Universität zu Köln

Institut für Digital Humanities

RASSISTISCHE SPRACHE MIT BERT ERKENNEN

—

EINE UNTERSUCHUNG AM BEISPIEL DEUTSCHER PLENARPROTOKOLLE

Judith Nester

Matrikelnummer: 5968232

MA. Informationsverarbeitung

ABSTRACT

Immer wieder kommt es vor, dass in Plenardebatten des Deutschen Bundestages rassistische Sprache verwendet wird. Gerade vor dem Hintergrund der Black-Lives-Matter-Demonstrationen, des rechtsextremistischen Terroranschlags von Hanau und der islamistischen Terroranschläge in Frankreich und Deutschland im turbulenten Jahr 2020 zeigt sich daher verstärkt die Notwendigkeit einer Auseinandersetzung mit rassistischer politischer Sprache. Plenarsitzungen sind meist sehr lang und unübersichtlich. Kaum jemand verfolgt alle Debatten und Reden. Diese zu überblicken und rassistische Sprache zeitnah zu identifizieren und zu kritisieren, erscheint in Anbetracht der großen Menge an Textdaten in Plenarprotokollen geradezu unmöglich. Es benötigt dementsprechend ein Tool, das den Text in Plenarprotokollen verarbeitet, versteht und automatisch rassistische Sprache erkennt.

Eine Möglichkeit für ein solches Tool birgt das Transformer-basierte BERT. Es stellt derzeit den State-of-the-Art im NLP dar. In dieser Arbeit soll evaluiert werden, ob und wie BERT für eine erfolgreiche binäre Textklassifikation zur Identifikation von rassistischer Sprache in Plenarprotokollen eingesetzt werden kann. Dazu erfolgt zunächst eine Auseinandersetzung mit Rassismus und rassistischer politischer Sprache, um jeweils Arbeitsdefinitionen entwickeln zu können. Nach einer Vertiefung in die theoretischen Grundlagen neuronaler Netze über verschiedene Netzarchitekturen wie RNN, LSTM und Transformer hinweg; wird näher auf die Funktionsweisen von BERT eingegangen. Im praktischen Teil der Arbeit werden schließlich auf Basis der festgelegten Arbeitsdefinitionen von Rassismus und rassistischer Sprache zwei möglichst differenzierte Textkorpora erstellt. Mit diesen Korpora werden fünf Experimente durchgeführt, die Aufschluss über die Forschungsfragen geben sollen.

Die Resultate zeigen, dass durchaus Potential für ein BERT-Model besteht, das rassistische Sprache in deutschen Plenarprotokollen identifiziert. Dennoch gibt es noch viele Möglichkeiten das Model zu verbessern. Diese sollten vor einem tatsächlichen Einsatz in der Politik auch genutzt werden.

INHALTSVERZEICHNIS

1 EINLEITUNG

1.1 MOTIVATION

Der Kern der vorliegenden Arbeit liegt in dem Versuch, rassistische Sprache in deutschen Plenarprotokollen zu erkennen. Dazu soll BERT, der momentane State-of-the-Art im Natural Language Processing verwendet werden. Zwei Gegebenheiten motivierten zur vertieften Auseinandersetzung mit dem Thema. Zum einen liegt das Interesse im Jahr 2020 begründet, welches von der Corona-Pandemie und einer Häufung rassistischer Ereignisse und politischer Geschehnisse geprägt war. Beispiele hierfür sind:

- Am 19. Februar 2020 starben elf Menschen in Hanau bei einem rechtsextremistischen Terroranschlag (Tagesschau 2020a).
- In den USA wurde am 25. Mai 2020 der Afro-Amerikaner George Floyd durch einen Polizisten ermordet. Zudem ereigneten sich weitere tragische, rassistisch motivierte Morde, die das *Black-Lives-Matter-Movement* bestärkten (Black Lives Matter o.A.). In Deutschland gab es trotz Pandemie im Frühjahr und Sommer 2020 stark besuchte Demonstrationen zu Black Lives Matter und gegen rassistisch motivierte Polizeigewalt (Zeit 2020).
- Weitreichende Kritik an der deutschen Polizei kam auf, nachdem wiederholt rechte Chatgruppen aufgedeckt wurden, in denen Polizist:innen rechtes Gedankengut geteilt hatten (Tagesschau 2020d).
- Es wurde starke Kritik am polizeilichen Vorgehen nach Krawallen von Jugendlichen in Stuttgart geäußert. Die Polizei hatte geplant Stammbaum und Abstammung der verdächtigen Personen zu erfassen und zu untersuchen (Geuther 2020).
- Bei islamistischen Terroranschlägen in Frankreich am 16. Oktober 2020 in Paris (Tagesschau 2020c) und am 29. Oktober in Nizza (Tagesschau 2020e) sowie in Deutschland am 04. Oktober in Dresden (MDR 2020) starben insgesamt sechs Menschen.
- Bei dem verheerenden Brand im Flüchtlingscamp Moria auf der griechischen Insel Lesbos wurde in der Nacht zum 09. September 2020 das Camp vollständig zerstört (Tagesschau 2020b).

All diese Ereignisse und einige mehr wurden weitreichend in der deutschen Politik diskutiert. Im Bundestag gab es viele Debatten, die sich mit den genannten Ereignissen und deren Folgen beschäftigten. Dabei kam es immer wieder vor, dass von Politiker:innen bei Plenarsitzungen offen rassistische Sprache verwendet wurde. Auch, wenn die Sprecher:innen nach rassistischen Aussagen oft von Öffentlichkeit und anderen Fraktionen verbalen Gegenspruch bekamen, sollte doch gesondert darauf hingewiesen werden, dass überhaupt erst einmal die Möglichkeit, beziehungsweise die Tatsache der Verwendung rassistischer Sprache in der deutschen Politik besteht. Nach Kritik an der Ausdrucksweise einer Politikerin oder eines Politikers wird gerne relativiert, es wäre nicht so gemeint gewesen und man sei falsch verstanden worden. Als Beispiel hierfür kann die Debatte um rassistische Äußerungen des Grünen-Politikers Boris Palmer im Mai 2021 genannt werden. Kritik an seinen Äußerungen wehrte Palmer ab. Er sprach von „Cancel Culture“ und wies den Rassismus-Vorwurf entschieden von sich (Schmid 2021; Tagesschau 2021b). Ebenso wurden mehrfach Politiker:innen der AfD kritisiert, die in Parlamentsdebatten des Bundestags vermehrt mit rassistischen Aussagen auffielen. Meist wurde die darauf folgende Kritik klar von sich gewiesen. So verteidigte Alexander Gauland beispielsweise in der 48. Plenarsitzung der 19. Wahlperiode am 12. September 2018 die polemische Wortwahl seiner Parteigenoss:innen mit folgendem Satz: „Hass ist erstens keine Straftat und hat zweitens in der Regel Gründe“ (Lehmann 2020).

Wie sich eine Wortwahl und Sprache allgemein auf Politik und Gesellschaft auswirken kann, haben auch die Ereignisse um den ehemaligen Präsidenten der U.S.A., Donald Trump, gezeigt. Seine Rede führte am 6. Januar 2020 zur Erstürmung des Kapitols durch aufgestachelte Anhänger:innen. Infolgedessen starben fünf Personen. Noch immer laufen Gerichtsverfahren (Tagesschau 2021a). All diese Ereignisse zeigen die Macht, die hinter politischer Sprache stecken kann. In Kombination mit rassistischer Sprache lässt sich explosives Potential erkennen. Zu rassistischer Sprache schrieben Ludger Hoffmann und Annika Frank:

> Rassismus und Hassrede liefern ideologische Rechtfertigung für Gewalttaten. Zu welchen Konsequenzen bis hin zum Völkermord das führen kann, hat das 20. Jahrhundert gezeigt. (Hoffmann und Frank 2021, S. 27)

Dementsprechend kann über Rassismus in politischer Sprache nicht einfach hinweggesehen werden. Da einzelne Aussagen jedoch schnell in den großen Textmengen langer Plenarsitzungen untergehen können, benötigt es eine sachlich-analytische Betrachtung der Aussagen, um rassistische Sprache zeitnah zu erkennen, zu kritisieren und zur Diskussion

stellen zu können. Gerade im Wahlkampf vor der Bundestagswahl im September 2021 besäße ein solches Tool große Praktikabilität und wäre ausgesprochen hilfreich, um eventuelle rassistische Gesinnungen bei Politiker:innen aufzudecken und den Wähler:innen mehr Transparenz in Hinsicht auf Parlamentsdebatten zu bieten.

Der zweite Motivationshintergrund dieser Arbeit ist die Aktualität des Themas *Natural Language Processing (NLP)* in der Computerlinguistik. In Zeiten von Big Data und immer weiter wachsenden Datenmengen ist NLP ein wichtiges Werkzeug, um Textdaten auszuwerten und automatisiert zu verarbeiten. Noch immer liegen bei NLP die größten Schwierigkeiten im Verständnis von Mehrdeutigkeiten, Metaphern, Synonymen und anderen Eigenheiten von natürlicher Sprache. Hinzu kommt das Erfordernis einer auf den Task angepassten und hochqualitativen Datenbasis, mit der das neuronale Netz trainiert wird. Es finden sich viele Tasks und Use Cases, für die NLP bereits eingesetzt wird, wie beispielsweise die Sprachvervollständigung der Smartphone-Tastatur oder Spamfilter im E-Mail-Postfach. Dennoch eröffnen sich stetig weitere Einsatzmöglichkeiten, auch bedingt durch die schnelle Weiterentwicklung von Architekturen neuronaler Netze. Der momentane State-of-the-Art im NLP ist das Transformer-basierte *BERT (Bidirectional Encoder Representations from Transformers)*. BERT wurde 2017 von Devlin et al. vorgestellt und erzielte bei vielen verschiedenen Tasks hervorragende Ergebnisse, die andere Modelle in Bezug auf das Verständnis der genannten Ambiguitäten noch immer übertreffen (Devlin et al. 2019). Dennoch sind häufig Ergebnisse mit Blick auf Datenbasis und Parametrisierung im Training nicht sofort zu erklären. Folglich gibt es noch viel zu NLP mit BERT zu erforschen und zu testen. Es ist meist die praktische Anwendung, die den größten Aufschluss über hilfreiche Konfigurationen und zielführende Datenbasen gibt.

Mit Blick auf die Problematik der Verwendung rassistischer Sprache in Plenarsitzungen ergaben sich folgende Forschungsfragen:

1. Kann mit BERT rassistische Sprache in deutschen Plenarprotokollen erkannt und korrekt klassifiziert werden?
2. Welche Konfigurationen im Fine-Tuning begünstigen die Ergebnisse des Models?
3. Wie wirkt sich die Datenbasis auf die endgültigen Resultate des Models aus?

Um diese Fragen beantworten zu können, wurden zu Beginn Aufsätze und Arbeiten mit ähnlichen Themen recherchiert. So konnte sich ein guter Überblick über verschiedene Arbeitsansätze im NLP verschafft werden.

1.2 VERWANDTE ARBEITEN

Die Recherche nach Literatur zu ähnlichen Themen wie das der vorliegenden Arbeit verdeutlichte, dass rassistische Sprache selten im expliziten Fokus der Forschungsarbeiten im NLP liegt. Eher befassen sie sich mit umfassenderen Formen von diskriminierender und beleidigender Sprache wie *Hate Speech* oder *Offensive Speech*. Dabei richtet sich die Aggression gegen viele verschiedene Menschengruppen. Hate Speech und Offensive Speech schließen beispielsweise auch Homophobie und Sexismus als Diskriminierungsformen mit ein. Als Datenbasen werden oft Tweets genutzt. Dies liegt vermutlich an der einfach zu handhabenden und frei zugänglichen Schnittstelle von Twitter, die es erlaubt große Mengen von Tweets auszulesen[1]. Zudem ist automatische Hate Speech Detection in sozialen Medien, bedingt durch den rauen Ton, der immer wieder auf Plattformen zu lesen ist, ein wiederkehrendes Thema in Forschung und Medien. Hierbei werden jedoch meist andere Formen neuronaler Netze als das Transformer-basierte BERT verwendet, beispielsweise LSTMs (Zhang und Luo 2018; Pitsilis et al. 2018). Ein Großteil wissenschaftlicher Aufsätze, denen ein dieser Arbeit ähnliches Thema zugrunde liegt, beschäftigen sich mit englischer Sprache. Die in der Recherche gefundene einzig aktuelle Arbeit zu Offensive Speech in deutscher Sprache und deren Klassifizierung durch BERT ist das Paper „Offensive Language Identification using a German BERT model" von Risch et al. (2019). Darin wurden deutsche Tweets mit Hilfe eines deutschen BERT-Models in drei Experimenten klassifiziert. Das erste Experiment („coarse-grained") teilte die Tweets in einem binären Klassifikationstask den Labels OFFENSIVE und OTHER zu. Im zweiten „fine-grained" Experiment wurden die Tweets in drei Kategorien eingeteilt: PROFANITY, INSULT und ABUSE. Im dritten Experiment wurden die als OFFENSIVE gelabelten Tweets in EXPLICIT oder IMPLICIT eingeteilt. Für das Pretraining wurde ein deutsches BERT-Model namens *German BERT* verwendet. Die Ergebnisse der Experimente zeigten, dass BERT erfolgreich für Klassifizierungstasks für Offensive Language in deutscher Sprache eingesetzt werden kann (Risch et al. 2019). Worauf Risch et al. sowie weitere Forscher:innen in ihren Aufsätzen zu NLP wenig eingehen, ist die Sammlung von Daten und Taktik bei der Erstellung der Textkorpora. Zu diesem Aspekt kann der mit „Developing a Multilingual Annotated Corpus of Misogyny and Aggression" betitelte Aufsatz von Bhattacharya et al. (2020) Aufschluss geben. Darin werden ausführlich der Prozess der Datensammlung und dabei auftretende

[1] Näheres zur Twitter API findet sich in folgender Dokumentation: https://developer.twitter.com/en/docs/twitter-api (Stand: 07.07.2021).

Schwierigkeiten, wie beispielsweise Subjektivität in der Auswahl der Daten, beschrieben. Bei der Erstellung des Textkorpus, der für die Experimente dieser Arbeit verwendet wurde, war die Beschreibung der Vorgehensweise von Bhattacharya et al. sehr hilfreich.

Die beiden Paper von Risch et al. und Bhattacharya et al. lieferten Leitfäden, nach denen teilweise vorgegangen wurde. So konnten ihre Erkenntnisse in diese Arbeit einfließen und in Bezug auf deutsche, politische, rassistische Sprache erweitert werden.

2 THEORETISCHER HINTERGRUND

2.1 RASSISTISCHE SPRACHE IN DER POLITIK

Eine linguistisch-grammatische Analyse von Rassismus in der deutschen Sprache gestaltet sich schwierig. So stellen sich Formulierungen je nach Umgebung (privat, beruflich, öffentlich, o.Ä.), Verbindung zwischen den Gesprächspartner:innen (Freunde, Familie, Fremde, o.Ä.) und Kanäle, über die Rassismus verbreitet wird (persönlich von Mensch zu Mensch, in sozialen Medien, etc.), unterschiedlich dar. In sozialen Medien sind immer wieder offen rassistische Beleidigungen zu beobachten. Demgegenüber stehen zurückhaltendere Formulierungen in professionelleren Kontexten, wie in der Arbeitswelt und der Politik. Gerade in letzterem Bereich ist Rassismus in der Sprache selten klar zu erkennen, hängt er doch vom größeren Bezugsrahmen innerhalb der politischen Situation, der Intention der Aussage sowie weiteren Faktoren, wie beispielsweise historischem und zeitgeschichtlichem Kontext ab. Im Folgenden soll ein möglichst klarer Rahmen gezogen werden, mittels dessen rassistische Sprache in der Politik erfasst werden kann. Hierfür soll zuerst genauer auf den Begriff des Rassismus eingegangen und eine Arbeitsdefinition etabliert werden.

2.1.1 Der Rassismus-Begriff

Es wurde oft versucht, Rassismus klar zu definieren, doch mangelt es bisher an einer universalen Definition. Die Fülle wissenschaftlicher Abhandlungen zu diesem Thema ermöglicht es allerdings, einen gewissen Konsens zu finden, der eine recht eindeutige Erklärung der Bedeutung liefert.

Die Einteilung von Menschen in sogenannte ‚Rassen', und damit in Kategorien, ist eine der bekanntesten Intentionen, die Rassismus konstituieren. Diese Kategorisierung impliziert

eine Hierarchie innerhalb des Sets der Rassen und damit eine Bewertung, die eine Rasse höher als eine andere wertet und folglich als überlegen darstellt. Dabei werden der einen Rasse positive Eigenschaften (hohe Intelligenz, Fleiß, o.Ä.) und der zweiten Rasse negative Eigenschaften (niedrige Intelligenz, Triebhaftigkeit, o.Ä.) zugeschrieben. Eine solche Klassifizierung dient letztlich der Absicht, klare Grenzen zwischen Menschengruppen zu ziehen. Sie soll „zu einer Vereinfachung faktischer Vielfalt und Unübersichtlichkeit verhelfen“ (Kimmich et al. 2016, S. 19). Meist basiert eine rassistische Abgrenzung auf der pseudowissenschaftlichen, veralteten Ansicht, die Menschheit wäre biologisch oder genetisch bedingt in Rassen einteilbar. Dass dies definitiv ein Trugschluss ist, ist wissenschaftlich zweifelsfrei belegt (Fischer et al. 2019; Hoffmann und Frank 2021, 18f.). Kimmich et. al. schreiben hierzu:

> „‚Rasse‘ ist keine reale Gegebenheit, aufgrund derer sich unterschiedliche Formen von Rassismus entwickeln können; sondern umgekehrt ist ‚Rasse‘ das Ergebnis von Diskursen der Klassifikation und Hierarchisierung, die ihre Ordnungskriterien als empirisch ausweisen und damit behaupten, deskriptiv und messend natürliche Gegebenheiten darzustellen.“ (Kimmich et al. 2016)

Auch in der Jenaer Erklärung von 2019, einem Aufsatz des Vorstands der Deutschen Zoologischen Gesellschaft und des Präsidenten der Friedrich-Schiller-Universität Jena, wird deutlich auf die gesellschaftliche Eigenkonstruktion von menschlichen Rassen hingewiesen:

> Die Einteilung der Menschen in Rassen war und ist zuerst eine gesellschaftliche und politische Typenbildung, gefolgt und unterstützt durch eine anthropologische Konstruktion auf der Grundlage willkürlich gewählter Eigenschaften wie Haar- und Hautfarbe. Diese Konstruktion diente und dient eben dazu, offenen und latenten Rassismus mit angeblichen natürlichen Gegebenheiten zu begründen und damit eine moralische Rechtfertigung zu schaffen. (Fischer et al. 2019)

Die Widerlegung menschlicher Rassen hat jedoch nicht zum Ende des Rassismus geführt. Vielmehr verschiebt sich der Fokus auf kulturelle und religiöse Unterschiede als Grundlage für Ausgrenzung und Diskriminierung. Es erfolgt also die generelle Einteilung in Menschengruppen nicht mehr allein anhand biologischer Merkmale, wie beispielsweise Hautfarbe, Körperbau oder Haarstruktur, sondern es werden vielmehr weitere Kategorien, wie Traditionen, Religion und Herkunft herangezogen, zu denen dann Stereotypen erzeugt werden, um die eigene Gruppe über die der ‚Anderen‘ zu erheben. Alikhani und Rommel vertreten in ihrem Aufsatz die These, der „nicht tabuisierte aber undifferenzierte Begriff ‚Kultur‘“ enthalte „dieselben wesenhaften Merkmale des Begriffs ‚Rasse‘“, woraus eine Art „kultureller Rassismus“ resultiere (Alikhani und Rommel 2018, S. 9). Zu ihrem Schluss

kommen Alikhani und Rommel anhand der Analyse und des Vergleichs von Samuel P. Huntingtons „Clash of Civilizations“ und Thilo Sarrazins „Deutschland schafft sich ab“. Die Autoren beider Bücher, Huntington in den USA und Sarrazin in Deutschland, kämen jeweils zum gleichen Ergebnis: Der Ursprung aller Konflikte liege im Unterschied der verschiedenen ‚Kulturen‘ sowie der angeblichen Verweigerung von Immigranten sich zu integrieren und an die westliche Gesellschaft anzupassen. Die Gruppe, auf die Huntington und Sarrazin meist abzielen, sind Muslime. Sarrazin lasse die Problematik der Diskriminierung aufgrund kultureller und religiöser Merkmale als weiteres Erklärungsmuster neben der Genetik zur klaren Abgrenzung von Menschengruppen außer Acht. Vielmehr sehe er „zusammenfassend ‚die deutsche Kultur‘ durch die Immigration von Muslimen als gefährdet an“ (Alikhani und Rommel 2018, S. 17). Sarrazin schafft also ein Feindbild der ‚integrationsunwilligen Muslime‘ und stellt die Minderheit als „alleinigen Aggressor der Mehrheit“ und „Auslöser für Konflikte“ dar (Alikhani und Rommel 2018, 18ff.). Auch der Philosoph Étienne Balibar beschäftigt sich mit einem ‚Rassismus ohne Rassen‘ (Balibar 2016, S. 30). 1989 veröffentlichte er den Text „Gibt es einen ‚Neo-Rassismus‘?“, in dem er von einer neuen Form von Rassismus sprach. Diese grenze sich von bisherigen Rassismus-Modellen ab. Balibar verweist auf den Zusammenhang der Immigration von Gastarbeitern und Xenophobie:

> Die Art und Weise, wie die Kategorie der Immigration als Ersatz für den Begriff der Rasse und damit als Agens einer Zersetzung des ‚Klassenbewußtseins“ funktioniert, liefert uns hierfür einen ersten Hinweis. [...] Schon seit langem sind die kollektiven Zusammenhänge der Arbeitsimmigranten Diskriminierungen und fremdenfeindlichen Gewalttätigkeiten ausgesetzt, die ihrerseits von den Stereotypen des Rassismus durchdrungen sind. (Balibar 2016, S. 29)

Weiter schreibt er:

> Ideologisch gehört der gegenwärtige Rassismus, der sich bei uns um den Komplex der Immigration herum gebildet hat, in den Zusammenhang eines ‚Rassismus ohne Rassen‘ [...]: eines Rassismus, dessen vorherrschendes Thema nicht mehr die biologische Vererbung, sondern die Unaufhebbarkeit der kulturellen Differenzen ist; eines Rassimus, der - jedenfalls auf den ersten Blick - nicht mehr die Überlegenheit bestimmter Gruppen und Völker über andere postuliert, sondern sich darauf ‚beschränkt‘, die Schädlichkeit jeder Grenzverwischung und die Unvereinbarkeit der Lebensweisen und Traditionen zu behaupten. (Balibar 2016, 30f.)

Ludger Hoffmann, Professor emeritus für deutsche Sprache an der Technischen Universität Dortmund, fasst in seinem Aufsatz „Zur Sprache des Rassimus“ von 2020 den Rassismus-Begriff so zusammen, dass er sowohl den biologisch begründeten, wie auch den kulturellen Rassismus umfasst und zieht historische und politische Parallelen, die seine Definition

unterstreichen. Als grundlegendes Konzept von Rassismus nennt er die Aufteilung in *Die-* und *Wir-Gruppen*. Die Die-Gruppe müsse sich dabei klar von der Wir-Gruppe unterscheiden. Dazu würden Eigenschaften als Indikatoren herangezogen, die unveränderlich und leicht identifizierbar (Hautfarbe, Gestalt, Kopfform, o.Ä.), sowie nicht sichtbar (Herkunft, Blut, Gene, Traditionen) sein können. Die Form von Rassismus, die beide Indikatoren verwendet, bezeichnet Hoffmann als *pseudowissenschaftlichen Rassismus*, der folglich kulturelle und religiöse Unterschiede als Indikatoren mit einschließt. Hoffmann zieht als Beispiel die politische Gruppe der ‚Identitären' heran. Deren Ansicht nach sei „jede Kultur [...] eine an Ort und Region gebundene Lebensform, die sich durch Homogenität ihrer Angehörigen auszeichnete. Jede Mischung sei für die aufnehmende Kultur schädlich und Migration daher abzulehnen" (Hoffmann 2020, S. 41).

Eine Arbeitsdefinition, die möglichst alle der genannten Felder abdeckt, muss also im Wesentlichen die folgenden drei Merkmale aufweisen:

1. Eine Gesellschaft wird in Gruppen eingeteilt. Dabei versteht sich eine Gruppe als Norm und stellt sich über eine andere.
2. Die hergestellte Hierarchie kann pseudowissenschaftlich begründet und so das Konzept von ‚Rasse' konstruiert werden.
3. Rassismus wird sowohl aufgrund biologischer Merkmale als auch kultureller und religiöser Unterschiede indiziert.

