

Heinrich Heine Universität
Philosophische Fakultät
Abteilung für Computerlinguistik



Bachelorarbeit zum Thema:

**Sondierung (Probing) von BERT-Modellen auf semantische
Nomenklassen**

zur Erlangung des akademischen Grades Bachelor of Arts

Vorgelegt von:

Aldi Halili
Bonner Platz 4
80803 München
Email: aldi.halili@uni-duesseldorf.de

Matrikelnummer: 3013735
Fachsemester: 7
Studiengang: BA Computerlinguistik

Betreut durch:
Erstgutachter: Apl.-Prof. Dr. Wiebke Petersen
Zweitgutachter: Dr. Kilian Evang
Abgabe: 03.11.2023

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die Arbeit „Sondierung (probing) von BERT-Modellen auf semantische Nomenklassen“ von mir eigenständig und nur unter den angegebenen Quellen erstellt worden ist.



Aldi Halili

Inhalt

Abbildungsverzeichnis

1	Einleitung	1
1.1	Forschungsziel	2
1.2	Aufbau der Arbeit	4
2	Theoretischer Teil.....	5
2.1	Forschungsstand	6
2.2	Semantische Nomenklassen bezüglich NLP	9
2.2.1	NLTK und WordNet	11
2.3	Coreference Resolution	13
2.4	Transformer	14
2.4.1	Bert-Modell	16
2.4.2	Bert Input-Repräsentationen.....	18
3	Methodik	19
3.1	Datenerhebung und -vorbereitung	19
3.2	Coreference Resolution Texten (Coref. Dateien)	20
3.3	Datenextraktion und -vorbereitung.....	21
3.3.1	Daten-Annotation	21
3.3.2	Extraktion von Embeddings	25
3.4	Probing-Implementierung.....	26
3.4.1	Leistungsindikatoren	26
4	Probing-Ergebnisse	27
5	Diskussion.....	30
5.1	Implikationen und zukünftige Forschung.....	31
6	Fazit.....	32
	Literaturverzeichnis.....	34

Abbildungsverzeichnis

Abbildung 1: Hyponyms und Hypernoms	10
Abbildung 2: NLTK WordNet.....	12
Abbildung 3: Coreference Resolution.....	13
Abbildung 4: Antezedent.....	14
Abbildung 5: Chronologische Entwicklung von Transformer-Modelle.....	15
Abbildung 6: MLM	17
Abbildung 7: Bert Input Repräsentation	18
Abbildung 8: OntoNotes annotiertes Korpus	20
Abbildung 9: Ursprüngliche Coreference Daten.....	21
Abbildung 10: get hypernoms Funktion.....	22
Abbildung 11: Extraktion von Hypernoms	23
Abbildung 12: Hierarchische Struktur von WordNet.....	23
Abbildung 13: Annotation mit belebt/nicht belebt	24
Abbildung 14: Annotation mit belebt	24
Abbildung 15: Annotation mit unbelebt	24
Abbildung 16: Annotation mit Unklar	25
Abbildung 17: Subtoken-Tokenizer Output	25
Abbildung 18: Confusion Matrix, LR-Model	28
Abbildung 19: Confusion Matrix, NN-Model.....	29

1 Einleitung

Große vortrainierte Sprachmodelle, wie BERT (Bidirectional Encoder Representations from Transformers) (Min et al., 2023) haben im Bereich des Natural Language Processing (NLP) eine revolutionäre Entwicklung eingeleitet, vergleichbar mit dem Einfluss von ImageNet im Bereich der Bilderkennung. In der sich ständig wandelnden Welt des NLP sind vortrainierte Sprachmodelle von immenser Bedeutung geworden. Sie haben die Art und Weise, wie Textdaten verarbeitet und interpretiert werden, grundlegend verändert. Unter diesen Modellen sind die von Hugging-Face besonders beliebt und setzen Maßstäbe in der NLP-Forschung sowie in der praktischen Anwendung. Daher besteht ein großes Interesse daran zu verstehen, wie diese Modelle entwickelt werden und welche Fähigkeiten sie besitzen.

Diese Modelle tragen nicht nur zur Optimierung etablierter Aufgabenfelder wie Question Answering (QA), Named Entity Recognition (NER), und maschinelle Übersetzung (eng. Machine translation) (MT) bei, sondern eröffnen zudem innovative Ansätze wie die Schließung von Wissenslücken. Sprache bildet ein komplexes Netz aus Worten, ihren Bedeutungen und dem jeweiligen Kontext. Das tiefgehende Verständnis dieser Nuancen und Feinheiten in der NLP war und ist stets eine Herausforderung. Durch die vortrainierten Sprachmodelle von Hugging-Face ist es möglich, Maschinen die menschliche Sprache nicht nur erfassen zu lassen, sondern diese auch zu interpretieren (Wallat et al., 2021). Im Mittelpunkt der Entwicklung steht BERT, ein Modell, das für sein cleveres Design und seine beeindruckenden Größe und Fähigkeiten bekannt ist.

Was genau ist BERT? Das Sprachmodell ist bekannt für sein besondere Architektur. Mit der Fähigkeit, bidirektionale Kontexte zu modellieren, erzielt ein auf Denoising-Autoencoding basiertes vortrainiertes Modell wie BERT bessere Ergebnisse als vortrainierte Modelle, die auf autoregressiver Sprachmodelle basieren. (Yang et al., 2019) Unter all seinen Fähigkeiten fällt besonders auf, wie es spezielle kontextualisierte Vektoren erstellt. Infolgedessen kann das vortrainierte BERT-Modell durch Hinzufügen nur einer zusätzlichen Ausgabeschicht (engl. output layer) feinabgestimmt werden, um hochmodernes Modell für eine Vielzahl von Aufgaben, wie zum Beispiel QA und Sprachinferenz (engl. language inference), zu erstellen, ohne dass wesentliche architekturenspezifische Änderungen erforderlich sind. (Devlin et al., 2018)

Zusätzliche charakteristische Eigenschaften dieses Modells sind außerdem seine Fähigkeit zur WordPiece-Tokenisierung und zur Generierung von kontextualisierten Vektoren. (Devlin et al., 2018) Der Schwerpunkt dieser Arbeit liegt auf diesen kontextualisierten Vektoren. Es wird davon ausgegangen, dass BERT, als ein weitverbreitetes und umfassendes Modell, bereits eine riesige Menge an Text verarbeitet hat. Dementsprechend sind die Erwartungen hoch, dass die kontextualisierten Vektoren von BERT auch Informationen über die semantische Klassifikation der Belebtheit von Nomen (belebten und unbelebten Substantiven) enthalten.

1.1 Forschungsziel

Das Hauptziel in der vorliegenden Arbeit geht von einer grundlegenden Frage aus, die von unsere intellektuelle Neugier angetrieben wird: “Haben die große Sprachmode (engl. Large Language Models) (LLMs) Wissen über Belebtheit und nicht Belebtheit der Nomen?” Diese Forschungsfrage fungiert als Leitfaden und ermöglicht eine Navigation durch das komplexe Netz des linguistischen Verständnisses, das BERT aufweist.

LLMs werden als die neueste Generation der „Distributional Semantic Models“ (DSMs) bezeichnet. (Lenci & Sahlgren, 2023) Diese Modelle lernen durch das Erfassen von Wort-Kookkurrenz Mustern (engl. word co-occurrence pattern) in Texten, was bedeutet, dass sie in der Lage sind, das nächste oder fehlende Wort aus einem gegebenen sprachlichen Kontext vorherzusagen. Eine Vielzahl von Studien hat gezeigt, dass DSMs ein breites Spektrum von menschlichen kognitiven Fähigkeiten erklären können. (Hill et al., 2015; Mandera et al., 2017) Daher sind sie sehr wertvoll, um zu verstehen, welche Art von Informationen im Text gelernt werden können.

Im Fokus dieser Arbeit stehen LLMs wie BERT, die auf großen, allgemeinen Textkorpora trainiert wurden, mit dem Ziel, Wörter im Kontext vorherzusagen. Diese Modelle werden daher als vortrainierte LLMs bezeichnet. Das Ziel der Wortvorhersage im Kontext ermöglicht es ihnen, umfangreiches Wissen zu erwerben, ohne durch spezifische Aufgabenanforderungen eingeschränkt zu sein. Der geheime Schlüssel dafür sind die kontextualisierten Wortrepräsentationen. (eng. contextualized word representations) (Devlin et al., 2018)

Modelle wie Word2Vec (Mikolov et al., 2013) und GloVe (Pennington et al., 2014) waren, aufgrund ihrer Fähigkeit, syntaktische und semantische Informationen von Wörtern aus großen Mengen nicht-gelabelte Texte zu erfassen, in den 2010er Jahren sehr beliebt. Andererseits ermöglichen jedoch diese Ansätze nur eine einzige kontextunabhängige Repräsentation für

jedes Wort. Das bedeutet, dass nur ein Embedding für jedes Wort erstellt wird. In Wirklichkeit können Wörter aber mehrere Bedeutungen haben. Um dieses Problem zu lösen, und um den Kontext zu berücksichtigen hat BERT kontextualisiertes Embedding erzeugt.

Die kontextualisierten Word Vektoren haben somit dazu beigetragen, dass in den letzten Jahren große Sprachmodelle wie BERT beeindruckende Ergebnisse in diversen Aufgabenfeldern der NLP erzielt haben. Trotz dieser Erfolge bleibt das genaue Ausmaß, welches von diesen Modellen erworbenen Wissens und die spezifischen sprachlichen Phänomene, die sie erfassen können, ein aktives Forschungsthema. Tatsächlich können lineare Sondierungsmodelle, die auf kontextualisierten Repräsentationen trainiert wurden, linguistische Eigenschaften von Wörtern vorhersagen. (Tenney et al., 2019; Hewitt and Manning, 2019) Vielen Studien haben anhand des Probing-Task unter anderen untersucht, wie semantische Informationen in BERT repräsentiert sind. Forscher haben gezeigt, dass BERT in der Lage ist, semantische Rollen, Entitätstypen und semantische Beziehungen zu repräsentieren. (Ettinger, 2020; Tenney et al., 2019) Eine weitere Studie von Jawahar et al. (2019) hat gezeigt, dass Embedding aus verschiedenen Schichten von BERT bei unterschiedlichen Aufgaben besser performen, wobei semantische Informationen tendenziell besser von den höheren Schichten repräsentiert werden. Die neuesten Studien zu diesem Thema, haben die Wichtigkeit der Interpretation von kontextualisierten Wort-Embeddings Modelle betont.

Die menschliche Sprache ist aber ein faszinierendes und sehr komplexes Kommunikationsmittel, das uns ermöglicht, unsere Gedanken und Vorstellungen miteinander zu teilen. Eine wichtige Rolle spielt dabei die Kategorisierung der semantischen Eigenschaften der Nomen, um die Vielfalt der Dinge und Konzepte in unserer Umwelt besser zu verstehen. Diese Gruppierung beschreibt, semantischen Merkmalen der Nomen wie Belebtheit, Geschlecht, Konsistenz, Form und funktionale Eigenschaften usw. (Aikhenvald, 2006)

Die Hypothese in der vorliegenden Arbeit behauptet, dass Sprachmodelle wie BERT, die eine Vielzahl von Texten verarbeitet haben und seine Architektur sich auf kontextualisierte Wort Repräsentationen basiert, effektiv in der Lage sein sollten, die feinen Unterschiede zwischen belebten und nicht belebten Objekten in der Sprache zu begreifen. Genauer betrachtet, fokussiert sich die Analyse nicht nur auf Nomen im Allgemeinen, sondern insbesondere auf Informationen im Rahmen der Coreference Resolution (CR). Hierbei geht es vor allem um die semantische Information (belebt/unbelebt) des Antezedens eines Pronomens und des Pronomens selbst. Wenn sich ein Pronomen auf ein Nomen bezieht, das als "belebt" klassifiziert ist, sollte auch das Pronomen als "belebt" klassifiziert werden.

