

Narr, S., Hulfenhaus, M. & Albayrak, S. (2012). Language-independent twitter sentiment analysis. *Knowledge discovery and machine learning (KDML), LWA*, 12–14.

Oliphant, T. E. et al. (2006). *A guide to numpy* (Bd. 1). Trelgol Publishing USA.

Parveen, H. & Pandey, S. (2016). Sentiment analysis on twitter data-set using naive bayes algorithm. In *2016 2nd international conference on applied and theoretical computing and communication technology (icatcct)* (S. 416–419).

Pawar, A. B., Jawale, M. & Kyatanavar, D. (2016). Fundamentals of sentiment analysis: concepts and methodology. *Sentiment analysis and ontology engineering: An environment of computational intelligence*, 25–48.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Ploch, D. (2015). Intelligent news aggregator for german with sentiment analysis. In *Smart information systems* (S. 5–46). Springer.

Prabowo, R. & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3* (2), 143–157.

Prettenhofer, P. & Stein, B. (2010). Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (S. 1118–1127).

Rajapakse, T., Raab, M., Franc, L. H., Carbin, M. A. & Sachan, D. S. (2019-2022). *Simple transformers: A high-level library for state-of-the-art nlp.* `https://github.com/ThilinaRajapakse/simpletransformers`. (Abgerufen am 5.02.2023)

Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*.

Rauh, C. (2018). Validating a sentiment dictionary for german political language—a workbench note. *Journal of Information Technology & Politics*, *15* (4), 319–343.

Sänger, M., Leser, U., Kemmerer, S., Adolphs, P. & Klinger, R. (2016). Scare—the sentiment corpus of app reviews with fine-grained annotations in german. In *Proceedings of the tenth international conference on language resources and evaluation (lrec'16)* (S. 1114–1121).

Schabus, D., Skowron, M. & Trapp, M. (2017). One million posts: A data set of german online discussions. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (S. 1241–1244).

Schmidt, T., Burghardt, M. & Dennerlein, K. (2018). *Sentiment annotation of historic german plays: An empirical study on annotation behavior.* RWTH Aachen.

Schmidt, T., Burghardt, M., Dennerlein, K. & Wolff, C. (2019). *Sentiment annotation for lessing's plays: Towards a language resource for sentiment analysis on german literary texts.* RWTH Aachen.

Schmidt, T., Dennerlein, K. & Wolff, C. (2021). Emotion classification in german plays with transformer-based language models pretrained on historical and contemporary language..

Schmidt, T., Fehle, J., Weissenbacher, M., Richter, J., Gottschalk, P. & Wolff, C. (2022, 12–15 September). Sentiment analysis on Twitter for the major German parties during the 2021 German federal election. In *Proceedings of the 18th conference on natural language processing (konvens 2022)* (S. 74–87). Potsdam, Germany: KONVENS 2022 Organizers. Zugriff auf `https://aclanthology.org/2022.konvens-1.9`

Schumaker, R. P., Zhang, Y., Huang, C.-N. & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems*, *53* (3), 458–464.

Sidarenka, U. (2016). Potts: the potsdam twitter sentiment corpus. In *Proceedings of the tenth international conference on language resources and evaluation (lrec'16)* (S. 1133–1141).

Sidarenka, U. (2019). *Sentiment analysis of german twitter* (Dissertation). doi: 10.25932/PUBLISHUP-43742

Siegel, M., Emig, K., ihringer, N., Kesim, S. & Yilmaz, T. (2017). Github repository: Sentiment analysis. ressources for sentiment analysis of german language..

Singh, J. & Gupta, V. (2016). Text stemming: Approaches, applications, and challenges. *ACM Computing Surveys (CSUR)*, *49* (3), 1–46.

Tang, H., Tan, S. & Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, *36* (7), 10760–10773.

Trueman, T. E., Jayaraman, A. K., Ananthakrishnan, G., Cambria, E. & Mitra, S. (2022). *An n-gram-based bert model for sentiment classification using movie reviews.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vijayarani, S., Ilamathi, M. J., Nithya, M. et al. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5* (1), 7–16.

Wang, H. & Castanon, J. A. (2015). Sentiment expression via emoticons on social media. In *2015 ieee international conference on big data (big data)* (S. 2404–2408).