2.1.2 Rassistische Sprache in Plenardebatten

Im vorherigen Kapitel wurde zusammengefasst, wie sich biologischer und kultureller Rassismus im Allgemeinen darstellen. Nun soll erarbeitet werden, wie sich Rassismus in Sprache manifestiert und in der Politik verwendet wird. Im Sinne dieser Arbeit steht dabei die rassistische Sprachverwendung in Plenarsitzungen des Bundestags im Fokus. Dazu sollte zuerst grundsätzlich politische Sprache betrachtet und deren Intention, beziehungsweise strategische Verwendung, diskutiert werden. Diese umfassen ebenso die Beanspruchung der Wahrheit der eigenen Aussage sowie der Überzeugung der Adressaten und der Meinungsbildung. Der Politolinguist Thomas Niehr verweist in diesem Zusammenhang auf die realitätskonstruierende Funktion von Sprache im Allgemeinen: „Einen direkten Zugang zur Realität haben wir nicht, sondern nur vermittels unserer Sprache. Mit den Ausdrücken unserer Sprache bezeichnen wir die Dinge der Welt" (Niehr 2014, S. 14). Es ist aber nicht die gesamte Welt, die so be- und gezeichnet wird, sondern die Welt aus dem subjektiven Blickwinkel des Sprechers oder der Sprecherin. Dazu passend schreibt der

Sprachwissenschaftler Heiko Girnth „[…] Das menschliche Denken [ist] grundsätzlich ideologiegebunden. Ideologischer Sprachgebrauch ist Ausdruck prinzipieller Seinsgebundenheit des Denkens“ (Girnth 2015, S. 4). Daraus folgt eine generelle Subjektivität von Sprache und, im Kontext von Politik, eine „Verflechtung von politischem und ideologischem Sprachgebrauch“ (Girnth 2015, S. 3). In einigen Politikbereichen, beispielsweise der Kommunikation zwischen politischen Systemen und zu politischen Prozessen (Gesetzestexte, Geschäftsordnungen, o.Ä.), steht weniger der strategisch-intentionale Sprachgebrauch zur Durchsetzung von Interessen im Mittelpunkt. Im Gegensatz dazu stehen Plenarsitzungen und Parteiprogramme. Um diesem Gegensatz Rechnung zu tragen, bietet sich eine begriffliche Unterscheidung zwischen Darstellungs- und Entscheidungspolitik an. Während letztere zu informationsgebenden und -manifestierenden Zwecken eingesetzt und in der Regel von der Öffentlichkeit ausgeschlossen stattfindet, geht es in der Darstellungspolitik – wie der Name schon sagt – um die öffentliche Darstellung der eigenen Ansichten und die Beeinflussung der Rezipienten (Girnth 2015, S. 41). Diekmann zieht hier Parallelen zur Werbesprache (Dieckmann 1975, S. 27)[2]. Die Rezipient:innen von politischer Sprache sind je nach Situation unterschiedlich.

Plenarreden werden geplant und im Vorfeld verfasst. Die Länge der Redezeiten sind abhängig von der Fraktionsgröße (Deutscher Bundestag 1980). Dementsprechend handelt es sich bei den Debatten selten um spontane Diskurse und Reaktionen auf vorherige Reden, sondern um die geplante Vermittlung des Standpunkts der Fraktion. Wie die Abgeordneten abstimmen, steht meist bereits vor den Debatten fest. Die wahre Zielgruppe der Reden und Debatten sind demgemäß nicht die anderen Abgeordneten, sondern die Konsument:innen der berichtenden Massenmedien, die es zu überzeugen gilt. In Bezug auf Plenardebatten kann insofern die Aussage getroffen werden, dass diese eher der Darstellungspolitik als der Entscheidungspolitik zuzuordnen sind. Dennoch kann nicht von leicht zu beeindruckenden, rein auf Konsum und Aneignung von Meinungen ausgerichteten Rezipient:innen ausgegangen werden. Politiker:innen im Allgemeinen, aber vor allem Beteiligte an Plenardebatten, befinden sich durch ihren Stand in der Öffentlichkeit unter deren ständiger Kontrolle, was wiederum über die Wähler:innen, als Rezipient:innen des Gesagten zur Gestaltung der deutschen Politik beiträgt (Amri-Henkel 2021, 106f.).

[2] Bezogen auf den realitätskonstruierenden Charakter von politischer Sprache stellt sich die Frage der Auswirkungen auf Rezipient:innen und der Gesellschaft in ihrer Gesamtheit. Girnth sieht hier eine Wechselwirkung zwischen Sprache und gesellschaftlicher Wirklichkeit. Letztere manifestiere sich in Sprache, würde aber erst durch Sprache ermöglicht und hergestellt (Girnth 2015, S. 6).

Befasst man sich mit Plenarprotokollen und darin enthaltenem Rassismus, ist festzustellen, dass sich keine allumfassenden, linguistisch-grammatischen Regeln zum Aufbau rassistischer politischer Sprache festlegen lassen, wie beispielsweise ein gängiger Satzbau oder fixes Vokabular. Oft sitzt der Rassismus tief in einer Rede versteckt und ist ohne Betrachtung des semantischen Kontextes schwer zu entdecken. Dies liegt meist auch an der Implizitheit vieler Äußerungen, die in einer direkteren Form, wie sie oft auf Social-Media-Plattformen gefunden werden kann, für die Darstellungspolitik zu explizit wären und als unprofessionell erachtet werden könnten. Dennoch sind in Plenarprotokollen immer wieder tief polemische Äußerungen zu lesen. Ludger Hoffmann und Annika Frank von der TU Dortmund liefern in ihrem Aufsatz „Zur Pragmatik rassistischer Beleidigungen“ [3] grundlegende Ansätze, mit denen dennoch Analysen rassistischer Aussagen erfolgen können. Dazu fassen die Autor:innen zuerst zusammen, wie sich rassistische Äußerungen generell darstellen. Sie verweisen darauf, dass rassistische Sprache „Merkmale eines Bildes“ besitzt und „den Eindruck erweckt, Wirklichkeit holistisch abzubilden und auch Prognosen über künftige Entwicklungen zu erlauben“ (Hoffmann und Frank 2021, S. 14). Die sich äußernde Person formuliert die rassistische Aussage in einer Art und Weise, als sei sie wahr und belegbar, ohne dass tatsächlich eine Kontrolle des Wahrheitsgehalts vorgenommen wurde oder überhaupt vorgenommen werden kann.

Im Folgenden sollen Beispiele rassistischer Sprache in Plenarprotokollen analysiert und unter Beachtung ihres politischen Kontextes untersucht werden.

Tabelle 1 spricht die Rednerin von einer „faktisch unbegrenzten und unkontrollierten Einwanderung“ und beansprucht dabei die Wahrheit dieser Aussage für sich, indem sie sie mit dem Adjektiv „faktisch“ (Faktum, Tatsache) unterstreicht. Es werden keine Daten oder Informationen geliefert, die diese Äußerung belegen. Sie erzeugt eine klar definierte Menschengruppe („[…] jungen Männern vornehmlich aus dem islamisch-orientalischen Kulturkreis […]“) und spricht ihr ein unbelegtes „kritisches Potenzial“ zu. Dabei suggeriert sie einen Zusammenhang zwischen Religion und Herkunft und einer davon angeblich ausgehenden Gefahr. Zuletzt wird der zuvor erstellten Menschengruppe eine „mehr oder minder offen[e] Verachtung“ gegenüber dem „deutschen Staat und [der] Mehrheitsgesellschaft“ unterstellt und damit eine nicht belegbare Allaussage getroffen.

[3] Die Autor:innen haben freundlicherweise ihr unveröffentlichtes, finales Manuskript für diese Arbeit zur Verfügung gestellt. Der Aufsatz soll 2021 in „Sprache/n, Institutionen und mehrsprachige Gesellschaften“ im Waxmann-Verlag erscheinen. Herausgeberin ist Christiane Hohenstein.

1	Faktisch unbegrenzte und unkontrollierte Einwanderung hat dazu geführt, dass es in vielen deutschen Großstädten inzwischen ein kritisches Potenzial an jungen Männern vornehmlich aus dem islamisch-orientalischen Kulturkreis gibt, die den deutschen Staat und die Mehrheitsgesellschaft mehr oder minder offen verachten. (Dr. Alice Weidel, AfD; 171. Plenarsitzung; 03.07.2020)
2	Die Bürger wissen aus eigener Erfahrung: Auch in der zweiten und dritten Generation legt eine beachtliche Zahl deutscher Staatsbürger mit Migrationshintergrund mangelnde Bereitschaft oder Fähigkeit zur Integration an den Tag. (Martin Hess, AfD; 189. Plenarsitzung; 05.11.2020)
3	Meine Damen und Herren, es ist abzusehen, wann Mohammed das Rennen auch in Deutschland endgültig gewinnen wird. Anstatt dass die Familienministerin gegen diesen tatsächlich von Menschen gemachten Bevölkerungswandel vorgeht, bejubelt sie, dass, wie bis vor Kurzem, jeden Freitag von deutschen Kindern, die im eigenen Land immer weniger werden, gegen eine vermeintlich menschengemachte CO2-Krise angehüpft wurde. (Frank Pasemann, AfD; 160. Plenarsitzung; 14.05.2021)
4	Da, wo der Islam zu Hause ist, gibt es keine Freiheit, keine Demokratie, keinen Rechtsstaat, der unseren Vorstellungen auch nur annähernd gleichkommt. Stattdessen findet man dort Intoleranz, Frauenfeindlichkeit, Hass auf Homosexuelle, Christenverfolgung, Steinigung, Enthauptung und beispiellose Grausamkeit gegenüber Tieren. (Jens Maier, AfD; 190. Plenarsitzung; 06.11.2020)
5	Ertüchtigen Sie doch lieber die Gesundheitsämter, größere Infektionsherde zu lokalisieren: Superspreader-Events in der großstädtischen Erlebnisszene, oft migrantisch geprägt. (Dr. Gottfried Curio, AfD; 189. Plenarsitzung; 05.11.2020)

Tabelle 1: Beispiele rassistischer Äußerungen in Plenarsitzungen aus 2020.

Tabelle 1 geht die Erzeugung eines Bildes einer Menschengruppe noch tiefer. Der Redner fasst alle „deutschen Staatsbürger mit Migrationshintergrund“ zusammen und schreibt ihnen eine „mangelnde Bereitschaft oder Fähigkeit zur Integration“ zu. Der Aussage des Redners

nach scheint eine tatsächliche Integration vom jeweiligen Individuum gar nicht erst gewollt. Es werden somit klare Grenzen erzeugt, die deutsche Staatsbürger in solche ohne und solche mit Migrationshintergrund einteilen. Das Bild der/des integrationsunwilligen Deutschen mit Migrationshintergrund entsteht und wird direkt den Bürger:innen als Adressat:innen vermittelt („Die Bürger wissen aus eigener Erfahrung […]"). Hoffmann und Frank betonen die Eignung solcher geschaffener Bilder, unhinterfragt von Menschen geglaubt und ins eigene Wissen übernommen zu werden. Dies sei besonders in den sozialen Medien zu beobachten. Implizite und explizite Generalisierungen sowie Hassreden lieferten bestimmten Gruppen „Voraussetzungen und normative Bezugspunkte sowie ein Formelrepertoire" (Hoffmann und Frank 2021, S. 14), auf das sie sich stützen können. Meist würden rassistische Aussagen pseudowissenschaftlich verargumentiert, indem sich auf „Schein-Evidenzen", „Pseudo-Schlüsse", „unkundige Messungen" und „Mythologisierungen" bezogen würde (Hoffmann und Frank 2021, 14f.).

Die bereits in den ersten beiden Beispielen aufgezeigten Generalisierungen nennen Ludger Hoffmann und Annika Frank als weiteres Merkmal rassistischer Sprache. Dabei wird, oft auch unter Verwendung eines bestimmten Artikels im Singular, von einem Individuum oder Menschenbild der sich äußernden Person auf eine ganze Menschengruppe geschlossen (z.B. ‚der Moslem', ‚der Schwarze'). Diese Gruppe erscheint somit als Einheit einer Person, der negative Eigenschaften und Absichten zugewiesen werden können wie sonst einem einzelnen Individuum. Hoffmann und Frank nennen die Verwendung eines „genuin individualisierenden Personennamens" (Hoffmann und Frank 2021, S. 20) als Steigerung dieser Praktik. Zu beobachten ist diese Praktik in Beispiel 3 (*Tabelle 1*). Im Kontext der gesamten Rede ist das Zitat in eine Debatte um die deutsche Familienpolitik eingebettet. Der Redner nutzt seine Redezeit, um auf Mohammed, einen weltweit gängigen Vornamen, einzugehen, der angeblich über die letzten Jahre hinweg auch im deutschen Ranking beliebter Vornamen aufgestiegen sei. Er bedient sich rassistischer Generalisierungen, indem er einen Vornamen als Sinnbild für eine ganze Menschengruppe verwendet und ihn negativ auflädt, zielt seine Rede doch darauf ab, auf einen vermeintlichen „Bevölkerungswandel" („[…] deutschen Kindern, die im eigenen Land immer weniger werden […]") hinzuweisen. Die angesprochene Menschengruppe scheint Muslim:innen, Migrant:innen und Deutsche mit Migrationshintergrund gleichermaßen zu umfassen und das unbestimmte Ziel zu haben „das Rennen auch in Deutschland endgültig" zu gewinnen, womit der Redner vermutlich auf den angeblichen Bevölkerungswandel und eine Verdrängung Deutscher durch die ebendiese Gruppe anspielt. Eine solche Unterstellung ist dementsprechend klar in die Kategorie eines religiös-kulturellen Rassismus einzuordnen.

Im nächsten Beispiel (Beispiel 4 in *Tabelle 1)* lässt sich der religiöse Bezug direkt zu Beginn des Zitats erkennen („Da, wo der Islam zu Hause ist […]"). Es folgen zwei Aufzählungen von Substantiven, die beide den Islam beschreiben sollen. Erstere Aufzählung spricht dem Islam in der Form, die „unseren Vorstellungen auch nur annähernd gleichkommt", positiv gewertete Begriffe, wie „Freiheit", „Demokratie", und „Rechtsstaat", pauschal ab. Letztere Substantivaufzählung schreibt allen Angehörigen der muslimischen Religion negative Charakteristika und Praktiken („Intoleranz, Frauenfeindlichkeit, Hass auf Homosexuelle, Christenverfolgung, Steinigung, Enthauptung und beispiellose Grausamkeit gegenüber Tieren") zu. Es lassen sich eine starke Verallgemeinerung sowie unbelegte Behauptungen mit religiös-rassistischer Intention erkennen.

Gelegentlich werden rassistische Bezüge auch an Satzenden oder in Nebensätzen aufgeführt. Sie finden auch in Debatten ohne jegliche Verbindung zu potenziell rassistisch aufgeladenen Themen Einzug. Feststellbar ist dies in Beispiel 5 aus Tabelle 1, in dem der Redner in einer Rede zur derzeitigen Lage in der Covid-19-Pandemie Parallelen zwischen „Superspreader-Events in der großstädtischen Erlebnisszene" und einer dortigen vermeintlichen „migrantischen Prägung" zieht. Er suggeriert somit eine kausale Verbindung zwischen beiden angeblichen Tatsachen. Die rassistische Aussage wird erst am Ende des Satzes in einem kurzen Nebensatz getroffen und auch im Rest der Rede nicht weiter ausgeführt.

Generell lässt sich anhand der Analyse der genannten Beispiele zusammenfassen[4], dass rassistische Aussagen in Plenarsitzungen

1. die Wahrheit der Aussage ohne Belege für sich beanspruchen.
2. generalisieren und negativ aufgeladene Begriffe und Sachverhalte auf ganze Menschengruppen beziehen, indem unbelegbare Allaussagen getroffen werden.
3. in den gesamten Kontext der Rede, gegebenenfalls aber auch in einen politischen Kontext (aktuelles politisches Geschehen) gestellt werden müssen.
4. gegebenenfalls nur kurz an Satzenden und in Nebensätzen getroffen und nicht weiter ausgeführt werden.
5. meist dem religiös-kulturellen Rassismus zuzuordnen sind.

[4] Für die Zusammenfassung der in den Analysen erarbeiteten Kriterien von rassistischer Sprache in Plenarprotokollen kann aufgrund der fehlenden, eindeutigen Definitionen von Rassismus und rassistischer Sprache kein Anspruch auf Vollständigkeit erhoben werden.

2.2 NLP MIT KÜNSTLICHEN NEURONALEN NETZEN

Das menschliche Gehirn kann Informationen und Reize in unglaublicher Geschwindigkeit verarbeiten. Im Alltag fällt dies meist nicht auf, ist es doch für die meisten Menschen vollkommene Normalität. Eine schnelle Auffassungsgabe verdanken wir einer der kleinsten Einheiten unseres Nervensystems, dem Neuron. In einem menschlichen Gehirn befinden sich circa 86 Milliarden Neuronen, die jeweils mehr als zehntausend Verbindungen mit anderen Neuronen bilden können (Cappy 2020, S. 48). Ein Neuron nimmt stetig Informationen auf und sammelt diese, bis ein elektrischer Impuls ausgelöst wird, der es aktiviert. Daraufhin wird die Information verarbeitet und an das nächste Neuron weitergeleitet. *Künstliche neuronale Netze* (*KNN*) machen sich ebendiesen Mechanismus der Aktivierung bei Überschreiten eines Schwellwerts sowie weitere Funktionsweisen des menschlichen Nervensystems zunutze und entwickeln sie weiter (Weidman und Lang 2020, 78f.) [5]. Auch die komplexesten Informationen und größten Datenmengen können so verarbeitet werden. KNNs sind bereits ein grundlegender Bestandteil des alltäglichen Lebens. Sie verbreiten sich durch Forschung und Weiterentwicklung in immer mehr Lebensbereichen. Vom autonomen Fahren über Spamfilter zu Autofokussierung in Handykameras gibt es die verschiedensten Einsatzmöglichkeiten.

Das Einsatzgebiet, das für diese Arbeit näher betrachtet werden soll, ist *Natural Language Processing (NLP)*. Möchte man NLP definieren, so beginnt man am besten erst einmal mit einer einfachen Übersetzung, lässt sich doch ein grundlegendes Verständnis aus dem recht selbsterklärenden Begriff ableiten: die Verarbeitung natürlicher Sprache. Dennoch ist es natürlich nicht so einfach, wie es scheint und eine tiefergehende Beschäftigung mit diesem Gebiet zeigt schnell, wie groß und weit die Thematik ist. Wissenschaftlich lässt sich NLP zwischen Linguistik, Informatik, Data Science, aber auch Psychologie und Neurowissenschaften verorten. Anwendungsbeispiele sind das automatisierte Beantworten von Fragen, Textzusammenfassung und -generierung, Übersetzung, Part-of-Speech-Tagging und automatische Spracherkennung. Darüber hinaus gibt es noch weitaus mehr Anwendungsbeispiele für NLP, die teilweise ganz eigene Forschungsgebiete darstellen[6].

[5] Künstliche neuronale Netze orientieren sich an ihrem biologischen Vorbild, dennoch sind nicht alle Funktionen im Machine Learning auch biologisch fundiert. Ein Beispiel hierfür ist die Backpropagation (vgl. Bengio et al. 2015; Ertel 2016, 301f.Ertel 2016, 301f., 308f.).

[6] Einen sehr guten Überblick über den State-of-the-Art von NLP-Tasks bietet die Webseite nlpprogress.com, die von dem Deep-Learning- und NLP-Forscher Sebastian Ruder gehostet wird.

Der Fokus dieser Arbeit liegt auf dem computerlinguistischen Teil von NLP, der sich auf automatisierte, maschinelle Verarbeitung von natürlicher Sprache spezialisiert, welche auf viele verschiedene Arten angegangen werden kann. Ein Weg ist die Verwendung von hartkodierten Strings und Regular Expressions, welche allerdings nur bis zu einem gewissen Grad hilfreich sind (Zhang und Teng 2021, 4ff.), was verständlich wird, wenn man sich vor Augen führt, wie kompliziert Sprache - im Sinne dieser Arbeit die deutsche Sprache - sein kann. Synonyme (z.B. Auto, Wagen, Fahrzeug), Mehrdeutigkeit (z.B. die Birne als Obst und in der Lampe), Wortanalogien (z.B. Zahl – Ziffer, Wort – Buchstabe), Metaphern, Konnotationen, sprachenabhängige Syntax und Semantik, aber auch Schreibfehler, Umgangssprache, Ironie sowie Social-Media-Sprache und Emojis sind nicht mehr mit festen Regeln zu erfassen und zu verarbeiten, sondern erfordern ein gewisses Textverständnis. Es sammeln sich schließlich unzählig viele zu beachtende Faktoren an, die in ihrer ungeheuren Menge die Rechenzeit exponentiell in die Höhe treiben können. Ab diesem Punkt benötigt es einen intelligenten, lernfähigen Ansatz wie Deep Learning als Teilgebiet von Künstlicher Intelligenz, um solche Tasks zu lösen. Dazu werden die Eingabedaten mitsamt aller Merkmale auf Vektoren abgebildet, um die Dimensionen des Inputs zu verringern. Über ein Netz mit einer hohen Anzahl an Schichten wird so eine tiefgehende Abstraktion zwischen Input und Output erreicht, die dazu beiträgt, die Rechenzeit um ein Vielfaches zu reduzieren (Ertel 2016, 300f.).

Der Task, der in dieser Arbeit näher betrachtet wird und praktische Anwendung findet, ist der der Textklassifikation. Das bedeutet, dass unter Anwendung von NLP ein Satz, Textabschnitt oder Dokument einer bestimmten Kategorie, im Falle dieser Arbeit *rassistisch* (RACISM) oder *nicht rassistisch* (OTHER), zugeordnet werden soll. Die Textklassifikation lässt sich dem Task des *Information Retrieval* zuordnen, also der Informationsgewinnung aus ungeordneten Daten (Zhang und Teng 2021, 19f.). Zur Klassifizierung von Texten benötigt es ein gewisses semantisches Verständnis ihres Inhalts. Die Bedeutung von Worten ergibt sich erst aus dem Kontext, in dem sie stehen. Für ein umfassendes Verständnis von Beispielsätzen und der darin enthaltenen Wörter müssen diese also zueinander sowie zum gesamten Text in Kontext gesetzt werden. Folglich zählt für eine erfolgreiche automatisierte Erfassung eines Textes der gesamte Text, aber auch einzelne Worte und Wortsequenzen und deren Beziehungen zueinander. Kompa et al. schreiben hierzu in ihrem Paper „Sprache, sprachliche Bedeutung, Sprachverstehen und Kontext“:

> Wir interpretieren die Äußerungen anderer mit Rückgriff auf erworbenes semantisches Wissen, gleichzeitig aber auch im Lichte kontextueller Interessen und Ziele und vor dem Hintergrund gewisser (geteilter) Annahmen und Erwartungen.

> Entsprechend kommt eine Schlüsselrolle in diesen Betrachtungen dem Begriff des Kontexts (Hintergrunds) zu. (Kompa et al. 2013, S. 11)

Um zu verstehen, wie mit künstlichen neuronalen Netzen natürliche Sprache verarbeitet werden kann, soll in den folgenden Kapiteln näher auf spezielle Formen von KNN eingegangen werden, nämlich *Feedforward-Netze*, *rekurrente Netze* und *Transformer*.

2.2.1 Feedforward-Netze

Wie bereits festgestellt, basiert ein künstliches neuronales Netz auf seiner kleinsten Recheneinheit. Diese erhält, wie sein biologisches Vorbild, das Neuron, Inputs, verarbeitet diese und gibt anschließend einen Output aus. Zuerst werden die Inputs $x_1 \ldots x_n$ mit Gewichtungen $w_1 \ldots w_n$ multipliziert und ein Bias-Wert b addiert (Jurafsky und Martin 2014, 124ff.).

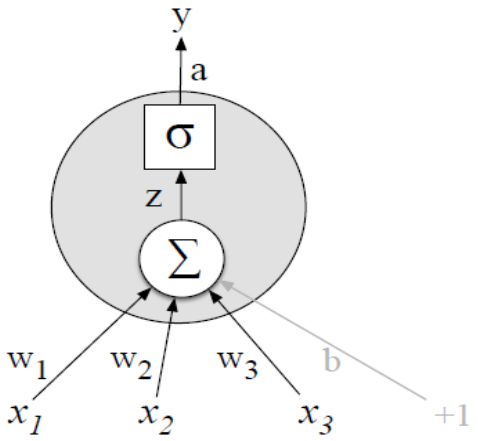

Abbildung 1: Eine Neuron mit drei Input-Werten x_1, x_2 und x_3, jeweils einer Gewichtung pro Input und einem Bias-Wert b. Durch Verarbeitung dieser Werte durch eine Aktivierungsfunktion wird der Output-Wert y erzeugt (Jurafsky und Martin 2014, S. 125).

Weights und Bias-Werte ermöglichen es, den Input so zu beeinflussen, dass der Output letztendlich einen für den Einsatz des Netzes sinnvollen Wert ergibt. So braucht es für manche Tasks, beispielsweise Predictions, einen beliebigen, reellen Wert und für andere, beispielsweise Klassifizierungstasks, einen eingegrenzten Output, der klar erkennen lässt, welcher Klasse der Input zuzuordnen ist (Jurafsky und Martin 2014, 130f.). Das Ergebnis z wird durch eine Aktivierungsfunktion geleitet. Im Beispiel von *Abbildung 1* handelt es sich um die Sigmoid-Funktion σ. Welche Aktivierungsfunktion verwendet werden soll, wird je nach Topologie, also der Struktur des Netzes, sowie der Art des Tasks, der mit Hilfe des Netzes gelöst werden soll, entschieden. Sie können sowohl linear als auch nicht-linear sein.

Die hier verwendete Sigmoid-Funktion ist nicht-linear und bildet das Ergebnis *a* einer Schicht des Netzes in einem Bereich zwischen 0 und 1 ab (Weidman und Lang 2020, S. 77). Handelt es sich um den endgültigen Ausgabewert des Netzes, so entspricht *a* dem Ausgabewert des gesamten Netzes *y*. Fasst man jeweils die Eingabewerte und Gewichtungen zu Vektoren zusammen, so lassen sich die beschriebenen Zusammenhänge mit folgender Formel beschreiben:

$$y = a = \sigma\,(w * x + b)$$

Indem mehrere dieser einzelnen Neuronen als Einheiten in einem Netz miteinander verbunden werden, kann über mehrere Schichten hinweg ein Lernprozess erzielt werden. Ein mehrschichtiges Netz, das die Outputs ausschließlich an die nächsthöhere Schicht weiterleitet und nie zurück zur niedrigeren Schicht, nennt man Feedforward-Netz oder auch *Multilayer-Perceptron (MLP)*. Es stellt die einfachste Form von neuronalen Netzen dar. Sein Aufbau besteht aus einer Input-Schicht, versteckten Schichten (sogenannte Hidden Layers) und einer Output-Schicht. Die einzelnen Schichten bestehen dementsprechend aus Input-Units, Hidden Units oder Output Units.