Beispielsweise wird in dem Satz aus dem Dataset „I asked to see the report and I found that it was for a 98 or 99 model car without a plate number“ das Nomen „*the report*“ als unbelebt annotiert. Es fungiert als Antezedent für das Pronomen „*it*“. Folglich sollte auch „*it*“ als unbelebt klassifiziert werden.

Um die Hypothese durch empirische Befunde zu untersuchen, werden diese annotierte Daten durch eine Probing-Aufgabe verifiziert.

1.2 Aufbau der Arbeit

Um eine fundierte und umfassende Antwort auf die im Rahmen dieser Bachelorarbeit aufgeworfene Forschungsfrage in der Einleitung zu erarbeiten, erfolgt in den nachfolgenden Kapiteln eine eingehende Analyse und detaillierte Betrachtung des Untersuchungsgegenstandes.

Der Kapitel 2 bildet theoretischen Teil der vorliegende Arbeit. Es beginnt mit einer Übersicht über den aktuellen Forschungsstand, gefolgt von einer Auseinandersetzung mit semantischen Nomenklassen im Kontext der NLP und den damit verbundenen Konzepten von NLTK und WordNet. Anschließend wird das Konzept der Coreference Resolution und die Rolle der Transformer-Architektur im NLP eingeführt. Hierbei wird insbesondere auf das BERT-Modell, den BERT-Tokenizer sowie auf die kontextualisierten Vektoren des BERT-Modells eingegangen. Des Weiteren wird das Probing als Methode zur Untersuchung des semantischen Wissens von Sprachmodellen wie BERT Model vorgestellt. Im Kapitel 3 wird die Methodik der Arbeit erläutert. Dies umfasst die Datenerhebung und -vorbereitung, die Durchführung der Coreference Resolution (Antezedent), die Datenextraktion und -vorbereitung sowie die Annotation von Nomen. Zudem wird die Implementierung der Methoden beschrieben. Die im Rahmen der Forschung gewonnenen Ergebnisse werden in Kapitel 4 präsentiert. In Kapitel 5 erfolgt eine Diskussion der Ergebnisse im Kontext der bestehenden Forschung und Theorie. Kapitel 6 fasst die wichtigsten Erkenntnisse der Arbeit zusammen und gibt einen Ausblick auf mögliche zukünftige Forschungsrichtungen in diesem Bereich.

2 Theoretischer Teil

Die vorliegende Arbeit unterteilt sich in zwei klar definierte Phasen. In der ersten Phase erfolgt eine theoretische Auseinandersetzung, die man mit einer Analyse des zu erforschenden Gebiets vergleichen kann, bevor man sich ins Unbekannte begibt.

Diese Phase, die in diesem Kapitel vorgestellt wird, umfasst eine detaillierte Betrachtung des aktuellen Forschungsstands zur Erkennung grammatikalischer Phänomene bei Large LLMs, sowie ein umfassendes theoretisches Verständnis semantischer Nomenklassen im Kontext von NLP. Im Rahmen dieser Auseinandersetzung wird das Konzept der CR inklusive der Identifikation von Antezedenten erläutert. Außerdem wird das BERT-Modell mit seiner Architektur und dem Tokenizer-System vorgestellt und analysiert. Ebenfalls wird aufgezeigt wie kontextualisierte Vektoren genutzt werden, um die Bedeutung von Wörtern im Satzzusammenhang zu erfassen und darzustellen. In der zweiten Phase erfolgt eine empirische Untersuchung, auf die im dritten Kapitel detailliert eingegangen wird.

Zu Untersuchungen des Konzeptes der Belebtheit bringen die Autoren Gelman et al. (1995) Aufschluss. In diesem Artikel aus der Physik wird dargelegt, dass Menschen die Bewegung von Objekten manchmal auf unterschiedliche Weise interpretieren können. Wenn sich ein Objekt so bewegt, wie es die Gesetze der Physik, genauer gesagt die Newtonschen Bewegungsgesetze, vorgeben, nehmen wir es oft als unbelebt wahr. Bewegt sich das Objekt jedoch anders, als es diese Gesetze erwarten lassen, erscheint es uns als belebt.

Aufgrund dieses Hintergrunds könnte man annehmen, dass die Unterscheidung zwischen diesen beiden semantischen Klassen nicht nur für Menschen, sondern auch für Sprachmodelle eine Herausforderung darstellt. Andererseits haben die Leistungsfähigkeit LLMs - wie BERT, die bereits eine enorme Menge an Text verarbeitet haben. Vor diesem Hintergrund bleibt die Frage offen, wie genau BERT diese Unterscheidungen in seiner Verarbeitung vornimmt. Diese Frage dient sowohl als Leitfaden, der den wissenschaftlichen Weg bestimmt, als auch als grundlegendes Prinzip, welches den Weg durch das komplexe Gebiet der linguistischen Fähigkeiten von BERT erleuchtet.

2.1 Forschungsstand

Menschliche Sprachen sind komplex und enthalten unzählige Informationen, die wir Menschen nur dank unserer kognitiven Fähigkeiten intuitiv interpretieren können. In der NLP waren und sind solche Phänomene stets eine große Herausforderung. Um nachvollzuziehen, wie die Maschinen diese komplexen Phänomene erfassen, basierten die alten NLP-Methoden auf vordefinierten linguistischen Merkmalen. Dies gewährleistete eine Verknüpfung zwischen linguistischer Theorie und Computeranwendung. Moderne (state-of-the-art) Sprachmodelle, wie BERT, haben jedoch diesen alten Ansatz verändert, da diese Modelle Vektorrepräsentationen von Wörtern statt linguistische Eigenschaften verwenden. Aufgrund dieser Entwicklung haben Forscher sogenannte Sondierungsaufgaben (engl. probing-task) entwickelt. Diese werden darauf so trainiert, dass linguistische Eigenschaften aus den Repräsentationen eines Modells genau hervorgesagt werden können. Dadurch können Aufgaben mit dem Ziel eingesetzt werden die sprachlichen Merkmalen aus den internen Repräsentationen des Modells zu extrahieren oder zu messen. (Hewitt & Liang, 2019)

Diese Bewertung hilft uns zu verstehen, welche Art von Informationen das Modell tatsächlich gelernt hat. Beispielsweise können diese Aufgaben entdecken, ob das Modell Satzstrukturen, Wortbedeutungen oder sogar komplexere Sprachkonzepte versteht. So zum Beispiel auf das Erkennen von früher genannten Wörter, welche sich auf Pronomen beziehen. Abhängig von den Ergebnissen dieser Aufgaben könnten Forscher das vortrainierte Modell spezifisch für NLP- oder NLU-Anwendungen optimieren oder tiefer in die Funktionsweise des Modells eintauchen, um besser zu verstehen, wie diese Modelle die menschliche Sprache interpretieren. (Tenney et al., 2019; Conneau et al., 2018)

Die Studie “A Closer Look at Linguistic Knowledge in Masked Language Models: The Case of Relative Clauses in American English“ von Mosbach et al. (2020) hat signifikante Fortschritte in dieser Thematik erzielt. Es wurden drei Modelle (BERT, RoBERTa und ALBERT) evaluiert, indem ihr grammatikalisches und semantisches Wissen durch Satzebene-Probing bzw. das Verstehen von Relativsätzen, die ein komplexes Phänomen darstellen und viel Kontext erfordern, getestet wurde. Im Anschluss daran wurde untersucht, ob die Modelle in der Lage sind, die Identifikation des Antezedenten, der durch den Relativsatz näher bestimmt wird, korrekt aufzulösen. Konkret wurde geprüft, ob die Modelle korrekte grammatikalische Kombinationen von Relativpronomen und bestimmten Typen von Antezedenten (belebt oder unbelebt) vorhersagen können. Ebenso wurde analysiert, ob sich grammatikalisch passende

Antezedenten anhand der vorgegebenen Relativpronomen (*who* vs. *which/that*) vorhersagen lassen. Die Antezedenten wurden auch auf semantischer Ebene untersucht, indem getestet wurde, ob die Modelle semantisch mögliche Antezedenten für gegebene Relativpronomen korrekt vorhersagen können. Dabei wurde der Grad der Spezifität der vorhergesagten Antezedenten im Vergleich zu den Ziel-Antezedenten berücksichtigt. Zum Beispiel im Satz: „*Children who eat vegetables are likely to be healthy*“ ist das Wort „*Boy*“ eine spezifischere Option als „*Children*“. Ihre Analyse, offenbart, dass diese Modelle in grammatikbezogenen Aufgaben hervorragend abschneiden, jedoch modellspezifische Schwächen, vor allem im Bereich des semantischen Verständnisses, aufweisen. Diese Studie hat sich lediglich auf die Antezedenten-Auflösung konzentriert und ist nicht auf die semantischen Eigenschaften der Antezedenten und der Pronomen, auf die sie sich beziehen, eingegangen. Auf der Problemstellung geht der vorliegende Arbeit ein.

Die Ergebnisse der Studie von Kauf et al. (2022) dienten als Vertiefung in Untersuchung der semantischen Eigenschaften von LLMs. Sie zeigen, wie vortrainierte große Sprachmodelle (LLMs) Wissen über häufige Ereignisse erwerben, indem sie mögliche von unmöglichen Beschreibungen von Agent-Patient-Interaktionen unterscheiden. Diese Studie hat festgestellt, dass vortrainierte LLMs erhebliches Wissen über Ereignisse verfügen und andere Sprachmodelle übertreffen. Insbesondere weisen sie fast immer höhere Wahrscheinlichkeiten für mögliche im Vergleich zu unmöglichen Ereignissen zu wie z.B. (*The teacher bought the laptop* vs. *The laptop bought the teacher*). Andererseits erzielen LLM-Modelle gute Ergebnisse auf syntaktischer Ebene (Aktiv- vs. Passivformen), jedoch wenig gute auf semantischer Ebene (Synonym-Sätze).

In der von Hawkins et al. (2020) durchgeführten Studie "Investigating representations of verb bias in neural language models" wurden die grammatikalischen Eigenschaften von Large LLMs eingehend analysiert und dabei bemerkenswerte Ergebnisse erzielt.

Diese vom EMNLP¹ anerkannte Forschung stellt den Distributed Artificial Intelligence Systems (DAIS) Benchmark-Datensatz vor, der 50.000 menschliche Bewertungen zu 5.000 unterschiedlichen Satzpaaren im Rahmen der englische Dativ enthält. Diese Datenbank enthält 200 einzigartige Verben und verschiedene Argumente, die unterschiedlich lang sind und durch unterschiedliche Formulierungen geäußert werden können. Beispielweise:

¹Empirical Methods in Natural Language Processing (EMNLP) ist eine führende Konferenz im Bereich der NLP und künstlicher Intelligenz (KI).

- a) Ava gave him something.
- b) Ava gave something to him.

Das zentrale Ziel ist es, festzustellen, ob die neuesten neuronalen Sprachmodelle menschliche Tendenzen in Bezug auf Verbpräferenzen (verb bias) adäquat repräsentieren. Den Erkenntnissen nach sind größere Modelle leistungsfähiger als kleinere. Es hat sich gezeigt, dass Transformer-Architekturen, wie GPT-2, die rekurrente Architekturen, wie LSTMs, übertreffen. Auch bei vergleichbaren Parameter- und Training Set. Weitergehende Analysen der internen Repräsentationen legen nahe, dass Transformer in der Lage sind, spezifische lexikalische Informationen besonders effizient mit grammatikalischen Konstruktionen zu verknüpfen. Die vorliegende Studie erweitert das bestehende Verständnis über neuronale Sprachmodelle und ihre Fähigkeit, linguistische Phänomene zu repräsentieren, insbesondere im Kontext von Verbpräferenz (verb-bias).