Wojatzki, M., Ruppert, E., Holschneider, S., Zesch, T. & Biemann, C. (2017). Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. *Proceedings of the GermEval*, 1–12.

Wunderlich, F. & Memmert, D. (2020). Innovative approaches in sports science—lexicon-based sentiment analysis as a tool to analyze sports-related twitter communication. *Applied Sciences*, *10* (2). Zugriff auf `https://www.mdpi.com/2076-3417/10/2/431` doi: 10.3390/app10020431

Yadav, P. & Pandya, D. (2017). Sentireview: Sentiment analysis based on text and emoticons. In *2017 international conference on innovative mechanisms for industry applications (icimia)* (S. 467–472).

Yue, L., Chen, W., Li, X., Zuo, W. & Yin, M. (2019). A survey of sentiment analysis in social media. *Knowledge and Information Systems*, *60*, 617–663.

Zainuddin, N. & Selamat, A. (2014). Sentiment analysis using support vector machine. In *2014 international conference on computer, communications, and control technology (i4ct)* (S. 333–337).

Zehe, A., Becker, M., Jannidis, F. & Hotho, A. (2017). Towards sentiment analysis on german literature. In *Joint german/austrian conference on artificial intelligence (künstliche intelligenz)* (S. 387–394).

Zhang, H., Gan, W. & Jiang, B. (2014). Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th web information system and application conference* (S. 262–265).

Zhang, L., Wang, S. & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8* (4), e1253.

Zuo, Z. (2018). *Sentiment analysis of steam review datasets using naive bayes and decision tree classifier.*

Çevikel, S. (2018). *emoji.* Zugriff auf `https://pypi.org/project/emoji/` (Abgerufen am 5.02.2023)

A. Laufzeiten der Methoden und Transformer-Varianten pro Domäne

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	00:00:00	00:00:00	00:00:04	00:00:08	00:01:47	00:01:47	00:00:39
Gemischte Domäne	00:00:18	00:00:00	00:00:12	00:00:51	00:16:01	00:16:02	00:05:12
Nachrichten-artikel	00:00:01	00:00:00	00:00:05	00:00:09	00:04:36	00:04:37	00:01:33
Produkt-bewertungen	03:15:30	00:00:07	00:03:39	00:17:08	03:33:57	03:34:12	01:08:10
soziale Medien	00:03:59	00:00:00	00:00:29	00:01:38	00:48:13	00:48:16	00:15:28

Tabelle 21.: Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von jedem Modell bei dichotomen Domänen

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	00:00:00	00:00:00	00:00:05	00:00:09	00:02:41	00:02:43	00:00:57
Gemischte Domäne	00:06:08	00:00:01	00:00:56	00:04:01	00:47:46	00:47:54	00:15:18
Nachrichten-artikel	00:00:08	00:00:00	00:00:08	00:00:15	00:11:13	00:11:14	00:03:41
Produkt-bewertungen	02:59:05	00:00:06	00:03:37	00:14:45	03:03:06	03:03:42	00:58:26
soziale Medien	00:28:10	00:00:01	00:01:04	00:04:00	01:48:33	01:48:42	00:34:42

Tabelle 22.: Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von jedem Modell bei trichotomen Domänen

Domäne	GBERT			GELECTRA		
	Base	Large	Differenz	Base	Large	Differenz
Literarische Texte	00:01:06	00:01:47	00:00:41	00:01:06	00:01:47	00:00:41
Gemischte Domäne	00:05:27	00:16:01	00:10:34	00:05:27	00:16:02	00:10:35
Nachrichten-artikel	00:01:58	00:04:36	00:02:38	00:01:57	00:04:37	00:02:40
Produkt-bewertungen	01:01:46	03:18:50	02:17:04	01:06:23	03:34:12	02:27:49
soziale Medien	00:10:06	00:31:07	00:21:01	00:10:13	00:31:09	00:20:56

Tabelle 23.: Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von den Varianten Base und Large im dichotomen Fall. Zusätzlich ist die zeitliche Differenz der beiden Varianten gegeben