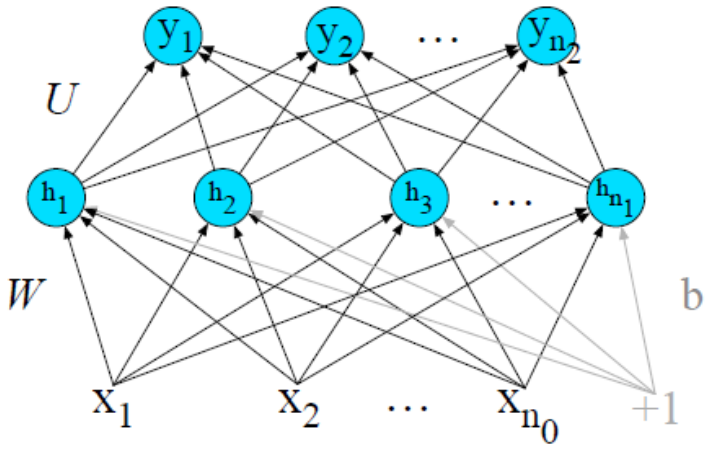

Abbildung 2: Ein einfaches Feedforward-Netz mit einer Input-Schicht, einer Hidden Layer und einer Output-Schicht (Jurafsky und Martin 2014, S. 130).

Die Input-Schicht erhält als Einstiegspunkt kombiniert mit Bias-Wert und Weights, den ersten Input und gibt sie an die erste Hidden Layer weiter. Jede Unit einer Schicht nimmt dabei als Input alle Outputs der vorherigen Schicht auf und kombiniert sie wiederum mit Weights und Bias-Wert. So sind alle Units zweier nebeneinander liegender Schichten miteinander verbunden (s. Abbildung 2). Der Output einer Hidden Layer stellt dementsprechend eine Repräsentation des Inputs dieser Hidden Layer dar.

Die Output-Schicht liefert das endgültige Ergebnis des Netzes. Es kann, wie bereits erwähnt, je nach Task eine reelle Zahl oder im Falle einer binären Klassifikationsaufgabe einen durch Wahl der Weights, des Bias Terms sowie der Aktivierungsfunktion eingegrenzten Wert darstellen, der einer bestimmten Klasse entspricht (1 oder -1, beziehungsweise 1 oder 0). Im Falle einer nicht-binären Mehrfachklassifikation, zum Beispiel bei Part-of-Speech-Tagging, gäbe es eine Output-Unit pro Klasse. Die Unit mit der entsprechenden Klasse hätte dann einen positiven Wert, während die restlichen Units für anderen Klassen ein negatives Ergebnis oder 0 ausgeben würden. So könnte eine Aussage über die Wahrscheinlichkeitsverteilung des Outputs getroffen werden (Jurafsky und Martin 2014, 129ff.). Beim Trainieren eines Feedforward-Netzes ist es das Ziel, den Systemoutput $\hat{y}$ so weit wie möglich dem korrekten Output y anzunähern. Der korrekte Output ist bei einem Feedforward-Netz bekannt. Der Systemoutput ist lediglich eine Schätzung darüber, was der korrekte Output ist. Um $\hat{y}$ und y so nah wie möglich einander anzunähern, werden die Weight-Parameter und die Bias-Werte aller Schichten angepasst. Zuerst muss also festgestellt werden, inwieweit der geschätzte Wert und der korrekte Wert voneinander entfernt sind. Dies geschieht mit Hilfe einer Loss-Funktion.

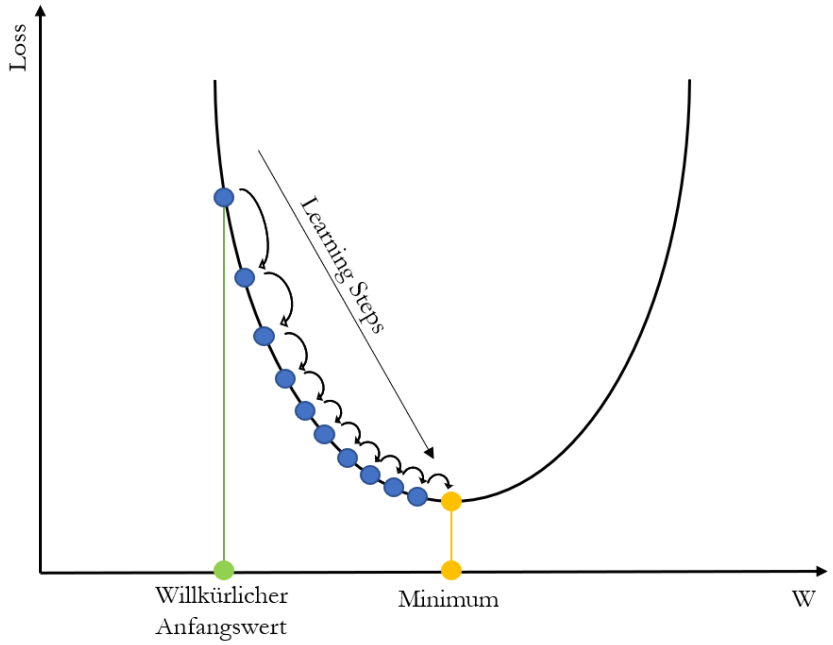

Abbildung 3: Die Funktionsweise des Gradientenabstiegverfahrens (Gradient Descent).

Im Falle einer binären Klassifikation bedeutet das, dass der Output $\hat{y}$ eines Netzes mit dem korrekten Ergebnis y verglichen wird und festgestellt wird, inwieweit sich beide unterscheiden. Um den Unterschied (bei binärer Klassifikation meist *Cross-Entropy-Loss* oder

auch *Log Loss*) so klein wie möglich zu halten und die Wahrscheinlichkeit einer korrekten Klassifikation zu maximieren, sollen anschließend mit Hilfe eines Optimierungsalgorithmus iterativ die Weights w und der Bias b angepasst werden. Der genannte Algorithmus heißt *Gradient Descent* oder *Gradientenabstieg* und wird dazu eingesetzt, das Minimum einer Funktion zu finden. Zur Beschreibung dessen Funktionsweise wird in der Literatur oft die Analogie eines nebligen Tals verwendet, in dem man ohne Sicht den tiefsten Punkt sucht. Dazu tastet man sich voran und stellt fest, in welcher Richtung es bergab geht, indem man die Differenz des Höhenunterschieds beider Füße einschätzt (Jurafsky und Martin 2014, S. 83). Auf eine Funktion übertragen (s. *Abbildung 3*) stellt sich dieser Vorgang wie folgt dar: Es wird ein beliebiger Punkt in der Loss-Funktion als Startpunkt gewählt. Die X-Achse stellt die Weights *W* dar und die Y-Achse den Loss. Unter Zuhilfenahme des Gradientenabstiegverfahrens wird ein Vektor erzeugt, der in die Richtung der Steigung zeigt. Indem schrittweise in die entgegengesetzte Richtung vorgegangen wird, werden so lange die Weights angepasst, bis das Minimum erreicht ist.[7] Die Schrittgröße, auch *Learning Rate* genannt, in der sich in Richtung des Minimums bewegt wird, ist ebenfalls von großer Wichtigkeit. Ist sie zu klein, benötigt das Training zu viele Epochen. Ist sie hingegen zu groß kann es passieren, dass das Minimum immer wieder verfehlt wird. Der Algorithmus würde sich infolgedessen ab einem bestimmten Punkt nicht mehr verbessern, obwohl das Optimum noch nicht gefunden worden wäre. Der gesamte beschriebene Prozess gehört zur Fehlerrückführung über alle Schichten eines künstlichen neuronalen Netzes hinweg und nennt sich auch *Backpropagation* (Jurafsky und Martin 2014, 83f.; Ertel 2016, S. 297). Ihr Erfolg kann mit der Wahl bestimmter Hyperparameter im Training des neuronalen Netzes beeinflusst werden (vgl. Kapitel 3.3.1).

Feedforward-Netze haben gleichzeitigen Zugriff auf alle Elemente des Inputs. Aspekte der Eingabe werden auf einmal erfasst. Sie verarbeiten lediglich Input-Vektoren einer festen Größe. In der Anwendung von NLP bietet dieses Konzept mehr Nach- als Vorteile, da Sprache generell einen sequenziellen Charakter hat. Das bedeutet, dass sinnvolle Inputs für sprachverarbeitende neuronale Netzwerke nicht immer in gleich große Vektoren gezwungen werden können. Jurafski und Martin sprechen in diesem Zusammenhang vom „zeitlichen Charakter" von Sprache, der sich durch ihre sequenzielle Verarbeitung ergibt (2014, S. 169). Feedforward-Netze können trotzdem für automatisierte Sprachverarbeitung eingesetzt

[7] Hierbei ist zu beachten, dass es sich in diesem Beispiel um eine konvexe Funktion handelt. Sollte dies nicht der Fall sein und die Funktion hat mehrere ‚Täler', so muss zwischen dem lokalen und dem globalen Minimum unterschieden und zuerst festgestellt werden, ob man sich im richtigen ‚Tal' befindet (Ertel 2016, 297f.; Jurafsky und Martin 2014, 82f.).

werden, indem die Sprachsequenz mit einem Fenster einer festen Token-Anzahl eingeteilt wird. Überschreiten die Tokens die bestimmte Anzahl, kann das Fenster Token für Token nach vorne verschoben werden. Dieser Ansatz wird *Sliding-Window* genannt. Doch löst er nicht alle Probleme, da dementsprechend Kontext von der Verarbeitung ausgeschlossen wird, der sich beliebig weit vom momentanen Fenster entfernt befindet, aber dennoch für das semantische Verständnis von großer Wichtigkeit sein kann. Zudem können im Text enthaltene Muster und Zugehörigkeiten verloren gehen (Jurafsky und Martin 2014, 169f.).

2.2.2 RNNs

Rekurrente neuronale Netze (RNN) und *Long short-term memory (LSTM)* bieten gegenüber Feedforward-Netzen grundlegende Vorteile in der Verarbeitung sequenzieller Daten. Bei Sequenzdaten handelt es sich um Listen von Vektoren, die einzelne Teile der gesamten Inputdaten enthalten. Die Reihenfolge der Tokens innerhalb der Vektoren ist dabei von großer Wichtigkeit, da sie beispielsweise bei Textdaten Beziehungen zwischen den Wörtern und Satzteilen und somit deren Bedeutung bestimmt (Goodfellow et al. 2017, 367ff.); (Tamura 2020). Sichtbar wird das an einem einfachen Beispiel:

Das Auto fährt schnell. Fährt das Auto schnell?

Der erste Satz ist eine einfache Aussage, während es sich bei dem zweiten Satz um eine Frage handelt. Abgesehen von den Satzzeichen, die nicht immer mit in die Interpretation und Verarbeitung von Textdaten übernommen werden, lässt sich der Unterschied nur am Satzbau, soll heißen an der Reihenfolge der Wörter, festmachen. Eine sequenzielle Verarbeitung der genannten Beispielsätze zur Feststellung der Bedeutung würde wie folgt ablaufen: Nacheinander werden Wörter eingelesen und vom System in einzelnen Zeitschritten verarbeitet, ohne dass dabei das vorangegangene Wort vergessen wird. Bereits verarbeitete Wörter können so in die Interpretation des momentanen Wortes und schließlich des ganzen Satzes mit einfließen. NLP-Tasks, die sich sequenzielle Daten zunutze machen, sind maschinelle Übersetzung, Spracherkennung, Named-Entity-Recognition, Part-of-Speech-Tagging und viele mehr (Jurafsky und Martin 2014, S. 176).

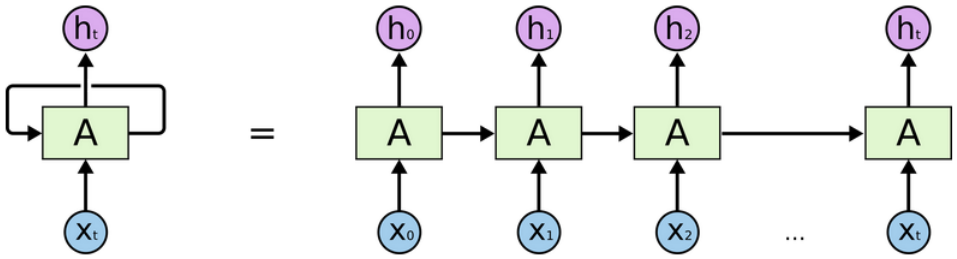

Abbildung 4: Ein einfaches rekurrentes Netzwerk, in dem die Hidden Layer *A* eine kreisläufige Verbindung enthält, mit der Outputs vorheriger Zeitschritte in den Input des aktuellen Zeitschrittes mit einbezogen werden (Olah 2015)

Um ihre Art der Verarbeitung und deren Vorteile genauer zu verstehen, soll zuerst die Funktionsweise eines einfachen RNN erklärt werden. RNNs und die dahinterstehenden mathematischen Zusammenhänge können hochkomplex sein. Deshalb wird für die folgenden Erklärungen ein stark vereinfachtes Modell als Beispiel genommen.

Die grundlegende Funktionsweise eines rekurrenten Netzes ähnelt zunächst einem Feedforward-Netz. Der Input ist ein Vektor, der das momentane Eingabeelement x_t darstellt (s. *Abbildung 4*). Dieser Vektor wird mit einer Weight-Matrix multipliziert und das Produkt durch eine Aktivierungsfunktion verarbeitet. Dessen Ergebnis wird an die nächste Hidden Layer *A* weitergegeben, welche eine kreisläufige Verbindung beinhaltet. Bei der Verarbeitung einer Sequenz wird an dieser Stelle der Input des aktuellen Zeitschritts *t* mit dem Aktivierungswert der Hidden Layer vom vorherigen Zeitschritt *t-1* ergänzt. Die vorherige ausgeblendete Schicht bildet den Kontext ab, der bis zum Anfang der Sequenz zurückreicht und sich auf nachfolgende Entscheidungen auswirkt. RNNs enthalten also, im Gegensatz zu Feedforward-Netzen, eine zeitliche Komponente in Form eines inkrementellen Algorithmus, der sich in einzelnen Zeitschritten über eine Sequenz hinweg bewegt (Chung et al. 2015). Jedes Netz, das einen Kreislauf innerhalb seiner Verbindungen enthält, ist ein rekurrentes Netz (Jurafsky und Martin 2014, S. 170). RNNs enthalten ein zusätzliches Set von Weights *U*, die festlegen, wie das Netz innerhalb der Hidden Layer den Kontext der vorherigen Zeitschritte verarbeiten soll und, ebenso wie die Weights der Input-Schicht zur Hidden Layer und die der Hidden Layer zur Output-Schicht, durch Backpropagation trainiert werden (Jurafsky und Martin 2014, 170f.). Aufgrund des zeitlichen Charakters des gesamten Prozesses spricht man auch von „Backpropagation Through Time" (Jurafsky und Martin 2014, S. 175). In der Theorie ermöglicht der rekursive Aufbau des Algorithmus RNNs demgemäß vorherige Informationen mit in den aktuellen Zeitschritt einzubeziehen. In der Praxis zeigt sich jedoch, dass dies nicht immer der Fall ist, beispielsweise wenn nicht nur der

nächstliegende Kontext benötigt wird, um den aktuellen Schritt erfolgreich durchzuführen, sondern auch Informationen, die sich weiter vom momentanen Wort entfernt befinden. Erkennbar wird das an folgendem Beispiel:

Ich lebe in Deutschland. … Ich spreche deutsch.

Soll das letzte Wort des zweiten Satzes vorhergesagt werden, so ist klar, dass es sich um eine Sprache handelt. Welche es nun genau ist, ist jedoch vom im vorherigen Satz genannten Land („Deutschland“) abhängig. Diese Information kann sich beliebig weit entfernt im Text befinden und der Abstand dabei unter Umständen sehr groß sein. RNNs können zu weit entfernten Kontext nicht mehr verarbeiten und den Bezug zwischen den Informationen verlieren, was hauptsächlich zwei Gründe hat. Der erste liegt in der doppelten Aufgabe der Verarbeitung der Information im aktuellen Zeitschritt sowie die Forwardpropagation von Informationen, die für Entscheidungen zukünftiger Zeitschritte vonnöten sind. Diese Komplexität führt dazu, dass RNNs im Training sehr langsam sind. Der zweite Grund der Schwierigkeiten beim Trainieren von RNNs sind die Gradienten, die, bedingt durch die exponentielle Abhängigkeit der Größe der Weights von der Fehlerrückführung über mehrere Zeitschritte hinweg, explodieren („blow up“) oder verschwinden („vanish“) können (Hochreiter und Schmidhuber 1997, 3ff.; Jurafsky und Martin 2014, S. 184). Eine Art von rekurrenten neuronalen Netzen, die diese Schwierigkeiten überwinden, sind LSTMs.

2.2.3 LSTMs

LSTMs (Long short-term memory) sind ebenfalls rekurrente neuronale Netze, die jedoch die Problematiken von RNNs überwinden und auch weit entfernte Informationen in die Verarbeitung mit einbeziehen können. Dazu werden Informationen, die für den aktuellen Schritt unwichtig sind und nicht mehr gebraucht werden, *vergessen*, und neue, wichtige Informationen gespeichert und weitergegeben (Hochreiter und Schmidhuber 1997). Um diese beiden Vorgänge leisten zu können, werden sogenannte Memory-Zellen verwendet. Anhand *Abbildung 5* lässt sich die Verarbeitung von Informationen innerhalb einer Memory-Zelle erklären: Auch in LSTMs werden Daten in eine Zelle einer Hidden Layer h_t aufgenommen, darin verarbeitet und zur nächsten Hidden Layer h_{t+1} weitergegeben.

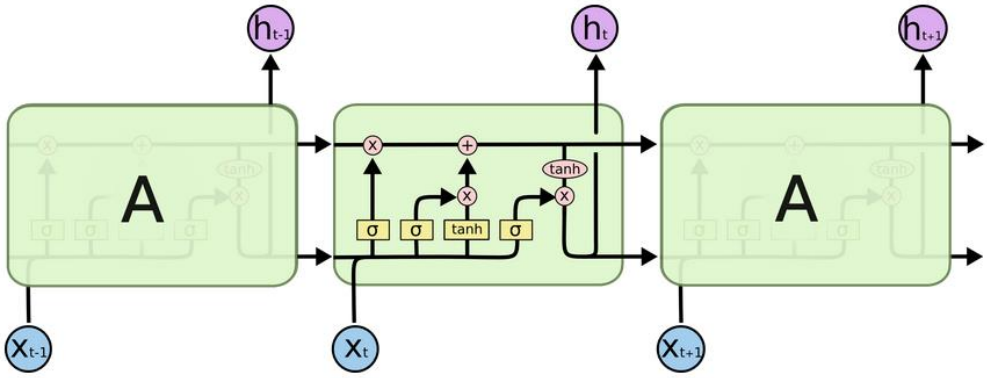

Abbildung 5: Aufbau einer Memory-Zelle und deren Funktionsweise innerhalb eines LSTMs (Olah 2015)

Im oberen Bereich der Zelle verläuft Zellstatus *C*, der als das *Gedächtnis* des LSTM-Netzes bezeichnet werden könnte. Er reicht über alle Schichten hinweg, so dass sich theoretisch Informationen vom ersten bis zum letzten Zeitschritt gemerkt werden können. Während des Verarbeitungsprozesses innerhalb der einzelnen Schichten können Informationen mit Hilfe verschiedener Gates dem Zellstatus hinzugefügt oder daraus entfernt und vergessen werden (Hochreiter und Schmidhuber 1997). Jede Zelle hat ein *Input-*, ein *Output-* und ein *Forget-Gate*. Sie sind selbst eigene neuronale Netze und können lernen, welche Informationen wichtig sind und sich gemerkt werden müssen und welche vergessen werden können. Die Aktivierungsfunktionen der Gates sind Sigmoid-Funktionen und bilden demgemäß die Aktivierungswerte zwischen 0 und 1 ab (Jurafsky und Martin 2014, S. 184). In vier Schritten werden nun das Ergebnis der vorherigen Hidden Layer h_{t-1} gemeinsam mit dem Input des aktuellen Zeitschritts x_t verarbeitet:

Zuerst werden beide Werte durch das Forget-Gate geleitet. Ergebnisse aus der Sigmoid-Funktion, die näher an 0 sind, werden vergessen. Ergebnisse, die näher an 1 sind, sollen sich gemerkt werden. Um zu entscheiden, welche Werte im zweiten Schritt zum Zellstatus hinzugefügt werden, werden h_{t-1} und x_t durch das Input-Gate verarbeitet. Wieder bestimmt die Sigmoid-Funktion, welche Werte upgedatet werden sollen und welche nicht. Die Informationen werden nun durch eine tanh-Funktion geleitet, die die Ergebnisse des Forget- und des Input-Gates zwischen -1 und 1 abbildet und einen Vektor C_t erstellt, der den neuen Zellstatus repräsentiert. Im dritten Schritt sollen die zuvor getroffenen Entscheidungen ausgeführt und der Zellstatus des vorherigen Zeitschrittes C_{t-1} in den neuen Zellstatus C_t aktualisiert werden. Dazu wird erst der vorherige Hidden State h_{t-1} gemeinsam mit x_t durch das eine Sigmoid-Funktion enthaltende Output-Gate geleitet. Der neue Zellstatus C_t wird durch eine tanh-Funktion verarbeitet. Das Ergebnis wird anschließend mit dem des Output-

Gates multipliziert, woraus sich der neue Hidden State h_t ergibt. Schließlich werden C_t und h_t an den nächsten Zeitschritt weitergereicht (Hochreiter und Schmidhuber 1997; Jurafsky und Martin 2014, 184ff.).

Der Prozess der Informationsverarbeitung innerhalb einer Memory-Zelle lässt sich am besten an einem einfachen Sprachmodell-Beispiel erklären, in dem das nächste Wort anhand der vorherigen Wörter vorhergesagt werden soll. Beispielsweise könnte in einem Text eine Person verschwinden und eine neue mit einem anderen Geschlecht hinzukommen. Im ersten Schritt würde der Entschluss gefasst, dass das alte Geschlecht, welches im Zellstatus existiert, vergessen werden soll. Anschließend würde mit Hilfe des Input-Gates entschieden, dass das neue Geschlecht dem Zellstatus hinzugefügt werden soll. Im dritten Schritt würden die vorher getroffenen Entscheidungen ausgeführt und das alte mit dem neuen Geschlecht ersetzt werden. Schlussendlich könnte der neue Zellstatus dazu genutzt werden, dem neuen Geschlecht entsprechend ein richtig konjugiertes Wort auszugeben (Olah 2015).

Auf den ersten Blick scheint die beschriebene Architektur von LSTMs also die Probleme der Langsamkeit und des schlechten Gedächtnisses von einfachen RNNs zu lösen. Leider ist das jedoch nicht der Fall, sind LSTMs doch durch die gesteigerte Komplexität noch langsamer im Training als einfache RNNs. Hinzu kommt, dass auch bei LSTMs die Sequenzlänge, bei der der gesamte Kontext noch mit hoher Qualität mit einbezogen werden kann, ihre Grenze hat. Ab einem gewissen Maximum geht also auch in diesem System der Kontext verloren. Eine weitere kritische Eigenschaft von RNNS und LSTMs ist die fehlende Fähigkeit parallele Prozesse auszuführen. Dies rührt von der auf sequenzielle Daten ausgerichteten Architektur her, bei der Token für Token nacheinander verarbeitet werden (Vaswani et al. 2017, 1f.).

Eine Modellarchitektur, die durch *Parallelisierung* und den *Attention-Mechanismus* eine weitaus höhere Performance ermöglicht, ist der *Transformer*.

2.2.4 Transformer und Attention

Wie bereits im vorherigen Kapitel erwähnt, bieten Transformer gegenüber RNNs den Vorteil der Parallelisierung. Während RNNs Sequenzen Token für Token aufnehmen und nacheinander verarbeiten, können Transformer diese Verarbeitung parallel durchführen und dadurch eine höhere Performance erreichen. Die Transformer-Modellarchitektur wurde 2017 von Vaswani et al. vorgestellt und für speziell Übersetzungstasks konzipiert. Man spricht auch von *Sequence-to-sequence-Models*, da eine Sequenz in einer Sprache eingelesen und

schließlich in einer anderen Sprache wieder eine Sequenz ausgegeben wird (Vaswani et al. 2017; Jurafsky und Martin 2014, S. 191).