Eine weitere Studie von Beloucif & Biemann (2021) hat sich darauf konzentriert, anhand von Probing-Methode herauszufinden, ob vortrainierte LLMs den Zusammenhang zwischen semantischen Attributen und ihren Werten erfassen. Ein Beispiel für semantische Attribute und deren Werte ist die Beziehung, die zwischen „alt“, „Alter“ und „Geburtsdatum“ besteht, oder die Beziehung zwischen „reich“, „Reichtum“ und „Nettovermögen“. Die Autoren verwenden drei LLMs Modelle (BERT, RoBERTa und XLNET), um maskierte Token mit Mustern (eng. pattern), die aus grundlegenden sprachlichen Phänomenen wie Hypernyme und Hyponyme bestehen, und Listen von Elementen aus Wikidata vorherzusagen. Allerdings zeigen die Ergebnisse, dass die drei LLMs bei dieser Aufgabe immer noch deutlich schlechter performen als der Mensch. BERT hat bewiesen, dass er gut bei Hypernymen und Hyponymen wie bei Gold, Silber und Eisen ist, schafft es jedoch nicht, das richtige Attribut-Wert-Paar genau vorherzusagen.

Im wissenschaftlichen Artikel von Zheng et al. (2023) wird das syntaktische Wissen des chinesischen BERT-Modells untersucht, das mithilfe von Probing-Methoden erworben wurde. Die Ergebnisse zeigen, dass bestimmte Attention-Heads und ihre Kombinationen effektiv spezifische und generelle syntaktische Relationen encoden. Das bedeutet, dass BERT sehr gut im Parsen eines Satzes ist. Die versteckten Schichten (engl. hidden Layer) enthalten ebenfalls in unterschiedlichen Maßen syntaktische Informationen. Zusätzlich zeigt die Analyse von fine-tuned Modellen, wie sie sich an linguistische Strukturen anpassen.

Diese Forschungsarbeiten haben maßgeblich dazu beigetragen, dass sich Forschungsbereiche rund um NLP und besonderes Interesse auf BERTs Verständnisfähigkeiten und seine Anwendungen in verschiedenen Domänen entwickelt wurde. Ihre Ergebnisse und Methoden dienen als wertvolle Referenzen. Für Recherchen in diesem Bereich im Hinblick auf die vorliegende Thematik ist es essenziell, sich einen Überblick über den aktuellen Forschungsstand zu verschaffen. Dabei wird der Umfang der Untersuchungen zu semantischen Nomenklassen bewertet, insbesondere im Kontext von kontextualisierten Vektoren, wie sie in Hugging-Face Modellen bzw. BERT verwendet werden. Dafür wird Probing spezifisch angewendet, um zu sehen, inwiefern die kontextualisierten Vektoren die Belebtheit und Unbelebtheit eines Nomens und deren Antezedenten erkennt.

2.2 Semantische Nomenklassen bezüglich NLP

Bei der Auseinandersetzung mit dem theoretischen Teil wird offensichtlich, dass die semantische Analyse im Rahmen der NLP eine Schlüsselrolle spielt. Weiterhin zeigt sich, dass fast alle Sprachen spezielle grammatikalische Mechanismen zur Kategorisierung von Substantiven und Nominalphrasen haben. Dieses Spektrum an Mechanismen zur Nomenkategorisierung erstreckt sich von lexikalischen Eigenschaften, die in den Sprachlandschaften Südostasiens dominieren, bis hin zu den detaillierten grammatikalischen Geschlechterstrukturen der indoeuropäischen Sprachfamilien. Trotz ihrer offensichtlichen Vielfalt teilen diese Mechanismen eine gemeinsame semantische Grundlage, wobei das Potenzial besteht, dass der eine sich in den anderen entwickeln kann. Aus diesem sprachlichen Spektrum ergibt sich eine tiefe Einsicht in die Fähigkeit des Menschen, die Welt mithilfe der Sprache zu kategorisieren. Hierbei kommen universelle semantische Parameter zum Einsatz, welche Faktoren wie Menschlichkeit, Belebtheit, Geschlecht, Form, Konsistenz und funktionelle Merkmale berücksichtigen. (Aikhenvald, 2006)

Diese Mechanismen zur Nomen-Kategorisierung äußern sich als Morpheme in oberflächlichen linguistischen Strukturen, treten unter spezifischen kontextuellen Bedingungen hervor und haben die Funktion, markante, wahrgenommene oder zugeschriebene Eigenschaften hervorzuheben. Diese Eigenschaften stehen in Bezug zu den Entitäten, auf die sich die zugehörigen Substantive beziehen. Sprache kann als dichtes Netzwerk von Bedeutungen verstanden werden, das auf semantischen Beziehungen aufbaut, um feinste Nuancen und diskrete Unterschiede zu vermitteln. (Allan, 1997)

Die semantische Analyse in NLP ist ein Verfahren, bei dem versucht wird, die Bedeutung aus Textinhalten zu gewinnen. Dies gestattet Computersystemen, den Sinn von Sätzen, Abschnitten und sogar umfassenden Dokumenten zu erfassen und zu deuten. Dies wird durch eine gründliche Analyse der grammatikalischen Struktur und das Erkennen von Beziehungen zwischen den einzelnen Wörtern im Kontext erreicht. In diesem Fachgebiet stoßen wir auf eine Vielzahl von grundlegenden Konzepten wie Hyponymie/Hyperonymie, Homonymie, Synonymie und Antonymie. (Goddard & Schalley 2010)

Jedes dieser Konzepte spielt eine wichtige Rolle bei der Interpretation und dem Verständnis von Texten. **Hyponyme** sind Wörter, die eine Unterordnungsbeziehung bezeichnen. Zum Beispiel ist "*mammal*" ein Hyponym von "*animal*". Im Bereich der Linguistik und Lexikografie wird der Begriff "Hyponym" verwendet, um auf ein spezifisches Mitglied einer allgemeineren Klasse hinzuweisen. (Cann, 2021, S. 456-467) Zum Beispiel können innerhalb der Kategorie "*Flower*" sowohl "*Rose*" als auch "*Jasmine*" als Hyponyme betrachtet werden, die spezifische Arten oder Beispiele vom Wort "*Flower*" bezeichnen. Während die **Hypernymie** die übergeordnete Beziehung beschreibt. Beide sind transitive Verhältnisse zwischen Wortgruppen. Da es in der Regel nur ein übergeordnetes Wort (Hypernym) gibt, ordnet diese semantische Beziehung die Bedeutungen von Nomen hierarchisch an. (Miller, 1995) Aus diesen Hierarchien wurden von WordNet alle Nome und Relationen für diese vorliegende Arbeit extrahiert. Sie werden ausführlich in der Kapitel 4 behandelt.

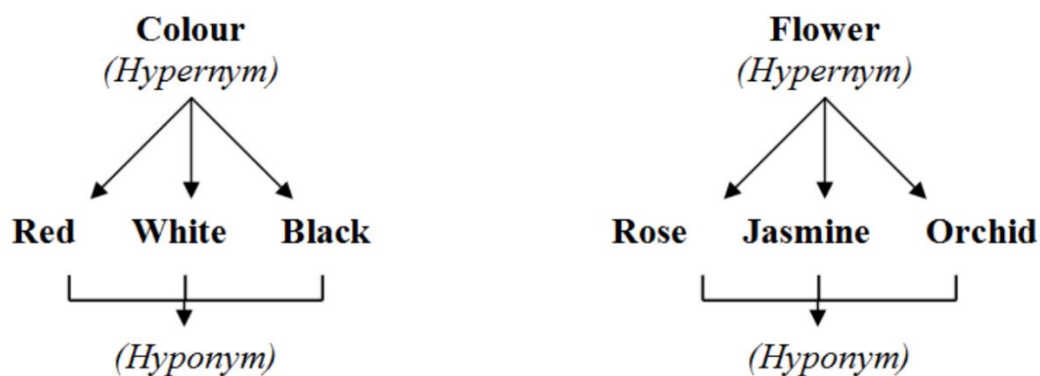


Abbildung 1: Hyponyms und Hypernyms [Quelle](#)

Wenn von **Homonymie** gesprochen wird, sind Wörter gemeint, die die gleiche Schreibweise oder Aussprache aufweisen, sich aber in ihren Bedeutungen unterscheiden. Ein Beispiel im Englischen hierfür ist das Wort "*Bank*", das sich auf eine Finanzinstitution oder das Ufer eines Flusses beziehen kann. Durch diese Art von Wörtern kann lexikalische Mehrdeutigkeit

entstehen, also Verwirrung, die durch Wörter verursacht wird, die mehr als eine mögliche Bedeutung haben (Goddard & Schalley 2010; Nordquist, 2023). Ähnlich verhält es sich bei der **Synonymie**. Hierbei handelt es sich um Wörter, die zwar unterschiedliche Schreibweisen und eventuell auch Aussprachen aufweisen können, aber dennoch ähnliche oder sogar gleiche Bedeutungen haben. Ein Beispiel sind die Wörter "*sofa*" und "*settee*". Im Kontext von WordNet wird die Synonymie als Hauptbeziehung genutzt, indem Gruppen von Synonymen, sogenannten Synsets, herangezogen werden, um die Bedeutungen von Wörtern abzubilden. (Cann, 2021, S. 458; Miller, 1995)

Schließlich wird die **Antonymie** als das Gegenstück zur Synonymie betrachtet. Hierbei handelt es sich um Wörter, die entgegengesetzte Bedeutungen aufweisen, wie etwa "*cold*" und "*hot*". (Cann, 2021, S. 460)

Das Verständnis dieser semantischen Relationen der Nomen ist von grundlegender Bedeutung, um nachvollzuziehen, wie BERT durch das komplexe Netzwerk linguistischer Bedeutungen agiert. Hierfür stehen dabei hilfreiche Tools und Python Bibliotheken zur Verfügung wie: NLTK (Natural Language Toolkit), wobei WordNet eine verlässliche Grundlage bildet.

2.2.1 NLTK und WordNet

Für die Untersuchung dieser komplexen Thematik wurde die robuste Natural Language Tool Kit (NLTK) -Bibliothek für natürliche Sprachverarbeitung eingesetzt. Der unten vorgestellte Codeausschnitt zeigt die eingesetzten Tools und Methoden. Das englische Wörterbuch von WordNet ist ein zentraler Bestandteil der NLTK-Bibliothek² in Python. Dieses umfassende Toolkit erleichtert signifikant die Aufgaben im Bereich der NLP und macht sie zugänglicher.

WordNet, entwickelt von George Miller (1995), ist ein graphenbasiertes lexikalisches Netzwerk, bestehend aus Knoten und Kanten. Die Knoten repräsentieren individuelle Bedeutungen und korrelieren mit Synsets (Synonymsets). Ein Synset ist eine Menge von Wörtern, die gleiche oder sehr ähnliche Bedeutungen haben. Diese Struktur kann zu Mehrdeutigkeiten führen, da einem Knoten genau ein Synset zugeordnet wird, einem Synset jedoch mehrere Knoten beziehungsweise Bedeutungen zugeordnet werden können. Synsets sind die zentralen Elemente von WordNet und enthalten nicht nur klassische Wörter, sondern auch Komposita, idiomatische Wendungen, Phrasalverben und Kollokationen, die semantisch

²NLTK for WordNet: [NLTK :: Sample usage for wordnet](#)

nicht weiter aufgeschlüsselt werden können. Zur Verdeutlichung der Bedeutungen enthalten die Knoten eine Glosse, die die Bedeutung erläutert, Beispiele, die die Verwendung des Wortes illustrieren, und syntaktische Informationen wie Wortkategorien. WordNet klassifiziert Wörter in vier Kategorien: Nomen, Verben, Adjektive, und Adverbien. Die Kanten in WordNet repräsentieren Relationen zwischen Bedeutungen und können in zwei Arten unterteilt werden:

- Kanten aus konzeptuellen Relationen beinhalten semantische Inhalte wie Hyperonyme, Hyponyme, Antonyme, Meronyme usw. Der Schwerpunkt der in dieser Arbeit gelabelten Daten liegt auf diesen konzeptuellen Relationen. Insbesondere wurden die Relationen in Bezug auf Hyperonyme und Hyponyme bei der Annotation berücksichtigt.
- Kanten aus lexikalischen Relationen, wie Normalisierung und zugehörige Verben. (Wurm, 2021)

Die zweite Kategorie wird jedoch nicht für die in dieser Studie verwendeten Daten berücksichtigt und daher in der vorliegenden Arbeit nicht weiter behandelt.

```
1  from nltk.corpus import wordnet 1
2  synonyms = [] 2
3  antonyms = []
4
5  for syn in wordnet.synsets("active"):
6      for l in syn.lemmas():
7          synonyms.append(l.name()) 3
8          if l.antonyms():
9              antonyms.append(l.
10                 antonyms()[0].name())
11 print(set(synonyms)) 4
12 print(set(antonyms))
```

Abbildung 2: NLTK WordNet [Quelle](#)

Diese umfangreiche Bibliothek erleichtert die Verarbeitung natürlicher Sprache erheblich. Für die vorliegende Arbeit wurden hauptsächlich Bibliotheken aus NLTK, die WordNet importieren, zur Annotation von belebten und unbelebten Nomen verwendet. Im Kapitel 3.3.1 wird dies genauer erläutert.