Domäne	GBERT			GELECTRA		
	Base	Large	Differenz	Base	Large	Differenz
Literarische Texte	00:01:22	00:02:41	00:01:20	00:01:22	00:02:43	00:01:20
Gemischte Domäne	00:15:14	00:47:46	00:32:32	00:15:15	00:47:54	00:32:38
Nachrichten-artikel	00:03:59	00:11:13	00:07:15	00:03:59	00:11:14	00:07:16
Produkt-bewertungen	00:55:51	02:59:32	02:03:41	00:55:35	03:00:06	02:04:31
soziale Medien	00:21:29	01:08:02	00:46:32	00:21:31	01:08:08	00:46:37

Tabelle 24.: Angabe der durchschnittlichen benötigten Zeit (hh:mm:ss) von den Varianten Base und Large im trichotomen Fall. Zusätzlich ist die zeitliche Differenz der beiden Varianten gegeben

B. Durchschnittliche Accuracy- und F1-Werte der einzelnen Algorithmen auf Domänen

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	62,04	62,10	65,05	61,16	68,52	69,79	64,93
Gemischte Domäne	72,42	73,19	69,02	69,54	91,03	85,04	79,93
Nachrichten-artikel	80,21	82,18	76,54	78,13	93,32	83,70	83,39
Produkt-bewertungen	84,68	84,48	84,38	84,73	92,99	93,31	86,64
soziale Medien	72,45	73,20	67,63	70,13	84,90	83,14	75,94
Gesamt	74,36	75,03	72,52	72,74	86,15	83,00	78,17

Tabelle 25.: Durchschnittliche Accuracy der einzelnen Algorithmen im dichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	43,38	46,11	50,65	49,20	53,36	52,72	49,53
Gemischte Domäne	57,33	57,92	53,95	52,78	72,96	62,45	61,49
Nachrichten-artikel	69,28	72,10	70,42	68,77	79,74	77,66	73,17
Produkt-bewertungen	71,27	71,61	69,43	72,17	79,20	77,79	73,40
soziale Medien	60,32	61,93	61,11	60,73	71,31	67,66	64,40
Gesamt	60,32	61,93	61,11	60,73	71,31	67,66	64,40

Tabelle 26.: Durchschnittliche Accuracy der einzelnen Algorithmen im trichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	60,60	58,12	51,33	52,26	60,55	61,56	59,04
Gemischte Domäne	71,56	71,53	58,87	64,44	91,03	82,90	79,74
Nachrichten-artikel	78,80	80,55	70,12	73,12	92,45	78,86	81,96
Produkt-bewertungen	83,80	82,59	81,82	83,50	92,77	93,06	86,40
soziale Medien	71,78	71,59	61,62	67,62	83,82	80,95	74,77
Gesamt	73,31	72,87	64,75	68,19	84,12	79,47	76,38

Tabelle 27.: Durchschnittliche F1-Werte der einzelnen Algorithmen im dichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen

Domäne	SVM	NB	CNN	RNN	GB	GE	BBMUS
Literarische Texte	42,41	41,13	37,56	40,63	51,37	47,83	44,27
Gemischte Domäne	56,27	56,41	47,36	48,52	75,42	71,93	60,40
Nachrichten-artikel	67,32	68,46	67,34	66,44	79,86	80,35	72,14
Produkt-bewertungen	69,72	68,70	68,41	69,93	79,59	81,46	72,85
soziale Medien	65,15	63,76	58,07	60,68	75,41	73,61	67,27
Gesamt	60,17	59,69	55,75	57,24	72,33	71,04	63,39

Tabelle 28.: Durchschnittliche F1-Werte der einzelnen Algorithmen im trichotomen Fall. Gesamt setzt sich aus dem Mittelwert der Domänen zusammen