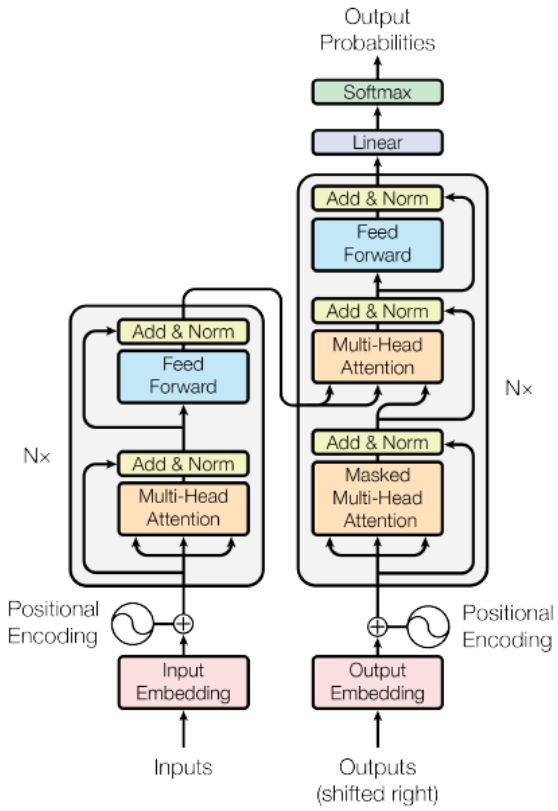

Abbildung 6: Architektur des Transformer-Modells mit dem Encoder-Block auf der linken und dem Decoder-Block auf der rechten Seite (Vaswani et al. 2017, S. 3)

Ein weiterer verwendeter Begriff für solche Modelle ist *Encoder-Decoder-Network*. Die Idee hinter Encoder-Decoder-Networks ist, das Potenzial von auf Parallelisierung ausgerichteten GPUs zu nutzen, indem eine Sequenz im Ganzen statt in einzelnen Zeitschritten eingelesen wird (Vaswani et al. 2017, S. 2). Für die entsprechende Sequenz wird mit Hilfe eines Encoders eine Repräsentation in Form eines Vektors erstellt und dieser anschließend durch einen Decoder verarbeitet. Schlussendlich wird eine dem Task entsprechende Sequenz ausgegeben. Repräsentationen eines Wortes in Form eines Vektors werden *Embeddings* genannt. Sie ermöglichen die rechnerische Verarbeitung eines Wortes indem sie dieses auf einen Punkt im Raum mappen. Wörter mit ähnlichen Bedeutungen befinden sich dabei näher beieinander als Wörter mit unterschiedlichen Bedeutungen. Durch Embeddings [8] können auch Mehrdeutigkeiten, Synonyme o. Ä. erfasst werden, die jedoch immer vom kontextuellen

[8] Mehr zum Thema Embeddings kann in „Speech and Language Processing“ von Jurafsky und Martin (2014) ab Seite 94 nachgelesen werden.

Bezugsrahmen des Satzes sowie von der Reihenfolge der Wörter abhängig sind, die ohne rekurrente Verarbeitung der Sequenz verloren gehen würden (Jurafsky und Martin 2014, 94ff.). Dieser Bezug wird mit *Positional Encodings* erfasst, die sich die Position des Wortes im Satz merken und dem *Input Embedding* (s. *Abbildung 6*) hinzugefügt werden, welches in das Transformer-Modell einspeist wird (Vaswani et al. 2017, 5f.).

Am Beispiel der Übersetzung eines Satzes lässt sich die Funktionalität von Encoder und Decoder erklären. Zuerst wird der Satz in der ersten Sprache als Input Embeddings mit Positional Encodings durch den Encoder-Block geleitet. Dieser besteht aus mehreren Schichten. Die erste Schicht, die im Encoder-Block den Input verarbeitet, ist eine *Multi-Head-Attention-Schicht*. Sie macht sich den sogenannten *Attention-Mechanismus* zunutze. Mit Attention kann festgestellt werden, welcher Teil des Inputs wichtig ist, sprich, es wird die Frage beantwortet: Wie relevant ist jedes Wort im Verhältnis zu allen anderen Wörtern innerhalb eine Satzes? Um das festzustellen, wird eine von Vaswani et al. *Scaled Dot-Product Attention* (Vaswani et al. 2017, S. 4) genannte Attention-Funktion verwendet. Die Formel dieser Funktion lautet:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q, K und V stehen hierbei für die abstrakten Vektoren Query, Key und Value. Sie repräsentieren verschiedene Elemente des Input-Wortes, für das der Attention-Wert errechnet werden soll. Dementsprechend gibt es für jedes Wort Q-, K- und V-Vektoren. Das Skalarprodukt des Querys wird mit allen Keys durch die Wurzel der Dimension von Q, K und V (d_k) geteilt. Anschließend werden durch die Anwendung der Softmax-Funktion[9] die Weights angepasst. Dadurch wird für jedes Wort ein Attention-Vektor erstellt, der die Relevanz eines Wortes in Verhältnis zu seinem Kontext repräsentiert. Im Multi-Head-Attention-Block werden mehrere Attention-Vektoren für verschiedene Repräsentationen des gleichen Worts erzeugt (s. *Abbildung 7*). Vaswani et al. schreiben zum Vorteil dieses Vorgehens:

[9] Die Softmax-Funktion ist eine exponentielle Funktion, mit der beliebig viele Werte eines Vektors auf eine Wahrscheinlichkeitsverteilung gemappt werden, bei der jeder Wert zwischen 0 und 1 liegt. Alle Werte dieses Output-Vektors summieren sich auf 1 (Jurafsky und Martin 2014, S. 89).

> Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this. (Vaswani et al. 2017, S. 5)

Damit die erzeugten Attention-Vektoren im nächsten Schritt in ein Feedforward-Netz geleitet werden können, das jedoch nur einen Vektor pro Zeitschritt aufnehmen kann, werden sie in Form einer gewichteten Matrix gebracht. Das Feedforward-Netz bringt diese Matrix in eine vom Decoder verwendbare Form.

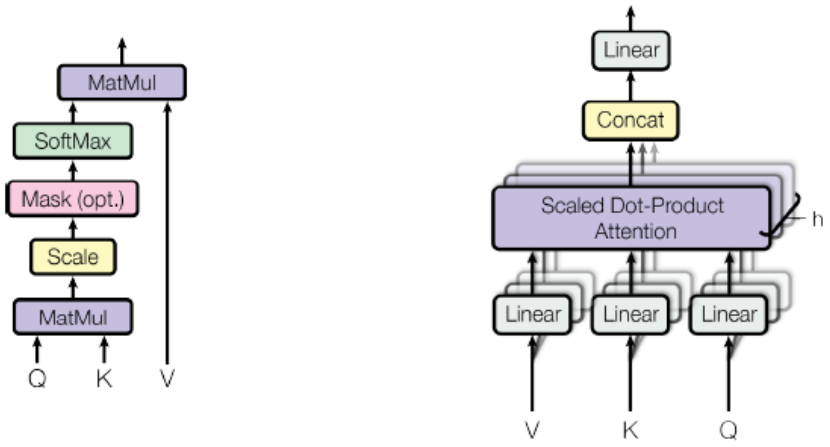

Abbildung 7: Scaled Dot-Product Attention auf der linken Seite, Multi-Head Attention mit mehreren parallelen Attention Layers auf der rechten Seite (Vaswani et al. 2017, S. 4)

Nun ist der Decoder an der Reihe. Er nimmt als Input den vorher generierten, übersetzten Satz Wort für Wort in einer zweiten Sprache als Input Embeddings auf und fügt, wie bereits zuvor beim Encoder, jeweils ein Positional Encoding hinzu. Der Decoder-Block besteht aus drei wichtige Komponenten: zwei Attention-Blöcke und ein Feedforward-Netz. Der erste Attention-Block produziert wieder Attention-Vektoren für jedes Wort und legt so die Relevanz des Wortes innerhalb des Satzes fest. Dieser Block heißt *Masked Multi-Head Attention* (s. *Abbildung 6*). Da alle Wörter des Satzes in der ersten Sprache, aber nur die bisherigen verarbeiteten Wörter des Satzes in der zweiten Sprache verwendet werden können, werden die folgenden Wörter ‚maskiert', um einen Lernerfolg zu erzielen (Jurafsky und Martin 2014, S. 196). Anschließend werden die produzierten Vektoren gemeinsam mit den Vektoren des Encoders durch einen weiteren Multi-Head-Attention-Block verarbeitet. Dieser nimmt alle Vektoren der beiden Sätze in den unterschiedlichen Sprachen auf und legt fest, wie stark die Bezüge zwischen den Wortvektoren beider Sprachen sind. Zuletzt wird das tatsächliche Mapping der Wörter in den verschiedenen Sprachen durchgeführt. Als

Output liefert der Block Attention-Vektoren für alle Wörter beider Sätze (Vaswani et al. 2017, S. 5). In der nächsten Schicht bringt das Feedforward-Netz die Vektoren für die folgenden Funktionen oder, falls benötigt, einen weiteren Decoder-Block, als gewichtete Matrix in eine leichter zu verarbeitende Form. Die folgenden Funktionen sind eine Linear-Schicht – ein weiteres Feedforward-Netz – und schließlich eine Softmax-Funktion, die mit Hilfe einer Wahrscheinlichkeitsverteilung als Output auf das wahrscheinlichste nächste Wort verweist. So wird ein Wort nach dem anderen vorhergesagt, bis der gesamte Satz übersetzt ist (Vaswani et al. 2017, S. 5).

Transformer mit ihrem Encoder-Decoder-Modell ermöglichen folglich eine Arbeitsteilung. Der Encoder erfasst eine Sprache samt deren Grammatik, Syntax und Kontext, während der Decoder Wörter einer Sprache auf die übersetzten Wörter einer anderen Sprache mappt. Diese Aufgabenteilung macht die Transformer-Architektur flexibel, so dass sie getrennt und neu angeordnet werden kann, um nicht mehr nur Übersetzungen lösen zu können. Ein Beispiel hierfür ist *BERT*, bei dem Encoder-Blöcke aneinanderreiht werden, wodurch es für viele verschiedene NLP-Tasks eingesetzt werden kann.

2.3 BERT

BERT steht für *Bidirectional Encoder Representation from Transformers* und wurde 2018 von Devlin et al. vorgestellt. Mit einer Architektur, die sich Transformer-Encoder zunutze macht und einem speziellen Trainingsprozess, mit dessen Hilfe auf tiefergehendes Sprachverständnis zurückgegriffen werden kann, erreicht es hervorragende Ergebnisse und stellt den momentanen State-of-the-Art im NLP dar.

2.3.1 Funktionsweise

Wie bereits erwähnt, wird für eine erfolgreiche Verarbeitung von Sprache der Einbezug des Kontextes innerhalb eines Textes und ein allgemeines Sprachverständnis der entsprechenden Sprache benötigt. BERT bezieht beides durch einen zweigeteilten Trainingsprozess (s. *Abbildung 8*) mit ein. Dazu werden zuerst Sprachmodelle auf große ungelabelte Textkorpora vortrainiert. Anschließend kann das Model je nach gewünschtem Task gefinetuned werden, sprich unter Verwendung des bereits im Pretraining gelernten Wissens mit gelabelten Daten auf einen Task trainiert werden (Devlin et al. 2019).

BERTs Aufbau besteht aus aufeinander geschichteten Encodern (vgl. *Transformer und Attention*). Es steht in zwei verschiedenen Größen zur Verfügung: $\text{BERT}_{\text{BASE}}$ und $\text{BERT}_{\text{LARGE}}$. Die BASE-Version enthält 12 und die LARGE-Version 24 Encoder.

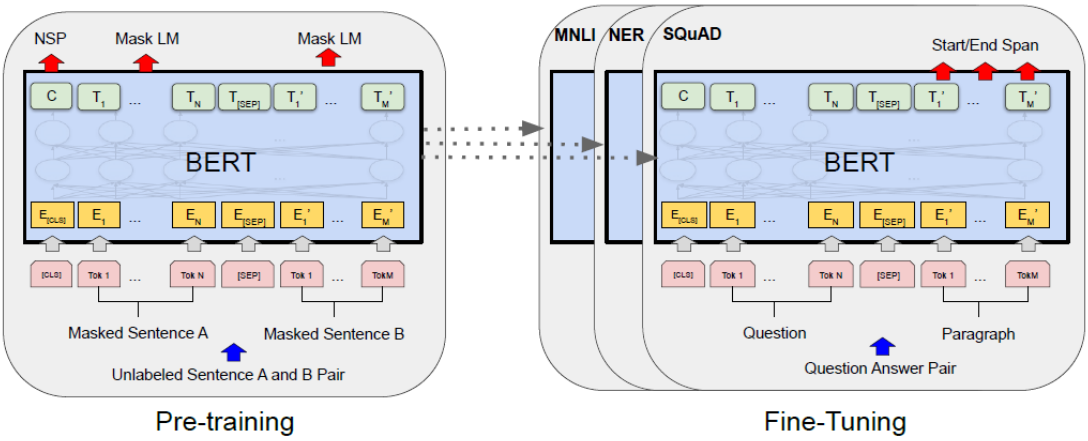

Abbildung 8: BERT zweigeteilter Trainingsprozess mit Pre-Training auf der linken und Fine-Tuning auf der rechten Seite (Devlin et al. 2019)

2.3.1.1 Pretraining

Ziel der ersten Phase des Trainingsprozesses von BERT, genannt Pretraining, ist, ein tiefgehendes Sprachverständnis aufzubauen. Beim ursprünglichen BERT-Model wurden für das Pretraining das englische Wikipedia (2500 Millonen Wörter) und der BooksCorpus (800 Millionen Wörter) verwendet (Devlin et al. 2019). Das Model soll im Pretraining lernen, was die entsprechende Sprache in Bezug auf Syntax, Grammatik und Kontext ausmacht. Um dies zu erreichen, wird das BERT-Model gleichzeitig auf zwei unüberwachte Tasks trainiert: *Masked Language Model (MLM)* und *Next Sentence Prediction (NSP)*. Zur Erklärung, wie beide Tasks funktionieren, muss zunächst darauf eingegangen werden, wie sich Input-Repräsentationen in BERT zusammensetzen (s. *Abbildung 9*).

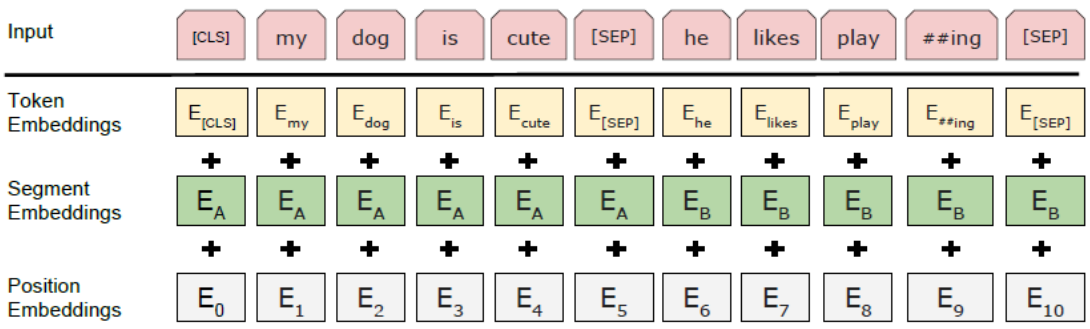

Abbildung 9: Zusammensetzung von Input-Repräsentationen in BERT (Devlin et al. 2019)

BERTs Input-Repräsentation ist fähig, sowohl einen einzelnen Satz wie auch mehrere aneinandergereihte Sätze in einer Token-Sequenz darzustellen. Der Input ist die Summe aus drei verschiedenen Vektoren. *Token Embeddings* sind vortrainierte *WordPiece-Embeddings*, die Wörter in kleinere Wort-Einheiten aufteilen. Letztere sind Teil eines Vokabulars von 30 Tausend Tokens (Wu et al. 2016). *Segment Embeddings* repräsentieren die Nummer eines Satzes in einem Vektor. Dazu wird jedem Token ein Embedding hinzugefügt, dass angibt, ob es zu Satz A oder Satz B gehört. *Position Embeddings* sind die in einem Vektor dargestellten Positionen der Worte innerhalb eines Satzes. Die resultierende Input-Sequenz beginnt immer mit [CLS]. Zwischen Sätzen steht ein [SEP]-Token. Basierend auf diesem Input wird das BERT-Model mit Hilfe von MLM und NSP trainiert (Devlin et al. 2019).

Bei MLM werden 15 Prozent der Wörter des Gesamtkorpus ‚maskiert' und somit versteckt (s. *Abbildung 8 links*). 80 Prozent davon werden mit einem [MASK]-Token verdeckt, 10 Prozent werden mit einem anderen willkürlich gewählten Wort ausgetauscht und 10 Prozent bleiben gleich. Aufgabe des Models ist, anschließend die maskierten Wörter zu benennen. So wird eine tiefe bidirektionale Repräsentation erreicht, was bedeutet, dass sowohl der der Sequenz vorausgehende als auch der der Sequenz folgende Kontext mit einbezogen wird. Die Anzahl der maskierten Tokens kann willkürlich gewählt werden. BERTs Output besteht aus den Wortvektoren T_i, die alle gleich groß sind und gleichzeitig generiert werden. Um den Loss zu minimieren, werden alle Wortvektoren der maskierten Wörter durch eine Softmax-Funktion geleitet und somit Wahrscheinlichkeitsverteilungen erzeugt. Anschließend wird jede Wahrscheinlichkeitsverteilung mit Hilfe eines Cross-Entropy-Loss mit dem Vektor des tatsächlichen Labels, also des Wortes aus dem Korpus, verglichen (Devlin et al. 2019; Wennker 2020, S. 34).

Bei NSP werden als Input zwei Sätze geliefert (A und B). Das Model soll schließlich ausgeben, welcher Satz auf den anderen folgt. In 50 Prozent der Fälle folgt A auf B und in der anderen Hälfte der Fälle B auf A. Durch NSP lernt BERT satzübergreifenden Kontext, die Beziehungen zwischen Sätzen und kann zudem sein Sprachverständnis erweitern. Output dieses Tasks ist ein binärer Wert C (s. *Abbildung 8*) (Devlin et al. 2019; Wennker 2020, S. 34).

Der beschriebene Prozess des Pretrainings hilft BERT, sowohl syntaktisches und grammatisches Sprachverständnis als auch semantisches Wissen aufzubauen. Auf das Pretraining folgt das Fine-Tuning für die Anforderungen eines spezifischen Tasks.

2.3.1.2 Fine-Tuning

Dank des Pretrainings geht der Fine-Tuning-Prozess recht schnell und einfach vonstatten. Die Transformer-Architektur bringt Self-Attention mit sich, was BERT ermöglicht, auf viele verschiedene Tasks angewendet zu werden. Für das Fine-Tuning wird das bereits vortrainierte Model verwendet. Es werden lediglich die Inputs durch taskspezifische Inputs und die letzte Output-Schicht durch eine auf den Task passende Schicht ausgetauscht, beispielsweise eine Textklassifikationsschicht oder eine Question-Answering-Schicht. Der neue Input ist ein gelabeltes Datenset. Devlin et al. zeigen dies in *Abbildung 8* am Beispiel eines Question-Answering-Tasks. Als Input werden eine Frage gestellt sowie die im zweiten Paragraph enthaltene Antwort mittels des Outputs vorhergesagt.

Nach Anpassung bestimmter Hyperparameter kann das Model auf einen Task trainiert werden. Die Hyperparameter beeinflussen die Art und Weise des Lernens des neuronalen Netzes und wirken sich somit auf das endgültige Model aus (Wennker 2020, S. 12). Die am häufigsten in der Praxis angepassten Parameter sind *Learning Rate, Epochs* und *Batch Size* (Devlin et al. 2019; Sun et al. 2019).

Learning Rate

Die Learning Rate legt die Größe der Schritte fest, in der beim Gradient Descent in Richtung des Minimums vorgegangen wird (s. *Abbildung 3)*. Ist die Learning Rate zu klein, kann die Suche nach dem Minimum zu lange dauern und ist damit ineffizient. Ist sie zu groß, kann das Minimum überschritten und immer wieder ‚verpasst' werden. Es gilt dementsprechend, eine Learning Rate zu finden, die zu einem steilen Abfall der Loss-Funktion führt, aber dabei nicht divergiert, sprich, immer wieder über das Minimum hinwegspringt (vgl. *Kapitel 2.2.1*). Zusätzlich zum Gradient Descent hat die Learning Rate auch Einfluss auf das sogenannte *Catastrophic Forgetting*, was bedeutet, dass das Model bereits gelernte Inhalte aus dem Pretraining vergisst, wenn bei der Umsetzung von Tasks neues Wissen hinzukommt (Chen et al., S. 7871).

Epochs

Mit der Anzahl der Epochen wird festgelegt, wie oft die gesamten Daten jeweils das neuronale Netz durchlaufen. So wird dessen Fähigkeit der Vorhersage beeinflusst. Über mehrere Epochen hinweg kann sich der Loss-Wert immer mehr verkleinern (Wennker 2020, S. 25).

Batch Size

Wie groß die Datenpakete sein sollen, die als Input auf einmal vom neuronalen Netz verarbeitet werden, wird durch die Batch Size angegeben. Dazu wird das Trainingsdatenset in kleine Pakete aufgeteilt, die jeweils die Größe der festgelegten Batch Size haben. Die Wahl der Batch Size ist abhängig von der Rechnerarchitektur, auf der das Training durchgeführt wird. Eine zu kleine Batch Size verlangsamt das Training, eine zu große kann die vorhandene Computing Power überlasten (Wennker 2020, S. 25).

Weitere Hyperparameter sind beispielsweise die maximale Sequenzlänge und der *Drop Out*. Die maximale Sequenzlänge legt das Maximum der Länge eingelesener Sequenzen fest. Sequenzen, die dieses überschreiten, werden am Grenzwert abgeschnitten. Der Drop Out bringt das neuronale Netz dazu, fixe Muster zu erlernen, indem ein Wert festgelegt wird, der die Wahrscheinlichkeit angibt, mit der ein willkürlich gewähltes Neuron während des Trainings inaktiv ist. So wird die Komplexität des Netzes verringert und eventuelles Rauschen in den Daten ignoriert (Wennker 2020, S. 26). Er ist ein hilfreiches Werkzeug, um *Overfitting* zu vermeiden. Overfitting bedeutet, dass das Model Regeln und Muster der Trainingsdaten lernt, diese aber nur schlecht oder gar nicht auf neue, bisher ungesehene Daten aus dem Fine-Tuning anwenden kann (Wennker 2020, S. 25). Im Gegensatz dazu steht *Underfitting*, bei dem das neuronale Netz weder die Trainingsdaten noch die Testdaten ausreichend modelliert und Zusammenhänge nur schlecht erfassen kann (Wennker 2020, S. 25).

Zusammenfassend lässt sich die Aussage treffen, dass das letztendliche Training des Models in der Fine-Tuning-Phase dank des Pretrainings schnell vonstattengeht. Die Ergebnisse und das endgültige Model hängen jedoch von den fürs Training gewählten Parametern ab. Spezifischere Eigenheiten der Anwendung von Fine-Tuning in der Praxis werden im Kapitel *Experimente* genauer betrachtet.

2.3.2 Anwendungsgebiete und Weiterentwicklungen

BERTs Flexibilität in Aufbau und Trainingsprozess ermöglicht den Einsatz vieler verschiedener Tasks. Über Textklassifikation, Named-Entity-Recognition und Question Answering kann BERT für verschiedenste NLP-Tasks eingesetzt werden. Die erfolgreiche Anwendung führte zur Verbreitung von und Forschung zu Transformer-ähnlichen Architekturen für NLP-Tasks. Ebenfalls regte sie zu weiterer Entwicklung und Verbesserung von BERT an. Beispiele hierfür sind *RoBERTa (Robustly optimized BERT approach)* und

DistilBERT. Das Model RoBERTa nutzt ebenfalls Transformer-Encoder, wendet im Pre-Training jedoch kein NSP an und ändert den Maskierungsprozess geringfügig ab. Nach einem längeren Training erreicht RoBERTa eine 2 – 20% bessere Performance als BERT (Liu et al. 2019). DistilBERT bricht die Architektur von BERT auf die wichtigsten Aspekte herunter und verwendet nur die Hälfte an Parametern sowie die Hälfte an Schichten. Mit einem kleineren neuronalen Netz kann der Output des großen, vortrainierten Netzes von BERT angeglichen und 97% der Performance beibehalten werden. Dabei ist DistilBERT bis zu viermal schneller als BERT (Sanh et al. 2019). Ist eine bessere Performance oder ein schnelleres Training vonnöten, bieten diese beiden Modelle eine mögliche Alternative zu BERT. Dennoch bietet BERT eine stabile Grundlage, die noch immer hervorragende Ergebnisse liefert und einwandfrei verwendet werden kann, wenn keine der Vorteile der Weiterentwicklungen dringend benötigt werden.

2.3.3 Syntaktisches und semantisches Wissen

BERT offenbart viele Vorteile, mittels derer im Bereich des NLP bei den verschiedensten Tasks hervorragende Ergebnisse und Fortschritte in der automatisierten Sprachverarbeitung erzielt werden, was unter anderem durch das vom Pretraining erzeugte Sprachverständnis bedingt ist. Sowohl in Syntax, Grammatik als auch Semantik besitzt BERT Wissen, welches sich während des Pretrainings aufbaut und sich im Fine-Tuning positiv auf Trainingsdauer und Ergebnisse auswirkt, indem auf vorher aufgebauten Kontext zugegriffen werden kann.

Wie und wie viel syntaktisches Wissen BERT genau aufnimmt, wird momentan genauer erforscht. Wu et al. schreiben hierzu:

> What we found is the "natural" syntax inherent in BERT, which is acquired from selfsupervised learning on plain text. We would rather say our probe complements the supervised probing findings in two ways. First, it provides a lowerbound (on the unsupervised syntactic parsing ability of BERT). By improving this lower-bound, we could uncover more "accurate" information to support supervised probes' findings. Second, we show that when combined with a down-stream application [...], the syntax learned by BERT might be empirically helpful despite not totally identical to the human design. (Wu et al. 2020)

Klar ist jedoch, dass BERT Syntax eher in einer Baumstruktur als linear erfasst. Während des MLM können Wörter, deren Syntax-Bäume große Ähnlichkeiten aufweisen, größere Auswirkungen aufeinander haben, wodurch die Vorhersage maskierter Wörter verbessert wird (s. *Abbildung 10*) (Lin et al. 2019; Wu et al. 2020).