2.3 Coreference Resolution

Die theoretischen Grundlagen dieser vorliegende Arbeit führt in die faszinierende Welt der CR. Wie bereits erwähnt, konzentriert sich die vorliegende Arbeit nicht nur auf die Identifizierung der Belebtheit von Nomen, sondern erstreckt sich auch auf CR-Task. Was ist aber CR und inwiefern wird das Thema in der vorliegenden Arbeit untersucht?

CR stellt die Beziehung zwischen zwei linguistischen Ausdrücken dar, die auf dieselbe Entität in einem Diskursmodell verweisen. Diese Beziehung ist entscheidend für das Verständnis von Struktur und Kohärenz in Sprache sowie für das Erkennen von Zusammenhängen zwischen verschiedenen Textpassagen. (Karttunen, 1976)

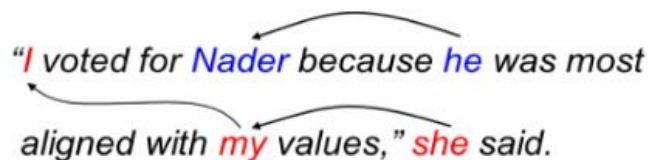


Abbildung 3: Coreference Resolution [Quelle](#)

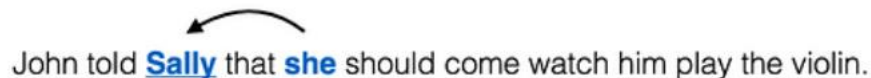
Im Bereich des NLP werden Techniken entwickelt, um Beziehungen zwischen Wörtern, insbesondere zwischen Substantiven und Pronomen, zu identifizieren und zu interpretieren. Die CR stellt eine anspruchsvolle Aufgabe in der NLP dar und kommt in verschiedenen Domänen zum Einsatz, einschließlich Maschinellem Übersetzung (MÜ), Sentiment-Analyse und automatischer Textzusammenfassung. (Mitkov et al., 2012)

Um die grundlegende Funktionsweise von CR zu verstehen, lässt sich der Prozess in drei Schritte unterteilen: Die CR startet mit dem Durchsuchen eines Textes, um linguistische Ausdrücke zu identifizieren, die potenziell auf dieselbe Entität verweisen. Zu diesen Ausdrücken zählen Pronomen (z.B. "he", "she", "it"), bestimmte Nominalphrasen (z.B. "the president", "the car") und unbestimmte Nominalphrasen (z.B. "a man", "an apple"). Dieser Schritt heißt **Identifikation**. Im nächsten Schritt erfolgt die **Verknüpfung**. Hier werden alle vorkommenden referierenden Ausdrücke, die sich auf reale Entitäten beziehen, nach ihrer Identifizierung verknüpft. Das bedeutet, dass das System sie mit einer einzigen Entität verbindet. Wenn zum Beispiel ein Text die Sätze "Johnson is a doctor. He works at a hospital," enthält, würde die Coreference Resolution "He" mit "Johnson" verknüpfen, weil erkannt wird, dass beide auf dieselbe Entität referieren. Nicht zuletzt spielt **Kontextuelles Verständnis** eine

Rolle bei der CR. Es berücksichtigt den Textkontext, um genaue Verknüpfungen herzustellen. Dabei werden grammatikalische Eigenschaften wie Genus- und Numerus- Zuordnung sowie semantische Kompatibilität berücksichtigt, um sicherzustellen, dass die verknüpften linguistischen Ausdrücke tatsächlich auf dieselbe Entität verweisen. (Jurafsky & Martin, 2023)

Allerdings stellt die CR-Aufgabe das System vor Herausforderungen, insbesondere in Fällen der Ambiguität oder komplexen Referenzen. Zum Beispiel muss das System im Satz *"I saw a cat. It was chasing a mouse."* bestimmen, ob *"It"* sich auf die Katze oder die Maus bezieht.

Im Bereich der CR tritt das Konzept des "Antezedens" als zentraler Orientierungspunkt hervor. Antezedenten dienen als linguistische Referenzen für Pronomen, das heißt sie sind die Entitäten, auf die sich Pronomen beziehen. (Jurafsky & Martin, 2023)



John told **Sally** that **she** should come watch him play the violin.

Abbildung 4: Antezedent Quelle

Die Signifikanz von Antezedenten im Kontext der CR nimmt einen zentralen Stellenwert in dieser Arbeit ein. Untersucht wird die Frage, inwieweit kontextualisierte Vektoren zur Übertragung semantischer Eigenschaften von Nomen (belebt oder unbelebt) eines Antezedenten auf das korrespondierende Pronomen beitragen. Die kontextualisierten Vektoren basieren auf der Analyse der kontextuellen Ebene des Textes, um linguistische Informationen zu extrahieren, wie es im BERT-Modell umgesetzt (Devlin et al., 2018). CR und die Erkennung von Antezedenten als linguistische Phänomene erfordern Kontextualität nicht nur, um erfolgreich aufgelöst zu werden (Kauf et al. 2022; Tenney et al., 2019), sondern vermutlich auch, um die semantischen Informationen der Nomenklassen (belebt vs. nicht belebt) des darauf bezogenen Antezedenten identifizieren zu können.

2.4 Transformer

Google präsentierte im Jahr 2017 einen innovativen Ansatz in der Architektur von neuronalen Netzen: den Transformer (Vaswani et al., 2017). Dieser erzielte im Bereich Natural Language Processing (NLP) herausragende Ergebnisse bei der Modellierung von Sequenzen und übertraf dabei deutlich die bisher entwickelten Sprachmodelle auf Basis von Rekurrenten Neuronalen

Netzen (RNNs). Dieser Fortschritt stellt einen Meilenstein in der Entwicklung dar und bildet eine wesentliche Grundlage für die Entstehung der beiden bedeutenden Sprachmodelle des Generative Pretrained Transformers (GPT) und BERT. Beide Modelle zeichnen sich durch die Kombination der Transformer-Architektur mit Ansätzen des unüberwachten Lernens (engl. unsupervised Learning) aus. Dadurch machten diese Modelle das Training aufgabenspezifischer Architekturen, die von Grund auf trainiert werden müssen, überflüssig. Auf Grundlage der GPT und BERT Modellen wurden weitere Transformer-Modelle entwickelt. (Tunstall et al., 2023)

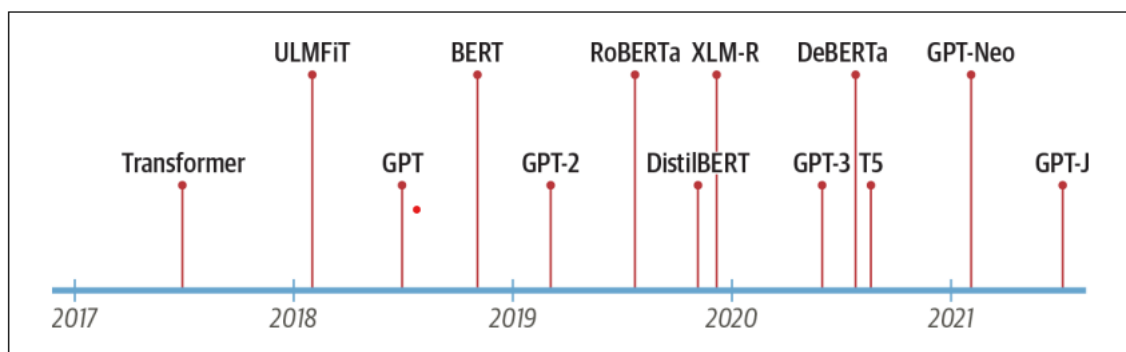


Abbildung 5: Chronologische Entwicklung von Transformer-Modelle (Tunstall et al., 2023)

Um die Architektur der Transformer-Modelle zu verstehen, ist es erforderlich, die drei Hauptkomponenten zu erläutern.

- Transformer besteht aus zwei verschiedenen Komponenten: einem Encoder, der den Eingabetext bearbeitet, und einem Decoder, der Vorhersagen für spezifische Aufgaben erstellt.
- Der Attention-Mechanismus ermöglicht es sich auf spezifische Teile der Eingabedaten zu konzentrieren, während eine Aufgabe bearbeitet wird. In Bezug auf Transformer Modelle wird dieses Mechanismus in Verbindung mit dem Encoder-Decoder-Framework verwendet.
- Transfer Lernen ist ein Konzept, das durch Transformer Modelle verstärkt wurde. Ein Modell, das für eine bestimmte Aufgabe entwickelt wurde und für eine andere, ähnliche Aufgabe verwendet wird. Das Modell kann dann von den bereits gelernten Informationen profitieren und muss nicht von Grund auf neu trainiert werden. (Tunstall et al., 2023)

2.4.1 Bert-Modell

Die Sprachfähigkeiten von BERT basieren auf seiner komplexen Architektur, die den innovativen Fortschritt in der Entwicklung der Sprachmodelle demonstriert. Um die Leistungsfähigkeit von BERT vollständig zu verstehen, wird in der vorliegenden Arbeit eine detaillierte Analyse seiner Struktur durchgeführt und die grundlegenden Komponenten, die das sprachliche Verständnis von BERT begründen, tiefgehend untersucht. Im Jahr 2018 stellten Forscher von Google Research ein revolutionäres Modell vor, das als BERT bekannt ist und die Abkürzung für "Bidirectional Encoder Representations from Transformers" steht. Dieses Modell stellte einen wichtigen Wendepunkt in der Welt des NLP dar. Nach seiner Einführung gewann BERT schnell an Anerkennung aufgrund seiner bemerkenswerten Fähigkeit, in einer Vielzahl von NLP- und Natural Language Understanding (NLU)-Aufgaben eine herausragende Genauigkeit (engl. accuracy) zu erreichen. (Devlin et al, 2018).

BERT nutzt die Transformer-Architektur, die für ihre Fähigkeit bekannt ist, sequenzielle Daten zu verarbeiten. Was BERT jedoch von anderen Modellen unterscheidet, ist seine bidirektionale Architektur. Während vorherige Modelle den Text in eine Richtung gelesen haben, kann BERT gleichzeitig sowohl den links als auch den rechts vom Wort liegenden Kontext berücksichtigen und bietet so einen breiten Blick auf die linguistische Domäne. Dieser einzigartige Ansatz verleiht BERT eine unvergleichliche Fähigkeit, Sprache zu verstehen, indem es komplexe Abhängigkeiten in ihr erfasst.