C. Accuracy-Werte der einzelnen Klassen pro Korpus

Domäne	Positive Klassifikation				Negative Klassifikation			
	Min		Max		Min		Max	
	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy
LT01-Zehe	CNN	0,00	SVM	52,63	SVM	70,79	CNN, GE	100,00
LT02-Schmidt	CNN	0,00	GE	58,42	SVM	68,92	CNN	100,00
LT03-Schmidt	CNN, RNN, GE	0,00	SVM	22,95	SVM	85,61	CNN, RNN, GE	100,00
MI01-Clematide	CNN	0,00	GB	84,06	BBMUS	80,00	CNN, RNN	100,00
MI02-Wojatzki	NB	35,98	GB	81,46	SVM	92,59	NB	98,40
MI03-Rauh	CNN	0,60	GB	93,09	RNN	68,84	CNN	100,00
NA01-Bütow	CNN	32,53	GB	95,70	RNN	68,45	GB	97,73
NA02-Ploch	MULTI	87,32	CNN, RNN, GE	100,00	CNN, RNN, GE	0,00	GB	68,42
NA03-Schabus	NB, CNN, RNN	0,00	GE	41,86	MULTI	99,25	NB, CNN, RNN	100,00
RE01-Klinger	MULTI	97,63	NB, CNN	100,00	NB, CNN, RNN	0,00	BBMUS	66,00
RE03-Du	NB	81,71	GE	92,16	SVM	84,55	GE	92,71
RE02-Sänger	RNN	86,21	CNN	97,08	CNN	8,97	GB	84,14
RE04-Guhr	SVM	92,18	NB	97,13	NB	64,47	GE	88,98
RE05-Prettenhofer	NB	79,65	GE	94,35	CNN	83,91	GE	93,49
SM01-Cieliebak	RNN	76,86	GB	92,48	NB	44,32	GB, GE	85,32
SM02-Sidarenka	BBMUS	84,05	NB	92,45	NB	35,36	GE	79,67
SM03-Narr	RNN	79,71	GE	94,57	CNN	17,30	GB	83,97
SM04-Mozetič	RNN	78,00	GB	88,42	CNN	63,41	GE	83,89
SM05-Siegel	CNN, RNN	0,00	NB	55,10	SVM, NB	88,79	CNN, RNN, GE	100,00
SM06-Momtazi	RNN	65,11	CNN	96,76	CNN	3,16	GE	71,05

Tabelle 29.: Angabe der besten und schlechtesten Accuracy-Werte innerhalb einer Klasse pro Korpus im dichotomen Fall. Außerdem Angabe des Ansatzes, der diese Werte erzeugte

Domäne	Positive Klassifikation				Negative Klassifikation				Neutrale Klassifikation			
	Min		Max		Min		Max		Min		Max	
	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy	Ansatz	Accuracy
LT01-Zehe	CNN, RNN	0,00	SVM	14,04	CNN	0,00	NB	47,19	NB	50,00	CNN	98,39
LT02-Schmidt	CNN	0,00	GB	47,52	SVM	52,43	GE	90,00	NB	3,03	CNN	62,12
MI01-Clematide	CNN, RNN	0,00	GB	62,32	BBMUS	56,36	CNN	100,00	CNN	0,00	GE	56,04
MI02-Wojatzki	NB	12,62	GB	58,17	RNN	47,15	GB	78,32	BBMUS	83,05	GE	89,50
MI03-Rauh	NB	27,33	GB	77,18	CNN	33,05	GE	76,42	RNN	54,29	GE	74,72
NA01-Bütow	NB	16,13	GB	76,88	NB	25,77	GB	77,11	RNN	81,11	NB	92,76
NA02-Ploch	CNN	0,00	BBMUS	64,79	NB, CNN, RNN, GE, BBMUS	0,00	GB	13,16	BBMUS	96,09	NB, CNN	100,00
NA03-Schabus	NB, CNN, RNN	0,00	GB	20,93	SVM	48,12	GE	67,11	RNN	60,44	GE	77,13
RE01-Klinger	BBMUS	96,05	NB, CNN, RNN	100,00	NB, CNN	0,00	GE	74,00	NB, CNN, RNN, GB, GE	0,00	SVM	5,88
RE02-Sänger	CNN	70,65	GB	81,92	CNN	62,41	GB	73,72	SVM	48,89	GB	64,63
RE03-Du	CNN	55,85	GE	89,42	NB	17,93	GE	72,07	CNN	60,71	GB	87,17
RE04-Guhr	SVM	81,79	NB	93,88	NB	53,89	GE	78,04	NB	33,46	GB	59,48
SM01-Cieliebak	RNN	53,55	GE	77,98	NB	17,73	GB	62,76	CNN	78,49	NB	89,08
SM02-Sidarenka	CNN	69,84	NB	85,85	NB	25,56	GB	69,27	NB	43,94	CNN	79,43
SM03-Narr	NB	39,71	GB	73,71	NB	13,50	GB	73,84	CNN	82,35	NB	91,32
SM04-Mozetič	RNN	45,96	GB	61,13	NB	29,43	GE	56,95	CNN	71,85	RNN	83,01
SM05-Siegel	CNN, RNN	0,00	NB	55,10	NB	87,85	CNN, RNN, GE	100,00	SVM, NB, CNN, RNN,GB, GE, BBMUS	0,00	SVM	0,00
SM06-Momtazi	SVM	77,34	CNN	98,92	CNN	3,16	GE	87,89	NB, CNN, RNN, GB, GE, BBMUS	0,00	SVM	4,55