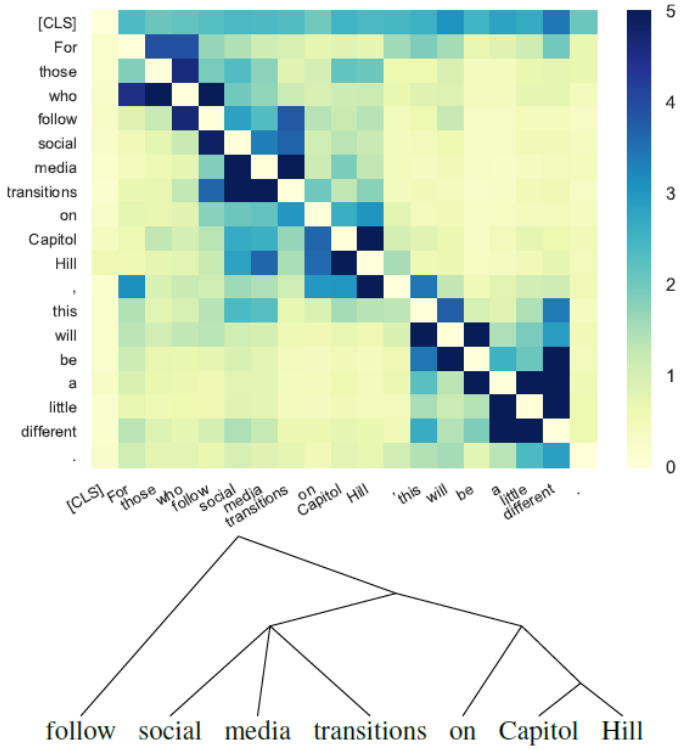

Abbildung 10: Aufnahme syntaktischen Wissens durch BERT in einer Baumstruktur (Wu et al. 2020)

Neben syntaktischem Wissen reichert BERT auch semantisches Wissen an. Es wurde festgestellt, dass während des Pretrainings Wissen zu semantischen Rollen, zu Präferenzen zwischen Entitäten und zu Beziehungen zwischen diesen erzeugt wird (Rogers et al. 2020). Im Gegensatz dazu zeigt BERT sowohl bei der Repräsentation als auch beim Verständnis von Zahlen Defizite. Zahlen, auf die nicht explizit trainiert wurden, werden schlecht verarbeitet (Wallace et al. 2019).

BERT baut in geringem Maße auch „world knowledge“ (Rogers et al. 2020) auf. Dieses Wissen übersteigt jedoch nicht einfache Beziehungen zwischen Entitäten und kann nicht für Schlussfolgerungen angewandt werden. Rogers et al. äußern sich dazu in ihrem Paper „A Primer in BERTology“ von 2020 folgendermaßen:

> […] BERT can „guess“ the affordances and properties of many objects, but can not reason about the relationship between properties and affordances. For example, it „knows“ that people can walk into houses, and that houses are big, but it cannot infer that houses are bigger than people. (Rogers et al. 2020)

Weiter schreiben sie in Bezug auf Poerner et al. und deren Paper „BERT is not a knowledge base (yet): Factual knowledge vs. Name-based reasoning in unsupervised qa“ von 2019:

> Some of BERT's world knowledge success comes from learning stereotypical associations, e.g., a person with an Italian-sounding name is predicted to be Italian, even when it is incorrect. (Rogers et al. 2020)

Bedingt durch die kurze Zeit, die BERT zur Verfügung steht, sind die Fähigkeiten, die in BERTs antrainiertem Wissen liegen, noch wenig getestet und erforscht. Es kann darum keine genaue Aussage dazu getroffen werden, wie sich beispielsweise das Erlernen von Stereotypen, gerade in Bezug auf das Thema dieser Arbeit, auf Ergebnisse von BERT-Models auswirkt.

3 PRAKTISCHER TEIL

Bei BERT handelt es sich um eine noch recht junge Entwicklung und ist im Verhältnis zu anderen Architekturen neuronaler Netze wenig erforscht. Dementsprechend ist in der Praxis oft nicht zu hundert Prozent nachvollziehbar, wie Ergebnisse zu Stande kommen und von welchen Faktoren und Parametern sie abhängen. Es ist daher nötig, weiter zu ergründen, wie BERT Sprache lernt und verarbeitet, indem man damit experimentiert und testet. Nicht nur eine Beobachtung des Trainingsprozesses ist von großer Wichtigkeit. Auch die Datenbasis hat, wie immer im Machine Learning, großen Einfluss auf die letztendlichen Resultate. Infolgedessen wurde im praktischen Teil dieser Arbeit ebenso viel Wert auf die Erstellung des Textkorpus gelegt, wie auf die Ausführung und die Evaluation der Experimente. Im ersten Teil wurden Plenarprotokolle auf ihre Sprache hin untersucht und Daten für die folgenden Experimente gesammelt. Inspiration und einen Leitfaden für den Prozess der Datensammlung lieferte dabei das Paper „Developing a Multilingual Annotated Corpus of Misogyny and Aggression" von Bhattacharya et al. (2020). Im Anschluss wurden mit den gesammelten Daten fünf Experimente durchgeführt, die Auskunft darüber geben sollten, wie BERT für die binäre Textklassifikation von rassistischer Sprache in deutschen Plenarprotokollen eingesetzt werden kann.

Das endgültige Korpus ist mit circa einhundert Datensätzen um einiges kleiner als die Korpora der meisten Studien zu BERT. Bei der Gestaltung und Durchführung der Experimente wurde sich deshalb an Risch et al. und ihrem Vorgehen im Paper „Offensive Language Identification using a German BERT model" von 2019 orientiert. Es wurde sowohl dasselbe deutsche BERT-Model (German BERT, vgl. *3.2.2*) als auch dasselbe Framework (FARM, vgl. *Kapitel 3.2.1*) verwendet. In Experiment 3 und 4 wurden zudem Daten aus dem von Risch et al. erstellten Korpus angewandt. Das Ziel der Orientierung an

ihrem binären „coarse-grained“ (Risch et al. 2019) Textkassifikationstask lag darin, eine gewisse Vergleichbarkeit zu schaffen, um unter anderem feststellen zu können, wie sich die Größe des Korpus auf die Resultate des Models auswirken kann.

3.1 ERSTELLUNG DER KORPORA

Notwendig für das Training eines Sprachmodels zum Zwecke eines bestimmten Tasks ist ein auf diese Aufgabe ausgerichtetes Textkorpus. Im Fall dieser Arbeit war es das Ziel, ein Sprachmodell auf einen binären Textklassifikationstask zu trainieren, so dass es Sätze oder Textabschnitte aus deutschen Plenarprotokollen möglichst fehlerfrei als *rassistisch* (RACISM) oder *nicht rassistisch* (OTHER) labelt. Dafür sollte ein Korpus erstellt werden, das häufige Ambiguität und Implizitheit von rassistischer Sprache in der Politik so detailreich wie möglich erfasst und eine gute Datenbasis liefert, um schlussendlich repräsentative Ergebnisse zu erhalten. Es wurde zudem ein weiteres, etwas größeres Korpus erstellt, um in Experiment 3 (s. *Kapitel 3.3.3*) testen zu können, wie es sich auf die Ergebnisse eines Models auswirkt, wenn dieses mit mehr Daten trainiert wird.

Zu Beginn der Arbeit wurde evaluiert, wie bei der Erstellung der Datensets vorgegangen werden kann. Die Suche nach bereits bestehenden Korpora für rassistische Sprache im politischen Kontext erwies sich als erfolglos, weshalb dieser Ansatz schnell verworfen wurde. Das im Paper von Bhattacharya et al. (2020) beschriebene Vorgehen zur Erstellung ihres Textkorpus konnte Aufschluss über das Vorgehen bei der Sammlung von Daten geben. Im Folgenden soll die Gestaltung des Prozesses der Korpuserstellung beschrieben werden.

3.1.1 Quellen

Für das erste, kleinere Textkorpus (*Korpus1*) wurden ausschließlich Sätze und kurze Textabschnitte aus Plenarprotokollen des Bundestags genutzt. Verwendet wurden die Plenarprotokolle 148 bis 215 der 19. Wahlperiode, also dem Zeitraum zwischen dem 04.03.2020 und dem 04.03.2021. Die Entscheidung, ausschließlich Protokolle aus dieser Zeitspanne zu nutzen, hat zwei Gründe. Zum einen gab es im Jahr 2020 weltweit viele Ereignisse, besonders aber auch in Deutschland, die große Debatten zum Thema Rassismus hervorriefen (vgl. Beschreibung der Ereignisse im Kapitel *Einleitung*). Diese Debatten erstreckten sich bis in den deutschen Bundestag und wurden über Wochen geführt. Die Hoffnung war dementsprechend in diesen Plenarprotokollen verstärkt rassistische Sprache

zu finden und in der begrenzten Bearbeitungszeit der vorliegenden Arbeit möglichst viele verwendbare Textdaten sammeln zu können. Für eine Bestätigung dieser Vermutung fehlen Vergleichswerte mit Plenarprotokollen aus anderen Zeiträumen. Dennoch konnte in manchen Debatten zu bestimmten Ereignissen vermehrt rassistische Sprache festgestellt werden.

Der zweite Grund zur Festlegung eines Zeitraums liegt in der zwangsweisen Einschränkung der Datenmenge begründet. Plenarprotokolle sind im Allgemeinen sehr umfangreich und können je nach Länge der Sitzung eine sehr große Wortanzahl erreichen. Die Plenarprotokolle aus dem genannten Zeitraum haben jeweils eine Wortanzahl, die sich in etwa zwischen 50.000 und 150.000 Wörtern bewegt[10]. Diese aufmerksam durchzulesen und auf rassistische Sprache zu untersuchen, bedeutet einen großen Arbeitsaufwand. Um dem Aufwand Grenzen zu setzen und ihn mit Hinsicht auf die restlichen Aufgaben dieser Arbeit in ein ausgewogenes Verhältnis zu setzen, wurde die Anzahl der berücksichtigten Protokolle wie beschrieben eingeschränkt.

Die Plenarprotokolle stehen frei auf der Webseite des Bundestags zur Verfügung und können in leicht lesbarer Form im PDF-Format heruntergeladen werden. Sie werden bei Plenarsitzungen von Stenograf:innen geführt und enthalten sowohl die einzelnen Reden, die Tagesordnungspunkte und Einwürfe der Bundestagspräsidentin oder des Bundestagspräsidenten sowie Kommentare und Reaktionen der Teilnehmer:innen der Plenarsitzungen (Deutscher Bundestag o.A.). Für computerlinguistische Funktionen erweisen sich die Protokolle im PDF-Format als nicht praktikabel. Dies liegt unter anderem daran, dass das Layout der Protokolle in PDF-Form in zwei Spalten aufgeteilt ist und viele Wörter durch die verkürzte Zeilenlänge mit einer automatischen Silbentrennung aufgeteilt werden. Kopiert man den Text, enthält folglich ein gewisser Teil der Wörter einen Bindestrich. Zudem können einzelne Abschnitte der Protokolle, wie beispielsweise Namen, Parteizugehörigkeit, Reden oder Kommentare nicht einzeln erfasst werden. Eine große Vereinfachung ist daher das Angebot des Bundestages Plenarprotokolle auch im sehr gut zu verarbeitenden XML-Format anzubieten[11]. Plenarprotokolle werden mit sehr detaillierten Metadaten getaggt. Das für diese Arbeit wichtigste Element ist <*rede*>, welches den Namen

[10] Diese Schätzung bezieht sich ausschließlich auf Reden und Kommentare. Weitere Elemente, wie beispielsweise Abstimmungen und Tagesordnungspunkte, wurden nicht mit einbezogen.

[11] Unter folgendem Link stellt der Deutsche Bundestag eine Auflistung der Elemente in Bundestags-Plenarprotokollen im XML-Format für Interessierte zur Verfügung: https://www.bundestag.de/resource/blob/577234/f9159cee3e045cbc37dcd6de6322fcdd/dbtplenarprotokoll_kommentiert-data.pdf (Stand: 18.06.2021).

und die Fraktion der Sprecherin oder des Sprechers sowie die gehaltene Rede selbst enthält. Im Textkorpus wurden weder Namen und Fraktionszugehörigkeiten noch Kommentare und Einwürfe erfasst. So sollte der Fokus auf die Sprache in der Rede selbst gelenkt werden und nicht auf Redner:innen und Parteien. Mittels eines Webscrapers wurden die Reden aus den XML-Dokumenten extrahiert und zur Erstellung der Sets der Trainings- und Testdaten weiter verarbeitet.

Für das zweite, etwas größere Korpus (*Korpus2*) wurde Korpus1 mit Textdaten erweitert. Dazu wurde das *GermEval-Korpus* von Risch et al. (2019) herangezogen und daraus rassistische Sätze und Textausschnitte ausgewählt. Da das GermEval-Korpus ausschließlich Beiträge von Twitter enthält, musste die Social-Media-Sprache korrigiert werden, sprich, es wurden Emojis, Umgangssprache und starke Beleidigungen entfernt und Rechtschreib- und Grammatikfehler ausgebessert. Zudem wurden die Sätze teilweise an politische Sprache angepasst, um eine Ähnlichkeit zur Sprache in Plenardebatten herbeizuführen. Zusätzlich wurden weitere nicht-rassistische Textdaten in Form von Sätzen und Textabschnitten aus den Plenarprotokollen gewählt.

3.1.2 Auswahl der Textdaten aus Plenarprotokollen

Die Annotation der Sätze und Textabschnitte wurde selbst vorgenommen. Ein erster Ansatz der Datenauswahl nach Die- und Wir-Gruppen, wie sie in *Kapitel 2.1.1* beschrieben wurden, wurde nach kurzer Zeit verworfen. In den wenigsten Sätzen konnten klar eine Die- und eine Wir-Gruppen identifiziert werden. Meist stand die oder der Sprecher:in und deren/dessen Fraktion stellvertretend für die Wir-Gruppe, die folglich nicht explizit in den einzelnen Textabschnitten erwähnt wurde (s. beispielsweise Textabschnitt 1 aus *Tabelle 2*). Es hätte also einen größeren Kontext und Hintergrundwissen zur momentanen Rede benötigt, um die Wir-Gruppe zu identifizieren. Als klare Eigenschaft und Struktur zur Erkennung rassistischer politischer Sprache stellten sich Die- und Wir-Gruppen also als unzureichend heraus.

Der zweite und endgültige Ansatz folgte so weit wie möglich dem Vorgehen von Bhattacharya et al.. Sie hatten zur Erstellung ihres Korpus die Datensammlung mit Hilfe von vier Personen vorgenommen. Diese konnten sich über die Entscheidung der Datenauswahl austauschen und darüber diskutieren, um Betriebsblindheit und zu große Subjektivität zu vermeiden (Bhattacharya et al. 2020). Da die Möglichkeit der Zusammenarbeit mit mehreren Personen bei dieser Arbeit nicht bestand, wurden möglichst klare Arbeitsdefinitionen von rassistischer Sprache in einem politischem Kontext erstellt, an denen sich bei der Suche nach

rassistischer Sprache in Plenarprotokollen orientiert werden konnte (s. *Kapitel 2.1)*. Trotz größter Sorgfalt muss bei der Verarbeitung der Daten und der Evaluation der Ergebnisse bedacht werden, dass die Auswahl der Daten in gewisser Weise von subjektiver Ansicht und eigener Weltanschauung beeinflusst sein können.

Es wurde darauf geachtet, dass die zusammengetragenen nicht rassistischen Sätze ebenso Schlagwörter enthalten, die meist in rassistischen Sätzen vorkamen. Beispiele für solche Schlagwörter sind ‚islamistisch', ‚Muslime', ‚Migrationshintergrund' und viele weitere. Durch dieses Vorgehen sollte verhindert werden, dass allein die Verwendung solcher Wörter zur Kategorisierung als rassistisch führt. Stattdessen sollte ein kompletter Satz mit dessen Aufbau und Grammatik verstärkt einbezogen werden, um darauf die Klassifizierungsentscheidung zu basieren. Dies kann an folgenden Textabschnitten näher dargestellt werden:

1	Diese Leute sind aber nicht friedlich, weil sie Muslime sind, sondern obwohl sie Muslime sind. (Jens Maier, AFD; 190. Plenarsitzung; 06.11.2020)
2	Wenn ich mich nicht ganz falsch erinnere, Herr Maier – und damit haben Sie im Grunde die Idee und den Geist Ihres Antrags bestmöglich zusammengefasst –, sagten Sie, dass die Mehrheit der Muslime friedlich sei, nicht weil sie Muslime seien, sondern obwohl sie Muslime seien. Das ist nicht nur zutiefst beleidigend für alle Menschen muslimischen Glaubens in diesem Land und auch für diejenigen, die als muslimisch gelesen und identifiziert werden, es ist auch noch blanker Rassismus und nichts anderes. (Helge Lindh, SPD; 190. Plenarsitzung; 06.11.2020)

Tabelle 2: Zwei Beispiele von Textabschnitten aus der 190. Plenarsitzung. Der erste Abschnitt ist rassistisch, der zweite Abschnitt zitiert den ersten lediglich indirekt und enthält keine rassistische Aussage.

Der Satz in Beispiel 1 ist klar islamophob und damit dem religiös-kulturellen Rassismus zuzuordnen. Der erste Satz des zweiten Beispiels enthält nicht nur mehrfach das gleiche Schlagwort („Muslime"), sondern zitiert Beispiel 1 sogar indirekt, lässt dabei aber insgesamt keinerlei Haltung zum zitierten Satz erkennen. Erst der folgende Satz stellt die Haltung des Redners zur zitierten Aussage dar. Beispiel 2 wurde folglich als nicht rassistisch gelabelt.

Wie bereits in *Kapitel 3.1.1* erwähnt, lag der Fokus zu einem großen Teil auf Debatten zu einschlägigen Themen, um gebündelt rassistische Sprache zu finden. Dennoch wurden alle Plenarprotokolle aufmerksam gelesen, da sich auch in Debatten zu potenziell nicht

rassistischen Themen manchmal rassistische Sprache versteckt (s. Beispiel 5 in *Tabelle 1*). Ebenso wurde versucht Textstellen auszuwählen, die sich klar von sonstiger Hate Speech und gruppenbezogener Diskriminierung sowie allgemein rechtem Gedankengut abgrenzt.

Die Länge der gewählten Textabschnitte bewegt sich bis maximal um die 100 Tokens. Manche Textabschnitte kommen nahe an diesen Grenzwert heran, andere sind mit um die zehn Tokens sehr kurz. Textstellen, die weitaus länger als die erwähnten 100 Tokens waren, konnten nicht beachtet werden, oder mussten, falls möglich, in einzelne Sätze aufgeteilt werden. Es wurde mehrfach beobachtet, dass sich Rassismus hinter einer Aussage oft erst durch Beziehungen zu Textstellen zu erkennen gab, die nicht in unmittelbarer Nähe lagen. Da sich im praktischen Teil der Arbeit für einen satzbasierten Ansatz entschieden wurde, konnte in diesen Fällen auf derart große Textabschnitte nicht geachtet werden. Gleiches gilt für rassistische Sprache, deren Verständnis ein größeres Weltwissen oder Kenntnis des aktuellen Zeitgeschehens benötigt.

Im Allgemeinen wurde versucht, ein Textkorpus zu erstellen, das rassistische Sprache in Plenarprotokollen aus so vielen Blickwinkeln und so umfassend wie möglich repräsentiert und darstellt. Dazu wurde nicht nur die Länge der Textabschnitte variiert, sondern auch die Formen von rassistischer Sprache, wie beispielsweise Islamophobie und Xenophobie. Zudem wurde darauf geachtet, dass spezielle Schlagwörter sowohl in rassistischen wie auch nicht rassistischen Sätzen vorkommen, um unter anderem den Unterschied zwischen Islamophobie und Kritik an Islamismus herauszukristallisieren.

3.1.3 Endgültige Textkorpora

Korpus1 enthält insgesamt 205 Datensätze. Davon wurden 105 Datensätze mit dem Label RACISM und 100 mit dem Label OTHER versehen. Korpus2 umfasst 500 Datensätze von denen 200 mit dem Label RACISM und 300 mit dem Label OTHER versehen sind. In Korpus2 wurde ein Ungleichgewicht zwischen OTHER und RACISM hergestellt, um sich mehr dem Verhältnis des binären „coarse-grained classification task“ von Risch et al. (2019) anzupassen. In Experiment 3 sollte so eine gewisse Vergleichbarkeit geschaffen werden.

Obwohl sich beim Sammeln der Daten für Korpus1 nicht explizit auf eine Partei konzentriert wurde, sondern Ansinnen war, rassistische Sprache verschiedener Parteien zu sammeln, um das Datenset möglichst neutral zu halten, ergab sich am Schluss ein Korpus, welches zum größten Teil aus Textabschnitten der AfD besteht. Parteizugehörigkeiten der Sprecher:innen wurden jedoch nicht erfasst und sind auch für diese Arbeit nicht von Belang. Dennoch muss dieser Aspekt der Datenbasis bei der Auswertung der Ergebnisse mit

einbezogen werden. Die in Korpus1 enthaltenen Daten sind in etwa jeweils zu einer Hälfte religiösem und zur anderen Hälfte kulturellem Rassismus zuzuordnen. Religiöser Rassismus richtet sich dabei ausschließlich gegen den Islam. Kulturell-rassistische Textabschnitte sind im allgemeinen xenophob und gegen Flüchtlinge und Migranten gerichtet. Es wurde in den Plenarprotokollen kein biologischer Rassismus gefunden. Dieser ist somit auch nicht im Korpus enthalten.

Die Daten wurden nicht bereinigt, sprich, es wurde keine Lemmatisierung durchgeführt und Stoppwörter und Zahlen wurden wie auch die Groß- und Kleinschreibung beibehalten[12]. Für die Umsetzung der geplanten Experimente wurden je nach Art des Experiments ein Trainings- und ein Testdatenset benötigt. Dazu wurden die Sätze und Textabschnitte bei Korpus1 und Korpus2 jeweils in zwei Textdateien gesammelt, eine für rassistische und eine für nicht rassistische Textabschnitte und Sätze. Anschließend wurden die beiden Dateien gelabelt und in einer CSV-Datei zusammengeführt, die anschließend im Verhältnis 75:25 in Trainings- und Testdatensets gesplittet wurde. Für beide Korpora stehen somit ein Trainings- und ein Testdatenset zur Verfügung.

3.2 IMPLEMENTATION

Die Implementation der Experimente richtete sich nach dem Vorgehen im von Risch et al. verfassten Paper (2019). Die Forscher:innen nutzten dafür das Framework *FARM* und das deutsche BERT-Model *German BERT*.

3.2.1 FARM

Das Open-Source-Framework FARM (*Framework for Adapting Representation Models*) wird von *deepset.ai*, einer Agentur für Machine Learning mit Schwerpunkt NLP, zur Verfügung gestellt. Es ermöglicht ein komfortables Modelling und Training mit verschiedenen BERT-Modellen sowie eine einfache Auswertung der endgültigen Ergebnisse. Neben der Textklassifikation werden noch mehrere andere Standardtasks unterstützt, wie zum Beispiel Named Entity

[12] Dies gilt für die ursprünglichen Korpora. In Experiment 2 (s. *Kapitel 3.3.2*) wurde getestet, wie sich Lemmatisierung auf die Ergebnisse eines Models auswirkt und das verwendete Korpus1 dementsprechend vor dem Training lemmatisiert.

Recognition und Question Answering. Es ist vollständig kompatibel mit den Transformer-Modellen von Huggingface[13] (FARM GitHub 2019).