Beim Training von BERT werden unüberwachten (eng. unsupervised) Hauptstrategien angewendet:

a. Masked Language-Modell (MLM)

Beim MLM Verfahren werden zufällig 15% der Wörter in jeder Sequenz durch ein [MASK]-Token ersetzt. Das Modell versucht anschließend, die ursprünglichen Werte dieser maskierten Wörter vorherzusagen, basierend auf dem Kontext der nicht maskierten Wörter. Um diese Implementierung durchzuführen, werden drei Schritte benötigt. 1- Das Hinzufügen einer Klassifikationsschicht (engl. Classification Layer) zum Encoder-Output. 2- Die Multiplikation der Ausgabevektoren mit der Embedding Matrix. 3-Die Berechnung der Wahrscheinlichkeit jedes Worts im Vokabular mittels Softmax.

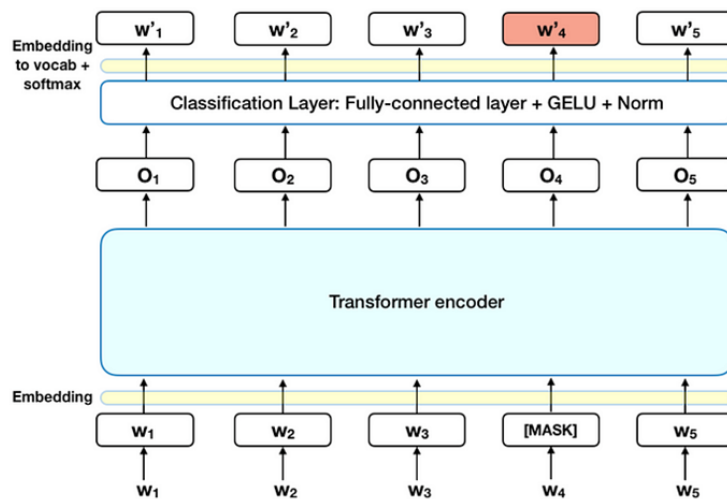


Abbildung 6: MLM (Paharia et al., 2021)

b. Next Sentence Prediction (NSP)

Beim (NSP) Verfahren wird dem Modell einige Satzpaare gefüttert, und es lernt vorherzusagen, ob der zweite Satz im Paar, dem ersten Satz im ursprünglichen Dokument folgt. Während dieses Trainings bestehen die Hälfte der Eingaben aus einem Paar, bei dem der zweite Satz tatsächlich der nachfolgende im ursprünglichen Dokument ist. In den anderen 50% der Fälle wird ein zufälliger Satz aus dem Korpus als zweiter Satz ausgewählt. Die Idee dahinter ist, dass der zufällige Satz voraussichtlich nicht in Beziehung zum ersten Satz steht. Um dem Modell zu helfen, während des Trainings zwischen diesen beiden Satztypen zu unterscheiden, wird die Eingabe vor dem Füttern in das Modell folgendermaßen transformiert: Am Anfang des ersten Satzes wird ein [CLS]-Token eingefügt und am Ende jedes Satzes ein [SEP]-Token. Jedem Token wird ein Satz-Embedding zugeordnet, das bestimmt, ob es zu Satz A oder B gehört. Zudem erhält jeder Token ein Position-Embedding (engl. positional embedding), das seine Position in der Sequenz kennzeichnet.

Das Fine-Tuning von BERT mit dem Self-Attention-Mechanismus des Transformers ermöglicht eine einfache Anpassung für verschiedene Aufgaben in NLP, egal ob für einzelne Texte oder Textpaare. Je nach Aufgabe werden spezielle Eingaben und Ausgaben in BERT verwendet, um das gesamte Modell anzupassen. Normalerweise um spezifische Modelle für bestimmten Aufgaben zu trainieren, wird eine zusätzliche Output-Layer zum bestehenden Bert hinzugefügt und das gesamte Modell feinabgestimmt (fine-tune). Während individuelle Token-Repräsentationen für spezifische Aufgaben auf Token Ebene, wie z.B. „sequence tagging“ or „question answering“ verwendet werden, dient die [CLS]-Repräsentation für Klassifizierung-Task. Das Fine-Tuning ist kosteneffizienter als das Pre-Training und kann innerhalb kurzer Zeit

mit modernen Hardware-Ressourcen wie TPU oder GPU durchgeführt werden. Bert kann, dank seine fine-tuning Technik, in NLP für folgende Aufgaben mit hervorragende Leistungs erfolgreich erzielen: Sentiment-Analyse, Question Answering, Text prediction, Text generation, Summarization-Task. In der vorliegenden Arbeit wird keine Fine-Tune Prozess gebraucht und aus diesem Grund wurde dieser Abschnitt knapp zusammengefasst. (Devlin et al, 2018).

2.4.2 Bert Input-Repräsentationen

Ein Besonderheit von BERT ist die Verwendung der WordPiece-Tokenisierung (Wu et al 2016). Diese Methode zerlegt Tokens in ihre jeweiligen Subtokens. Dies ist besonders nützlich, um mit unbekannten Token umzugehen, die nicht im Wortschatz vorhanden sind. Ein Beispiel hierfür ist das Wort *"playing"*, welches, wie in der genannten Abbildung 7 dargestellt, in *"play"* und *"##ing"* tokenisiert wird. Dieser Prozess wird durchgeführt, bevor die Tokens encodet werden.

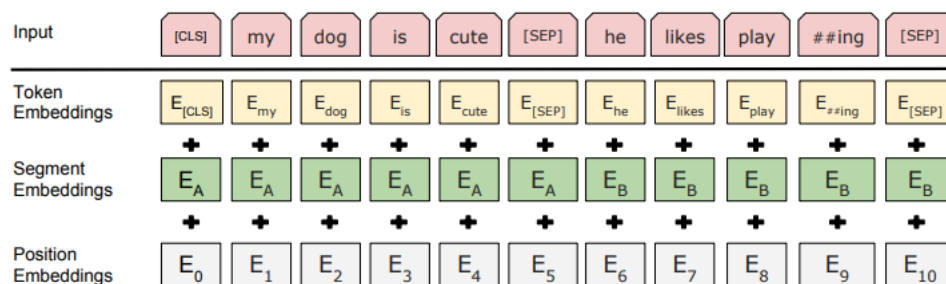


Abbildung 7: Bert Input Repräsentation (Devlin et al, 2018)

Die neuronalen Netzwerke (NN) funktionieren grundsätzlich mit Zahlen. Aus diesem Grund werden alle Tokens als Zahlen dargestellt. Beispielsweise, wenn man einen Wortschatz von 100.000 Wörtern hat, sind jedem Wort ein einzigartiger Index zwischen 1 und 100.000 zugeordnet. Somit kann jedes Wort durch seinen festgelegten Index repräsentiert werden. Das entspricht in der Abbildung 7 der Token Embedding Repräsentation. Wie oben erwähnt wurde, können damit die Sätze unterschieden werden, indem sie mit einem speziellen Token ([SEP]) voneinander getrennt werden. Zusätzlich dazu wird jedem Token ein gelerntes Embedding hinzugefügt (engl. segment embeddings), das anzeigt, ob es zum ersten Satz oder zum zweiten Satz gehört. Am Ende fügt BERT die positionelle Einbettungen (eng. positional embedding) hinzu um die Position jedes Wortes in der Sequenz zu berücksichtigen. Die Eingabedarstellung

eines bestimmten Tokens entsteht durch die Addition der zugehörigen Token-, Segment- und positionelle Embedding. (Devlin et al, 2018)

3 Methodik

Im Abschnitt zur Methodik wird detailliert erläutert, wie Daten gesammelt und verarbeitet wurden. Des Weiteren werden die Techniken und Tools vorgestellt, die im Rahmen dieser Arbeit eingesetzt wurden. Konkrete Schritte, beginnend mit der Datenerfassung, über die Vorverarbeitung, bis hin zu eigenen für die Analyse implementierten Funktionen werden dargelegt. Dieser Abschnitt dient als ausführlicher Leitfaden, um den methodischen Ansatz dieser Arbeit nachzuvollziehen.

3.1 Datenerhebung und -vorbereitung

Die Daten wurden ursprünglich aus OntoNotes Release 5.0³ extrahiert, das von der Heinrich-Heine-Universität Düsseldorf (HHU) bereitgestellt wurde. OntoNotes Release 5.0 ist ein annotiertes Korpus und repräsentiert die finale Version des OntoNotes-Projekts. Dieses Projekt wurde in Kooperation von BBN Technologies, der Brandeis University, der University of Colorado, der University of Pennsylvania und dem Information Sciences Institute der University of Southern California durchgeführt. Das annotierte Korpus umfasst eine Vielzahl von Textgenres, darunter Nachrichten, Weblogs, Rundfunk, Talkshows und Telefonate, und ist in drei Sprachen verfügbar: Englisch, Chinesisch und Arabisch. Insgesamt beinhaltet das Korpus 2,9 Millionen Wörter, von denen 300.000 auf Arabisch, 1.200.000 auf Chinesisch und 1.445.000 auf Englisch sind. Diese Aufteilung wird in der Abbildung 8 ausführlich dargestellt.

³ [OntoNotes Release 5.0 - Linguistic Data Consortium \(upenn.edu\)](https://www.linguisticdataconsortium.org/ontonotes/)

	Arabic	English	Chinese
News	300k	625k	250k
BN	n/a	200k	250k
BC	n/a	200k	150k
Web	n/a	300k	150k
Tele	n/a	120k	100k
Pivot	n/a	n/a	300

Abbildung 8: OntoNotes annotiertes Korpus mit 2,9 Millionen Wörter [Quelle](#)

Die linguistischen Annotationen aus "OntoNotes Release 5.0" berücksichtigten strukturelle Informationen und eine flache Semantik für Substantive und Verben, inklusive ihrer Verknüpfung mit einer Ontologie (z. B. WordNet) sowie Koreferenzen. Folgend wurden nur die originalen englischen Daten aus "OntoNotes Release 5.0" herangezogen, die etwa 1,5 Millionen Wörter umfassen (Weischedel et al., 2012). Dabei wurden ausschließlich die Coreference-Dateien aus HTML/XML-Dokumenten mithilfe der Python-Bibliothek BeautifulSoup extrahiert und anschließend mit verschiedenen Python-Bibliotheken verarbeitet. Das Hauptinteresse dieser Arbeit lag auf Substantiven (Nomen und Eigennamen), Pronomen und der Identifikation ihrer Coreference-Verknüpfungen (Coref-IDs). Um diese Elemente zu extrahieren, wurde Part-of-Speech-Tagging (POS) mit der spaCy-Bibliothek durchgeführt. Die Beziehungen zwischen Nomen (Antezedent) und ihren Pronomen wurden schließlich als "belebt" oder "nicht belebt" annotiert.