Tabelle 30.: Angabe der besten und schlechtesten Accuracy-Werte innerhalb einer Klasse pro Korpus im trichotomen Fall. Außerdem Angabe des Ansatzes, der diese Werte erzeugte

Erklärung zur Urheberschaft

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle Zitate und Übernahmen von fremden Aussagen kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.

Regensburg, 13.02.2023

Unterschrift

Erklärung zur Lizenzierung und Publikation dieser Arbeit

Name: Niklas Donhauser

Titel der Arbeit: *Implementierung und Evaluation verschiedener Machine Learning-Ansätze für die Sentiment-Analyse im Deutschen*

Hiermit gestatte ich die Verwendung der schriftlichen Ausarbeitung zeitlich unbegrenzt und nicht-exklusiv unter folgenden Bedingungen:

- ☐ Nur zur Bewertung dieser Arbeit
- ☐ Nur innerhalb des Lehrstuhls im Rahmen von Forschung und Lehre
- ☒ Unter einer Creative-Commons-Lizenz mit den folgenden Einschränkungen:
  - ☒ BY – Namensnennung des Autors
  - ☐ NC – Nichtkommerziell
  - ☐ SA – Share-Alike, d.h. alle Änderungen müssen unter die gleiche Lizenz gestellt werden.

(An Zitaten und Abbildungen aus fremden Quellen werden keine weiteren Rechte eingeräumt.)

Außerdem gestatte ich die Verwendung des im Rahmen dieser Arbeit erstellten Quellcodes unter folgender Lizenz:

- ☐ Nur zur Bewertung dieser Arbeit
- ☐ Nur innerhalb des Lehrstuhls im Rahmen von Forschung und Lehre
- ☐ Unter der CC-0-Lizenz (= beliebige Nutzung)
- ☒ Unter der MIT-Lizenz (= Namensnennung)
- ☐ Unter der GPLv3-Lizenz (oder neuere Versionen)

(An explizit mit einer anderen Lizenz gekennzeichneten Bibliotheken und Daten werden keine weiteren Rechte eingeräumt.)

Ich willige ein, dass der Lehrstuhl für Medieninformatik diese Arbeit – falls sie besonders gut ausfällt - auf dem Publikationsserver der Universität Regensburg veröffentlichen lässt.

Ich übertrage deshalb der Universität Regensburg das Recht, die Arbeit elektronisch zu speichern und in Datennetzen öffentlich zugänglich zu machen. Ich übertrage der Universität Regensburg ferner das Recht zur Konvertierung zum Zwecke der Langzeitarchivierung unter Beachtung der Bewahrung des Inhalts (die Originalarchivierung bleibt erhalten).

Ich erkläre außerdem, dass von mir die urheber- und lizenzrechtliche Seite (Copyright) geklärt wurde und Rechte Dritter der Publikation nicht entgegenstehen.

- ☒ Ja, für die komplette Arbeit inklusive Anhang
- ☐ Ja, für eine um vertrauliche Informationen gekürzte Variante (auf dem Datenträger beigefügt)
- ☐ Nein
- ☐ Sperrvermerk bis (Datum):

Regensburg, 13.02.2023

Unterschrift

Inhalt des beigefügten Datenträgers

`/1_Ausarbeitung`	Die schriftliche Ausarbeitung als PDF und Rohdaten von Overleaf
`/2_Code`	Quellcode der Algorithmen
`/3_Quellen`	Alle in der Arbeit zitierten Quellen im PDF-Format
`/4_Tabellen`	Tabellen als PDF und als Excel-Datei
`/5_Vorträge`	Folien von Antrittsvortrag im PDF-Format