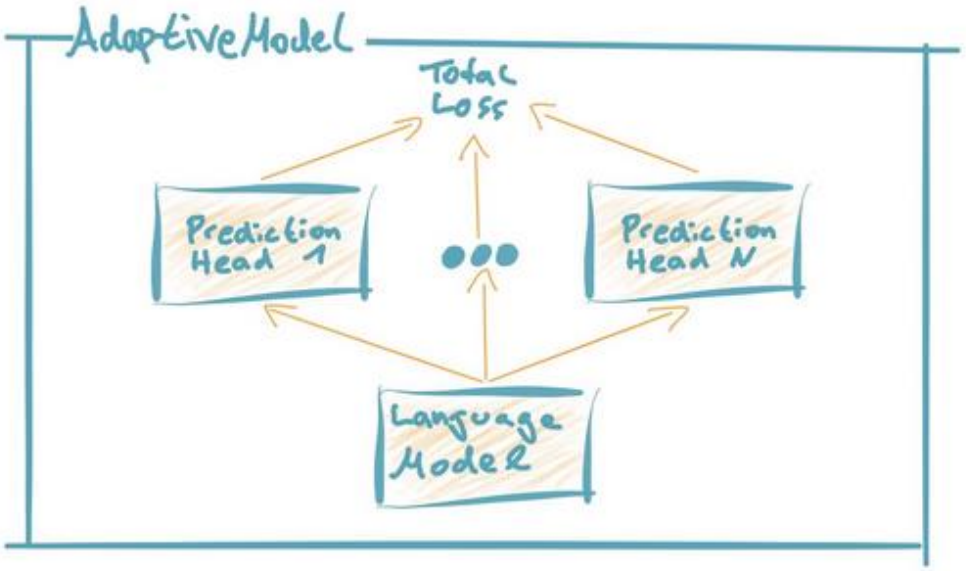

Abbildung 11: FARMs Adaptive Model, bestehend aus einem vortrainierten Sprachmodell und einer gewünschten Anzahl von Prediction Heads (FARM Dokumentation 2019a)

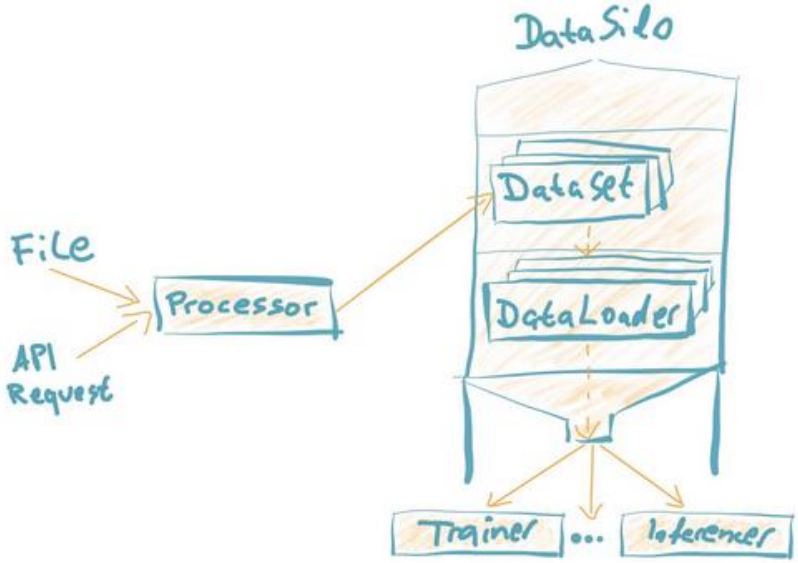

Abbildung 12: Data Handling mit FARM: Die Daten werden als Train- und Testset, falls vorhanden auch als Dev-Set in ein DataSilo geladen. Von diesem aus können die Daten für verschiedene Funktionen geladen werden (FARM Dokumentation 2019b).

[13] Das Unternehmen Huggingface stellt eine Vielzahl von Transformer- und BERT-Modellen umsonst unter folgendem Link zur Verfügung: https://huggingface.co/models (Stand: 26.06.2021).

Der große Vorteil des FARM-Frameworks ist sein modularer Aufbau, der *Prediction Heads*, standardisierte Klassen für verschiedene Tasks, beispielsweise für Named Entity Recognition oder Textklassifizierung, und das vortrainierte Sprachmodell voneinander trennt. Durch die daraus resultierende Flexibilität können mehrere Prediction Heads mit einem gewünschten vortrainierten Model (BERT, RoBERTa, DistilBERT, o.a.) kombiniert werden. Gemeinsam bilden sie das *Adaptive Model* (s. *Abbildung 11*), welches ermöglicht, leicht zwischen Sprachmodellen zu wechseln und Änderungen am System einspielen zu können. Der Loss wird während des Trainings durch das gesamte Adaptive Model backpropagiert (FARM Dokumentation 2019a).

Ebenfalls viele Vorteile bietet das Data Handling mit FARM. Durch Erweiterung der *Processor*-Klasse kann ein eigenes Datenset verwendet werden[14]. Mittels des Processors wird das Datenset in Pytorch-Datensets umgewandelt und lädt das Train-, Test- und Dev-Set in ein *DataSilo*. Falls kein Dev-Set vorhanden ist, wird es durch Teilen des Train-Sets erzeugt (FARM Dokumentation 2019b). Anschließend können die Daten aus dem DataSilo geladen und weiterverwendet werden (s. *Abbildung 12*).

3.2.2 German BERT

Neben FARM stellt deepset.ai auch das deutsche BERT-Model German BERT zur Verfügung (deepset.ai 2019). German BERT wurde auf den aktuellen deutschen Wikipedia-Dump, den OpenLegalData-Dump und auf Nachrichtenartikel trainiert. Insgesamt enthält das Datenset ungefähr 12 Gigabyte Textdaten, die mit *spaCy*[15] bereinigt wurden. German BERT verwendet für die Tokenisierung die *sentencepiece*-Bibliothek[16]. Mit den gewählten Hyperparameter-Einstellungen [17] dauerte das Training circa neun Tage. Zum Hyperparameter-Tuning und zur Evaluierung der Ergebnisse schreiben die Entwickler:innen in ihrem Blog:

[14] In der vorliegenden Arbeit wurde der TextclassificationPredictionHead mit einem erweiterten Processor verwendet, um das eigene Datenset nutzen zu können.

[15] Spacy ist eine Bibliothek, mit der sich Textdaten für NLP-Tasks bereinigen lassen. Für die deutsche Sprache stehen unter folgendem Link eigene Pakete zur Verfügung: https://spacy.io/models/de (Stand: 07.07.2021).

[16] Vgl. https://github.com/google/sentencepiece (Stand: 07.07.2021)

[17] Batch Size 1024, Learning Rate 1e-4, maximale Sequenzlänge 128 und später 512, n_steps 810000 (deepset.ai 2019).

> There does not seem to be any consensus in the community about when to stop pre-training or how to interpret the loss coming from BERT's self-supervision. We took the approach of BERT's original authors and evaluated the model performance on downstream tasks. (deepset.ai 2019)

Downstream-Tasks, mit denen die Performance von German BERT getestet wurden, waren u. a. *germEval18Fine* und *germEval18Coarse* mit Macro-F1-Score, welche im Paper von Risch et al. näher beschrieben werden (Risch et al. 2019). Die weiteren getesteten Tasks waren *germEval14* und *CONLL03* mit Seq-F1-Score für Named Entity Recognition und *10kGNAD* mit Accuracy für Document Classification. Alle diese Tasks ergaben trotz ungenauem Hyperparameter-Tuning stabile Resultate (deepset.ai 2019).

Model	germEval18Fine	germEval18Coarse	germEval14	CONLL03	10kGNAD
multilingual cased	0.441	0.71	0.834	0.792	0.888
multilingual uncased	0.461	0.731	0.823	0.786	0.901
German BERT cased (**ours**)	**0.488**	**0.747**	**0.84**	**0.804**	**0.905**

Tabelle 3: Ergebnisse von German BERT in Kombination mit verschiedenen Tasks im Vergleich mit anderssprachigen BERT-Models (deepset.ai 2019)

3.3 EXPERIMENTE

Bei der Implementation der Experimente wurde sich am binären „coarse-grained" Textklassifikationstask von Risch et al. (2019) orientiert. Zuerst geplante Tasks, die mehr auf den Inhalt der Daten eingehen, wie beispielsweise ein Multilabel-Task, der rassistische Sprache nach der Art des Rassismus (Islamophobie, Antisemitismus, Xenophobie, Antiziganismus, biologisch, etc.) einteilt, wurden verworfen, nachdem festgestellt wurde, dass dafür zu wenig Daten zur Verfügung stehen. Ebenso wurde eine Einteilung nach Die- und Wir-Gruppen sowie nach einer rassistischen Behauptung verworfen, da die Wir-Gruppe in den meisten Fällen nicht klar identifizierbar und dazu mehr Kontext vonnöten gewesen wäre (vgl. *Kapitel 3.1.2*).

Insgesamt wurden fünf Experimente durchgeführt[18], mit denen evaluiert werden sollte, wie erfolgreich die Anwendung von BERT bei einem binären Textklassifikationstask mit

[18] Das Training in den ersten drei Experimenten wurde mit dem MLFlowLogger von deepset.ai getrackt. Die Ergebnisse sind unter folgendem Link einsehbar: https://public-mlflow.deepset.ai/#/experiments/410 (Stand:07.07.2021).

Daten aus deutschen Plenarprotokollen sein kann. Im ersten Experiment wurden verschiedene Einstellungen von Hyperparametern im Fine-Tuning getestet. Das zweite und dritte Experiment behandelten die Auswirkungen des Preprocessings der Daten sowie eines größeren Datensets auf die Ergebnisse des Models. Im vierten und fünften Experiment wurden die Fähigkeiten des Models in Bezug auf das Erkennen rassistischer Sprache mittels synthetischer Daten getestet. Es sollte erkennbar gemacht werden wie explizit formuliert rassistische Aussagen sein müssen, um erkannt zu werden und welche Form von Rassismus am ehesten erkannt wird[19].

3.3.1 Experiment 1 – Fine-Tuning

Mit dem ersten Experiment sollte festgestellt werden, welche Fine-Tuning-Einstellungen des bestehenden binären Textklassifizierungstasks die besten Ergebnisse bringen. Bei der Wahl der Konfigurationen orientierte man sich am Aufsatz „How to Fine-Tune BERT for Text-Classification“ (Sun et al. 2019), am Paper von Risch et al. zu Offensive Language (2019) sowie am ursprünglichen BERT-Paper von Devlin et al. (2019). Risch et al. richteten sich beim Fine-Tuning stark nach Devlin et al. und wählten die gleichen Parameter.

Insgesamt wurden fünf Modelle mit unterschiedlichen Konfigurationen auf den kleineren Korpus1 trainiert. Die Daten wurden nicht vor dem Training bereinigt, sondern jeweils die Hyperparameter *Batch Size, Epochs* und *Learning Rate* angepasst. Devlin et al. verweisen auf eine gesteigerte Empfindlichkeit des neuronalen Netzes bei kleinen Datenmengen in Hinsicht auf die Wahl der Hyperparameter. Sie empfehlen ausführliche Versuche mit verschiedenen Einstellungen und die anschließende Wahl der erfolgreichsten Kombination (Devlin et al. 2019). Im ersten Experiment wurden darum fünf verschiedene Konfigurationen getestet (s. *Tabelle 4*).

	Model 1	**Model 2**	**Model 3**	**Model 4**	**Model 5**
Batch Size	8	16	16	4	8
Epochs	2	3	5	5	3
Learning Rate	2e-5	3e-5	3e-5	4e-4	5e-5

[19] Der Software-Code kann unter folgendem Link auf GitHub eingesehen werden: https://github.com/JuNeHasIssues/racism-in-german-plenary-protocols.

Tabelle 4: Übersicht der Hyperparameter, mit denen die fünf Modelle aus Experiment 1 trainiert wurden

Learning Rate

Aufgrund der Auswirkungen der Learning Rate auf das Catastrophic Forgetting richtete man sich bei Model 1 unter anderem nach Sun et al.. Sie weisen in ihrem Paper auf eine Begünstigung des Catastrophic Forgettings durch eine aggressive Learning Rate von 4e-4:

> We find that a lower learning rate, such as 2e-5, is necessary to make BERT overcome the catastrophic forgetting problem. With an aggressive learn rate of 4e-4, the training set fails to converge. (Sun et al. 2019)

Um dies mit den verwendeten Daten zu testen und da Devlin et al. in ihrem ursprünglichen BERT-Paper Learning Rates von 2e-5, 3e-5 und 5e-5 verwenden, wurden diese vier Rates in den Tests verwendet.

Epochs

Da Devlin et al. in ihrem Aufsatz zum Fine-Tuning meist kleine Anzahlen, wie zwei bis fünf Epochen wählten, wurde sich bei den Models 1 bis 5 danach gerichtet und ebenso kleine Epochenanzahlen verwendet (Devlin et al. 2019). Nachdem bei Model 2 ein kleinerer Test-Loss als bei Model 1 errechnet wurde, wurde die Epochenzahl auf 5 erhöht, um zu testen, ob die zusätzlichen Iterationen den Loss noch weiter verringern können.

Batch Size

Beim Training der fünf Models wurden Batch Sizes von 4, 8 und 16 getestet. Diese Werte ergaben sich aus früheren Tests, bei denen festgestellt wurde, dass eine Batch Size, die den Wert 16 übersteigt, die verwendete GPU überlastet. Darum wurden drei verschiedene Batch Sizes gewählt, die nicht größer als 16 sind.

max_seq_len

Für alle Models wählte man die maximale Sequenzlänge von $BERT_{BASE}$, welche 512 Tokens beträgt (Sun et al. 2019; Devlin et al. 2019). Da keine der verwendeten Textabschnitte dieses Maximum überschreitet, wurden auch keine Daten unbeabsichtigt gekürzt.

	Model 1		Model 2		Model 3		Model 4		Model 5	
	macro avg	**weighted avg**	**macro avg**	**weighted avg**	**macro avg**	**weighted avg**	**macro avg**	**weighted avg**	**macro avg**	**weighted avg**
precision	0.7048	0.7133	0.7300	0.7417	0.7993	0.8116	0.2155	0.1858	0.6382	0.6536
recall	0.7085	0.7069	0.7333	0.7241	0.8036	0.7931	0.5000	0.4310	0.6273	0.6034
f1-score	0.7047	0.7082	0.7238	0.7251	0.7929	0.7938	0.3012	0.2597	0.6005	0.5957
accuracy	0.7069		0.7241		0.7931		0.4310		0.6034	
test loss	*0.5935*		*0.5561*		*0.7959*		*0.6977*		*2.3911*	
train loss	*0.167*		*0.503*		*0.001*		*0.671*		*0.004*	

Tabelle 5: Ergebnisse aller Models aus Experiment 1

Model 1 stellt den Ausgangspunkt des Versuchs dar, die erfolgreichsten Einstellungen von Batch Size, Epochenanzahl und Learning Rate zu finden und das neuronale Netz so zu konfigurieren, dass es die bestmöglichen Ergebnisse erzielt. Es zeigt sich, dass grundsätzlich gute Ergebnisse bei Accuracy und F1-Score erreicht werden, dennoch muss beachtet werden, dass die Beurteilung, ob die Ergebnisse gut oder schlecht sind, von vielen Faktoren abhängt (s. Kapitel *Diskussion*). Der Test-Loss liegt bei 0.5935, weist jedoch eine große Differenz zum Train-Loss von 0.167 auf, was auf Overfitting schließen lässt. Besonders stark ist dies bei Model 5 mit einem Test-Loss von 2.3911 und einem Train-Loss von 0.004 zu beobachten. Im Durchschnitt erreicht der Test-Loss bei allen Models mittelmäßige Werte. Das Ziel ist, den Test-Loss möglichst weit in Richtung 0 zu bringen und diesen dem Train-Loss dabei anzunähern. Die beste Möglichkeit, das zu erreichen, wäre, BERT mit mehr Trainingsdaten zu trainieren. Wenn jedoch, wie hier gegeben, nur eine begrenzte Menge an Daten zur Verfügung steht, kann versucht werden, mit einem Drop Out dem Overfitting entgegenzuwirken (Howard und Ruder 2018). Bei allen fünf Models stand der Dropout-Wert auf 0.1[20]. Neben Model 1 zeigen auch Model 3 und 5 starke Anzeichen von Overfitting. Obwohl Model 3 bei F1-Score und Accuracy die besten Werte erreicht, spricht die große Differenz zwischen Train- und Test-Loss für eine schlechte Verlässlichkeit dieser Ergebnisse und eine Instabilität des Sprachmodells. Infolgedessen erfolgten erneut zwei Tests mit den Konfigurationen von Model 3 mit jeweils einem Dropout von 0.3 und 0.5.

[20] Bei FARM wird der Dropout über den Parameter *embeds_dropout_prob* im *AdaptiveModel* reguliert.

	Model 3 (DO 0.1)		Model 3.1 (DO 0.3)		Model 3.2 (DO 0.5)	
	macro avg	**weighted avg**	**macro avg**	**weighted avg**	**macro avg**	**weighted avg**
precision	0.7993	0.8116	0.6552	0.6647	0.6485	0.6552
recall	0.8036	0.7931	0.6582	0.6552	0.6485	0.6552
f1-score	0.7929	0.7938	0.6535	0.6568	0.6485	0.6552
accuracy	0.7931		0.6552		0.7931	
test loss	0.7959		0.6171		0.6437	
train loss	0.001		0.461		0.79	

Tabelle 6: Die durchschnittlichen Ergebnisse von Model 3 mit einem Dropout-Wert von 0.1 sowie Model 3.1 und 3.2 mit jeweils einem Dropout von 0.3 beziehungsweise 0.5

Die Ergebnisse (s. *Tabelle 6*) zeigen zwar eine Verbesserung des Test-Loss und eine Annäherung des Train-Loss bei Model 3.1, jedoch verschlechtern sich Accuracy und Macro-F1-Score. Sie sinken im Verhältnis zu Model 3. Model 3.2 erreicht die Accuracy von Model 3, verschlechtert sich jedoch sowohl bei Train- als auch bei Test-Loss. Zusätzlich übersteigt hier der Train-Loss den Test-Loss, was auf Underfitting hinweist. Ein Dropout von 0.5 bedeutet allerdings auch, dass die Anzahl der Neuronen im neuronalen Netz halbiert werden, was eine eklatante und gegebenenfalls zu starke Verringerung der Komplexität des neuronalen Netzes bedeutet. Somit erscheinen die starken Unterschiede der Werte logisch und nachvollziehbar.

Die schlechtesten Werte aller Models hat Model 4 (s. *Tabelle 5*), was sich nach Sun et al. mit der aggressiven Learning Rate von 4e-4 erklären lässt. Betrachtet man die Funktion des Train-Loss (s. *Abbildung 13*), ist zu erkennen, dass sie divergiert, was letztendlich bedeutet, dass sie immer wieder am Optimum vorbeispringt (Sun et al. 2019).

Als Model mit der größten Stabilität erscheint Model 2. Trotz nur geringfügig schlechterer Werte bei Accuracy und Macro-F1-Score im Vergleich zu Model 3 erreicht es doch mit einem Macro-F1-Score von 72,38% und einer Accuracy von 72,41% gute Ergebnisse. Damit kommt der F1-Score dem Ergebnis des „corse-grained" Task von Risch et al., bei dem 76,4 % erreicht wurden, sehr nahe. Hinzu kommt, dass bei Model 2 Train- und Test-Loss sehr nah beieinander stehen. Der Train-Loss ist zwar mit 0,503 geringfügig kleiner als der Test-Loss (0,5561) und weist damit ein wenig Overfitting auf, dennoch bietet Model 2 damit mehr Stabilität als Model 3.

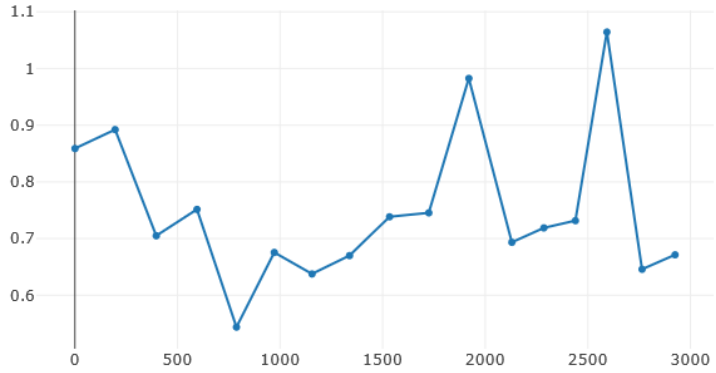

Abbildung 13: Divergierende Loss-Funktion des Train-Loss von Model 4. Die y-Achse stellt den Loss und die x-Achse den zeitlichen Verlauf des Trainings dar.

	Model 2		**Model 3**	
	OTHER	**RACISM**	**OTHER**	**RACISM**
precision	0.6452	0.8148	0.7097	0.8889
recall	0.8000	0.6667	0.8800	0.7273
f1-score	0.7143	0.7333	0.7857	0.8000

Tabelle 7: Aufschlüsselung von Precision, Recall und f1-Score nach den Labels

In der Aufschlüsselung der F1-Scores nach Labels lässt sich erkennen, dass sowohl bei Model 2 als auch bei Model 3 die Klassifizierung mit dem Label RACISM ein wenig verlässlicher ist, als die mit dem Label OTHER. Model 3 erreicht zudem beim Label RACISM den höchsten F1-Score von 80%. Da Model 2 jedoch mehr Stabilität aufweist, wird es in den folgenden Experimenten weiterverwendet.

3.3.2 Experiment 2 – Auswirkungen von Lemmatisierung

Bei einer Lemmatisierung wird ein Wort auf seine Grundform reduziert, sprich, Zeiten, Plural, Fälle, Konjunktionen u.a. werden entfernt (Kupietz und Schmidt 2018, S. 44; Chakrabarty et al.). Aus ‚gelernt' wird durch Lemmatisierung beispielsweise ‚lernen' und aus ‚Häuser' wird ‚Haus'. Dies kann den Vorteil haben, dass komplexe Sprachen mit vielen Flexionen und Konjunktionen vereinfacht werden und gegebenenfalls syntaktisch und semantisch leichter zu verarbeiten sind. Lemmatizer nutzen lexikale Quellen, um ein Wort

auf seine Grundform zurückzuführen (Chakrabarty et al.). Je nachdem welcher Lemmatizer genutzt wird, kann es allerdings passieren, dass nicht die zum Kontext passende, sinnvolle Grundform gewählt wird und somit semantische Zusammenhänge verloren gehen. Für Experiment 2 wurde ein deutscher Lemmatizer von spaCy genutzt. Er basiert auf dem deutschen Textkorpus ‚de_core_news_sm'[21]. Dieser Lemmatizer von spaCy reduziert stets alle Personalpronomen auf ‚ich', auch wenn im Text ein Personalpronomen im Plural steht. Es stellt sich folglich die Frage, ob durch Lemmatisierung semantische Zusammenhänge und Beziehungen zwischen Entitäten verloren gehen und ob das Fine-Tuning mit lemmatisierten Daten schlechtere Ergebnisse liefert als jenes mit nicht lemmatisierten Daten.

Um diese Frage beantworten zu können, wurden die Konfiguration von Model 2 aus Experiment 1 für das Training auf lemmatisierte Daten angewandt und die Ergebnisse anschließend mit dem bereits von Model 2 erreichten Resultat verglichen.

Feststellbar sind geringfügig niedrigere F1-Scores des auf lemmatisierte Daten trainierten Models (‚Lemma') im Gegensatz zum Model, das auf nicht lemmatisierte Daten trainiert wurde (Model 2 oder ‚No Lemma'). So erreichte Model ‚Lemma' einen Macro-F1-Score von 68,51% im Vergleich zu den 72,38% des Models ‚No Lemma' (s. *Tabelle 8*). Beide Models erreichen eine Accuracy von 72,41%. Im Gegensatz zu Model ‚No Lemma' zeigt Model ‚Lemma' Zeichen von Overfitting, da der Test-Loss bei weitem größer ist als der Train-Loss.

	No Lemma		**Lemma**	
	macro avg	**weighted avg**	**macro avg**	**weighted avg**
precision	0.7300	0.7417	0.6868	0.6901
recall	0.7333	0.7241	0.6876	0.6852
f1-score	0.7238	0.7251	0.6851	0.6855
accuracy	0.7241		0.7241	
test loss	0.5561		0.7061	
train loss	0.503		0.195	

Tabelle 8: Die durchschnittlichen Ergebnisse einmal ohne und einmal mit Lemmatisierung

[21] Vgl. https://spacy.io/models/de (Stand:08.07.2021)

	No Lemma		Lemma	
	OTHER	**RACISM**	**OTHER**	**RACISM**
precision	0.6452	0.8148	0.6429	0.7308
recall	0.8000	0.6667	0.7200	0.6552
f1-score	0.7143	0.7333	0.6792	0.6909

Tabelle 9: Die Ergebniswerte ohne und mit Lemmatisierung aufgeschlüsselt nach Labeln

Ein Blick auf die nach Labeln aufgeteilten Resultate bestätigt die bisherige Beobachtung, dass Model ‚Lemma' einen geringfügig schlechteren F1-Score aufweist als Model ‚No Lemma'. Die Unterschiede zwischen beiden Labels sind dabei minimal und das Verhältnis von RACISM zu OTHER ist bei beiden Models nahezu gleich. Dennoch wird bei beiden das Label RACISM ein wenig besser erkannt als OTHER.

Es lässt sich die Aussage treffen, dass sich eine Lemmatisierung des Datensets, auf das trainiert wird, negativ auf die Ergebnisse auswirken kann. Vermutlich gehen durch die Reduzierung aller Wörter auf ihre Grundform Zusammenhänge und Kontext verloren, welche jedoch für eine korrekte Erfassung der Semantik eines Textes durch das kontextuelle BERT-Model vonnöten sind.

3.3.3 Experiment 3 – Auswirkungen eines größeren Datensets

Mit 205 Datensätzen in Korpus1 und 500 Datensätzen in Korpus2 sind die selbst erstellten Datensets im Verhältnis zu Korpora anderer mit BERT arbeitenden Studien und Experimente extrem klein (vgl. *Kapitel 3.1.3*). Risch et al. (2019) nutzten beispielsweise ein Korpus von 3980 Tweets, von denen 1282 als OFFENSIVE und 2698 als OTHER gelabelt wurden. Korpus2 enthält eine ähnliche Verteilung der Labels.

In ihrem Paper „How to Fine-Tune BERT for Text Classification" schreiben Sun et al. (2019) zu den Ergebnissen eines Experiments mit verschieden großen Datensets:

> One of the benefits of the pre-trained model is being able to train a model for downstream tasks within small training data. […] This experiment result demonstrates that BERT brings a significant improvement to small size data. (Sun et al. 2019)

Die Größen der Datensets, die für alle Experimente von Sun et al. genutzt wurden, bewegen sich zwischen 500 und 70000 Datensätzen. Die Aussage, BERT bringe deutliche

Verbesserungen bei kleinen Datensätzen, bezieht sich dementsprechend nicht auf Datensets mit unter 500 Datensätzen.