3.2 Coreference Resolution Texten (Coref. Dateien)

Das Coreference- Annotationsprojekt wurde bei BBN Technologies durchgeführt. (Weischedel et al., 2012) Wie in Abschnitt 2.3 erläutert, zielt das Projekt darauf ab, das Textverständnis durch die Identifikation von Koreferenzen zu verbessern. Beispielweise für den Satz "*She had a good suggestion and it was unanimously accepted*" wird die Verbindung zwischen „a good suggestion“ und „it“ in einem Satz erfolgreich durchgeführt und dadurch, wie oben in CR Theorie erklärt wurde, können neben Nomen auch nominale Mentions, Pronomen und sogar einige Verben als koreferent markiert werden. Das erfolgte in Englisch, Chinesisch und Arabisch. Die Coref-Dateien enthalten Informationen zur Coreference-Resolution in einem Text wie folgt:

Die Datei beginnt mit einer „DOC“-Markierung, die Informationen zur Dokumentennummer („DOCNO“) enthält. Die Dokumentennummer bestehen aus mehreren Teilen, die durch Schrägstriche getrennt sind. Der Hauptinhalt des Textes befindet sich im <TEXT> Abschnitt. Des Weiteren gibt es in der Datei die Coreferenzen, die mit <COREF> Tags identifiziert werden, wobei diese Tags eine eindeutige Coref-ID und einen TYPE-IDENT haben. Dargestellt in der Abbildung 9.

```
<DOC DOCNO="wb/a2e/00/a2e_0000@0000@a2e@wb@en@on">
<TEXT PARTNO="000">
Celebration Shooting Turns <COREF ID="1__2" TYPE="IDENT">Wedding</COREF>
Into a Funeral in <COREF ID="1__4" TYPE="IDENT">Southern Gaza Strip</COREF>
```

Abbildung 9: Ursprüngliche Coreference Daten

3.3 Datenextraktion und -vorbereitung

Um nur die Coref-Dateien vom ganzen Ontonotes-Projekt Dateien zu extrahieren, wurde, gezielt mithilfe der „glob“ Python-Standardbibliothek gearbeitet, die bestimmte Dateien im Dateisystem durchsucht. Es wurden insgesamt 1787 Coreference-Dateien extrahiert. Zunächst wurden weitere Python-Bibliotheken importiert, die für diese Daten erforderlich waren wie: BeautifulSoup, spaCy, „re“ (RegEx) (Reguläre Ausdrücke) und Pandas. BeautifulSoup aus bs4 wurde für die Verarbeitung bzw. Parsen von HTML-oder XML-Dokumenten angewendet. Für Extraktion der Coref-IDs wurde explizit eine Funktion erstellt, die die Coref-IDS aus den Coref. Mentions extrahiert. Viele überflüssige Mentions für die vorhandenen Daten wurden mittels der RegEx Bibliothek bereinigt. Des Weiteren wurde gezielt mit spaCy gearbeitet, um die Coref-Texten zu zerlegen und Informationen über POS-Tags zu extrahieren. Mit der Pandas-Bibliothek wurde es ermöglicht, die gewünschte Daten als CSV-Datei zu speichern und nachträglich zu bearbeiten.

3.3.1 Daten-Annotation

Für eine präzise Extraktion der gewünschten Informationen wurden die semantischen Relationen der Substantive (Nomenklassen) mithilfe der WordNet Ontologie analysiert. Diese

Ontologie wurde in Python durch die NLTK-Bibliothek importiert, um alle Hypernoms für jedes entsprechende Wort zu extrahieren. Zusätzlich wurden diese Substantive durch selbst erstellte Funktionen in Kategorien für belebte oder unbelebte Objekte annotiert. Die durchgeführte Datenaufbereitung erlaubte es, jedes Substantiv und Pronomen, die durch eine Coref_ID miteinander verknüpft sind, als belebt oder unbelebt zu annotieren. Dabei wurden die Wörter betrachtet, die von der spaCy-Bibliothek als Nomen, Pronomen oder Eigennamen kategorisiert und die gleiche Coref_ID aufwiesen. Dadurch war es ebenfalls möglich, den Antezedenten eines Pronomens zu identifizieren. Zur Durchführung der Annotation wurden verschiedene Funktionen erstellt, darunter:

get_hyponyms(synset): Diese Funktion, inspiriert von WordNet, ermöglicht das Erfassen von übergeordneten Kategorien, sogenannten Hypernymen, auf rekursive Weise. Damit kann die hierarchische Struktur von semantischen Relationen innerhalb von WordNet erkundet und umfassendere Bedeutungen aufgedeckt werden. Wie in der Theorie erläutert wurde, ist ein Hypernym ein Wort, dass in seiner Bedeutung andere Wörter in einer hierarchischen Weise umfasst.

```
# Function to get hyponyms recursively
def get_hyponyms(synset):
    hyponyms = set()
    for hyponym in synset.hyponyms():
        hyponyms |= set(get_hyponyms(hyponym))
    return hyponyms | set(synset.hyponyms())
```

Abbildung 10: get hyponyms Funktion [Quelle](#)

extract_hyponyms(w): Mit dieser Funktion erfolgt das Extrahieren der Hyperonymen für ein gegebenes Wort und bieten Einblicke in dessen umfassenderen semantischen Kontext. Die WordNet-Datenbank dient für diese Herausforderung als wertvolle Ressource. Die Funktion verwendet das wn.synsets Modul, um die Synsets (Bedeutungen) des Wortes zu erhalten. Wenn es Synsets für das Wort gibt, nimmt das erste Synset. Dann wird die Funktion get_hyponyms aufgerufen, um die Hyperonyme dieses Synsets zu erhalten. Für jedes gefundene Hypernym fügt es das Hauptlabel (lemma) des Hypernoms zur hyponyms_ Liste hinzu. Schließlich gibt die Funktion die hyponyms_ Liste zurück. Siehe Abb. 11.

```
# Function to extract hypernyms from WordNet
def extract_hypernyms(w):
    hypernyms_ = []
    synsets = wn.synsets(str(w), pos=wn.NOUN)
    if len(synsets) != 0:
        word = synsets[0]
        hypernyms = get_hypernyms(word)
        for hypernym in hypernyms:
            hypernyms_.append(hypernym.lemma_names('eng')[0])
    return hypernyms_
```

Abbildung 11: Extraktion von Hypernyms [Quelle](#)

Der gesamte Code dient als Tool, um die Hypernyme (höherstufige Kategorien) eines gegebenen Wortes zu finden, wobei der Schwerpunkt auf der ersten Bedeutung des Wortes als Nomen in der WordNet-Datenbank liegt.

Abbildung 12 veranschaulicht, wie die Hierarchien der Nomen „car“ und „wife“ aus WordNet mittels Python extrahiert werden.

```
for el in final_[final_["Word"]=="car"][:1]["hypernyms"]:
    print("The hierarchical structure of WordNet of word car : ")
    print()
    print(el)
```

The hierarchical structure of WordNet of word car :

```
['whole', 'conveyance', 'object', 'entity', 'artifact', 'physical_entity', 'wheeled_vehicle', 'self-propelled_vehicle', 'container', 'vehicle', 'motor_vehicle', 'instrumentality']
```

```
for el in final_[final_["Word"]=="wife"][:1]["hypernyms"]:
    print("The hierarchical structure of WordNet of word wife : ")
    print()
    print(el)
```

The hierarchical structure of WordNet of word wife :

```
['causal_agent', 'whole', 'organism', 'object', 'entity', 'female', 'spouse', 'woman', 'adult', 'physical_entity', 'domestic_partner', 'relative', 'person', 'living_thing']
```

Abbildung 12: Hierarchische Struktur von WordNet [Quelle](#)

label_animate (hypernyms): Diese Funktion ist entscheidend für die Klassifikation von Daten in "belebt" oder "unbelebt". Sie analysiert detailliert die Hypernyme, die mit jeweiligen Nomen in Relationen stehen, und unterstützt so bei der Klassifikation nach Belebtheit. Die Belebtheit wird dabei der Kategorie "living things" zugeordnet oder, wenn nicht zutreffend, als unbelebt annotiert. Abb. 13.

```
# Function to Label animate and inanimate
def label_animate(hypernyms):
    if type(hypernyms) != float:
        if len(hypernyms) == 0:
            return "wordnet_false"
        if "living_thing" not in hypernyms:
            return "inanimate"
        else:
            return "animate"
```

Abbildung 13: Annotation mit belebt/nicht belebt [Quelle](#)

3.3.1.1 Annotationsformen

Wie bereits oben erläutert, wurden die Pronomen als belebt (Label 1) oder unbelebt (Label 0) annotiert, basierend auf den Antezedenten, auf die sie sich beziehen. Es stellt sich jedoch die Frage, was geschieht, wenn der Antezedent aus mehreren Substantiven oder Eigennamen besteht, die unterschiedlich, als belebt oder unbelebt, annotiert wurden. In solchen Fällen wird zunächst geprüft, ob der Antezedent aus mehreren Nomen oder Pronomen besteht. Falls dies zutrifft, wird anschließend überprüft, ob eines dieser Wörter als unbelebt annotiert wurde. Ist dies der Fall, erhält das Pronomen das Label 2. Somit erhält das Pronomen das Label 1 /0 nur, wenn alle Nomen bzw. Eigennamen als belebt/unbelebt annotiert sind. Als Beispiel kann der Satz "He said: there is a women's coffee shop in Barida, and he gave me its name - 'Jothun'" herangezogen werden. Hier ist "women's coffee shop" der Antezedent für das Pronomen "its". Der Antezedent besteht aus drei Nomen (women, coffee, shop), wobei eines davon als belebt und die beiden anderen als unbelebt annotiert wurden. Das Pronomen erhält somit das Label 2 und wird im endgültigen Datensatz nicht berücksichtigt. Dabei werden nur die Daten mit den Labels 1 und 0 einbezogen.

Sentence	Sentence_no	Word	POS	coref_id	antecedent	hypernyms	labels	end_labels
('Celebration', 'Shooting', 'Turns', 'Wedding'...	0	celebrators	NOUN	1__16	NaN	['physical_entity', 'causal_agent', 'whole', '...	animate	1.0
('Celebration', 'Shooting', 'Turns', 'Wedding'...	0	his	PRON	1__16	one of the celebrators	NaN	Pronouns	1.0

Abbildung 14: Annotation mit belebt (1) Beispiel genommen vom Datensatz

('I', 'sat', 'imagining', 'that', 'coffee', 's...	65	its	PRON	1__74	that coffee	NaN	Pronouns	0.0
---	----	-----	------	-------	-------------	-----	----------	-----

Abbildung 15: Annotation mit unbelebt (0) Beispiel genommen vom Datensatz

('He', 'said', ',', 'there', 'is', 'a', 'women...')	59	its	PRON	1__21	a women 's coffee shop in	NaN	Pronouns	2.0
---	----	-----	------	-------	------------------------------	-----	----------	-----

Abbildung 16: Annotation mit Unklar (2) Beispiel genommen vom Datensatz

Nach dieser Filterung verbleiben 20.583 Daten, bestehend aus Substantiven, Eigennamen und den zugehörigen Pronomen. Die Daten wurden für den Probing-Klassifikator in Trainings- und Testdaten aufgeteilt. Es verblieben 16.466 Datensätze für das Trainingsdaten und 4.117 Datensätze für die Testdaten.

3.3.2 Extraktion von Embeddings

Um die Embeddings für jedes Wort zu extrahieren, wurden die Sätze aus den Zeilen als Input für den Tokenisierer (engl. Tokenizers) verwendet. Da der BERT den WordPiece-Ansatz verwendet, werden die Wörter, die nicht im Vokabular des Tokenisierers enthalten sind, in Subtokens zerlegt. Um die Embeddings für jedes Wort zu extrahieren, wurden die Sätze im String-Format als Input für den Tokenisierer verwendet. Der BERT- Tokenisierer, der auf dem WordPiece-Ansatz basiert, zerlegt die Wörter, die nicht in seinem Vokabular enthalten sind, in Subtokens. Genauso wie die Autoren des offiziellen BERT-Papers die Embeddings von Subworten für Wortklassifikationsaufgaben wie den NER-Task verwendet haben, wird in dieser Studie die gleiche Methodik angewendet. Schließlich handelt es sich bei dieser Studie auch um eine Aufgabe zur Wortklassifikation. Die Autoren haben lediglich das erste Embedding eines Subtokens als Repräsentation für das gesamte Wort betrachtet. (Devlin et al, 2018)

```
[ '[CLS]', 'Celebration', 'Shooting', 'Turn', '##s', 'Wedding', 'Into', 'a', 'Fun', '##eral', 'in', 'Southern',
  'Gaza', 'Strip', 'As', '##ad', '1', '/', '20', '/', '2007', 'Gaza', '-', 'UP', '##I', 'The', 'cheer', '##s',
  'and', 'hail', '##s', 'of', 'happiness', 'at', 'a', 'wedding', 'in', 'Khan', 'You', '##nes', 'in', 'the',
  'southern', 'Gaza', 'Strip', 'turned', 'into', 'screams', 'and', 'moan', '##s', 'of', 'pain', 'after', 'one',
  'of', 'the', 'c', '##ele', '##bra', '##tors', 'lost', 'control', 'of', 'his', 'weapon', ',', 'from', 'which', 'a',
  'number', 'of', 'bullets', 'were', 'released', 'that', 'killed', 'the', 'groom', '"', 's', 'brother', 'and',
  'hit', 'three', 'other', 'relatives', 'of', 'his', ',', 'turning', 'the', 'wedding', 'into', 'a', 'funeral', 'in',
  'moments', 'c', '##ele', '##bra', '##tors', '[SEP]' ]
```

Main tokens: ['Turns', 'Funeral', 'Asad', 'UPI', 'cheers', 'hails', 'Younes', 'moans']
 Subtokens: ['##s', '##eral', '##ad', '##I', '##s', '##s', '##nes', '##s']

Abbildung 17: Subtoken-Tokenizer Output eines Satzes vom Datensatz [Quelle](#)

3.4 Probing-Implementierung

Für die Implementierung des Probing-Tasks wurden zwei Methoden verwendet: Zum einen die logistische Regression (LR), eine einfache Vorhersagemethode, die mit einer grundlegenden Methode des maschinellen Lernens vergleichbar ist. (Chung, M. K. 2020) Dabei kann man sich vorstellen, dass eine gerade Linie gezogen wird, um belebte von unbelebten Wörtern zu trennen. Zum anderen wurde das einfache neurale Netzwerk (NN) (Srinivas. et al., 2014) eingesetzt. Dieses Verfahren ist fortschrittlicher und kann mit einem intelligenten Roboter verglichen werden, der lernt, Worte in einem tieferen Zusammenhang zu verstehen. Beide Klassifikatoren werden mit BERTs kontextualisierter Wortrepräsentation gefüttert und darauf trainiert, vorherzusagen, ob ein Nomen oder ein Pronomen belebt oder unbelebt ist.

Der NN-Klassifikator, setzt sich aus einer Eingabeschicht zusammen, deren Größe den BERT-Embedding entsprechen, einer Schicht mit ReLU-Aktivierung und einer Ausgabeschicht mit Sigmoid-Aktivierung für die binäre Klassifizierung.

3.4.1 Leistungsindikatoren

Um die Effizienz der Modelle zu bewerten, werden drei Metriken verwendet.

- Genauigkeit (eng. Accuracy): Diese gibt an, wie häufig das Modell korrekte Vorhersagen trifft.
- F1-Score: Dieser Wert kombiniert Präzision (wie viele der als belebt vorhergesagten Elemente tatsächlich belebt sind) und Sensitivität (eng. Recall) (wie viele der tatsächlich belebten Elemente vom Modell identifiziert wurden). Man kann ihn als eine Art Ausgewogenheit zwischen Präzision und Sensitivität betrachten.
- Konfusionsmatrix (engl. confusion Matrix): Mit dieser Matrix lässt sich nachvollziehen, in welchen Bereichen das Modell Fehler gemacht hat. Sie zeigt, wie viele belebte und unbelebte Elemente korrekt bzw. inkorrekt klassifiziert wurden. (Sokolova et al., 2016)

4 Probing-Ergebnisse

Das LR-Modell erzielte eine Accuracy von 86% und einen F1-Score von 79%. Es ist ersichtlich, dass die Accuracy in diesem Fall signifikant besser abschneidet als der F1-Score. Allerdings kann Accuracy nicht mehr als verlässliches Maß betrachtet werden, wenn der Datensatz unausgeglichen ist, das heißt, die Anzahl der Proben in einer Klasse deutlich größer ist als die Anzahl der Proben in den anderen Klassen. In solchen Fällen liefert die Accuracy eine überoptimistische Einschätzung der Klassifizierungsleistung für die Mehrheitsklasse. (Sokolova et al., 2016) Dies trifft auch auf die vorliegenden Ergebnisse zu, bei der die Anzahl der unbelebten Stichproben mit 13.156 deutlich höher ausfällt als die Anzahl der belebten Stichproben, die 7.427 beträgt. Das Modell könnte die Mehrheitsklasse die meiste Zeit vorhersagen und dennoch eine hohe Accuracy erzielen. In dem Fall könnte das Modell sagen, dass alles unbelebt ist und eine hohe Punktzahl erreichen, weil die meisten Wörter unbelebt sind. Es ist also von Vorteil, eine hohe Genauigkeit zu haben, jedoch nicht der einzige Parameter, welchen man betrachten sollte. Der F1-Score hingegen würde in diesem Fall niedriger sein, weil die Präzision oder der Recall (oder beides) niedriger wäre, was darauf hinweist, dass das Modell Probleme mit dem Umgang in Bezug auf die Minderheitsklasse hat. Daher gibt der F1-Score ein ausgewogeneres und realistischeres Bild von der Leistung eines Modells bei unausgebalancierten Datensätzen. Der F1-Score hilft, ein Gleichgewicht zu finden. In logistischen Regressionsmodell liegt er bei etwa 78%. Das bedeutet, dass die kontextualisierte Wortvektoren von BERT gute Information über Belebtheit der Antezedent und der Pronomen ausweist, aber es besteht noch Verbesserungspotential.

Die Konfusionsmatrix dient dazu, zu erkennen, bei welchen Klassifizierungen das LR-Modell Schwierigkeiten hat. Sie ist ein hilfreiches Instrument, um das Gleichgewicht zwischen Präzision und Recall zu finden. In unserem Fall zeigt die Matrix, dass das Modell gut darin ist, unbelebte Objekte (Nomen, Pronomen) korrekt zu identifizieren, während die Belebtheit dagegen nicht richtig erkannt wird.

True Positives (TP): Das sind Wörter, die tatsächlich belebt sind, und das Modell hat sie korrekt als belebt vorhergesagt. Im Logistischen Regressionsmodell sind es etwa 1182.

True Negatives (TN): Das sind Wörter, die tatsächlich unbelebt sind, und das Modell hat sie korrekt als unbelebt vorhergesagt. Im Logistischen Regressionsmodell sind es etwa 2324.

False Positives (FP): Das sind Wörter, die tatsächlich unbelebt sind, aber das Modell hat sie fälschlicherweise als belebt vorhergesagt. Im Logistischen Regressionsmodell sind es etwa 311.

False Negatives (FN): Das sind Wörter, die tatsächlich belebt sind, aber das Modell hat sie fälschlicherweise als unbelebt vorhergesagt. Im Logistischen Regressionsmodell sind es etwa 300. Hier besteht noch Verbesserungspotenzial.

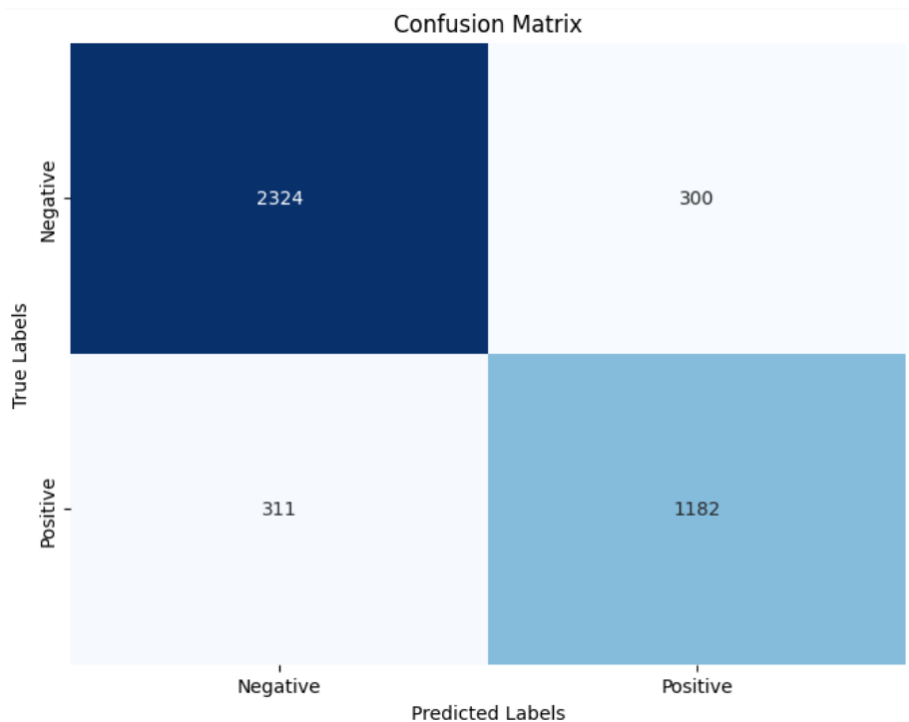


Abbildung 18: Confusion Matrix, LR-Model [Quelle](#)

Das NN-Modell hingegen erreichte eine Accuracy von 87% und einen F1-Score von 82%. Die Ergebnisse weisen hinsichtlich der Accuracy nur minimale Unterschiede auf. Beim F1-Score zeigt das NN-Modell deutlich bessere Ergebnisse. Es ist zu erwähnen, dass das NN-Modell durch die Erhöhung der Anzahl der Durchläufe (Epochen) eine verbesserte Genauigkeit erzielen konnte. Die Ergebnisse wurden nach 40 Epochen erreicht. Im Deep Learning spielt die Anzahl der Trainingsepochen eine entscheidende Rolle für die Leistung des Modells. Mit der Erhöhung der Epochenanzahl auf bis zu 40 vertieft das Modell seinen Lernprozess und optimiert sein Datenverständnis. Es identifiziert dabei feine Muster und komplexe Beziehungen zwischen den einzelnen Merkmalen, was zu einer verbesserten Modellleistung führt. (Haerder, 2020).

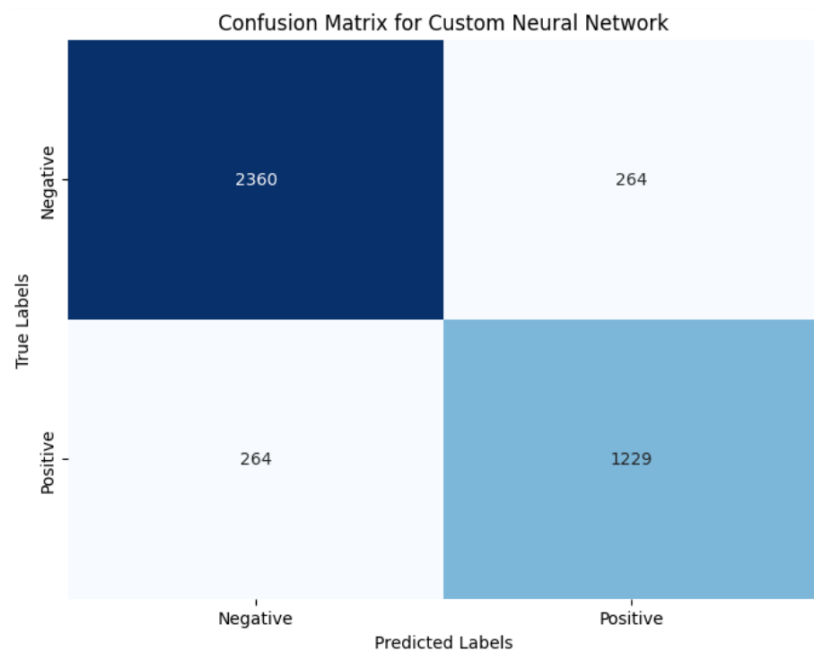


Abbildung 19: Confusion Matrix, NN-Model [Quelle](#)

Wie in Abbildung 19 ersichtlich, erzielt das NN-Modell bessere Ergebnisse bei der Reduzierung von False Positives und False Negatives.

5 Diskussion

Ein zentraler Punkt in der vorliegenden Arbeit war vor allem die Erkennung semantischer Eigenschaften (Belebtheit) von Pronomen, die sich auf ein Antezedent beziehen. Basierend auf den Ergebnissen des LR-Modells ist es von großem Interesse zu sehen, wie gut die Pronomen im Vergleich zu den Nomen klassifiziert wurden.