In diesem Experiment wurde getestet, ob sich bei verschieden großen Korpora bis 500 Datensätzen erhebliche Unterschiede in den Ergebnissen erkennen lassen. Es wurden zwei Versuche mit Korpus2 und den Hyperparameter-Konfigurationen der Models 1 und 2 aus Experiment 1 (s. *Tabelle 1*) durchgeführt, bei dem Korpus1 verwendet wurde. Eine solche Auswahl wurde getroffen, um durch unterschiedliche Konfigurationen bei allen drei Hyperparametern gute Vergleichswerte zu erhalten.

Mit den Hyperparametern wie sie in *Tabelle 10* zu sehen sind, erreichte Test 1 mit Korpus2 einen Macro-F1-Score von 62,05%. Dagegen erreichte Test 2 nach 100 Batches 93,95%, jedoch endgültig nach 118 Batches nur 59,79% (s. *Tabelle 11*). Beide Tests zeigen Zeichen von Overfitting, da jeweils der Test-Loss weitaus größer ist als der Train-Loss.

	Test 1	**Test 2**
Batch Size	8	16
Epochs	2	3
Learning Rate	2e-5	3e-5

Tabelle 10: Hyperparameter beider Tests mit einem größeren Datenset

	Test 1		**Test 2.1**		**Test 2.2**	
	macro avg	**weighted avg**	**macro avg**	**weighted avg**	**macro avg**	**weighted avg**
precision	0.6928	0.7279	0.9473	0.9498	0.6524	0.6829
recall	0.6700	0.6240	0.9325	0.9500	0.6400	0.6000
f1-score	0.6205	0.6132	0.9395	0.9496	0.5979	0.5921
accuracy	0.6240		0.9500		0.6000	
test loss	2.4556		0.3048		2.4322	
train loss	0.0004				0.0003	

Tabelle 11: Durchschnittlichen Ergebniswerte von Test 1 sowie Test 2 nach 100 Batches und 118 Batches. Beide Tests wurden auf Korpus 2 trainiert.

	Test 1		**Test 2.1**		**Test 2.2**	
	OTHER	**RACISM**	**OTHER**	**RACISM**	**OTHER**	**RACISM**
precision	0.8684	0.5172	0.9412	0.9535	0.8049	0.5000
recall	0.4400	0.9000	0.8889	0.9762	0.4400	0.8400
f1-score	0.5841	0.6569	0.9143	0.9647	0.5690	0.6269

Tabelle 12: Ergebniswerte von Test 1 und Test 2 aufgeschlüsselt nach Labeln

Beide Tests erreichen einen besseren Macro-F1-Score beim Label RACISM als bei OTHER (s. *Tabelle 12*). Dieser Umstand könnte von der Verteilung der Labels RACISM zu OTHER in Korpus2 herrühren (2:3). Die Models, die mit Korpus1 trainiert wurden, der fast ein Verhältnis von 1:1 aufweist, haben über die Labels hinweg geringfügig ausgeglichenere Ergebnisse (s. *Tabelle 7*).

Fazit ist, Model 1 und 2 aus Experiment 1 mit dem kleineren Korpus1 erreichen bessere Ergebnisse, als die Models, die auf den etwas größeren Korpus2 trainiert wurden. Dies liegt entgegen der Erwartung, größere Datensets würden bessere Ergebnisse bedeuten. Auch wurde der von Risch et al. erreichte Macro-F1-Score von 76,4% verfehlt. Für einen direkten Vergleich der Ergebnisse beider Arbeiten müssten mehr Gegebenheiten als die Verteilung der Labels im Textkorpus übernommen werden, beispielsweise die Größe des Datensets.

3.3.4 Experiment 4 – Implizit vs. explizit

Nachdem sich mit Experiment 1 nachweisen ließ, dass auch mit einem kleinen Datenset gute Metriken erzielt werden können, sollte Model 2 in Experiment 4 auf die Probe gestellt werden. Um herauszufinden, wie explizit rassistische Aussagen sein müssen, damit sie vom Sprachmodell als solche erkannt zu werden, wurden 15 gelabelte Textabschnitte im Inference-Modus [22] getestet. Die Textabschnitte enthalten zu je einem Drittel unterschiedliche Rassismus-Levels (s. *Tabelle 13*) [23]:

1. Expliziter Rassismus mit sehr deutlichen Beleidigungen und Ausdrücken (dunkelblau): Die Textabschnitte wurden aus dem Korpus von Risch et al. kopiert und abgeändert. Es handelt sich um Social-Media-Sprache und nicht um politische Sprache, wie sie in Plenarprotokollen verwendet wird.
2. Impliziter Rassismus ohne Ausdrücke (hellblau): Diese fünf Textabschnitte sind Sätze aus Plenarprotokollen, die miteinander vermischt und umformuliert wurden.

[22] Vgl. https://farm.deepset.ai/api/running.html#module-farm.infer (Stand: 08.07.2021)

[23] Disclaimer: Die Textabschnitte enthalten extrem beleidigende Ausdrücke, wie beispielsweise das N-Wort. Es soll verdeutlicht werden, dass keinem der Tweets, die aus dem Korpus von Risch et al. kopiert wurden, Beleidigungen hinzugefügt wurden. Lediglich wurden teilweise Beleidigungen entfernt und Rechtschreib- und Satzbaufehler korrigiert. Die Textabschnitte spiegeln nicht die Ansichten und Meinungen der Autorin wieder und sind nur zum Zweck dieses Experiments ausgewählt worden.

3. Textabschnitte, die nach der Arbeitsdefinition als nicht rassistisch gelabelt worden wären (grau): Sie stammen ebenfalls aus Plenarprotokollen und wurden leicht abgeändert.

	text	label	prediction	probability
1	Pfui Teufel. Die Weiber sollte man nach Afrika mit samt Negern abschieben!	RACISM	RACISM	0.7717
2	Die Afrika-Sache fliegt uns bald um die Ohren. Wir sind übersät mit afrikanischem Unkraut und das werden wir nie wieder los!	RACISM	OTHER	0.9097
3	Und deutsche Schlampen lassen sich immer häufiger mit ekelhaften Arabern ein.	RACISM	RACISM	0.9428
4	In unserer Nachbarschaft wohnt eine weitere deutsche Familie. Vor ein paar Jahren war das noch ganz anders, bevor bei uns die Deutschen verdrängt wurden. Jetzt kann man sich bei den ganzen Moslems und Schwarzafrikanern abends gar nicht mehr auf die Straße trauen.	RACISM	RACISM	0.9354
5	Solche Menschen fordern, dass Deutschland „bunt" wird. Bald siehts hier aus wie in Innerafrika.	RACISM	RACISM	0.9549
6	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in migrantischen Gebieten.	RACISM	RACISM	0.9506
7	Eine lebenswerte islamische Gesellschaft gibt es nirgends.	RACISM	RACISM	0.9354
8	Eine Therme in Duisburg etwa wollte Frauen untersagen, knappe Bikinis zu tragen: Freizügigkeit werde nicht von allen toleriert. So geht es los. Bald sitzen wir alle in der Koranschule!	RACISM	OTHER	0.7662
9	Weil man, wenn man sich nachts in Frankfurt im Bahnhofsviertel bewegt, bei den ganzen dunklen Gesichtern einfach nicht mehr sicher fühlt.	RACISM	OTHER	0.7301
10	Es bräuchte jetzt doch Solidarität mit den schon Geschwächten. Aber während deutsche Rentner benachteiligt werden, geht es den ganzen Wirtschaftsflüchtlingen wunderbar.	RACISM	RASCISM	0.8177
11	Sie sagen zwar Menschen, die dem Islam angehören oder aus Afrika kommen, wären 'dümmer', aber selbstverständlich stimmt das nicht!	OTHER	RACISM	0.7743
12	Sie spinnen doch! Wie kann man sowas sagen?	OTHER	OTHER	0.9183
13	Die Grünen sind antideutsch! Sie sind gegen ihr eigenes Volk. Wenn man so etwas nicht Volksverrat nennen kann, was denn dann?	OTHER	OTHER	0.5694
14	Und doch handelt Frankreich jetzt endlich. Es gibt Razzien gegen Islamisten, Abschiebungen von Gefährdern, Schließung radikaler Moscheen.	OTHER	RASCISM	0.6543
15	Wir müssen Lehren aus den Attentaten ziehen, müssen die Wertevermittlung sichern, und – Frankreich macht es vor – wir müssen endlich dem Islamismus den Kampf ansagen und handeln, bevor es auch hier zu spät ist.	OTHER	OTHER	0.9480

Tabelle 13: Fünfzehn unterschiedlich explizit rassistische Textbeispiele

Von den Sätzen der ersten Kategorie wurden alle bis auf einen richtig als rassistisch erkannt. Lediglich der zweite Satz wurde mit einer Wahrscheinlichkeit von 90,97% als OTHER gelabelt. Zum Grund der falschen Kategorisierung können nur Vermutungen angestellt werden. Unter Umständen wurde die Bezeichnung „afrikanische[s] Unkraut" vom Sprachmodell nicht als Bezug auf eine Menschengruppe und als deren Beleidigung

verstanden. Zudem fehlt es scheinbar an weltpolitischem Kontext, der dem Model hätte vermitteln können, was genau mit der „Afrika-Sache“ gemeint ist.

In der zweiten Kategorie wurden die Abschnitte 8 und 9 fälschlicherweise mit OTHER gelabelt. Alle anderen Textabschnitte wurden korrekt der Kategorie RACISM zugeordnet. Textabschnitt 8 ist islamophob, es wird aber keine klare Aussage gegen den Islam getroffen, wie beispielsweise in Satz 7. Stattdessen wird eine Metapher verwendet, die scheinbar für eine angebliche Islamisierung in Deutschland stehen soll: „Bald sitzen wir alle in der Koranschule!“ Für eine korrekte Klassifizierung durch das Sprachmodell war die Metapher scheinbar zu implizit. Ähnlich verhält es sich bei Satz 9. Es wird zwar keine Metapher verwendet, jedoch könnte die falsche Klassifizierung von fehlendem Kontext herrühren. Um zu verstehen, dass es sich um eine rassistische Aussage handelt, müsste verstanden werden, dass mit „den ganzen dunklen Gesichtern“ potenziell Schwarze Menschen gemeint sind. Zudem müsste Hintergrundwissen zur Geschichte und zur medialen Darstellung des Frankfurter Bahnhofsviertels vorhanden sein. Vermutlich liegt die fehlerhafte Klassifizierung am Fehlen dieses Wissens.

In Kategorie 3 wurden die Textabschnitte 11 und 14 fälschlicherweise als RACISM klassifiziert. Satz 11 ist ein indirektes Zitat einer rassistischen Aussage und negiert diese am Ende des Satzes. Die Negierung scheint nicht eindeutig genug gewesen zu sein, um vom Sprachmodell als OTHER gelabelt zu werden. Vermutlich gibt es in den Trainingsdaten wenig bis gar keine Datensätze mit ähnlichem Satzbau. Abschnitt 14 richtet sich gegen Islamismus und ist nicht klar islamophob. Korpus1 enthält ähnliche Sätze, deren Inhalt sich gegen Islamismus richtet und nicht gegen den Islam. Sie wurden mit OTHER gelabelt, um den Unterschied zwischen Islamismuskritik und Islamophobie zu verdeutlichen. Abschnitt 15, der ebenfalls Kritik an Islamismus enthält, wurde demgemäß auch richtig gelabelt. Trotzdem wurde Abschnitt 14 falsch klassifiziert. Es ist daher nicht nachvollziehbar, wo die Gründe für die falsche Klassifizierung liegen könnten.

Bei den richtig gelabelten Textabschnitten lag die geringste Wahrscheinlichkeit bei Satz 13 (0,5694). Hier wurde ein nicht rassistischer Abschnitt korrekt als OTHER gelabelt. Er besitzt einen ähnlichen Aufbau wie viele in Korpus1 gesammelte rassistische Aussagen, richtet sich aber gegen die Partei der Grünen. Darin liegt vermutlich die niedrige Wahrscheinlichkeit begründet, mit der das Label OTHER vorhergesagt wurde. Der Durchschnitt der Wahrscheinlichkeiten der richtig gelabelten Beispiele liegt bei 0,7796.

Bei den impliziten Beispielen wurden zwei Textabschnitte inkorrekt gelabelt. Bei den expliziten Beispielen war es hingegen ein Abschnitt weniger. Man könnte folglich vermuten, dass expliziter Rassismus eher erkannt wird als impliziter Rassismus. Für eine Bestätigung

dieser Vermutung müssten jedoch weitere Tests mit einer größeren Anzahl an diverseren Textbeispielen durchgeführt werden. So könnte gegebenenfalls auch ein eindeutigerer Unterschied in den Ergebnissen von rassistischen Aussagen in Social-Media-Sprache und politischer Sprache erkannt werden.

3.3.5 Experiment 5 – Einfluss der Dateninhalte

Bei der Sammlung der Daten für Korpus1 und Korpus2 stellte sich religiös-kultureller Rassismus gegen den Islam als die scheinbar häufigste Form von Rassismus in deutschen Plenarprotokollen heraus. Daraus ergab sich die Frage, ob - beeinflusst durch diese unausgeglichene Repräsentation in der Datenbasis - das darauf trainierte Model dazu neigt, eher Islamophobie zu erkennen als andere Formen von Rassismus. Diese Frage sollte mit einem Experiment beantwortet werden, in dem im Inference-Modus mit zwei verschiedenen Sätzen getestet wurde, die jeweils eine diskriminierende Aussage gegenüber einer Menschengruppe enthalten. Diese Menschengruppe wurde jeweils durch eine von einer anderen Rassismusform betroffenen Gruppe ersetzt. Die erwähnten Rassismusformen sind Islamophobie, Xenophobie, Antisemitismus, biologischer Rassismus sowie Christenfeindlichkeit. Es wurde vermutet, dass basierend auf dem erstellten Textkorpus Islamophobie und Xenophobie besser erkannt würden als Antisemitismus, biologischer Rassismus oder Christenfeindlichkeit, die nicht im Trainingsdatenset vorkommen. Demgemäß erhoffte man sich eindeutige Vergleichswerte.

In beiden Satzbeispielen wurde der xenophobische Satz mit der höchsten Wahrscheinlichkeit erkannt (s. *Tabelle 14*). Sowohl der islamophobische Satz wie auch der biologischen Rassismus enthaltende Satz wurde jeweils mit einer nahezu gleichen Wahrscheinlichkeit erkannt. Die Sätze mit der geringsten Wahrscheinlichkeit im Verhältnis zu anderen Rassismusformen waren die antisemitischen Sätze und die christenfeindlichen Sätze.

Es lässt sich dementsprechend ein Muster erkennen, nach dem Rassismusformen in eine Rangfolge gebracht werden können, die zeigt, wie groß die Wahrscheinlichkeit ist, vom Model richtig klassifiziert zu werden. Die Rangfolge ist:

1. Xenophobie
2. Islamophobie und biologischer Rassismus
3. Antisemitismus und Christenfeindlichkeit

Obwohl es also ein Ranking nach der Wahrscheinlichkeit gibt, wurden alle Sätze der richtigen Kategorie zugewiesen. Zudem ist die durchschnittliche Wahrscheinlichkeit mit 0,9275 bei

allen Sätzen sehr hoch, auch bei den Formen gruppenbezogener Diskriminierung, die nicht in Korpus1 vorkommen. Infolgedessen kann angenommen werden, dass der Satzbau eine mindestens ebenso große Rolle für die Klassifizierung spielt, wie die bezeichnete rassistisch diskriminierte Gruppe.

	text	label	prediction	probability
1	Eine lebenswerte islamische Gesellschaft gibt es nirgends.	RACISM	RACISM	0.9354
2	Eine lebenswerte Gesellschaft mit Migranten gibt es nirgends.	RACISM	RACISM	**0.9558**
3	Eine lebenswerte jüdische Gesellschaft gibt es nirgends.	RACISM	RACISM	0.8928
4	Eine lebenswerte christliche Gesellschaft gibt es nirgends.	RACISM	RACISM	0.8774
5	Eine lebenswerte Gesellschaft mit Schwarzen, Asiaten und Braunen gibt es nirgends.	RACISM	RACISM	0.9338
6	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in islamischen Gebieten.	RACISM	RACISM	0.9464
7	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in migrantischen Gebieten.	RACISM	RACISM	**0.9506**
8	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in jüdischen Gebieten.	RACISM	RACISM	0.9155
9	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in christlichen Gebieten.	RACISM	RACISM	0.9216
10	Der deutsche Bildungsstandard sinkt und damit ist die Katastrophe für den Arbeitsmarkt in ein paar Jahren vorprogrammiert, besonders in Gebieten mit Schwarzen, Asiaten und Braunen.	RACISM	RACISM	0.9463

Tabelle 14: Ergebnis-Tabelle von Experiment 5. Es wurden zwei Sätze getestet, in denen die diskriminierte Menschengruppe ausgetauscht wurde. Getestet wurde auf die Erkennung von Islamophobie, Xenophobie, Antisemitismus, biologischem Rassismus sowie Christenfeindlichkeit.

3.3.6 Diskussion

Bei der Evaluation der Experimente richtete sich deer Fokus zur Einschätzung der Performanz eines Sprachmodells hauptsächlich auf den Macro-F1-Score[24] gelegt. Es stellt sich jedoch die Frage, wie dieser Richtwert zu verstehen und zu beurteilen ist. Ist der vom Model 2 erreichte F1-Score von 72,38% erfolgreich genug oder noch nicht? Um diese Frage zu beantworten, ist es nötig zu verstehen, dass er einen Mittelwert zwischen Precision und

[24] Der Macro Average F1 berechnet sich, indem die Metriken für beide Labels ermittelt werden und deren ungewichteter Mittelwert genommen wird. Vgl. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (Stand: 09.07.2021).

Recall abbildet (Jurafsky und Martin 2014, S. 483). Er vereint dementsprechend beide Werte und spiegelt dadurch umfassend die Performanz des Models wider. Im Fall von Experiment 1 und 2 kann auch die Accuracy betrachtet werden. Da Korpus1 nahezu gleichviele Datensätze beider Labels enthält und folglich sehr ausgeglichen ist, ist die Accuracy in den genannten Experimenten repräsentativ. In Experiment 3 wurde Korpus2 verwendet, der eine ungleiche Verteilung der Labels enthält. Hier kann sich demgemäß nicht auf die Accuracy verlassen werden und es sollte mehr auf den F1-Score geachtet werden.

Zur Ermittlung einer Einschätzung des F1-Scores ist es sicherlich sinnvoll, diesen mit jenen anderer Arbeiten zu vergleichen, die sich ebenfalls mit einem ähnlichen Task und einer ähnlichen Datenbasis beschäftigen. Aus diesem Grund wurden die Resultate der Experimente vor allem mit der Arbeit von Risch et al. (2019) verglichen. Es ließ sich feststellen, dass in Experiment 1 ähnliche F1-Scores erreicht wurden, obwohl Korpus1 um einiges kleiner ist, als der von Risch et al..

Zur Interpretation des Ergebnisses ist es ebenso wichtig, in Betracht zu ziehen, wie das Model eingesetzt wird und wie sich die gefällten Entscheidungen des Models auswirken. Ist die Entscheidung eines neuronalen Netzes bedeutend und wirkt sie sich auf das Leben von Menschen aus, ist eine hohe Verlässlichkeit dieser Entscheidung umso wichtiger. Als vergleichbares Beispiel für einen solchen Fall kann man den Test zur Gesichtserkennung am Berliner Südkreuz im Jahr 2018 heranziehen. Nach mehreren Test-Wochen zur Gesichtserkennung mit speziellen Kameras veröffentlichte das Bundesinnenministerium eine Pressemitteilung, in der folgende Ergebnisse veröffentlicht wurden:

> Die durchschnittliche Trefferrate liegt bei dem besten getesteten System unter realistischen Testbedingungen bei über 80%. Das heißt: In über 80% der Fälle wurden die Testpersonen durch die Systeme zuverlässig erkannt. Weitere Optimierungen und höhere Trefferraten sind technisch möglich. […] Die Systeme haben sich damit für einen Einsatz im Polizeialltag bewährt. Der Präsident des Bundespolizeipräsidiums, Dr. Dieter Romann, sagt: "Die Technik erleichtert es, Straftäter ohne zusätzliche Polizeikontrollen zu erkennen und festzunehmen. Dies bedeutet einen erheblichen Sicherheitsgewinn. (Bundesinnenministerium 11.10.2018)

Für eine genaue Einschätzung der Performanz der Gesichtserkennung fehlen wichtige Metriken. Dennoch erscheint eine Trefferrate von circa 80% im Falle eines Systems, das darüber entscheiden kann, ob eine Person als Gefährder:in eingeschätzt und „ohne zusätzliche Polizeikontrollen“ festgenommen wird, doch als deutlich zu gering. Auf das Sprachmodell aus Experiment 1 übertragen, hängt eine endgültige Einschätzung der Fähigkeiten von dessen Einsatz ab. Wie in Kapitel *Einleitung* erwähnt, könnte ein solches

Tool eine größere Transparenz für Wähler:innen im Hinsicht auf unübersichtliche Plenardebatten bieten und eventuelle rassistische Gesinnungen bei Politiker:innen sichtbarer machen. Auch dieser Einsatz kann bei einer Fehleinschätzung des neuronalen Netzes große negative Auswirkungen für die Person bedeuteten, deren Rede oder Textabschnitte vom Sprachmodell fälschlicherweise als rassistisch gelabelt würden. Wünschenswert, wenn nicht gar obligatorisch, wäre für einen solchen Einsatz folglich ein nahezu hundertprozentiger F1-Score und dadurch eine maximale Verlässlichkeit. Sieht man von diesem möglichen Einsatz jedoch ab und konzentriert sich ausschließlich auf das Erreichen bestmöglicher Ergebnisse unter den gegebenen Umständen, erscheinen die Ergebnisse im Vergleich mit Risch et al. (2019) bereits gut. Mit Änderung der Umstände (sehr kleine Datenbasis, begrenzte Arbeitsmittel u.a.) könnten gegebenenfalls noch bessere Resultate erzielt werden. Dies meint Verbesserungen der Versuchsaufbauten und die Verwendung größerer und diverserer Datensets. Für letzteres müssten entweder mehr rassistische Textdaten aus Plenarprotokollen gesammelt werden oder auf deren Grundlage ähnliche synthetische Daten erzeugt werden.

Experiment 2 könnte mit unterschiedlichen Lemmatizern, die verschiedene deutsche Datenbasen nutzen, ausgeführt werden, um festzustellen zu können, ob und wie sich diese auswirken. Die allgemeine Auffassung in der Literatur ist, dass Preprocessing der Daten eher Kontext entfernen kann, der nötig ist, um die Semantik eines Textes fehlerfrei zu verstehen (Qiao et al. 2019). Zudem erübrigt sich eine Lemmatisierung unter Umständen durch die von BERT verwendeten WordPiece Embeddings (Devlin et al. 2019). In mehreren Foren oder Blogs, zum Beispiel Stack Overflow[25] oder TowardsDataScience[26], wird allgemein von einem Preprocessing der Daten abgeraten, sollte mit einem kontextbasierten Model wie BERT gearbeitet werden.

In Experiment 3 hat sich gezeigt, dass ein größeres Datenset nicht automatisch bessere Ergebnisse bedeutet. Es sollte jedoch bedacht werden, dass die Größe der Korpora ungewöhnlich klein ist und keine Studie zu Textklassifikation mit BERT mit einem vergleichbar kleinen Datenset gefunden werden konnte, zu dem Parallelen gezogen werden könnten. Es müssten dementsprechend weitere Tests durchgeführt werden, um feststellen zu können, warum die Models, die auf den kleineren Korpus1 trainiert wurden, bessere

[25] Vgl. https://stackoverflow.com/questions/54938815/data-preprocessing-for-nlp-pre-training-models-e-g-elmo-bert (Stand: 15.07.2021)

[26] Vgl. https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03 (Stand: 15.07.2021)

Ergebnisse mit einem kleineren Loss und weniger Overfitting erreichten. Beispielsweise sollte untersucht werden, wie sich die Verteilung der Labels in einem Datenset auswirken kann.

Weiterhin wäre es interessant zu untersuchen, ob mehr Kontext, sprich, das Training mit größeren Korpora, in Experiment 4 höhere Wahrscheinlichkeiten bei den Untersuchungen verschiedener Textabschnitte im Inference-Modus bewirken kann. Wäre zudem mit politischer Sprache gepretrained worden, könnte eventuell eine Aussage zu den Unterschieden im Erkennen von rassistischer Social-Media-Sprache und rassistischer politischer Sprache getroffen werden. Dafür könnten Plenarprotokolle von außerhalb des untersuchten Zeitraums verwenden werden. Sun et al. schreiben zum Vorteil eines weiteren Pretrainings, „that almost all further pre-training models perform better on […] datasets than the original BERT base model" (Sun et al. 2019). Zwar wurde nicht das originale BERT Model für die Experimente verwendet, sondern German BERT, trotzdem kann angenommen werden, dass Pretraining mit ähnlicher Sprache wie die des Trainingsdatensets, in diesem Fall politische Sprache, von großem Vorteil für das Fine-Tuning ist.

Die Untersuchung der Fähigkeiten des Models in Experiment 5 legt weitere Versuche nicht nur mit mehr Daten, sondern auch eine Multilabel-Klassifikation nahe. Dazu könnte ein Korpus angelegt werden, der mit zusätzlichen Labels für weitere Rassismusformen wie beispielsweise Antiziganismus und Diskriminierungen gegen weitere Religionen annotiert würde. Es wäre zudem interessant zu beobachten, wie sich die Verteilung verschiedener Rassismusformen im Korpus, auf das trainiert wird, auf die Wahrscheinlichkeiten zu jeder Kategorie in einem Versuch wie Experiment 5 auswirkt.