Von insgesamt 2380 Pronomen wurden 1816 korrekt und 564 inkorrekt klassifiziert. Die Wahrscheinlichkeit der korrekt klassifizierten Pronomen beträgt ca. 76,4 %. Das bedeutet, dass etwa 3 von 4 Pronomen richtig klassifiziert wurden. Dies ist ein gutes Ergebnis, zeigt aber auch, dass es noch Raum für Verbesserungen gibt. Die Wahrscheinlichkeit der inkorrekt klassifizierten Pronomen beträgt ca. 23,7 %. Das heißt, dass etwa 1 von 4 Pronomen falsch klassifiziert wurde. Obwohl diese Zahl relativ klein ist, zeigt sie doch, dass es eine nicht unerhebliche Anzahl an Fehlklassifikationen gibt, die in zukünftigen Modellen oder durch die Verbesserung des bestehenden Modells angegangen werden sollten.

Im Rahmen der vorliegenden Studie wurden von Test-Datensatz 1.726 Nomen analysiert, von denen 1.644 korrekt und 82 inkorrekt klassifiziert wurden. Dies resultiert in einer Erfolgsquote von etwa 95,2 % und einer Fehlerquote von rund 4,8 %. Diese Ergebnisse heben die hohe Präzision des Modells bei der Klassifizierung von Nomen hervor, insbesondere im Vergleich zur Klassifizierung von Pronomen. Des Weiteren umfasste der Test-Datensatz 6 Eigennamen, die alle korrekt kategorisiert wurden.

Wie in der Theorie, Kapitel 2.4.2 erläutert, zerlegt der BERT-Tokenizer die Wörter in Subtokens. Da in dieser Arbeit das Embedding des ersten Subtokens das gesamte Wort repräsentiert, erscheint es sinnvoll, eine Analyse der vorhergesagten Wörter durchzuführen, die in Subtokens zerlegt wurden. Dies sollte im Vergleich zu den Wörtern erfolgen, die nicht in Subtokens untergliedert wurden.

Die Ergebnisse zeigen, dass von den 102 Nomen, die in Subtokens zerlegt wurden (all_noun_subtoken), 85 korrekt und 17 inkorrekt klassifiziert wurden. Dies ergibt eine Erfolgsquote von etwa 83,3 % und eine Fehlerquote von etwa 16,7 %. Im Vergleich dazu wurden von den 1.624 Nomen, die nicht in Subtokens zerlegt wurden (all_noun_without_subtoken), 1.559 korrekt und 65 inkorrekt klassifiziert, was eine Erfolgsquote von etwa 96 % und eine Fehlerquote von etwa 4 % bedeutet.

5.1 Implikationen und zukünftige Forschung

Die vorliegende Arbeit leistet einen Beitrag zur Forschung im Bereich des kontextuellen Word Embeddings hinsichtlich semantischer Informationen wie Belebtheit der Nomen und Pronomen. Es ist jedoch zu beachten, dass die Ergebnisse unter verschiedenen Voraussetzungen interpretiert werden müssen.

Der Datensatz, der aus 16.466 Trainingsdatensätzen besteht, wird in der Welt des Maschinellen Lernens und Deep Learning als mittelgroß bis klein angesehen. Es ist wahrscheinlich, dass ein größerer Datensatz akkuratere Ergebnisse liefern könnte.

Ein weiterer Aspekt, der in zukünftigen Forschungen berücksichtigt werden sollte, betrifft die Datenannotation. In der vorliegenden Arbeit wurden die Daten automatisch annotiert, sowohl hinsichtlich der Wortart (Part-of-Speech, POS) als auch im Hinblick auf die Belebtheit und Unbelebtheit der Nomen. Es bleibt zu untersuchen, welche Ergebnisse man mit manuell gelabelten Daten erzielen würde, und ob diese zu einer Verbesserung der Genauigkeit und Zuverlässigkeit der Analysen führen würden.

Die Statistik über die vorhergesagten Wörter, die vom Tokenizer in Subtokens zerlegt wurden, besagt, dass die Klassifizierung von Nomen, die in Subtokens zerlegt wurden, eine deutlich höhere Fehlerquote aufweist als die Klassifizierung von Nomen, die nicht zerlegt wurden. Dies könnte darauf hinweisen, dass der Zerlegungsprozess in Subtokens die Genauigkeit der Klassifizierung beeinträchtigt und somit ein Optimierungspotential darstellt.

In der vorliegenden Arbeit wurden lediglich die Embeddings für das erste Subtokens herangezogen und als Repräsentation für das gesamte Wort verwendet. Für zukünftige Forschungen wäre es daher ein wichtiger Aspekt, den Durchschnitt der Embeddings aller Subtokens zu berechnen, anstatt sich lediglich auf das erste Embedding zu stützen.

6 Fazit

Große Sprachmodelle wie BERT haben eindrucksvoll demonstriert, dass sie in verschiedenen Domänen der NLP erfolgreich eingesetzt werden können. Insbesondere hat ihre Architektur, vor allem die Technik der Word Embeddings, die Forschung dazu inspiriert, zu untersuchen, inwiefern diese Modelle die Feinheiten oder Nuancen einer Sprache erfassen können. Das Hauptziel der vorliegenden Arbeit ist es, diesem Forschungsinteresse nachzugehen, indem gefragt wird, wie viele Informationen in den Word Embeddings enthalten sind und ob Word Embeddings umfangreiche semantische, morphologische und syntaktische Eigenschaften einer Sprache abbilden, wie es in Kapitel 2.1 zum Forschungsstand dargestellt wurde. Die Large Language Models (LLMs) haben bewiesen, dass sie über ausreichendes Wissen bezüglich bestimmter sprachlicher Phänomene verfügen. So hat BERT beispielsweise in der semantischen Repräsentation von Informationen erfolgreiche Ergebnisse erzielt, was auch in dieser Arbeit bestätigt wurde.

Wie in Kapitel 1.1 zum Forschungsziel dargelegt, strebt die vorliegende Arbeit danach, eine Antwort auf die zentrale Frage zu finden: Inwieweit verfügt BERT über Wissen bezüglich der Belebtheit und Unbelebtheit von Nomen (Antezedenten) und deren korrespondierenden Pronomen? Die Ergebnisse der Probing-Tasks im Rahmen dieser Bachelorarbeit heben deutlich die Vorteile und Grenzen von kontextualisierten Word Embeddings bei der Erkennung semantischer Eigenschaften von Wörtern hervor, insbesondere bei der Unterscheidung zwischen belebten und unbelebten Antezedenten und den dazugehörigen Pronomen. Konkret verfügt BERT über ausreichende Kenntnisse in Bezug auf die Belebtheit und Unbelebtheit von Nomen. Ein relativ einfaches Logistisches Regressionsmodell (LR-Modell) erreichte dabei eine Genauigkeit (Accuracy) von 86%. Mit einer Fehlerquote von 4,8 % hat BERT bei der Klassifizierung von Nomen hinsichtlich Belebtheit und Unbelebtheit hervorragend abgeschnitten. Auch bei der Performanz von Pronomen zeigt BERT mit einer Fehlerquote von etwa 23,7 % eine gute Generalisierungsfähigkeit, obwohl hier noch Verbesserungspotenzial besteht. Das Phänomen der Subtokens durch den BERT-Tokenizer scheint kein wesentliches Hindernis darzustellen; Nomen, die nur durch das erste Subtoken repräsentiert werden, wiesen eine Fehlerquote von ungefähr 16,7 % auf.

Diese Arbeit kommt jedoch nicht ohne Einschränkungen aus. Eine Limitation ist die Größe des in dieser Studie verwendeten Datensatzes. Eine weitere Einschränkung ergibt sich aus der Methodik der automatischen Datenannotation. Zukünftige Forschungen könnten sich darauf

konzentrieren, diese Einschränkungen zu überwinden und dadurch die Qualität und Verlässlichkeit der Forschungsergebnisse zu verbessern.

Literaturverzeichnis

- Aikhenvald, A. Y. (2006). *Classifiers and noun classes: semantics*. Elsevier.
- Allan, K. (1977). Classifiers. *Language*, 53(2), 285-311.
- Beloucif, M., & Biemann, C. (2021, November). Probing pre-trained language models for semantic attributes and their values. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 2554-2559).
- Cann, R. (2011). Sense Relations. In C. Maienborn, K. Von Heusinger, & P. Portner (Eds.), *Semantics: An International Handbook of Natural Language Meaning* (Vol. 1, pp. 456-478). (Handbook of Linguistics and Communication Science). Mouton de Gruyter.
- Chung, M. K. (2020). Introduction to logistic regression. arXiv preprint arXiv:2008.1356
- Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. arXiv preprint arXiv:1805.01070.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Gelman, R., Durgin, F. H., & Kaufman, L. (1995). Distinguishing between animates and inanimates: Not by motion alone.
- Goddard, C., & Schalley, A. (2010). Semantic analysis. In *Handbook of natural language processing*. Chapman & Hall/ CRC.
- Haerder, C. (2020). Deep Learning und klassisches Machine Learning Ein Vergleich anhand des Boston Housing Value Datensatzes. *Reading Processing Applying*, 1
- Halili, A. (2023). BERT_Probing. Durchgeföhrt an der HHU. Verfügbar unter https://github.com/itsmeeeeeee/BERT_Probing
- Hawkins, R. D., Yamakoshi, T., Griffiths, T. L., & Goldberg, A. E. (2020). Investigating representations of verb bias in neural language models. arXiv preprint arXiv:2010.02375.
- Hewitt, J., & Liang, P. (2019). Designing and interpreting probes with control tasks. arXiv preprint arXiv:1909.03368.
- Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4), 665-695.
- Jawahar, Ganesh, Sagot, Benoît, & Seddah, Djame. (2019). What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

- Karttunen, L. (1976). Discourse referents. In *Notes from the linguistic underground* (pp. 363-385). Brill.
- Kauf, C., Ivanova, A. A., Rambelli, G., Chersoni, E., She, J. S., Chowdhury, Z., ... & Lenci, A. (2022). Event knowledge in large language models: the gap between the impossible and the unlikely. *arXiv preprint arXiv:2212.01488*.
- Lenci, A., & Sahlgren, M. (2023). *Distributional semantics*. Cambridge University Press.
- Mandera, P., Keuleers, E., & Brysbaert, M. (2017). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, 92, 57-78.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.
- Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., ... & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2), 1-40.
- Mitkov, R., Evans, R., Orăsan, C., Dornescu, I., & Rios, M. (2012). Coreference resolution: To what extent does it help NLP applications?. In *Text, Speech and Dialogue: 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings 15* (pp. 16-27). Springer Berlin Heidelberg.
- Mosbach, M., Degaetano-Ortlieb, S., Krielke, M. P., Abdullah, B. M., & Klakow, D. (2020). A closer look at linguistic knowledge in masked language models: The case of relative clauses in American English. *arXiv preprint arXiv:2011.00960*.
- Nordquist, Richard. (2023, April 5). Homonymy: Examples and Definition. Retrieved from <https://www.thoughtco.com/homonymy-words-and-meanings-1690839>
- Paharia, N., Pozi, M. S. M., & Jatowt, A. (2021). Change-Oriented Summarization of Temporal Scholarly Document Collections by Semantic Evolution Analysis. *IEEE Access*, 10, 76401-76415.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR abs/1802.05365* (2018). *arXiv preprint arXiv:1802.05365*.
- Şahin, G. G., Vania, C., Kuznetsov, I., & Gurevych, I. (2020). LINSPECTOR: Multilingual probing tasks for word representations. *Computational Linguistics*, 46(2), 335-385.
- Srinivas, S., & Babu, R. V. (2015). *Deep learning in neural networks: An overview*. Computer Science.
- Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006, December). Beyond accuracy