Generell lässt sich anhand der Analyse aller durchgeführten Experimente feststellen, dass rassistische Aussagen in Plenarsitzungen in den gesamten Kontext der Rede sowie in einen politischen Kontext gestellt werden müssen. Insofern schließt ein satz- beziehungsweise textabschnittbasierter Ansatz gegebenenfalls zu viel Kontext aus, der dazu benötigt werden kann, eine Aussage mit rassistischer Intention als solche zu identifizieren. Es könnte darum von Vorteil sein, einen weiteren Korpus anzulegen, der nicht auf Sätze und Textabschnitte ausgelegt ist, sondern auf ganze Plenarreden. Bei dem Task würde also nicht mehr nur ein kurzer Text, sondern ein ganzes Dokument klassifiziert. Es kann vermutet werden, dass durch den Einbezug einer kompletten Rede mehr Beziehungen zwischen Entitäten erfasst werden könnten. Allgemein würde es sich empfehlen, mit sehr viel mehr Daten zu trainieren und zu testen, um zu analysieren, wie sich der Loss weiter verringern lässt. So würden auch weitere, nicht nur sprachliche, sondern auch inhaltliche Experimente ermöglicht. Bezüge mancher Aussagen auf aktuelles politisches Zeitgeschehen, das nicht explizit in einer Rede

erläutert wird, können aktuell nicht in den Entscheidungsprozess des neuronalen Netzes mit einbezogen werden. Hier läge das weitere Pretraining von German BERT oder auch die Nutzung anderer deutscher BERT-Modelle nahe, um zu erforschen, wie sich diese auf das Kontextverständnis auswirken. Grundsätzlich bedarf es weiterer Nachforschungen zu BERTs Weltwissen. Die Erkenntnis, dass BERT Wissen über stereotype Namen besitzt, kann nicht auf Plenarprotokolle angewandt werden, da diese kaum Namen von Einzelpersonen enthalten. Vielmehr werden hauptsächlich Menschengruppen erwähnt, wie Rentner:innen, Lehrer:innen, Schüler:innen aber eben auch Muslim:innen und Migrant:innen. Weltwissen zu Menschengruppen zu besitzen, scheint von großer Wichtigkeit in Bezug auf das Verständnis rassistischer Sprache zu sein und eine Untersuchung mit rein sprachlichem und grammatisch-linguistischem Ansatz als nicht ausreichend.

4 FAZIT UND BLICK IN DIE ZUKUNFT

Zu Beginn der vorliegenden Arbeit wurden drei Forschungsfragen festgelegt:

1. Kann mit BERT rassistische Sprache in deutschen Plenarprotokollen erkannt und korrekt klassifiziert werden?
2. Welche Konfigurationen im Fine-Tuning begünstigen die Ergebnisse des Models?
3. Wie wirkt sich die Datenbasis auf die endgültigen Resultate des Models aus?

Zur Beantwortung dieser Fragen wurde sich zuerst mit Rassismus und rassistischer politischer Sprache auseinandergesetzt, um jeweils Arbeitsdefinitionen entwickeln zu können. Nach einer Auseinandersetzung mit den theoretischen Grundlagen neuronaler Netze über verschiedene Netzarchitekturen wie RNN, LSTM und Transformer wurde näher auf die Funktionsweisen von BERT eingegangen. Der praktische Teil erforderte schließlich die Erstellung zweier möglichst differenzierter Textkorpora auf Basis der festgelegten Arbeitsdefinitionen von Rassismus und rassistischer Sprache. Mit diesen Korpora wurden fünf Experimente durchgeführt, die Aufschluss über die Forschungsfragen geben sollten. Nach Analyse der Ergebnisse aller Experimente kann die erste Forschungsfrage wie folgt beantwortet werden: Es ist definitiv möglich, mit BERT rassistische Sprache in deutschen Plenarprotokollen korrekt zu klassifizieren.

Die zweite und dritte Forschungsfrage können nicht einzeln beantwortet werden, hängen doch die Ergebnisse des Trainings nicht nur von den Konfigurationen, sondern ebenso stark von der Datenbasis ab. In Experiment 1 wurden die besten Resultate (Macro-F1-Score von 72,38% und Accuracy von 72,41%) mit einer Learning Rate von 3e-5, drei Epochen und einer Batch Size von 16 erzielt. Mit einem etwas größeren Korpus und den gleichen Hyperparameter-Einstellungen fielen die Ergebnisse jedoch schlechter aus. Folglich gibt es keinen ‚goldenen Weg' im Fine-Tuning mit BERT. Die bestmöglichen Einstellungen für eine Textklassifikation sind immer von der Datenbasis, deren Größe und dem darin enthaltenen Gleichgewicht zwischen den Labels abhängig. Der Erstellungsprozess des Korpus kann dementsprechend bereits Auswirkungen auf die Ergebnisse des trainierten Sprachmodells haben. Gegebenenfalls wäre eine eventuell von Subjektivität beeinflusste Datensammlung sogar gefährlich, würde dem neuronalen Netz doch damit eine politische Richtung gegeben, obwohl absolute Neutralität anzustreben ist. Da dies jedoch schwer zu erreichen ist, wäre beispielsweise die Zusammenstellung eines diversen Teams mit unterschiedlichen politischen Blickrichtungen sinnvoll. Die Teammitglieder könnten sich besprechen und über die Auswahl der Daten diskutieren, um ein möglichst neutrales und ausgeglichenes Textkorpus zu gestalten.

Trotz der recht klaren Ergebnisse zeigte die Analyse und Diskussion der Experimente, dass diese in Bezug auf NLP mit BERT nur die Spitze des Eisbergs darstellen. Es können und müssen noch deutlich mehr Experimente und Tests durchgeführt werden, um vollkommen zu verstehen, wie BERT arbeitet und Entscheidungen trifft. Immer wieder entsteht bei der Arbeit mit BERT der Eindruck, nur an der Oberfläche des tatsächlichen Potentials zu kratzen. Nicht nur gibt es bisher nur wenige Aufsätze und Studien, die sich mit der Arbeit mit deutschen BERT-Modellen beschäftigen, vielmehr sind besonders die Untersuchungen zu sehr speziellen Formen von Sprache, wie der rassistischen Sprache, selten. Auch liegt der Fokus meist auf Social-Media-Sprache und nicht auf politischer Sprache. Gerade die Konzentration auf letztere bietet viele interessante Einsatzmöglichkeiten mit unter Umständen starken positiven Auswirkungen. Es wurde in der vorliegenden Arbeit bereits die Idee vorgestellt, ein Tool zu entwickeln, welches eine zeitnahe Kritik an rassistischer Sprache in Plenarreden ermöglichen und Wähler:innen so transparentere Plenardebatten bieten würde. Weitere Möglichkeiten der Verwendung von BERT zu rassistischer aber auch beispielsweise sexistischer Sprache in der deutschen Politik sind denkbar. In erster Linie sind sie aber vor allem sinnvoll und sogar von großer Notwendigkeit, denn diskriminierende Sprache findet immer stärker Einzug in die Sprachverwendung deutscher Politiker:innen (Stecker et al. 2021). Aus diesem Grund sind

das Thema dieser Arbeit sowie die weitere Beschäftigung mit diesem nicht nur aus Gründen der technischen Weiterentwicklung im NLP von großer Wichtigkeit, sondern auch aus ethisch-moralischen Gründen. Sprache formt die deutsche Politik und daraus folgend auch unsere Gesellschaft. Der Politolinguist Thomas Niehr formuliert es wie folgt: „Sprache ist nicht nur irgendein Instrument der Politik, sondern überhaupt erst die Bedingung ihrer Möglichkeit“ (Niehr 2014, S. 11). Sprache im Bundestag muss dem deutschen Grundgesetz (Artikel 3) entsprechen und diskriminierungsfrei und neutral sein. Eine Auseinandersetzung mit Diskriminierungen in politischer Sprache kann jedoch nicht vermieden werden, denn sie kann, bedingt durch die Vorbildsfunktion der Politiker:innen, starke Auswirkungen auf Rezipient:innen haben.

Das Ziel des Einsatzes von BERT zur Erkennung rassistischer Sprache in Plenarprotokollen ist, Menschen zu einen, an die Würde eines jeden Menschen zu erinnern und in turbulenten Zeiten wieder mehr Sachlichkeit in die deutsche Politik zu bringen. In diesem Zusammenhang ist es zu schwach formuliert zu sagen, man könne über Rassismus nicht hinwegsehen. Vielmehr darf man nicht über Rassismus hinwegsehen. Insofern soll diese Arbeit mit den Worten von Ludger Hoffmann und Annika Frank abgeschlossen werden: „Wer Frieden möchte, darf zum Rassismus nicht schweigen“ (Hoffmann und Frank 2021, S. 27).

5 ABBILDUNGSVERZEICHNIS

6 TABELLENVERZEICHNIS

7 LITERATURVERZEICHNIS

Alikhani, Behrouz; Rommel, Inken (2018): Aufstieg des Kulturrassismus. Von Huntington zu Sarrazin. In: *Zeitschrift für Vergleichende Politikwissenschaft* (1), S. 9–24. Online verfügbar unter https://www.springerprofessional.de/zeitschrift-fuer-vergleichende-politikwissenschaft-1-2018/15482234, zuletzt geprüft am 04.04.2021.

Amri-Henkel, Andrea (2021): Die Energiewende im Bundestag: ein politisches Transformationsprojekt? Eine Diskursanalyse aus feministischer und sozial-ökologischer Perspektive. 1. Auflage. Bielefeld: transcript; transcript Verlag (Edition Politik, 106).

Balibar, Étienne (2016): Gibt es einen "Neo-Rassismus"? In: Dorothee Kimmich, Stephanie Lavorano und Franziska Bergmann (Hg.): Was ist Rassismus? Kritische Texte. Stuttgart: Reclam (19220), S. 23–31.

Bengio, Yoshua; Lee, Dong-Hyun; Bornschein, Jorg; Mesnard, Thomas; Lin, Zhouhan (2015): Towards Biologically Plausible Deep Learning. Online verfügbar unter https://arxiv.org/pdf/1502.04156.

Bhattacharya, Shiladitya; Singh, Siddharth; Kumar, Ritesh; Bansal, Akanksha; Bhagat, Akash; Dawer, Yogesh et al. (2020): Developing a Multilingual Annotated Corpus of Misogyny and Aggression. Online verfügbar unter https://arxiv.org/pdf/2003.07428, zuletzt geprüft am 07.07.2021.

Black Lives Matter (o.A.): About. Hg. v. Black Lives Matter. Black Lives Matter. Online verfügbar unter https://blacklivesmatter.com/about/, zuletzt geprüft am 26.06.2021.

Bundesinnenministerium (11.10.2018): Projekt zur Gesichtserkennung erfolgreich. Testergebnisse veröffentlicht - Systeme haben sich bewährt. Berlin. Online verfügbar unter https://www.bmi.bund.de/SharedDocs/pressemitteilungen/DE/2018/10/gesichtserkennung-suedkreuz.html, zuletzt geprüft am 09.07.2021.

Cappy, Alain (2020): Neuro-inspired information processing. London, Hoboken, NJ: ISTE Ltd; John Wiley & Sons, Inc (Electronics engineering series).

Chakrabarty, Abhisek; Pandit, Onkar Arun; Garain, Utpal: Context Sensitive Lemmatization Using Two Successive Bidirectional Gated Recurrent Networks. In: Regina Barzilay und Min-Yen Kan (Hg.): Proceedings of the 55th Annual Meeting of the

Association for. Proceedings of the 55th Annual Meeting of the Association for. Vancouver, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics, S. 1481–1491.

Chen, Sanyuan; Hou, Yutai; Cui, Yiming; Che, Wanxiang; Liu, Ting; Yu, Xiangzhan: Recall and Learn: Fine-tuning Deep Pretrained Language Models with Less Forgetting. In: Bonnie Webber, Trevor Cohn, Yulan He und Yang Liu (Hg.): Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online. Stroudsburg, PA, USA: Association for Computational Linguistics, S. 7870–7881.

Chung, Junyoung; Kastner, Kyle; Dinh, Laurent; Goel, Kratarth; Courville, Aaron; Bengio, Yoshua (2015): A Recurrent Latent Variable Model for Sequential Data. Online verfügbar unter https://arxiv.org/pdf/1506.02216, zuletzt geprüft am 17.06.2021.

deepset.ai (2019): Open Sourcing German BERT. Insights into pre-training BERT from scratch. Hg. v. deepset.ai. Berlin. Online verfügbar unter https://www.deepset.ai/german-bert, zuletzt geprüft am 08.07.2021.

Deutscher Bundestag (o.A.): Protokolle. Deutscher Bundestag. Online verfügbar unter https://www.bundestag.de/protokolle, zuletzt geprüft am 18.06.2021.

Deutscher Bundestag (1980): Geschäftsordnung. Hg. v. Deutscher Bundestag. Online verfügbar unter https://www.bundestag.de/parlament/aufgaben/rechtsgrundlagen/go_btg, zuletzt geprüft am 29.05.2021.

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (2019): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (1), S. 4171–4186. Online verfügbar unter https://www.aclweb.org/anthology/N19-1423/, zuletzt geprüft am 23.06.2021.

Dieckmann, Walther (1975): Sprache in der Politik. Einführung in die Pragmatik und Semantik der politischen Sprache ; mit einem Literaturbericht zur 2. Aufl. 2. Aufl. Heidelberg: Winter (Sprachwissenschaftliche Studienbücher : Abt. 2).

Ertel, Wolfgang (2016): Grundkurs künstliche intelligenz. Eine praxisorientierte einfhrung. 4., überarbeitete Auflage. [Place of publication not identified]: Morgan Kaufmann (Computational Intelligence).

FARM Dokumentation (2019a): Building Blocks. Hg. v. FARM. Online verfügbar unter https://farm.deepset.ai/modeling.html, zuletzt geprüft am 26.06.2021.

FARM Dokumentation (2019b): Data Handling. Hg. v. FARM. Online verfügbar unter https://farm.deepset.ai/data_handling.html, zuletzt geprüft am 26.06.2021.

FARM GitHub (2019): Core features. Hg. v. FARM. Online verfügbar unter https://github.com/deepset-ai/FARM#core-features, zuletzt geprüft am 26.06.2021.

Fischer, Martin S.; Hoßfeld, Uwe; Krause, Johannes; Richter, Stefan (2019): Jenaer Erklärung. Das Konzept der Rasse ist das Ergebnis von Rassismus und nicht dessen Voraussetzung, 2019. Online verfügbar unter https://www.uni-jena.de/190910_JenaerErklaerung, zuletzt geprüft am 29.05.2021.

Geuther, Gudula (2020): Polizei-Praktiken in Stuttgart bleiben heftig umstritten. „Stammbaumforschung“. Hg. v. Deutschlandfunk. Deutschlandfunk. Online verfügbar unter https://www.deutschlandfunk.de/stammbaumforschung-polizei-praktiken-in-stuttgart-bleiben.1773.de.html?dram:article_id=480479, zuletzt geprüft am 26.06.2021.

Girnth, Heiko (2015): Sprache und Sprachverwendung in der Politik. Eine Einführung in die linguistische Analyse öffentlich-politischer Kommunikation. 2., überarbeitete und erweiterte Auflage. Berlin: DE GRUYTER (Germanistische Arbeitshefte, Band 39).

Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2017): Deep Learning. Cambridge, Mass.: MIT Press Ltd (Adaptive Computation and Machine Learning Series). Online verfügbar unter http://www.deeplearningbook.org/.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. In: *Neural computation* 9 (8), S. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Hoffmann, Ludger (2020): Zur Sprache des Rassismus. In: *Sprachreport* 36 (4), S. 40–47. DOI: 10.14618/SR-1-2020-HOF.

Hoffmann, Ludger; Frank, Annika (2021): Zur Pragmatik rassistischer Beleidigungen. In: Chr. Hohenstein (Hg.): Sprache/n, Institutionen und mehrsprachige Gesellschaften. Münster: Waxmann.

Howard, Jeremy; Ruder, Sebastian (2018): Universal Language Model Fine-tuning for Text Classification. Online verfügbar unter https://arxiv.org/pdf/1801.06146, zuletzt geprüft am 26.06.2021.

Jurafsky, Daniel; Martin, James H. (2014): Speech and language processing. 2. ed., Pearson new internat. ed. Harlow: Pearson Education (Always learning).

Kimmich, Dorothee; Lavorano, Stephanie; Bergmann, Franziska (Hg.) (2016): Was ist Rassismus? Kritische Texte. Stuttgart: Reclam (19220).

Kompa, Nikola; Moll, Henrike; Eckardt, Regine; Grassmann, Susanne (2013): Sprache, sprachliche Bedeutung, Sprachverstehen und Kontext.

Kupietz, Marc; Schmidt, Thomas (2018): Korpuslinguistik: DE GRUYTER.

Lehmann, Armin (2020): „Gaulands Sprache ist der schlecht verkleidete Jargon von Gangstern“. Gefährliche Rhetorik der AfD. Hg. v. Tagesspiegel. Online verfügbar unter https://www.tagesspiegel.de/politik/gefaehrliche-rhetorik-der-afd-gaulands-sprache-ist-der-schlecht-verkleidete-jargon-von-gangstern/25569590.html, zuletzt geprüft am 11.07.2021.

Lin, Yongjie; Tan, Yi Chern; Frank, Robert (2019): Open Sesame: Getting Inside BERT's Linguistic Knowledge. Online verfügbar unter https://arxiv.org/pdf/1906.01698.

Liu, Yinhan; Ott, Myle; Goyal, Naman; Du Jingfei; Joshi, Mandar; Chen, Danqi et al. (2019): RoBERTa: A Robustly Optimized BERT Pretraining Approach. Online verfügbar unter https://arxiv.org/pdf/1907.11692.

MDR (2020): Tödlicher Messerangriff in Dresden offenbar islamistisch motiviert. Generalbundesanwalt eingeschaltet. Hg. v. MDR. MDR. Online verfügbar unter https://www.mdr.de/nachrichten/sachsen/dresden/dresden-radebeul/tatverdaechtiger-toetungsdelikt-dresden-100.html, zuletzt geprüft am 26.06.2021.

Niehr, Thomas (2014): Einführung in die Politolinguistik. Gegenstände und Methoden. Stuttgart, Göttingen: Vandenhoeck & Ruprecht (UTB, 4173). Online verfügbar unter http://www.utb-studi-e-book.de/9783838541730.

Olah, Christopher (2015): Understanding LSTM Networks. Online verfügbar unter http://colah.github.io/posts/2015-08-Understanding-LSTMs/, zuletzt geprüft am 17.06.2021.

Pitsilis, Georgios K.; Ramampiaro, Heri; Langseth, Helge (2018): Detecting Offensive Language in Tweets Using Deep Learning. In: *Appl Intell* 48 (12), S. 4730–4742. DOI: 10.1007/s10489-018-1242-y.

Poerner, Nina; Waltinger, Ulli; Schütze, Hinrich (2019): E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. Online verfügbar unter https://arxiv.org/pdf/1911.03681.

Qiao, Yifan; Xiong, Chenyan; Liu, Zhenghao; Liu, Zhiyuan (2019): Understanding the Behaviors of BERT in Ranking. Online verfügbar unter https://arxiv.org/pdf/1904.07531.

Risch, Julian; Stoll, Anke; Ziegele, Marc; Krestel, Ralf (2019): hpiDEDIS at GermEval 2019: Offensive Language Identification using a German BERT model. In: German Society for Computational Linguistics & Language Technology (Hg.): Proceedings of the 15th Conference on Natural Language Processing (KONVENS). Erlangen, S. 403–408.

Rogers, Anna; Kovaleva, Olga; Rumshisky, Anna (2020): A Primer in BERTology: What We Know About How BERT Works. In: *Transactions of the Association for Computational Linguistics* 2020 (8), S. 842–866. Online verfügbar unter https://www.aclweb.org/anthology/2020.tacl-1.54/, zuletzt geprüft am 23.06.2021.

Sanh, Victor; Debut, Lysandre; Chaumond, Julien; Wolf, Thomas (2019): DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Online verfügbar unter https://arxiv.org/pdf/1910.01108.

Schmid, Mirko (2021): Boris Palmer will sich nicht „mundtot“ machen lassen: „Cancel Culture“ als wahre Bedrohung. Grüne Baden-Württemberg. Hg. v. Frankfurter Rundschau. Online verfügbar unter https://www.fr.de/politik/boris-palmer-rassismus-eklat-gruene-parteiausschluss-baden-wuerttemberg-dennis-aogo-habeck-90527722.html, zuletzt geprüft am 11.07.2021.

Stecker, Christian; Müller, Jochen; Blätte, Andreas; Leonhardt, Christoph (2021): The evolution of gender-inclusive language. Evidence from the German Bundestag, 1949-2021.

Sun, Chi; Qiu, Xipeng; Xu, Yige; Huang, Xuanjing (2019): How to Fine-Tune BERT for Text Classification? Online verfügbar unter https://arxiv.org/pdf/1905.05583, zuletzt geprüft am 26.06.2021.

Tagesschau (2020a): Entsetzen, Schock und Trauer. Anschlag in Hanau. Hg. v. Tagesschau. Online verfügbar unter https://www.tagesschau.de/inland/hanau-morde-zusammenfassung-101.html, zuletzt geprüft am 26.06.2021.

Tagesschau (2020b): Feuer verwüstet Flüchtlingslager Moria. Griechenland. Hg. v. Tagesschau. Tagesschau. Online verfügbar unter https://www.tagesschau.de/ausland/brand-moria-105.html, zuletzt geprüft am 26.06.2021.

Tagesschau (2020c): Macron spricht von islamistischem Terrorakt. Attacke in Frankreich. Hg. v. Tagesschau. Online verfügbar unter https://www.tagesschau.de/ausland/frankreich-anschlag-lehrer-101.html, zuletzt geprüft am 26.06.2021.

Tagesschau (2020d): Rechtsextreme Chatgruppen aufgeflogen. Polizei in NRW. Hg. v. Tagesschau. Tagesschau. Online verfügbar unter https://www.tagesschau.de/regional/nordrheinwestfalen/nrw-rechtextreme-polizei-netzwerk-101.html, zuletzt geprüft am 26.06.2021.

Tagesschau (2020e): Tote bei Messerangriff in Nizza. Frankreich. Hg. v. Tagesschau. Tagesschau. Online verfügbar unter https://www.tagesschau.de/ausland/nizza-angriff-105.html, zuletzt geprüft am 26.06.2021.

Tagesschau (2021a): Erstes Urteil nach Sturm auf US-Kapitol. Drei Jahre auf Bewährung. Hg. v. Tagesschau. Online verfügbar unter https://www.tagesschau.de/ausland/amerika/erstuermung-kapitol-erstes-urteil-101.html, zuletzt geprüft am 11.07.2021.

Tagesschau (2021b): Palmer droht Parteiausschluss. Rassismusvorwürfe. Hg. v. Tagesschau. Online verfügbar unter https://www.tagesschau.de/inland/tuebingen-palmer-rassismusvorwurf-101.html, zuletzt geprüft am 11.07.2021.

Tamura, Yasuto (2020): Prerequisites for understanding RNN at a more mathematical level. Hg. v. Data Science Blog. Online verfügbar unter https://data-science-blog.com/blog/2020/06/01/prerequisites-for-understanding-rnn-at-a-more-mathematical-level/, zuletzt geprüft am 06.06.2021.

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N. et al. (2017): Attention Is All You Need. Online verfügbar unter https://arxiv.org/pdf/1706.03762.

Wallace, Eric; Wang, Yizhong; Li, Sujian; Singh, Sameer; Gardner, Matt (2019): Do NLP Models Know Numbers? Probing Numeracy in Embeddings. Online verfügbar unter https://arxiv.org/pdf/1909.07940.

Weidman, Seth; Lang, Jørgen W. (2020): Deep Learning - Grundlagen und Implementierung. Neuronale Netze mit Python und PyTorch programmieren. 1. Auflage. Heidelberg, Ann Arbor: O'Reilly; ProQuest Ebook Central.

Wennker, Phil (2020): Künstliche Intelligenz - Anwendungsszenarien für alle Unternehmensbereiche (AT). Impulse, wie Sie effizient und wettbewerbsfähig bleiben (AT). 1. Auflage 2020. Wiesbaden: Springer Fachmedien Wiesbaden GmbH; Springer Gabler.

Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le V, Quoc; Norouzi, Mohammad; Macherey, Wolfgang et al. (2016): Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Online verfügbar unter https://arxiv.org/pdf/1609.08144.

Wu, Zhiyong; Chen, Yun; Kao, Ben; Liu, Qun (2020): Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. Online verfügbar unter https://arxiv.org/pdf/2004.14786.

Zeit (2020): Zehntausende Menschen protestieren deutschlandweit gegen Rassismus. Hg. v. Zeit. Zeit. Online verfügbar unter https://www.zeit.de/gesellschaft/zeitgeschehen/2020-06/demonstration-anti-rassismus-polizeigewalt-deutschland-protest-black-lives-matter, zuletzt geprüft am 26.06.2021.

Zhang, Yue; Teng, Zhiyang (2021): Natural language processing. A machine learning perspective. Cambridge: Cambridge University Press.

Zhang, Ziqi; Luo, Lei (2018): Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. Online verfügbar unter https://arxiv.org/pdf/1803.03662, zuletzt geprüft am 07.07.2021.

Hiermit versichere ich an Eides Statt, dass ich diese Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.
Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken und Quellen, einschließlich der Quellen aus dem Internet, entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.
Diese Arbeit habe ich in gleicher oder ähnlicher Form oder auszugsweise nicht im Rahmen einer anderen Prüfung eingereicht.
Ich versichere zudem, dass der Text der eingereichten elektronischen Fassung mit dem Text der vorgelegten Druckfassung identisch ist.

Köln, _____ Unterschrift: _____